A Dynamically Reconfigurable Transceiver for Software Defined Radio

Rehan Muzammil
Dept. of Electronics Engineering,
Aligarh Muslim University,
Aligarh, India

M. Salim Beg
Dept. of Electronics Engineering,
Aligarh Muslim University,
Aligarh, India

Mohsin M. Jamali
Dept. of Elect. Engg. & Comp. Sc.
The Univ. of Toledo,
Toledo, Ohio, USA

ABSTRACT
Software Defined Radio (SDR) is an emerging field in the area of Digital Communications. The ever growing interest in SDR systems lies in the fact that it can provide multi-standard architecture. This paper describes a dynamically reconfigurable architecture of an SDR system on a Model Based Development (MBD) platform which is a new approach and saves a considerable amount of time in design, testing and implementation. In this work, four modulation schemes are chosen for this application of SDR namely BPSK, QPSK, 16-QAM, 256-QAM. The paper describes the implementation of these four modulation / demodulation schemes on an MBD platform. The receiver in this architecture is made intelligent enough to determine which modulation scheme the transmitter is transmitting and switches on to the corresponding demodulation scheme. Finally the results obtained are presented in graphical form.

General Terms
Software Defined Radio, Quadrature Amplitude Modulation, Field Programmable Gate Arrays, Model based Development.

Keywords
FPGA, SDR, DSP, MBD, QAM.

1. INTRODUCTION
Software Defined Radio is a general hardware / software platform which supports inter-communication between systems using different wireless communication standards. The basic idea of an ideal software defined radio receiver is to digitize the received signal using high-speed analog-to-digital converters (ADCs) and to process it by sophisticated programmable system consisting of reconfigurable hardware in the form of high speed Field Programmable Gate Arrays (FPGAs) and Digital Signal Processors (DSPs) [1].

In an SDR system most of the radio receiver/processor functions would be defined by user written software programs to be run on a General Purpose Programmable Processor rather than the functions being implemented strictly in non programmable hardware. With the advances made in the semiconductor technology, future wireless baseband processors move towards Multiprocessor System-On-Chip (MP SoCs) which integrate heterogeneous processing elements tailored for different processing tasks. MP SoCs offer high performance, re-configurability and energy efficiency [2, 3].

In order to meet the high computation and adaptability requirements, multi-core platform has been widely used for the advances in SDR systems due to their high performance of signal processing. As compared to single core processor, a multi-core processor is not that fast enough but has the ability to perform more efficiently by handling more work in parallel [4]. It may be noted that the work being reported in the current paper uses an SDR platform which has a System-on-Chip comprising a TI DSP 64x+ and an ARM 9 General Purpose Processor (GPP).

SDR systems concept is to replace most of the analog signal processing with digital signal processing to provide flexibility through reconfiguration in future transceivers. The long term objectives of SDR systems lie in things such as the wideband antenna, efficient power amplification of a multi-standard signal, wide band analog-to-digital conversion (ADCs) etc. Initially, the idea in SDR systems was to place the ADCs right after the antenna in order to achieve higher overall flexibility. However, this demands a very high speed ADC in the range of Giga Samples per second which is practically not feasible and an ADC of this speed is not currently available in the market. Hence, an Intermediate Frequency (IF) stage is kept after the antenna to bring down the frequency to workable range which can be handled by the commercially available ADCs. [5 - 9].

The scarcity of spectrum has become a major bottleneck of the development of next generation communication systems. To counteract this problem the concept of SDR system was introduced. SDR is an intelligent wireless communication system where the wireless system changes its parameters for transmission or reception to communicate efficiently. This active alteration in parameters more frequently called dynamic reconfiguration is based on several factors in radio environment such as the frequency spectrum, behavior of the user and state of the network [10].

Research on modulation classification (MC) has been carried out for at least two decades. These are considered as important radio techniques for reconfigurable radio convergence in mobile systems and, have recently attracted interest due to their capabilities of replacing several receivers with one universal receiver. The advent of realizable SDR systems allows the implementation of creative transceiver designs, which can dynamically adapt to the communications channel and user applications. One such transceiver design involves a receiver that automatically determines the modulation scheme used in an incoming signal [11], and this demonstrates the flexibility of the SDR system.

Recently various forms of telecommunication services are beginning to extend more intelligent communications. Most of the system functions including wide-band digital filter, directly digital frequency synthesizing, digital down conversion, modulation and demodulation, coding etc. are accomplished by software in an SDR system [12].

Today the wireless communication systems are statically specified by their built in link and physical layer functions. Up until now, the dedicated DSPs have been well known technology associated with the development of SDR systems. However the implementation complexity and cost increases when the digitization of the signal is as close as possible to the antenna. With the recent growth in demand for cellular services and the proliferation of micro and pico-cells, there is an urge to reduce the cost of the base stations. For this reason there is an increasing interest in the SDR systems. The main idea behind SDR systems is that to employ a wideband ADC and all subsequent processing is implemented digitally. By deploying SDR systems we can reduce substantially the cost of the base stations, since a single transceiver is required instead of a transceiver for each channel. Moreover, the flexibility of the base stations is increased in the sense that it can be programmed for different existing standards and for future standards [13, 14].

Present technologies oblige their users to buy a particular device for each type of communication standard, and the operators have to deploy base stations for each system. These circumstances require users to switch from one cellular standard to another. For this case, emerging multi-standard multi-bands, multi-modes and multi-protocols systems have been deployed. Therefore, SDR systems can serve as the fundamental technology. The present work on SDR systems includes hardware adaptability by the software means that is incorporated by downloading the necessary software from the network and installing. Thus SDR is the mechanism supporting system rebuild on request [15].

Component software technology for maximizing software reuse is greatly helpful to overcome the extreme complexity of embedded software and reduce time-to-market. The SDR system has adopted SCA (Software Communication Architecture) of the JTRS (Joint Tactical Radio System) as the standard structure of SDR embedded systems. The SCA also provides a flexible environment for integrating hardware and software written in various languages by adopting Common Object Request Broker Architecture (CORBA) as its base middleware. The SCA specification establishes and implementation-independent framework with baseline requirements for the development of JTRS SDR systems. The requirement includes both interface and behavioral specifications that ensure maintenance, portability and configurability across vendor platforms [16, 17].

Keeping in view of all these challenges faced in the design of a feasible SDR the authors have proposed a design of an SDR transceiver where different modulation schemes, namely BPSK,4,16,256-QAM, are transmitted in turn and the receiver is made intelligent enough to correctly choose the De-Modulation scheme automatically. This is a good example of dynamic reconfiguration which is one of the requirements of the SDR transceiver.

Section 2 describes SDR Architecture; Section 3 illustrates the very new type of design and development of Communication

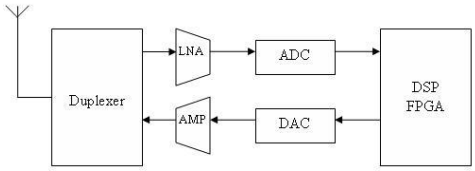

Fig 1: Ideal Software Defined Radio Architecture

Circuits known as Model Based Development; Section 4 describes the reconfigurable transceiver architecture designed by the authors; Section 5 & 6 describes the modulator and de-modulator respectively as designed by the authors; Section 7 illustrates the real time results; Section 8 gives the conclusions.

2. SOFTWARE RADIO ARCHITECTURES

2.1 Ideal Software Defined Radio

The ideal SDR system is given in Fig. 1. In this architecture the digital part of the receiver is placed as close to the antenna as possible. However, due to implementation constraints the ideal SDR system is not feasible [5].

2.2 Feasible SDR Architecture

A feasible SDR or practical SDR is one in which there is an IF stage between the RF and the base-band. This is depicted in Fig. 2. The IF is responsible to bring down the frequency which can be easily sampled by an ADC of about 125 Msps. This receiver is divided into two parts, Analog Front End (AFE) and Digital Front End (DFE). The AFE shifts the frequency from RF to IF with which the ADC has to act upon. The DFE is that part of the SDR which is used to perform all the functions digitally which is done by conventional radio receiver in analog form. For this to be practical, the DFE should be very high speed digital circuit. Fortunately, DSPs and FPGAs now are commercially available which can do the job quite effectively [5].

3. MODEL BASED DEVELOPMENT

In this work, the authors have used a Model Based Development platform. This is Small Form factor (SFF) SDR low power tunable equipment. This SFF SDR platform is composed of three boards: RF module, Data Conversion Module, Data Processing Module. The board is illustrated in the Fig. 3.

The SFF SDR platform comes with two board support packages – Board Software Development Kit (BSDK) and Model Based Development kit (MBDK). The BSDK allows users to quickly become fully functional developing C, C++, or assembly language codes for the DSP and GPP, or HDL code for the FPGA by giving users an understanding of all the platform's major interfaces such as VPSS, audio codec, data

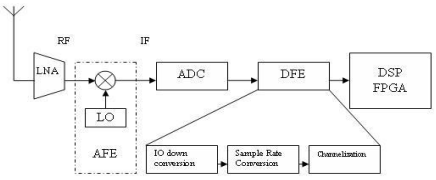

Fig 2: Feasible Software Radio Receiver Architecture

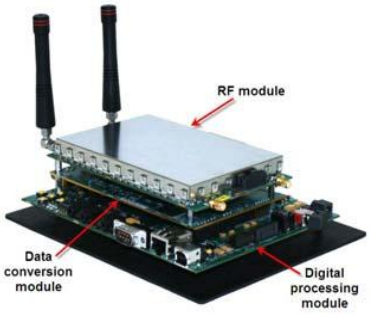

Fig 3: The SFF SDR Platform

conversion module, or RF module. Similarly, the MBDK allows users to develop applications for the platform with Simulink within MATLAB. By targeting the DSP and FPGA with MBDK tools, users can deploy and validate algorithms on the hardware more rapidly.

Unlike the SDR development platforms on the market, the SFF SDR development platform is a hybrid hardware-software system that supplies the necessary full-signal chain for multi-protocol software defined radios. By separating the base-band, IF, and RF from one another as distinct modules (rather than maintaining a single, fixed architecture), developers can extend their radio development capabilities and optimize costs and power consumption. The SFF SDR development platforms can be used to perform four types of development – FPGA, DSP, GPP, and Model based (combination of the above three). This paper describes model based development using this platform, though only FPGA and DSP are exploited [18]. The FPGA is Xilinx Virtex4 SX35 and the DSP is TI C64x+.

As can be seen from Fig. 4 [19], the traditional design involves three processes before production, often performed by three distinct groups. These processes are linked to each other as shown by arrows. Because of this it takes much more time in the final outcome at the production stage. On the other hand in the model based system design, all these three processes are performed by a single group as shown in Fig. 5 [19]. Hence, a substantial amount of the design and development time is saved [19].

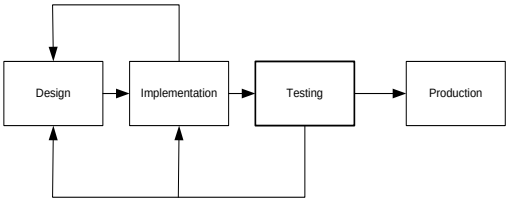

Fig 4: Traditional System Design Methodology

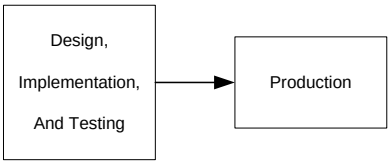

Fig 5: Model Based system Design Methodology

4. RECONFIGURABLE TRANSCEIVER ARCHITECTURE

The whole system for the transceiver is designed and developed in two parts, Front-End Module and the Back-End Module. The front end module is where all the control of the DACs, ADC, and RF reside. The back end module is where almost all of the baseband processing of the transceiver in the form of forward error correction, interleaving and de-interleaving [20], modulation and de-modulation [18], and detection is performed. The front end is the DSP model and the back-end is the FPGA model. Both these models collectively form the transceiver. The front and the back end models are illustrated in Fig. 6(a) & (b) respectively.

4.1 Front End Model

The front model is the model where the DAC, ADC, RF control resides. It is called front end model because of this reason. As can be seen from the front end model in Fig. 6 (a), the audio codec interface digitizes the analog signal which is given to the "line-in" jack of the data processing module. Each sample is 32-bits. Each of these samples is then passed on to the back-end through a Video Processing Sub System (VPSS) interface for the baseband processing. After the baseband processing in the receiver portion of the system, the generated 32-bit samples are sent back to the front end through VPSS interface for converting it back to analog audio through the audio codec interface which is forwarded to the "line-out" jack of the data processing module which is connected to speaker for listening. The modulator, demodulator and the real time results will be described in the following sections.

4.2 Back End Model

The back-end model is where almost all the baseband data processing is done. The various processes involved are Block Coding, Interleaving, P/S conversion, Modulation, De-Modulation, Detection, De-Interleaving, Block Decoding, S/P conversion. The Block Coding / Block DeCoding used is extended (7,4) Block Codes or (8,4) Block Codes [24]. An 8x8 Interleaver / De-Interleaver is implemented. The Modulation and Demodulation / Detection are described in the next two sections.

5. MODULATOR

The modulator for the system is illustrated in Fig. 7. Here there is a subsystem named Custom_Register which takes the button selection input from the front-end as shown in Fig. 6 (a). When a button is pushed, its corresponding output line is set to '1' and the rest of the lines are at '0' state, a signal is generated which is pushed onto the Modulation Select Block. This modulation select outputs a 2-bit word for each MUX4 & MUX5 which outputs the corresponding modulated signal. This is illustrated in Table 1.

From Table 1 it is clear that if button 2 is pressed, QPSK Modulation will be chosen and the transmitter will transmit the bits using this modulation scheme. When none of the buttons is pressed, the transmitter will transmit BPSK signal. As shown in Fig. 6 (b) the transmitter part consists of all the blocks up to the DAC. From the ADC onwards it is the receiver part. No matter which modulation scheme is selected for transmission of the information, the receiver has to be made intelligent enough to decide which modulation scheme is transmitted and set the receiver to the corresponding

demodulation scheme. This will be taken up in the next section which describes the demodulator.

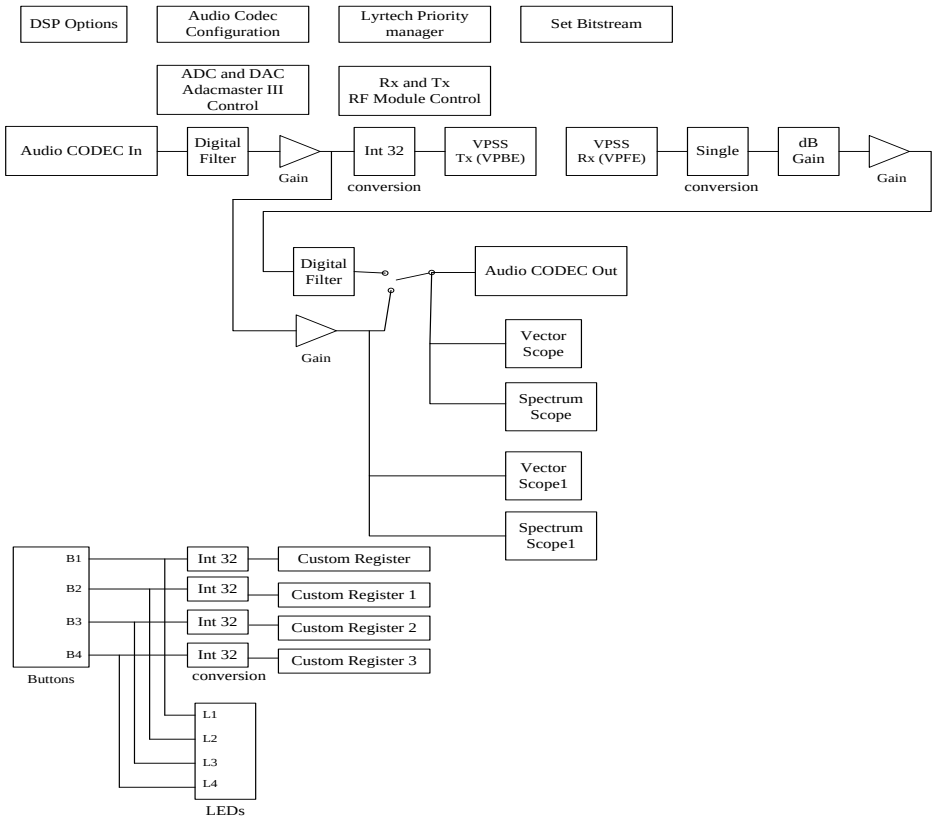

Fig 6(a): Front End Model

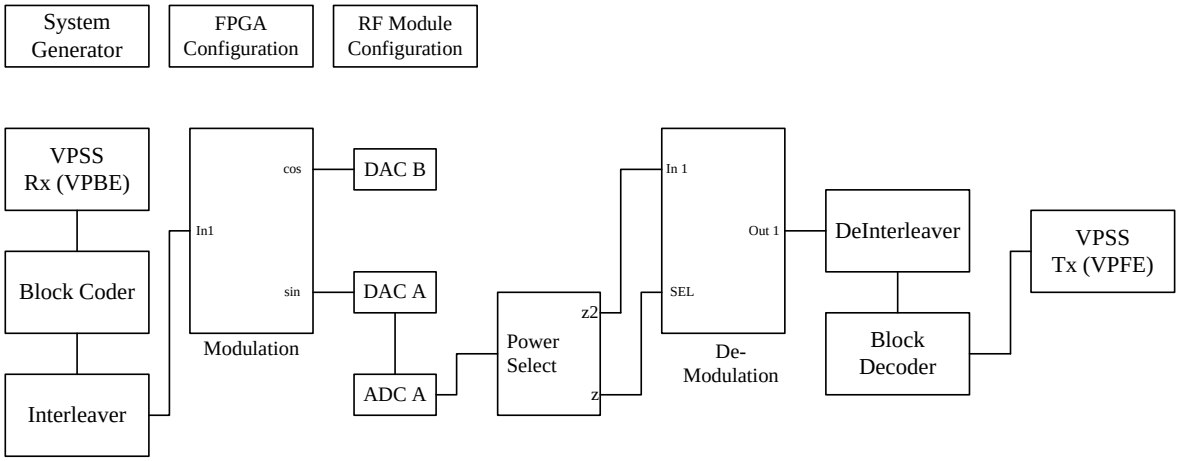

Fig. 6(b): Back End Model

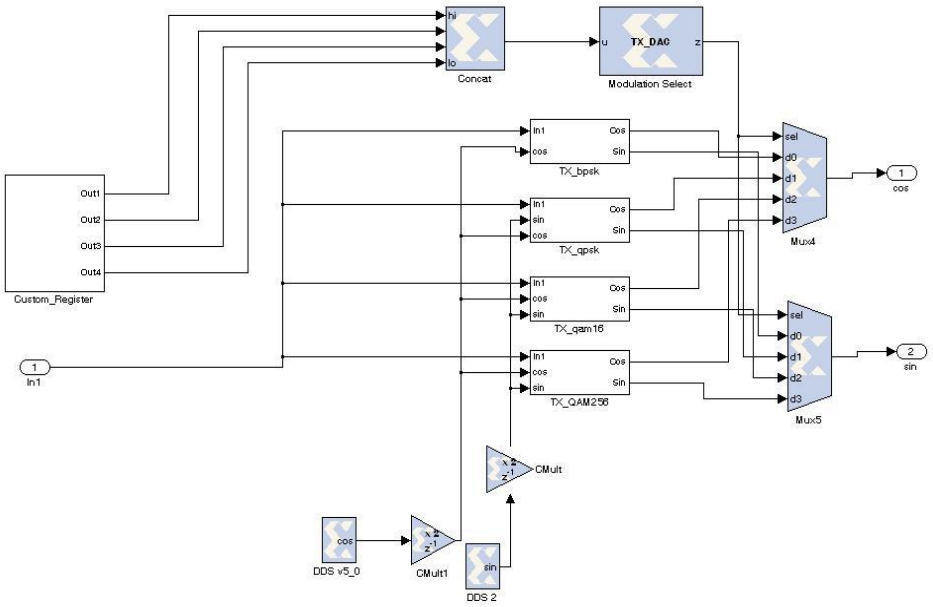

Fig 7: Modulator Architecture

Table 1. Modulation Select

Button Selected	Custom_Register Output	Modulation Select	Modulation Selected
none	0000	00	BPSK
B1	1000	00	BPSK
B2	0100	01	QPSK
B3	0010	10	QAM-16
B4	0001	11	QAM-256

From Fig. 7 it is clear that when QAM-256 is selected, there are two channels, cosine and sin for the in-phase (I) and quadrature-phase (Q) respectively. To generate the cosine and sin signals, two Direct Digital Synthesizers (DDS) are employed in the design of the system, one for the cosine and one for the sine. The various amplitude values for the various modulation schemes are depicted in Table 2. Power is taken to be $\frac{A^2}{2}$ combined for I and Q channels.

As can be seen from Fig. 8 the slice block chooses 4-bit symbol for the 'I' channel. Likewise, another slice block chooses 4-bit symbol for the 'Q' channel in the sine subsystem. In all there are 8 bits per symbol in the QAM-256 modulation scheme. Since this circuit follows gray coding, the amplitude values are not in order. This is depicted in Table 3.

Table 2. Modulation Scheme, Amplitude and Power Transmitted

Modulation Scheme	Amplitude Range (A) for I and Q	Total Transmitted Power range (P)
BPSK	± 0.125	0.0078125
QPSK	±0.1875, ±0.1875	0.03515625
QAM-16	±0.3125±0.875,	0.09766 – 0.7656
QAM-256	±1±3±5±7±9±11±13±15, ±1±3±5±7±9±11±13±15	1.0 – 225.0

The amplitude thus selected is multiplied sample by sample with the samples generated by the DDS. Hence, the output is modulated cosine wave. Likewise the process for the sine wave is executed and we get a modulated sine wave. These modulated sine and cosine waves are then pushed onto the two DACs (A & B) as shown in the back-end model in Fig. 6 (b). The next block is ADC which comes under the receiver section which is the next section to be discussed.

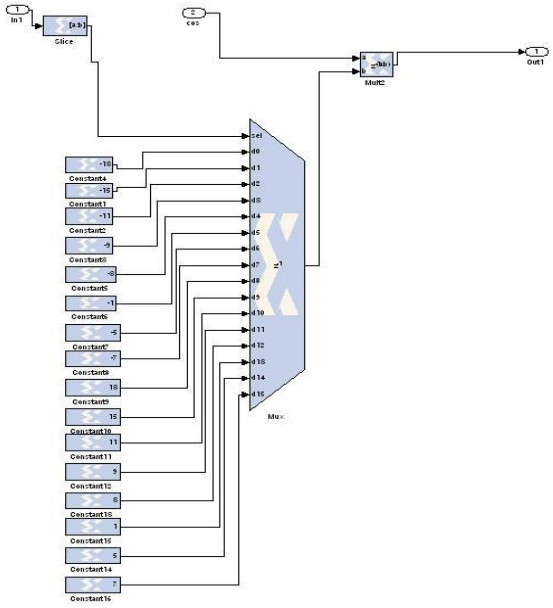

Fig 8: Modulator Cosine Subsystem showing Amplitude values for QAM-256

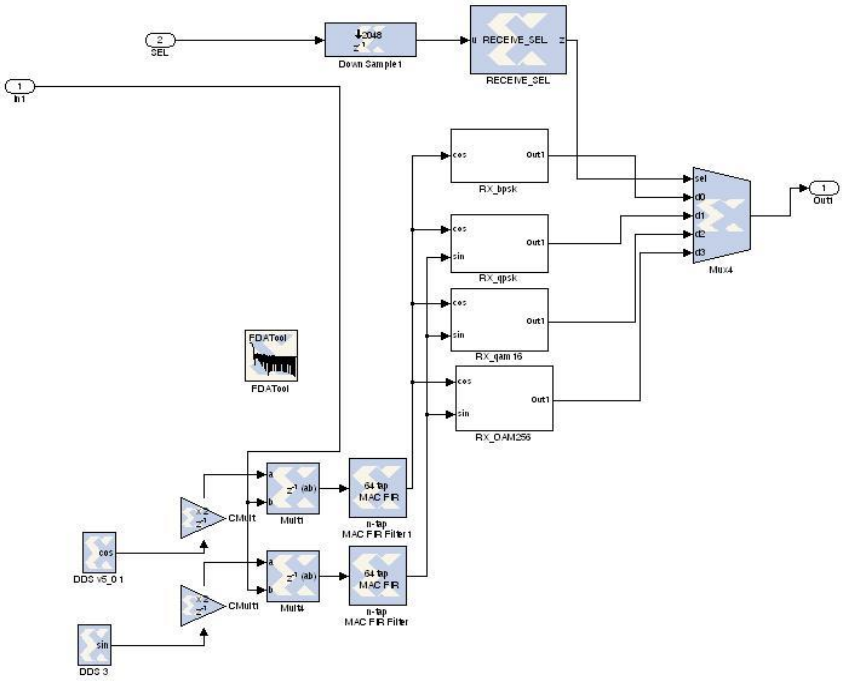

Fig 9: DeModulator Architecture

Table 3. Slice Output and Amplitude Selected for carrier modulation

SLICE OUTPUT for Gray Coded constellation	AMPLITUDE SELECTED
0001	-15
0000	-13
0010	-11
0011	-9
0111	-7
0110	-5
0100	-3
0101	-1
1101	+1
1100	+3
1110	+5
1111	+7
1011	+9
1010	+11
1000	+13
1001	+15

6. DEMODULATOR

The demodulator subsystem is depicted in the Fig. 9. As can be seen in the back-end model there is a power select block just after the ADC. The power select block calculates the power of the incoming signal by squaring each sample received and then summing them up after 64 samples are received and dividing by 64. This is depicted in Equation 1.

$$P = \frac{\sum_{n=1}^{64} |x(n)|^2}{64} \quad \ldots (1)$$

where x(n) is given by

$$x(n) = A_n \cos(\omega n) + B_n \sin(\omega n) \quad \ldots (2)$$

and represents the samples of the received signal, 64 is the number of samples per cycle of the sine / cosine signals generated by DDS and A_n and B_n are the amplitudes per sample. Hence, this power P calculated is passed onto the RECEIVE_SEL block which determines which modulation scheme is transmitted and gives a two-bit output for demodulation selection. The receiver takes 64 samples to decide which modulation scheme is transmitted. This is only one cycle of the received signal. Hence, the maximum period there is an error in modulation select is one cycle or 64 samples. Since the frequency of the sine / cosine wave is 1.25 MHz which has 0.8 µs time period. Hence, there can be an error in demodulation for only 0.8 µs. This shows that virtually there is no error in the demodulation. The modulation select according to the power received is depicted in Table 4. Hence, it can be seen that just by observing the power over one cycle of the sine / cosine signal the modulation scheme is determined. After the decision taken, the corresponding demodulated signal is fed to the detector for detection. Hence, after the detection process, the generated 4 bits each for I (cosine) & Q (sine) channels are concatenated to form an 8-bit symbol and this is pushed onto the S/P converter which converts it into 64-bit word and is fed to De-interleaver. After this process the resultant 64-bit word is fed to the Block Decoder which strips off the redundant 32 bits and makes necessary bit corrections to give a 32-bit word. This is the data word and is fed back to the front-end through the VPSS interface to be converted back to the analog audio through the AUDIO CODEC interface as can be seen in Fig. 6(a). This is fed to the speakers for listening.

Table 4. Power Range, Receive Select, Demodulation selected

Power Range (P)	RECEIVE_SEL output	De-Modulator Selected
0 – 0.01	00	BPSK
0.01 – 0.05	01	QPSK
0.05 – 0.9	10	QAM-16
Greater than 0.9	11	QAM-256

7. REAL TIME RESULTS

The Real-Time results are taken by generating the C and VHDL codes for DSP and FPGA respectively. The code for C language for DSP is generated automatically by Code Composer Studio invoked by Real Time Workshop. The code for VHDL for FPGAs is generated automatically by System Generator. Both these processes are executed from Simulink Environment. After C and VHDL code generation, they are built and compiled into ".out" and ".bit" files respectively and then they are burnt onto the DSP and FPGAs respectively from the Simulink Environment and the circuit becomes operational. Then as an mp3 audio signal is given to the "line-in" jack an output is received at the "line-out" jack. The transmitted and received audio waveforms are shown in Fig. 10 (a) and (b) for BPSK, Fig. 11 (a) and (b) for QPSK, Fig. 12 (a) and (b) for QAM-16, Fig. 13 (a) and (b) for QAM-256. From these figures it is quite clear that the input and output waveform for all the modulation schemes are similar to each other. The only difference is the small latency and amplitude level which may be due to transmission errors. The quality of the received audio has also been checked through the speakers, and has been found to be satisfactory. Since, the system so designed can detect and correct 1 bit error per 4 data bits transmitted, the total number of errors detected and corrected is 8 per 32-bit word, theoretically. Hence, the BER allowed in this case is roughly 8/32 which is 0.25 or 25 bit errors per 100 bits transmitted. Hence, the system so designed can to a large extent counteract the high BER introduced due to low power transmitted. To get detailed information of the Block Coding / De-Coding, Interleaving / De-Interleaving the readers are advised to consult [20] where the authors have described the FEC in detail. But despite the FEC and interleaving / de-interleaving there are some errors which pass through undetected or corrected. Because of these errors there is certain noise in the received signal. The channel used in this work is Additive White Gaussian Noise (AWGN) channel. Moreover, the circuit implemented in this work saves considerable amount of resources both in the form of DSP48s and FPGA chip area (slices) as shown in Table 5. This table clearly shows that the reconfigurable hardware (shown in second last column) gives rise to resource saving compared to all the modulation schemes taken separately. The SliceM parameter in the table occupies exactly 50 % of the total number of slices and it stores the parameters such as Look Up table, Multiplexers, Shift Registers and responsible for high speed arithmetic operation like addition and subtraction etc. The total field signifies the total number of the resources shown in the feature column.

Table 5. Device Utilization Summary for Virtex 4 SX35 FPGA

Feature	BPSK	QPSK	16-QAM	256-QAM	Reconfig	Tota
DSP 48	9	13	13	13	19	192
Slices	4128	4491	4766	5938	6678	1536
SliceM	215	290	300	300	300	7680

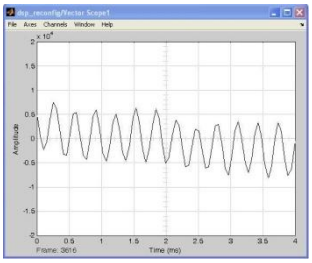

Fig 10(a): For BPSK at Audio Codec IN

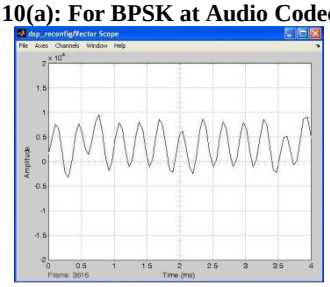

Fig 10(b): For BPSK at Audio Codec OUT

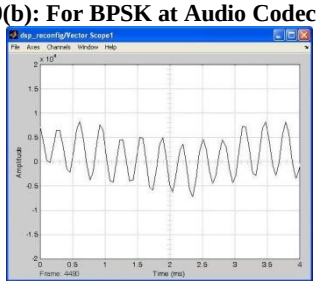

Fig 11(a): For QPSK at Audio Codec IN

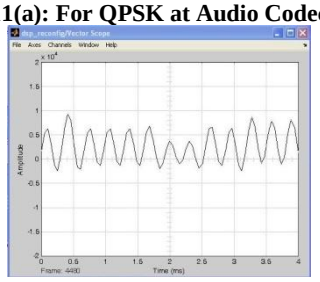

Fig 11(b): For QPSK at Audi Codec OUT

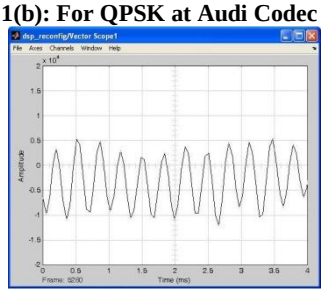

Fig 12(a): For QAM-16 at Audio Codec IN

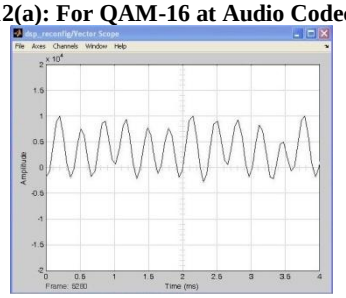

Fig 12(b): For QAM-16 at Audio Codec OUT

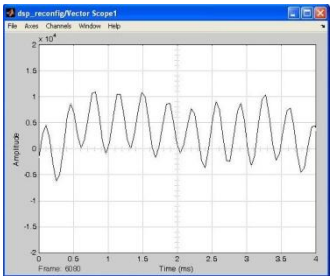

Fig 13(a): For QAM-256 at Audio Codec IN

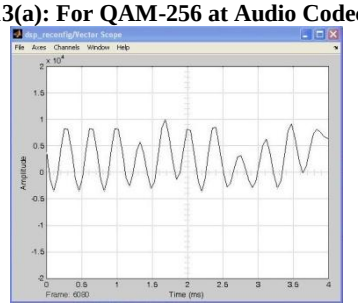

Fig 13(b): For QAM-256 at Audio Codec OUT

8. CONCLUSIONS

The dynamically reconfigurable transceiver has been successfully designed and implemented in this work on a model based development platform for SDR. The circuit is tested in real time mode and is found to be working satisfactorily. The work has demonstrated a considerable amount of saving in the FPGA chip area. Moreover, the model based development is a new way of development which saves a considerable amount of time in the form of design, implementation and testing.

9. REFERENCES

[1] Yeung, K.S.; Chan, S.C., "The design and multiplier-less realization of software radio receivers with reduced system delay", Regular Papers, IEEE Transactions On Circuits and Systems, 2004, Vol: 51, Issue: 12., pp 2444–2459.

[2] Alluri, V.B.; Heath, J.R.; Lhamon, M., "A New Multichannel, Coherent Amplitude Modulated, Time-Division Multiplexed, Software-Defined Radio Receiver Architecture, and Field-Programmable-Gate-Array Technology Implementation", IEEE Transactions on Signal Processing, VOL: 58, Issue: 10, pp 5369 – 5384.

[3] Qiwei Zhang; Kokkeler, A.B.J.; Smit, G.J.M., "A System-level_Design_Method_for_Cognitive_Radio_on_a Reconfigurable_____Multi-processor_____Architecture", International Symposium on System On Chip, 2007, ISSOC 2007, Nov. 19–21, Tampere, Finland, 2007, pp 1–4.

[4] Qingguo Zhou; Shuwei Bai; McGuire, N.; Jianwen Chen., "Isolated_network_model_based_on_Cell_for software_radio_system", IEEE International Symposium on Circuits and Systems 2009, ISCAS 2009, May 24–27 2009, Taipei, Taiwan, pp 2069–2072.

[5] Al Ghouwayel, A.; Louet, Y., "FPGA implementation of a re-configurable FFT for multi-standard systems in software radio context", IEEE Transactions on Consumer Electronics, Vol: 55, No. 2, May 2009, pp 950–958.

[6] Yuce, M.R.; Liu, W., "Design_and_implementation_of_a multirate_sub-sampling_front-end_in_software_radio systems", 2004 IEEE Radio and Wireless Conference, Atlanta, Georgia, USA, 19-22 September 2004, pp 529–532.

[7] Rivet, F.; Deval, Y.; Begueret, J.-B.; Dallet, D.; Cathelin, P.; Belot, D., "The first experimental demonstration of a SASP-based full Software Radio receiver", IEEE Radio Frequency Integrated Circuits Symposium, 2009, RFIC '09, 7–9 June, 2009, Boston , MA, USA, pp 25–28.

[8] Pham, D.-M.; Premkumar, A.B.; Madhukumar, A.S., "Efficient_Sample_Rate_Conversion_in_Software_Radio Employing_Folding_Number_System", IEEE International Conference On Communications, 2009, ICC '09, June 14–18, Dresden, Germany, pp 1 – 5.

[9] Chopra, P.; Kumar, N.; Paily, R., "Ultra_wideband_AGC and_VGA_designs_for_software_radio_applications", 2008 IEEE Region Ten Conference, TENCON 2008, 18–21 November, 2008, Hyderabad, India, pp 1–5.

[10] Rallapalli, H.; Lal Kishore, K., "Dynamic spectrum management scheme in a smart software radio", Fourth International Conference on Wireless Communication and Sensor Networks, 2008. WCSN 2008, Dept. of ECE, Osmania Univ., Hyderabad, India,27–29 Dec, pp 141–147.

[11] Cheol-Sun Park; Dae Young Kim., "A_novel_robust feature_of_modulation_classification_for_reconfigurable software_radio", IEEE Transactions on Consumer Electronics, 2006, VOL: 52, Issue: 4, pp 1193–1200.

[12] Saravanan, M.; Ravi, S., "Software radio technology in FH/MC DS-CDMA communication system", IET-UK International Conference On Information and Communication Technology in Electrical Sciences, ICTES 2007, 20–22 December, 2007, Chennai, India, pp 762–765.

[13] Kan Zheng; Jianfeng Wang; Lin Huang; Decarreau, G., "Open Wireless Software Radio on Common PC", IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, 2006, PIMRC '06, 11– 4 September 2006, Helsinki, Finland, pp 1 – 5.

[14] Araujo, T.; Dinis, R., "Analytical_evaluation_and optimization_of_the ADC (analog-to-digital converter) in software radio___architectures", IEEE Global Telecommunications Conference, 2004, GLOBECOM 2004, 29 Nov – 3 Dec, 2004, Dallas, USA, pp 1066 – 1070.

[15] Zlydareva, O.; Sacchi, C., "Multi-Standard WIMAX/UMTS System Framework Based on SDR", 2008 IEEE Aerospace Conference, 1 – 8 March 2008, Big Sky, MT, USA, pp 1 – 13.

[16] Saehwa Kim; Masse, J.; Seongsoo Hong; Naehyuck Chang., "SCA-based component framework for software defined radio", IEEE Workshop on Software Technologies for Future Embedded Systems, 2003, WSTFES '03, 15-16 May 2003, Hokkaido, Japan, pp 3 – 6.

[17] Noseworthy, Joshua; Kulp, James., "Standard Interfaces for FPGA Components", Military Communication Conference, MILCOM 2007, October 29-31, Orlando, Florida, USA, pp 1-5.

[18] Muzammil, R.; Salim Beg, M.; Jamali, M.M., "Design and implementation of BPSK transmitter and receiver for

Software Defined Radio on a Model Based Development Platform", Proc. IEEE Symposium on Industrial Electronics and Automation, 25 – 28 September, 2011, Langkawi, Malaysia, pp 89–94 (available on IEEEXplore).

[19] Ahmadian, M.; Nazari, Z.J.; Nakhaee, N.; Kostic, Z., "Model_based_design_and_SDR", The 2nd IEE/EURASIP Conference on DSP enabled Radio, 2005, 19–20 September, 2005, Southampton, UK, pp 19/1 – 19/6.

[20] Muzammil, R.; Beg, M.S.; Jamali, M.M., "Design and implementation of Forward Error Correction in Software Defined Radio on a Model Based Development platform", Proc. International Conference on Multimedia, Signal Processing and Communication Technologies, IMPACT–2011, 17–19 December, 2011, Aligarh, India, pp 304–307 (available on IEEEXplore).