LUT-Based Efficient Impulse Shaping for Direct Synthesizing Digital Communication Signals at Arbitrary Symbol Rate

Ziyao Liu[1](✉), Zhijie Wang[2], Jun Wang[2], Di Huang[1], and Nangen Zhang[1]

[1] School of Information and Electronics,
Beijing Institute of Technology, Beijing, China
liuziyaochn@163.com, huangdibit@163.com,
zhangnangen@sina.com

[2] Institute of Telecommunication Satellite,
China Academy of Space Technology, Beijing, China
1554264456@qq.com, bitwj@163.com

Abstract. In this paper, we present a LUT-based efficient impulse shaping for direct synthesis of digital communication signal at arbitrary symbol rate. Compared with general approaches where sampling clock changes according to the symbol rate thus signal quality degrade as result of a complex analog hardware structure, or involving a fractional interpolation which consumes considerable computational resource, this new approach allows FPGA to synthesis variable high data rata signal with high quality directly at a fixed sampling rate which simplify the hardware structure and saves computational resource consumption. With some little modifications, the presented scheme could be easily adapted to 8PSK, 16QAM and other arbitrary amplitude-phase-modulation constellations. A hardware prototype has been built to verify the presented algorithms. In particular, we have achieved 4.8 Gsps parallel impulse shaping which supports input symbol rate ranging from 100 Ksps to 600 Msps.

Keywords: LUT-based · Impulse shaping · Direct synthesis
Variable symbol rate

1 Introduction

Producing digital communication signals such as the BPSK, QPSK, 8PSK or 16QAM signals at variable symbol rate is one of the fundamental functions provided by the state-of-the-art Vector Signal Generators (VSG) or Arbitrary Waveform Generators (AWG). For example, the famous KeySight™ (formerly known as Agilent) VSG E8267D is capable of generating QPSK signals with the symbol rate ranging from 100 Ksps to 50 Msps (when using basic kit) [1]. The more advanced Zodiac™ Cortex High Data Receiver XXL (when used as a simulator) may generate digital communication waveforms with various constellation patterns at the symbol rate of 1 Ksps–600 Msps [2]. Besides its application to advanced testing equipment, real-time synthesis of variable rate communication signals is also one of the enabling techniques for various Software Defined Radio (SDR) systems [3, 4].

B. Li et al. (Eds.): ChinaCom 2017, LNICST 237, pp. 215–226, 2018.
https://doi.org/10.1007/978-3-319-78139-6_23

To further elaborate on the challenges and solutions for variable symbol rate signal generation, let us examine the conceptual diagram of a simplest BPSK transmitter in Fig. 1. As illustrated in the figure, before modulating the Local Oscillator (LO), the baseband Non-Return-to-Zero (NRZ) signal is generated by passing the information carrying pulse-train through a pulse shaping filter, which usually has a Squared Root Raised Cosine (SRRC) function as its unit impulse response, to eliminate both out-of-band power leakage and Inter-Symbol Interferences (ISI) [5]. On most SDR platforms, pulse shaping operation is carried out in the digital domain by interpolating a symbol-rate binary pulse train with a digital SRRC Low-Pass Filter. The interpolation rate is often an integral to avoid fractional rate resampling. Due to the constraint of integral interpolation rate, a following quadrature (I/Q) modulator becomes a prevailing choice [6, 7].

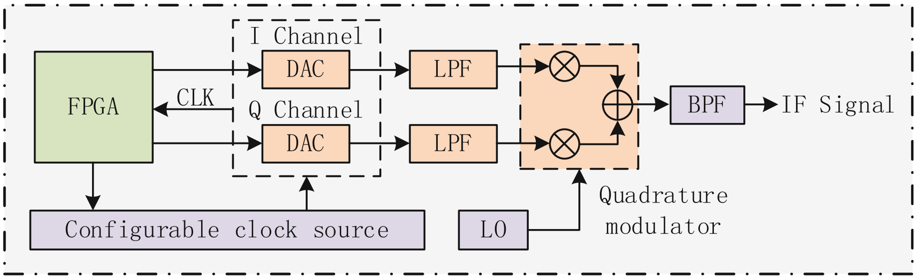


Fig. 1. Conceptual diagram of a simplest BPSK transmitter with integral rate sampling

As can be seen in the figure, the Field Programmable Gate Array (FPGA) and the double-channel DACs are driven by a reconfigurable clock f_{clk}, which satisfies $f_{\text{clk}} = k \cdot R_s$, where R_s is the symbol rate, and k is a positive integral. When the transmitter needs to work on a new symbol rate, it may simply change f_{clk}. As f_{clk} is independent of the RF or IF center frequency f_c, the topology of Fig. 1 allows for great flexibility to support variable symbol rate [8].

However, as modulation is accomplished in the analog domain in Fig. 1, the quality of the IF or RF output will inevitably suffer from the amplitude imbalance between the I/Q channels, as well as the non-orthogonality of the cos & sin LO pair. A more integrated, "Direct Synthesizing" transmitter solves this problem by modulating the digital LO waveforms directly inside the FPGA, and then generate IF or RF output by a single-channel DAC, as can be seen in Fig. 2. Direct synthesis complies with the general SDR principle of "placing the DAC or ADC as close to the RF front-end as possible". Besides improved output quality, it also saves cost, volume and power consumption as it cancels a DAC channel, a stand-alone analog modulator and a reconfigurable clock source.

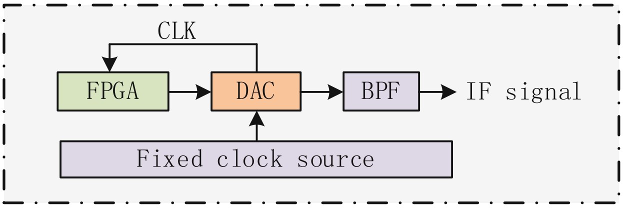


Fig. 2. Hardware diagram of the “Direct Synthesizing” transmitter

The challenge of direct synthesis when applied to variable symbol rate signal generation is, the FPGA and DAC’s driving clock has now to be determined by the output frequency f_c, and usually it is no longer an integral multiple of the symbol rate R_s, hence fractional resampling becomes a requisite in this case which is shown in Fig. 3.

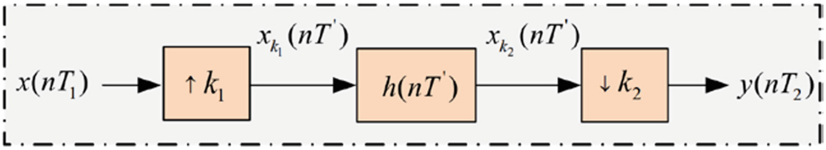


Fig. 3. A common structure of fractional resampling

Suppose $f_{clk}/R_s = k_1/k_2$, fractionally resampling a symbol stream at the rate of R_s into a digital baseband waveform at the sampling of f_{clk} means first interpolating by k_1 times with the pulse shaping filter, and then decimating the output by k_2 times. The computational overhead incurred by this process soon becomes formidable as R_s or f_c increases. Even worse, the “decimate after interpolate” method cannot gives us a unified implementation framework as the interpolation and decimation rate, namely k_1 and k_2, vary with R_s.

The motivation of this paper is to develop an efficient pulse shaping algorithm that makes direct synthesis practical for variable rate signal generation. As a low-complexity alternative to the on-line interpolating & decimating computation, a Look-Up-Table (LUT) based fractional resampling scheme is presented, which almost only relies on reading the content out from the proper address in a pre-defined memory space. We demonstrate the effectiveness of our pulse-shaping algorithm by directly synthesizing a 100 Ksps–600 Msps SRRC pulse-shaping QPSK signal at an IF center frequency of $f_c = 1.2\,\text{GHz}$, using a DAC (MD622H) working at 4.8 GHz and a FPGA (Xilinx XC7VX690T) with clock rate 150 MHz (×32 parallel channels). It is observed the transmitter only takes 12% of the logic, computational and storage of the targeted FPGA. The output signal is examined by a vector signal analyzer. Both the spectrum and the Error-Vector-Magnitude (EVM) results are provided. With some minor modifications, our scheme can also be applied to other more complex constellations such as QPSK, 8PSK or 16QAM or other pulse-shaping functions than SRRC waveforms.

The remainder of the paper is organized as follow. We first describe a diagram of the "Direct Synthesizing" transmitter and illustrate the model of the LUT-based impulse shaping in Sect. 2.1. We dedicate Sect. 2.2 to the derivation of the relationship between LUT address updating and impulse shaping at variable symbol rate, based on Direct Digital Synthesis (DDS) and then present the algorithm in detail (Sect. 2.3). Next, we investigate the computational complexity of the proposed algorithm and introduce our hardware experiment platform in Sect. 3. The test result, such as spectrum and EVM, obtained by hardware test are also given in the last section (Sect. 3), followed by conclusion.

2 Proposed Scheme

2.1 Direct-Synthesizing Transmitter

As can be seen in the topology of "Direct Synthesizing" transmitter shown in Fig. 4, the parallel digital up conversion and Oserdes & FIFO is based on a conventional parallel DDS structure, which is out of this paper's scope. In the sequel we focus on is the parallel impulse shaping module whose schematic diagram is illustrated in Fig. 5. Besides, as shown in Fig. 4, Data stream (a PN sequence) provided by Data source module in a parallel way will be constellation-mapped in accordance to the modulation type thus with little modification of mapping pattern, this "Direct Synthesizing" transmitter could be applied to arbitrary amplitude-phase-modulation constellations, such as BPSK, QPSK, or 16QAM.

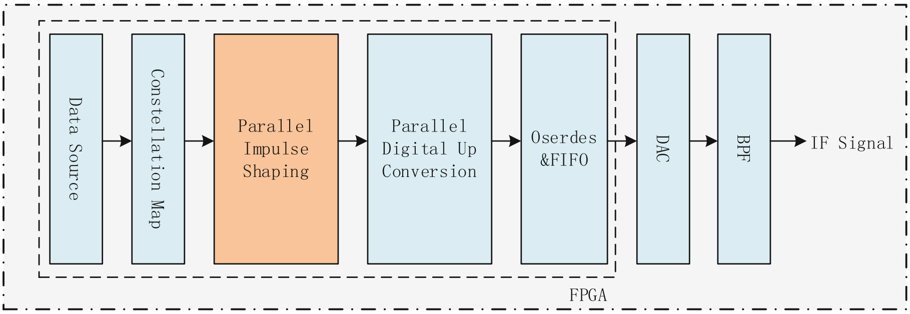


Fig. 4. Topology of "Direct Synthesizing" transmitter

As shown in Fig. 5, the schematic diagram is one channel of parallel impulse shaping module which simulate a convolution process involving 6 symbols (from Data ROM) and coefficients (from LU1 to LUT6) in the same amount. The reason why we assign 6 to the number of symbols involved in the convolution process will be explained in the next part. Therefore, our problem boils down to find the proper address of Data ROM and LUTs to obtain the proper data and coefficients.

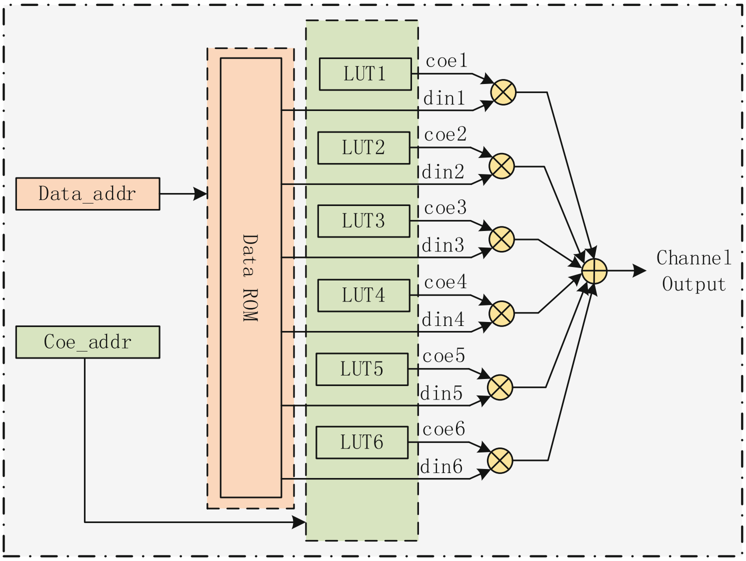


Fig. 5. Schematic diagram of the impulse shaping in one channel

2.2 Relationship Between LUT Address Updating and Impulse Shaping

In order to find the proper address of Data ROM and LUTs, we propose an algorithm updating the address on the basis of DDS theory. It is well known that classic DDS technique is capable of generating arbitrary-frequency sinusoid wave at a fixed sampling rate, given the frequency of the generated sinusoid is below 1/3 or 1/4 of the sampling rate. The structure of a classic DDS is shown in Fig. 6.

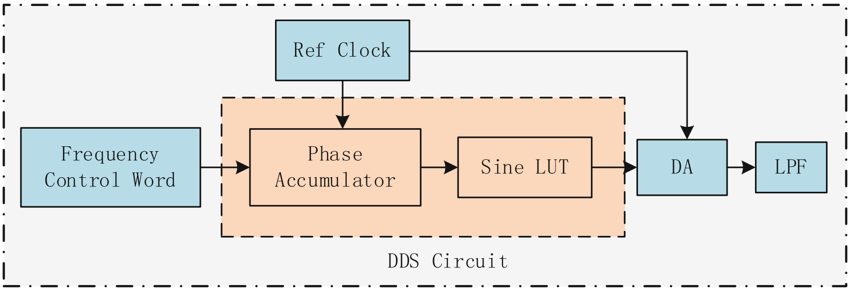


Fig. 6. A common DDS structure

As Fig. 6 shows, the sum of the Frequency Control Word (FCW) and phase data at last clock will be output from Phase Accumulator as look-up-table address to next part thus waveform could be generated from Sine LUT continuous output.

Considering the generation of digital I/Q baseband waveforms often relies on a pulse-shaping filter, or an interpolator when the output sampling frequency is integral multiples, i.e. $F_s = N * R_s$, where N $\in N^+$. But when $F_s \neq N * R_s$, as in the common case, producing baseband waveforms is often regarded as difficult. As in this case, a fractional resampling has to be adapted, and more particularly, if $F_s/R_{\mathrm{s}} = k_1/k_2$, then a k_1 times interpolation will be followed by a k_2 times decimation to implement a k_1/k_2 resampling which bring about unaffordable computational consumption.

Therefore, we assume that if we can store the result of impulse shaping in a LUT and output it in a regular manner, specifically, based on DDS theory, unaffordable real-time computational resource consumption could be avoidable. Let us examine a SRCC (Square Root Raised Cosine) waveform which shown in Fig. 7.

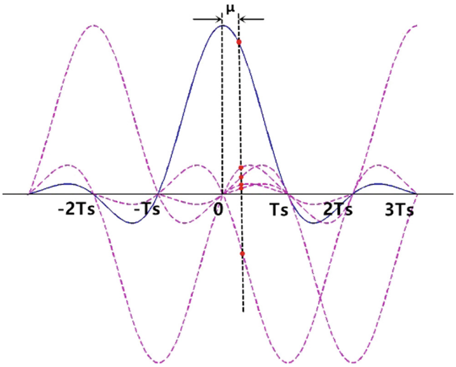


Fig. 7. A SRCC (Square Root Raised Cosine) waveform

It is well known that direct synthesis is on the basis of outputting the pre-stored approximate value of result, which can be expressed as a formula below.

$$\mathrm{y}(\mathrm{k}T_i) = \mathrm{y}[(m_k + \mu_k)T_s] = \sum\nolimits_{i=I_1}^{I_2} x[(m_k - i)T_s]h_I[(\mu_k + i)T_s] \tag{1}$$

$$\approx \sum\nolimits_{i=I_1}^{I_2} x[(m_k - i)T_s]h_I[iT_s] \tag{2}$$

The difference of Eqs. 1 and 2 is the $h_I(t)$, assume the error of them is

$$\upmu = (\mu_k + i)T_s - iT_s = \mu_k T_s \tag{3}$$

where $\upmu \in [0, 1]$ thus $\upmu = \mu_k T_s \leq T_s$. Since T_s is depend on the sampling rate, we can optimizing the interpolation performance by improving it. The initial value $\mathrm{k} = 0$; If the current $\upmu > 1$, then

$$\mathrm{k} = \mathrm{k} + 1 \tag{4}$$

$$\mu = \mathrm{mod}\,[\mu, 1] \tag{5}$$

So if we discretize the waveform in t $\in (0, T_s)$ at a resolution of $T_S/1024$, then a 32*1024-depth (the number of channel is 32) LUT is capable of covering any possible value of μ, at a resolution $T_S/1024$, which is in general sufficient in most practical contexts.

Meanwhile, when we fix the waveform of impulse shaping filter, the number of symbol involving in one clock is determined. Take the waveform of Fig. 7 for instance, the blue curve is the waveform of impulse shaping filter in time domain which has 6 x-intercept, we can observe that pulse shaping result of any point at t $\in (0, T_s)$ is determined by 6 symbol, as the red point of intersection illustrated in Fig. 7, which can be formulated to Eq. 4. This means that every impulse shaping process only need 6 symbols and 6 corresponding coefficients which explain the problem proposed in Part 2.1.

$$\mathrm{y(k)} = \sum\nolimits_{k=-2}^{3} x(k)h\left(t - nT_S^{'}\right) \tag{6}$$

Therefore, we pre-store the result of pulse shaping at very high resolution in LUTs, and take it out according to the symbol rate on the basis of DDS theory, a continuous waveform of impulse shaping could be obtained. A parallel impulse shaping scheme based on our algorithm is given in Fig. 8.

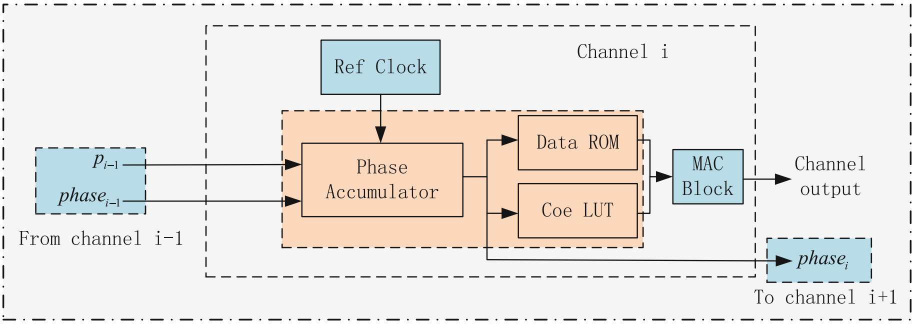


Fig. 8. A parallel impulse shaping scheme based on our algorithm

Besides, since the pulse shaping is based on a LUT process, unaffordable real-time computational resource consumption could be avoidable. This benefit will be observed in detail in Part 3.

2.3 Implementation Steps of LUT-Based Efficient Impulse Shaping

Through the discussion above, implementation steps of LUT-based efficient impulse shaping can be summarized as follows for example.

1. Store PN sequence generated by PN sequence generator in parallel structure into a FIFO. Update the FIFO according to the requirement proposed by address-updating module and output 8bit data into a register with a RFD (ready for data) signal.
2. Generate S LUT storing the coefficient and divide them into T ROM, the depth of ROM is 2^n. In general, consider the influence of side-lobe and computing quantity, S = T = 6, n = 10.
3. Parallel process the wideband signal that divide it into M channel. If the symbol rate is R_s, working clock is F_1, sampling clock is F_s, then M = F_s/F_1, and the symbol control word of each channel can be expressed as

$$P_i = \left\lfloor i * R_s * \frac{M}{F_S} * 2^{32} \right\rfloor \tag{7}$$

Where 2^{32} is the normalization, and $\lfloor\rfloor$ is defined as rounding operation.

4. Update the phase of each channel to obtain the address of ROM to get the data and coefficient for convolution operation. The updating process is defined as below

$$\mathrm{Phase}_i = 2 * 2^{32} \tag{8}$$

$$Phase_i = Phase_{i-1} + P_{i-1}, \mathrm{i} = 2, 3, 4, \cdots, \mathrm{M} \tag{9}$$

Where $Phase_1$ is the phase of 1^{st} channel, $Phase_i$ is the phase of i^{th} channel and it is expressed in 32bit. Denote the data address as Data_addr, coefficient address as Coe_addr, then Data_addr is equal to the highest 4 bit of $Phase_i$ and Coe_addr is equal to the highest n bit of the lowest 32bit of $Phase_i$.

5. According to the address we have obtained from last step, get the data and coefficient and make convolution operation of each channel.
6. Update the phase of the 1st channel as below

$$Phase_1 = Phase_1 + P_{32} \tag{10}$$

7. After $Phase_1$ have updated, generate a RFD signal and go back to the first step.

3 Implementation, Results and Discussion

A LUT-based efficient impulse shaping for direct synthesis of digital communication signal at arbitrary symbol rate is implemented using the algorithm mentioned above. The hardware platform on which the presented algorithm is tested is given below in Fig. 9. At the bottom of the figure is a Compact PCI Express (CPCIe) interface which allows the platform to be connected along a CPCIe bus, and become a part of a host computer. Since the on-board DAC is working at 4.8 GHz, it has to accept digital waveform data from the FPGA through 32 parallel channels at the rate of 150 MHz each. Quantization width of SRRC filter is 32, roll-off factor is 0.35, thus a group of filter coefficients whose width = 32, depth = 6144 can be produced in advance by MATLAB, and then stored into 6 LUT with width = 32 and depth = 1024. Input symbol rate has been tested from 100 Ksps to 600 Msps, and the IF center frequency is fixed at 1.2 GHz. Without loss of generality the modulation type is chosen as 8PSK in our test.

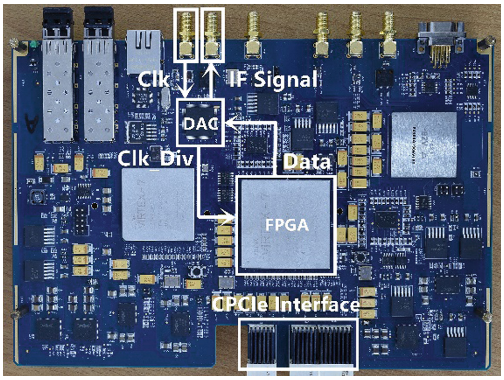


Fig. 9. Photograph of testing platform

As indicated in Fig. 9, DAC uses its internal clock divider using an exterior clock ("Clk") to provide a fixed working clock ("Clk_Div") to FPGA. FPGA sends the data ("Data") generated by our algorithm to DAC which transforms it to an IF analog signal ("IF Signal").

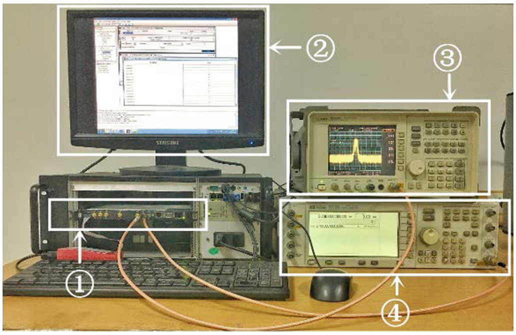


Fig. 10. Photograph of the testing system

The photograph of our testing system is shown in Fig. 10, and for the detailed models of the key components please refer to Table 1. As shown in the figure, the hardware platform (the PCB board) is inserted into an industrial PC through CPCIe bus. The PC provides power for the ADC + FPGA board, and also runs a GUI to facilitate real-time status monitoring and data analysis. The signal generator (HP E4433B) produces external clock for the testing system, and the IF output is fed to the spectrum analyzer (Agilent 8563EC).

Table 1. Critical components of the hardware system

Identifier	Component	Model number
1	DAC	Euvis MD622H
	FPGA	Xilinx XC7VX690T
2	Software interface	Xilinx Chipscope Pro
3	Spectrum analyzer	Agilent 8563EC
4	Signal generator	HP E4433B

Table 2. LUT-based impulse shaping consumption (Targeted FPGA model: Xilinx XC7VX690T)

Resource type	Used	Available	Utilization
Slice	9308	607200	12%
BlockRam18E	384	2060	18.6%
DSP48E1s	384	2800	13%

In Table 2 we summarize the resource consumption of the presented LUT-based impulse shaping algorithm for the targeted FPGA model Xilinx XC7VX690T. In particular, we are concerned about three different kinds of resources, namely the logic resource (slices), the storage resource (Block RAM 18E) and the computational resource (DSP48E1s). For all these three different categories of resources, we are glad to find the presented algorithm occupies less than 20% of the total available amount inside the FPGA, which suggests it may strike as a highly cost-efficient choice in practice.

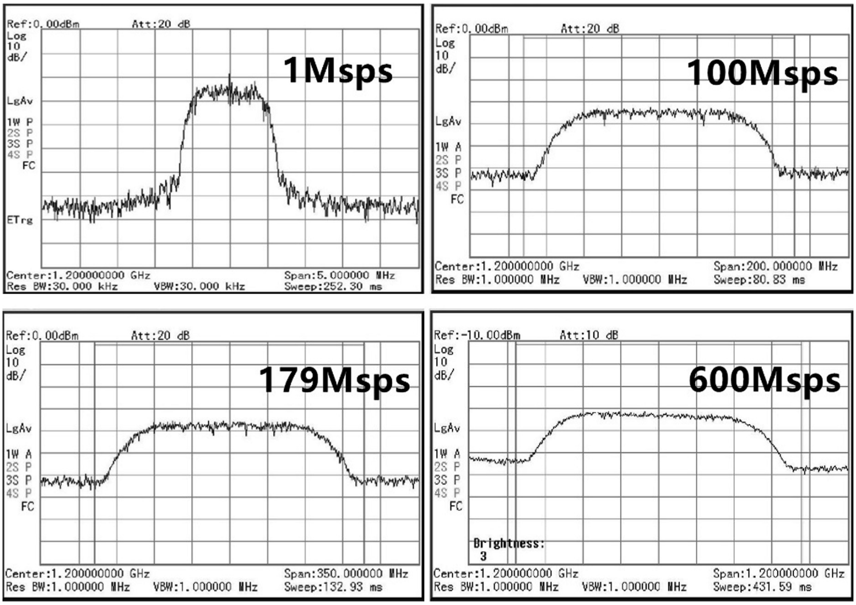
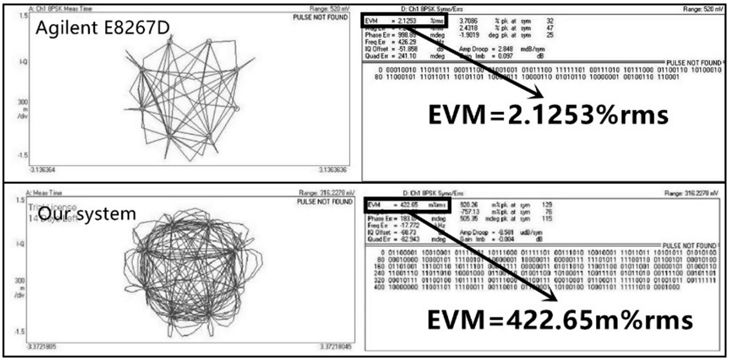


Fig. 11. Frequency spectrum of IF signal at different symbol rate



```
D: Ch1 8PSK Syms/Errs                                         Range: 520 mV
EVM       = 2.1253   %rms     3.7086   % pk at sym   32
                              2.4318   % pk at sym   47
Phase Err = 998.89   mdeg    -1.9019   deg pk at sym 25
Freq Err  = 426.29
IQ Offset = -51.858  dB       Amp Droop = 2.848  mdB/sym
Quad Err  = 241.10   mdeg          Imb  = 0.097  dB
                                                         PULSE NOT FOUND
 0 00010010 11010111 00011100 [illegible] 01011100 11111101 00111010 10111000 01100110 10100010
80 11000101 11011011 10101101 [illegible] 10000110 01010110 10000001 00100110 110001
```

```
D: Ch1 8PSK Syms/Errs                                         Range: 316.2278 mV
EVM       = 422.65   m%rms    920.26   m% pk at sym  129
                             -757.13   m% pk at sym  76
Phase Err = [illegible] mdeg  505.35   mdeg pk at sym 115
Freq Err  = -17.772  kHz
IQ Offset = -68.73            Amp Droop = -8.581  udB/sym
Quad Err  = -82.943  mdeg     Gain Imb  = -0.004  dB
                                                         PULSE NOT FOUND
  0 01100001 10010001 [illegible] 10111000 01111101 00111010 10010001 11011011 10101011 01010100
 80 00010000 10000101 11100101 [illegible] 10000011 00000111 11101011 11100110 01100101 00010100
160 01101001 11100110 10111101 [illegible] 00000011 01011010 11001100 01011001 00000101 01000110
240 11001110 11011010 10001000 [illegible] 01001100 10100011 10011101 01011010 00111100 00101101
320 00010111 01100100 10111111 00111000 [illegible] 00010001 01100011 11110010 01001011 00111111
400 10000000 11001101 11100011 00110010 [illegible] 10100100 10001101 11111010 0001000
```

Fig. 12. Constellation, frequency spectrum and EVM of an 8PSK signal at 5 Msps symbol rate tested by Agilent 89600 Vector Signal Analyzer

In Fig. 11 we present the result obtained on the spectrum analyzer, when the IF output is generated for symbol rate 1 Msps, 100 Msps, 179 Msps and the maximum 600 Msps. Note that we choose 179 Msps deliberately to show the presented algorithm may work perfectly for the case where the sampling rate of the DAC (4.8 GHz) is not an integral multiple of the symbol rate (179 Msps). It is observed, the corresponding bandwidth is approximately, 1.4 MHz, 140 MHz, 252 MHz and 820 MHz, respectively, due to the 0.35 roll-off factor of the adopted SRRC baseband pulse. It should be noted when the output signal bandwidth is above 500 MHz, the far-end of the IF output spectrum is a little bit attenuated. The reason for this phenomenon is the DAC has a low-pass "Sinc" output characteristic [9]. Using digital pre-emphasis, it is feasible to compensate for the non-uniform output characteristic of the DAC inside the FPGA. This work is now underway (Fig. 12).

From a digital modulation point of view, the IF signal quality is usually evaluated by the Error Magnitude Vector [10], which characterize the deviation of the signal under-test with respect to a standard, error-free constellation. EVM could be measured by a Vector Signal Analyzer, and in our test, we use KeySight™ 89600 VSA software running on an E8563E. KeySight™ Vector Signal Generator E8267D is used as a performance benchmark, i.e. we compare the EVM of the IF signal generated by our presented algorithm and hardware with that of E8267D. Without loss of generality, the

Table 3. EVM of other higher symbol rate signal

Symbol rate	Error-Vector-Magnitude (EVM)
100 Msps	3.0470% rms
200 Msps	3.7903% rms
250 Msps	4.2828% rms
350 Msps	5.1685% rms
500 Msps	6.2878% rms
550 Msps	6.9851% rms
600 Msps	7.9340% rms

symbol rate is chosen as 5 Msps, which falls in both the available range of E8267D (10 Ksps–50 Msps) and our system (100 Ksps–600 Msps). It is found the EVM of our system is 0.422%, and is significantly better than that of E8267D, which is 2.12%. For other higher symbol rate, the EVM of the presented system is listed by Table 3.

4 Conclusion

The communication signal generation at arbitrary symbol rate is one of the most difficult problems on SDR platform. To solve this problem, we propose an efficient impulse shaping algorithm based on a low-complexity hardware system. Through test data compared with the counterpart of KeySight™ Vector Signal Generator E8267D, it is demonstrated that our experimental results are in consistence with the theoretical derivation. In particular, we have achieved 4.8 Gsps modulation with symbol rate ranging from 100 Ksps to 600 Msps. Moreover, with some little modifications, the scheme we proposed could be easily adapted to 8PSK, 16QAM and other arbitrary amplitude-phase-modulation constellations, or other baseband pulse-shaping functions in high speed modem.

Acknowledgement. This work is supported by National Natural Science Foundation of China under contract No. 61471360.

References

1. KeySight Technologies: Agilent E8267D PSG User Manual (2012)
2. ZODIAC AEROSPACE, Cortex HDR XXL-High Data Rate Receiver User Manual (2016)
3. Peiqing, C.: Digital Signal Processing Tutorial. Tsinghua University Press, Beijing (2007). pp. 182–188
4. Gongli, Z.: All Digital Receivers Theory and Techniques. Science Press, Beijing (2005). pp. 86–106
5. Tseng, B.D.: Directly realization of the structure for FIR and IIR filters. In: 23th Asilomar Conference on Signals, Systems and Computers, pp. 233–235 (1989)
6. Xiao, Z., Su, L., et al.: Fractional sampling rate transformation for wideband all digital receivers. Tsinghua Univ. (Sci. Tech.) **50**(10), 1643–1645 (2010)
7. Yang, J., Cui, S., Liu, C.: A joint implementation for bit synchronization and filtering in high-speed digital receivers, CN101789858A (2010)
8. Li, K.: Design and implementation of a variable data rate signal generator. In: The Eleventh Satellite Communication Conference (2015)
9. Vandenbussche, J., Van Der Plas, G., Gielen, G., et al.: Behavioral model of reusable DA converters. IEEE Trans. Circ. Syst. Analog Digital Sig. Process. **46**(10), 1323–1326 (1999)
10. Mckinley, M.D., Remley, K.A., Myslinski, M., et al.: EVM calculation for broadband modulated signals, pp. 45–52 (2004)