Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS 2018)
IEEE Xplore Compliant Part Number: CFP18K74-ART; ISBN:978-1-5386-2842-3

Implementing PSK and FSK Scheme in Zynq-7000 FPGA for SDR applications

Raj Mamarde
VLSI & Embedded,
MITCOE Pune 411038, India.
Email: rajmamarde@gmail.com

Suchitra Khoje
Electronics and Telecommunication,
MITCOE Pune 411038, India.
Email: suchitra.khoje@mitcoe.edu.in

Abstract—**This paper presents a versatile approach to implement BPSK (Binary Phase Shift Keying), QPSK (Quadrature Phase Shift Keying), FSK (Frequency Shift Keying), QFSK (Quadrature Frequency Shift Keying), M-PSK (M-ary Phase Shift Keying) and M-FSK (M-ary Frequency Shift Keying) on a Zynq 7000 All Programmable SoC which can find application in SDR (Software Defined Radio). The designs are implemented using Verilog HDL (Hardware Description Language) and the tool used is Vivado IDE (Integrated Development Environment) 2017.4 by Xilinx. The system follows a modular approach to make the design easy to modify and reuse. The design approach followed is a top-down portraying the primary modules and then moving into details of each module implementation. The Sine wave is generated using a look-up-table and the phase or frequency values of the sine wave produced is dependent upon the modulating message bits. With symbol sample frequency selection input the design is able to switch the rate at which the input message symbols are sampled facilitating the design to switch between message symbol sampling rate on the fly. These features of the design can find potential applications in SDR (Software Defined Radio).**

Keywords-**QPSK; M-PSK; QFSK; M-FSK; Zynq-7000 AP SoC; Vivado; SDR.**

I. Introduction

With increased application of digital communication techniques in fields like IoT, satellite communication, satellite broadcasting, DTH (Direct To Home) services etc. and advent in SDR techniques, the need for designing software defined or partially reconfigurable modulation and demodulation schemes has become a priority. In an SDR the properties like modulation, network access, carrier frequency and signal bandwidth are defined by software. This term software-defined radio was previously known as Software radio and was first introduced by prof. Joseph Mitola [6]. After the development of a large number of standards for wireless communication networks using SDR, the advantages and benefits of SDR transceiver become evident. At the operator's end, the terminal equipment is supposed to support multiple specifications simultaneously, this poses a challenge to design a transceiver system that supports multiple encryptions, modulation and signal processing. Transceiver systems implemented with SDR system presents a solution to these problems and challenges.

Reconfigurability or partial reconfigurability of the FPGA's (Field Programmable Gate Arrays) or DSP's (Digital Signal Processors) makes them a suitable choice to implement SDR systems. The AP SoC's like Zynq-7000 provided by Xilinx comprising Dual-core ARM Cortex-A9 MPCore and Artix-7 or Kintex-7 based programmable logic which is powerful enough to implement the entire SDR systems with remarkable flexibility. If SDR systems are implemented on an AP SoC like Zynq-7000 with Hardware-Software co-design approach the entire chain of multiple encryption-decryption, modulation-demodulation and signal processing techniques can be implemented on a single SoC utilizing the flexibility of software and reconfigurable hardware along with the reliability of the hardware.

For effective transmission of information wireless communication systems demand high data rate. Modulation techniques are introduced to increase data transmission and reception efficiency within the same bandwidth. QPSK is the most commonly used digital modulation technique which changes the phase according to the baseband data while keeping the frequency and amplitude unchanged [7]. In FSK frequency of the high-frequency carrier changes according to the baseband data while phase and amplitude remain unchanged.

In QPSK two input data bits are represented with four discrete phase variations in the carrier signal as shown in TABLE I [1]. The first bit represent the in-phase (I) components and the second bit represent the quadrature (Q) components [3].

$$S_{QPSK}(t) = d_I(t)cos(2\pi f_0 t) - d_Q(t)sin(2\pi f_0 t) \quad (1)$$

$$d_I(t) = \sqrt{\frac{2E}{T}}cos\left[(2i-1)\frac{\pi}{4}\right] \quad (2)$$

$$d_Q(t) = \sqrt{\frac{2E}{T}}sin\left[(2i-1)\frac{\pi}{4}\right] \quad (3)$$

Where E is the energy per symbol, $i = 1, 2, 3, 4, 0 \leq t \leq T$.

TABLE I
QPSK PHASE VALUES FOR DIFFERENT INPUT BIT COMBINATION

Input Bits	Phase
00	225°
01	315°
10	135°
11	45°

QPSK modulation is formed from two distinct BPSK signals

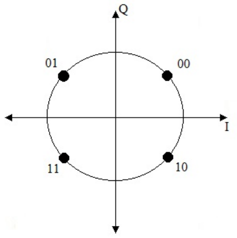

Fig. 1. *Constellation Diagram For QPSK*

combined into one. However, in QPSK the data transmission is enhanced while the BER (Bit Error Rate) over the SNR (Signal to Noise Ratio) is maintained at the same level with the original BPSK [10]. The symbol period of QPSK is 2× that of bit period, $T_s = T_b$. In [11] impact of the phase noise on the performance of digital communication system is studied. As described in [2] implementation of QPSK signal in full digital domain has the potential to save cost for long-term as well as increase wireless data immunity over the environmental noise.

FSK modulation changes the frequency of the high-frequency carrier according to the baseband data while keeping the amplitude and phase unchanged. When compared with QPSK, QFSK is capable of transmitting the baseband signal at a comparatively higher data rate but at the cost of increased power consumption. Thus for applications that demand higher transmission rate and power consumption is not a concern QAM is preferred.

This paper is arranged in six *Sections*. *Section I* has highlighted the importance and emergence of completely digital modulation techniques from the standpoint of SDR. The architecture and general model of these completely digital schemes are described in *Section II* that is followed by the architecture of models that we have implemented in *Section III* where different design versions and their features are also enlisted. Design matrix and version specific design details are covered in *Section IV*. Furthermore, the simulation and implementation results are included in *Section V* for few selected test cases and designs. Lastly, in *Section VI*, the designs are compared for their resource utilization and conclusions are drawn.

II. The Architecture of Completely Digital PSK and FSK designs

DDS (Direct Digital Synthesis) is the technology of frequency synthesis developed as the third generation of frequency synthesis technology after Direct and Indirect Frequency Synthesis. DDS has gained popularity as the method of generating sinusoidal signals and modulated signals in digital systems [12]. Remaining part of this section describes the general block diagram of DDS system and a slit variant of DDS system that we have designed to implement PSK and ASK modulation schemes.

A. DDS System General Model

In the simplest case, a DDS can be constructed using a ROM that has one or many sine samples stored in it, inside sine LUT (Look-up table) and it was introduced in [4][9]. Fig.2 shows the block diagram of DDS system which at its output produces sinusoidal signals at given frequency by integrating higher clock frequency digitally [8]. The frequency of the sine wave to be produced is set using the FSW (Frequency Setting Word) which is an input to the phase accumulator block. The phase accumulator is then used to ”calculate” the successive addresses of the sine look-up table which generates a digital sine-wave output. The step is dependent on the FSW. The DDS translates the resulting phase to a sinusoidal waveform via the look-up table and converts the digital representation of the sine-wave to analog form using a Digital-to-Analog converter followed by LPF (Low Pass Filter). The digital part of the DDS, i.e. the phase accumulator and the LUT, is usually called a Numerically Controlled Oscillator (NCO). The frequency of the output signal for an M bit system is determined by the following equation:

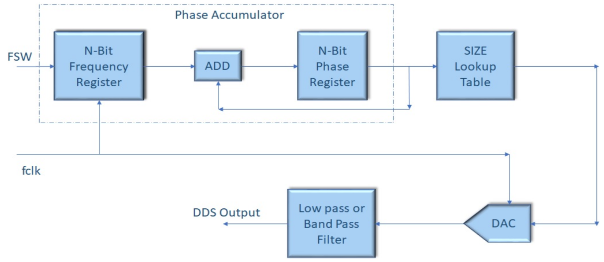

Fig. 2. *DSS System-Block Diagram*

$$f_0 = \frac{K \times f_{clk}}{2^M} \tag{4}$$

Where K is the FSW, M is the number of bits that the phase accumulator can handle and f_{clk} is the generator clock frequency in Hz.

B. Implemented PSK model

To generate PSK slight modifications are done in the general DDS model as shown in Fig.3.

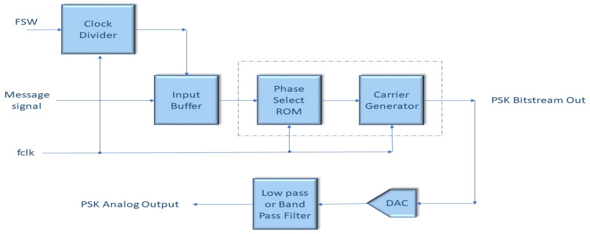

Fig. 3. *PSK System-Block Diagram*

Message signal or Baseband digital information bits are input to the Input Buffer block that performs two primary functions:

- Sample the in-phase and quadrature phase message bits.

- Operates at sampling frequency decided by Clock Divider.

Carrier Generate blocks perform a very similar operation to that of a Phase accumulator block shown in DDS and it also includes the size look up table. N-Bit phase register provides the phase jump word to the Carrier generator block according to the PSK scheme being implemented. For BSK the Phase Select block will provide two distinct phase select words according to input message sequence, whereas in case of QPSK it will provide four distinct phase select words according to the combination of input message sequence.

C. Implemented FSK model

FSK system block diagram is shown in Fig.4 and it is quite similar to that of the PSK system Fig.3. There is a slight difference in implementation of the frequency select ROM and the Carrier Generator blocks. Clock Divider and input buffer perform the same functions as explained in *Section II-B*. Frequency Select ROM generates the FSW's in accordance with the input message sequence. These FSW's intern varies the frequency of the generated sinusoidal wave producing the FSK Bit-stream Out. The bit-stream is a 16-Bit word that drives the DAC to finally produce the FSK analog output. The Low pass or Bandpass filter filters out the noisy spikes i.e. high-frequency components.

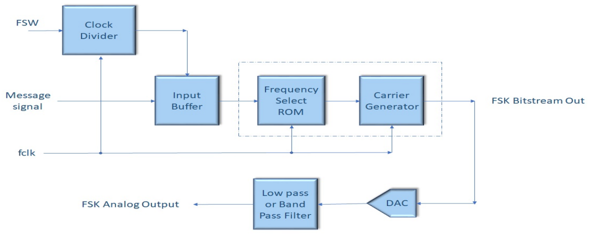

Fig. 4. *FSK System-Block Diagram*

III. Architecture Implemented on FPGA

For BPSK and BFSK we have implemented only one Version i.e. Revision 1 whereas for every design of QPSK and QFSK we have implemented three versions namely Rev. 1 (Revision 1), Rev. 2 (Revision 2) and Rev. 3 (Revision 3) respectively. Every version is implemented with slight modification and the impact of these modifications are studied based on comparison between their resource Used and power consumption. These three revisions have following highlighting features and modifications:

TABLE II
QPSK phase values for different input bit combination

Revision	Features
Rev. 1	Clock Divider is implemented using D-FF chain. Design is implemented for single sampling frequency.
Rev. 2	Clock Divider is implemented using Counter and FSM. Design is implemented for single sampling frequency.
Rev. 3	Clock Divider is implemented using Counter and FSM. Design is implemented for **Eight** distinct Sampling frequencies that **can be changed on the-fly**.

This feature of changing the sampling frequency of the message signal on-the-fly can be quite useful in SDR's which have multiple message signals with different baud rate. These designs can find application in SoC based implementations [5].

A. The Architecture of PSK Models implemented on FPGA

Design of BPSK and QPSK is quite similar except for the clock divider block. BPSK design does not use the clock divider block as there are only two symbols logic-1 and logic-0 that modulates a high-frequency sinusoidal carrier by 180° phase shift differentiating these logic symbols.

1) BPSK Model Architecture: The RTL (Register Transfer Logic) schematics of the BPSK model is depicted in Fig.5. $SIPO$ (Serial In Parallel Out) buffer samples the change in the message signal and produces a control signal to select a corresponding $phasedata_{out}$ from the $PhROM$. The $PhROM$ (Phase ROM) holds the phase select word that corresponds to the message signal. Carrier generator has 256 locations each having a 16 Bit sine wave values that produce the sinusoidal carrier wave controlled by the $phasedata$ input. The output $bpsk_{out}$ can be given to the DAC and filter circuit to produce the smooth analog BPSK signal.

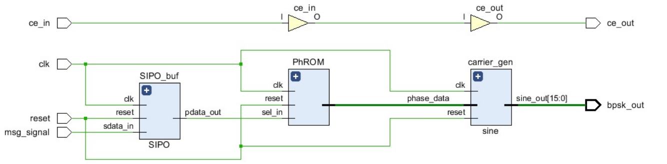

Fig. 5. *BPSK RTL schematic*

2) QPSK Model Architecture: The Fig.6 shows the RTL schematic of QPSK model which is similar to the Fig.5 whereas, in addition, it has a $MessageClockGenerator$ block that samples the input message symbols. Rest of the blocks perform the same functions that are explained in the *Section A. 1.*

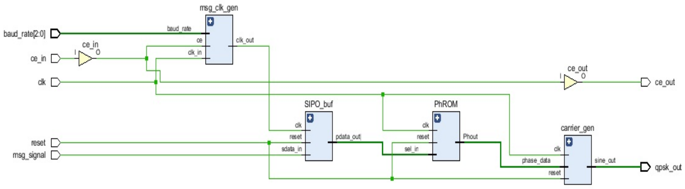

Fig. 6. *QPSK RTL schematic*

$PhaseROM$ has four phase select control words corresponding to four different message symbol. For all three implemented versions of QPSK, the architecture is same only changing the internal implementation of the

$MessageClock_{generator}$ block according to the revision implemented.

B. The Architecture of FSK Models implemented on FPGA

BFSK and QFSK designs are similar except few differences in clock divider block. There is no clock divider block in BFSK because of two logic symbols.

1) BFSK Model Architecture: Fig.7 shows BFSK RTL schematic where *SIPO* performs the same function as it performs in BPSK Fig.5 and selects the corresponding frequency select word from the $frequencyROM$ that drives the carrier generator block to produce $bfsk_{out}$ having two distinct frequencies for two message symbols.

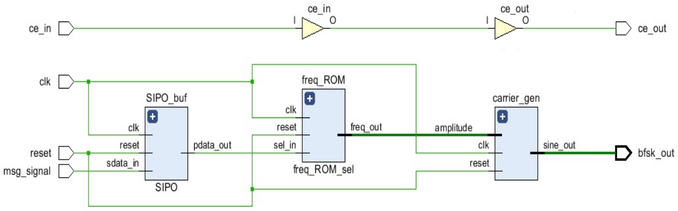

Fig. 7. *BFSK RTL schematic*

2) QFSK Model Architecture: Fig.8 shows QFSK RTL schematic that is quite similar to the Fig.6 where instead of the $PhaseROM$ a $FrequencyROM$ is implemented that generates frequency select word to change the frequency of the generated sinusoidal carrier in accordance with the input message symbols. $MessageClockGenerator$ block generates symbol sampling rate and as per the revision of implemented design, it has different features to either have a fixed rate or on-the-fly programmable symbol rates.

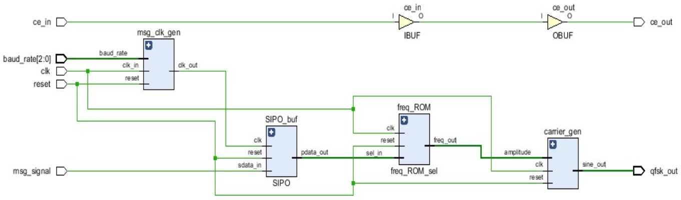

Fig. 8. *QFSK RTL schematic*

IV. Design Implementation on FPGA

Design matrix considered while designing the systems are enlisted in TABLE III.

TABLE III
DESIGN MATRIX

Parameter	Description	Unit
f_{system}	system or board frequency	Hz
$f_{carrier}$	carrier or digital sine wave frequency	Hz
R_b	Bit rate	bps
R_s	Symbol rate	bps
$B.W$	Bandwidth	Hz
E_b	Energy per Bit	W

A. Implemented PSK Model Design Specifications

BPSK and QPSK modulators are designed for the following specifications and their results are evaluated by simulating each version of design in Xilinx Vivado IDE 2017.4 then implemented on Zybo Z7-20 Zynq-7000 FPGA evaluation board.

1) Implemented BPSK Design Specifications: TABLE IV enlists the specifications of BPSK design implemented.

TABLE IV
BPSK DESIGN SPECIFICATIONS

Parameters	f_{system} (MHz)	$f_{carrier}$ (KHz)	R_b (max) $(Kbps)$	$B.W$ (KHz)	E_b (μW)
BPSK	125	488	244	488	22.315

2) Implemented QPSK Design Specifications: TABLE V enlists the specifications of QPSK design implemented.

TABLE V
QPSK DESIGN SPECIFICATIONS

Parameters **Revision**	*Message Sampling options (Hex)*	f_{system} (MHz)	$f_{carrier}$ (KHz)	R_b $(Kbps)$	R_s $(Kbps)$	$B.W$ (KHz)	E_b (μW)
$Rev.1$	-	125	488	488	244	488	11.157
$Rev.2$	-	125	488	488	244	488	11.157
$Rev.3$	Default	125	488	484.49	242.248	484.49	11.238
	1	125	488	243.19	121.595	243.19	22.389
	2	125	488	121.83	60.915	121.83	44.693
	3	125	488	60.97	30.485	60.97	89.306
	4	125	488	30.50	15.250	30.50	178.524
	5	125	488	15.25	7.625	15.25	357.049
	6	125	488	7.62	3.810	7.62	714.566
	7	125	488	3.81	1.905	3.81	1429.133

B. Implemented FSK Model Design Specifications

Following are the specifications and their results evaluated by simulating each version of the design in Xilinx Vivado IDE 2017.4 then implemented on Zybo Z7-20 Zynq-7000 FPGA evaluation board for BFSK and QFSK modulators we have designed.

1) Implemented BFSK Design Specifications: TABLE VI enlists the specifications of BFSK design implemented.

TABLE VI
BFSK DESIGN SPECIFICATIONS

Parameters	f_{system} (MHz)	$f_{1carrier}$ (KHz)	$f_{2carrier}$ (KHz)	R_b (max) $(Kbps)$	$B.W$ (MHz)	E_b (μW)
BFSK	125	1953.125	976.562	488	1.464563	11.157

2) Implemented QFSK Design Specifications: TABLE VII enlists the specifications of QFSK design implemented.

TABLE VII
QFSK DESIGN SPECIFICATIONS

Parameters **Revision**	*Message Sampling options (Hex)*	f_{system} (MHz)	$f_{1carrier}$ (KHz)	$f_{2carrier}$ (KHz)	$f_{3carrier}$ (KHz)	$f_{4carrier}$ (KHz)	R_b $(Kbps)$	R_s $(Kbps)$	$B.W$ (KHz)	E_b (μW)
$Rev.1$	-	125	488	984	1470	1953	1953.125	973.56	5371.25	2.787
$Rev.2$	-	125	488	984	1470	1953	1953.125	973.56	5371.25	2.787
$Rev.3$	Default	125	488	984	1470	1953	968.98	484.49	3402.96	5.619
	1	125	488	984	1470	1953	486.38	243.190	2437.76	11.194
	2	125	488	984	1470	1953	243.664	121.832	1952.328	22.346
	3	125	488	984	1470	1953	121.95	60.975	1708.9	44.649
	4	125	488	984	1470	1953	61.004	30.502	1587.008	89.256
	5	125	488	984	1470	1953	30.51	15.255	1526.02	178.46
	6	125	488	984	1470	1953	15.256	7.628	1495.512	356.908
	7	125	488	984	1470	1953	7.628	3.814	1480.256	713.817

V. Simulation And Implementation Results

Designed PSK and FSK models are simulated in Vivado IDE 2017.4 to verify their behavior and functionality. Behavioral simulations are performed in the initial stage of designing to verify the correct operation of the design and perform timing analysis. In the later stage of designing prior to implementation of the development board, functional simulations are performed to verify critical timing constrains post-implementation.

Finally the designs are implemented on the Zybo Z7-20 development board by Digilent that has a Xilinx Zynq-7000 series FPGA. The output from the board is observed on the logic analyzer. The snapshots of simulation and implemented waveform observed on logic analyzer are included in *section* below.

A. PSK Simulation and Implementation results

In the first part of this *section* BPSK simulation and it's implementation waveforms are included whereas in second part simulation and implementation waveforms for QPSK are presented:

1) BPSK Simulation and Implementation results: For the implemented BPSK design simulation is shown in Fig.9. The binary input message is a digital wave of $100KHz$ with 70% duty cycle.

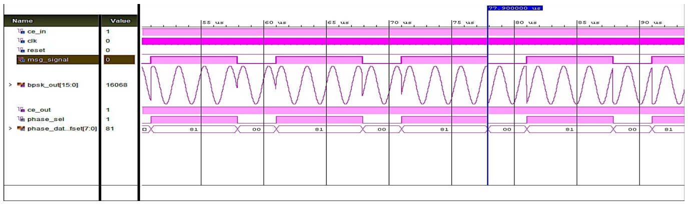

Fig. 9. *BPSK Simulation*

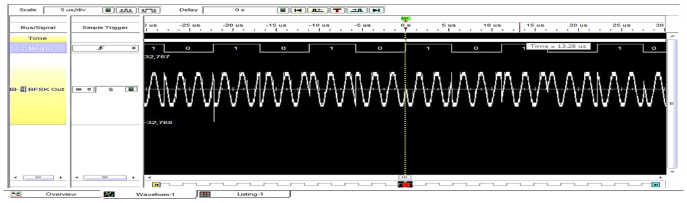

Fig. 10. *BPSK Implemented Output Waveform*

Output waveform of the implemented BPSK design on Xilinx Zynq-7000 is shown in Fig.10 where input message wave is generated by a function generator having $120KHz$ frequency and 50% duty cycle.

2) QPSK Simulation and Implementation results: As seen in Fig.11 message sampling option is 7 i.e. default so the R_b*(max)* is 484.49 *Kbps*. There are all possible symbol combinations that are evaluated and each has unique phase representation. We have implemented all the Revisions of the QPSK design on Xilinx Zynq-7000 and observed the output waveforms for them which are working satisfactorily as per

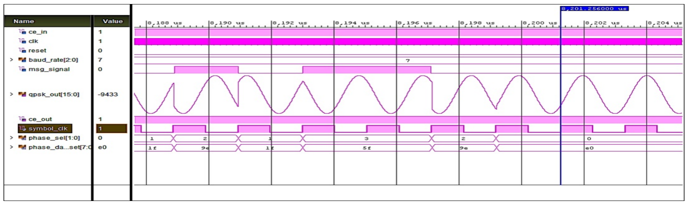

Fig. 11. *QPSK Simulation*

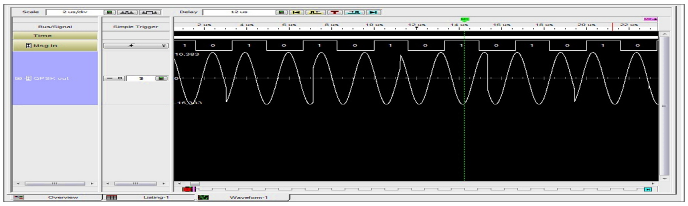

Fig. 12. *QPSK-Rev.1 Implemented Output Waveform at 300KHz*

the design. One such output waveform for *Rev.1* with message signal at *300KHz* is shown in Fig.12.

B. FSK Simulation and Implementation results

This *Section* has two parts that include simulation and implemented waveform for our BFSK and QFSK designs.

1) BFSK Simulation and Implementation results: Fig.13 is a snapshot of BFSK module simulated for message wave of frequency $100KHz$ with a duty cycle of 60%.

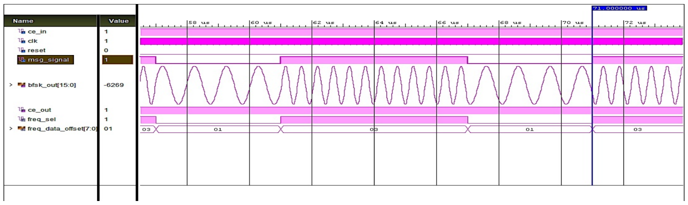

Fig. 13. *BFSK Simulation*

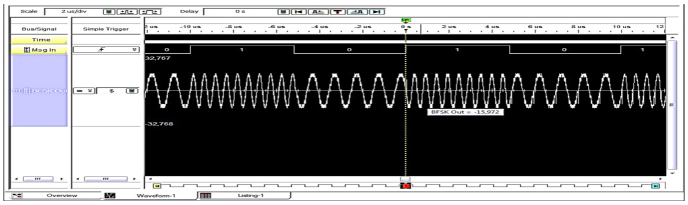

Fig. 14. *BFSK Implemented Output Waveform*

The output waveform of the implemented BFSK design on Xilinx Zynq-7000 is shown in Fig.14 where input message wave is generated by a function generator having $100KHz$ frequency and 50% duty cycle.

2) QFSK Simulation and Implementation results: As seen in Fig.15 message sampling option is 1 so the R_b*(max)* is 486.38 *Kbps* and the message wave has frequency of $30.003KHz$ with duty cycle of 44%.

TABLE VIII
RESOURCES UTILIZED BY ALL DESIGNS

Resource	**Available**	**BPSK**		**QPSK**						**BFSK**		**QFSK**					
		Rev.1		*Rev.1*		*Rev.2*		*Rev.3*		*Rev.1*		*Rev.1*		*Rev.2*		*Rev.3*	
		Used	Used %	Used	Used %	Used	Used %	Used	Used %	Used	Used %	Used	Used %	Used	Used %	Used	Used %
LUT	53200	9	0.02	21	0.04	15	0.03	68	0.13	10	0.02	27	0.05	57	0.11	67	0.13
FF	106400	10	0.01	33	0.03	17	0.02	48	0.05	10	0.01	34	0.03	45	0.04	45	0.04
BRAM	140	0.50	0.36	0.50	0.36	0.50	0.36	0.50	0.36	0.50	0.36	0.50	0.36	0.50	0.36	0.50	0.36
IO	125	21	16.80	21	16.80	21	16.80	24	19.20	21	16.80	21	16.80	21	16.80	24	19.20
BUFG	32	1	3.13	1	3.13	1	3.13	2	6.25	1	3.13	1	3.13	1	3.13	2	6.25

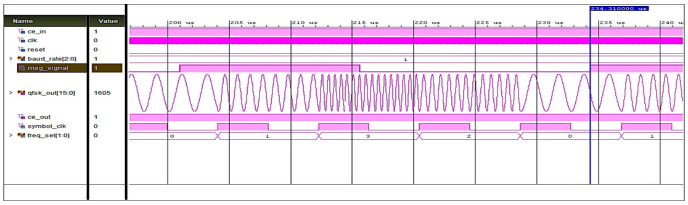

Fig. 15. *QFSK Simulation*

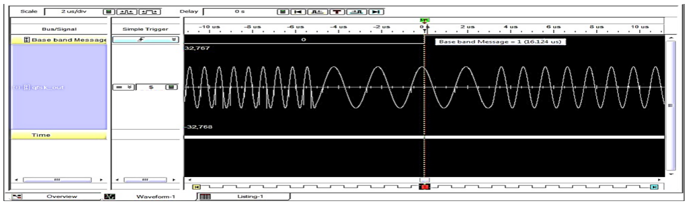

Fig. 16. *QPSK-Rev.3 Implemented Output Waveform at 30KHz*

There are all possible symbol combinations that are evaluated and each has unique phase representation. We have implemented all the Revisions of the QFSK design on Xilinx Zynq-7000 and observed the output waveforms for them which are working satisfactorily as per the design. One such output waveform for *Rev.3* with message signal at *30KHz* is shown in Fig.16.

VI. CONCLUSION

This paper presents a new highly versatile programmable symbol sampling rate circuit for BPSK, QPSK, BFSK and QFSK modulation which can be scaled for more generic M-ary PSK and M-ary FSK. This allows the modulator circuit to switch the symbol sampling rate on-the-fly. These circuits are designed in Verilog HDL language and implemented on Xilinx Zynq-7000 xc7z020clg400-1 device. Its versatility was tested by changing the symbol sampling rate at runtime.

TABLE VIII enlists resource utilization for each design and their respective implemented versions. Where it is observed that the modulator designed for same symbol sampling frequency (see QPSK), one using *Rev.1* features utilize more resources when compared to the design implemented for *Rev.2*. The functionalities of *Rev.1* and *Rev.2* are similar but there is a difference in design approach of the Clock Divider block that has enabled savings in resource utilization which in turn reduces power consumption.

If there is a difference in message sampling frequency (see QFSK) of the implemented design versions then resource utilization is inversely proportional to the implemented message sampling frequency. It is observed that among *Rev.1* and *Rev.2* resource utilization of the design implementing lower message sampling rate will be significantly higher compared to the design that implements a high sampling rate design (See QFSK).

QFSK *Rev.1* has implemented high symbol sampling rate design and QFSK *Rev.2* has implemented a low symbol sampling rate design thus *Rev.1* utilizes less resources compared to *Rev.2*.

The functionality to change the message symbol sampling rate on-the-fly is provided by *Rev.3* which utilizes the maximum amount of resources among the three revisions for both QPSK and QFSK. Thus it is concluded that the increased functionality comes at the cost of increased resource utilization and power consumption.

From an application point of view, this design can be potentially implemented in SDR radios.

REFERENCES

[1] K Anitha, BK Sujatha, et al. Fpga implementation of high throughput digital qpsk modulator using verilog hdl. *International Journal of Advanced Computer Research*, 4(1):217, 2014.

[2] Gihad Elamary, Graeme Chester, and Jeffrey Neasham. A simple digital vhdl qpsk modulator designed using cpld/fpgas for biomedical devices applications. In *Proceedings of the World Congress on Engineering*, volume 1, pages 978–988, 2009.

[3] Tarik Kazaz, Merima Kulin, and Mesud Hadzialic. Design and implementation of sdr based qpsk modulator on fpga. In *Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on*, pages 513–518. IEEE, 2013.

[4] CS Koukourlis, JE Plevridis, and JN Sahalos. A new digital implementation of the rds in the fm stereo. *IEEE transactions on broadcasting*, 42(4):323–327, 1996.

[5] KA Arun Kumar. Soc implementation of a modulation classification module for cognitive radios. In *Communication Systems and Networks (ComNet), International Conference on*, pages 87–92. IEEE, 2016.

[6] J. Mitola. Software radios-survey, critical evaluation and future directions. In *[Proceedings] NTC-92: National Telesystems Conference*, pages 13/15–13/23, May 1992.

[7] SO Popescu, AS Gontean, and D Ianchis. Qpsk modulator on fpga. In *Intelligent Systems and Informatics (SISY),*

2011 IEEE 9th International Symposium on, pages 359–364. IEEE, 2011.

[8] Wenmiao Song and Qiongqiong Yao. Design and implement of qpsk modem based on fpga. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, volume 9, pages 599–601. IEEE, 2010.

[9] Wenmiao Song, Jingying Zhang, and Qiongqiong Yao. Design and implement of bpsk modulator and demodulator based on modern dsp technology. In *Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, 2009 3rd IEEE International Symposium on*, pages 1135–1137. IEEE, 2009.

[10] D. Taggart and R. Kumar. Impact of phase noise on the performance of the qpsk modulated signal. In *2011 Aerospace Conference*, pages 1–10, March 2011.

[11] David Taggart and Rajendra Kumar. Impact of phase noise on the performance of the qpsk modulated signal. In *Aerospace Conference, 2011 IEEE*, pages 1–10. IEEE, 2011.

[12] Matthew L Welborn. Direct waveform synthesis for software radios. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, volume 1, pages 211–215. IEEE, 1999.