An FPGA Based All-Digital Transmitter with Radio Frequency Output for Software Defined Radio

Conference Paper · May 2007

DOI: 10.1109/DATE.2007.364561 · Source: IEEE Xplore

3 authors, including:

John Grosspietsch
Institute of Electrical and Electronics Engineers
23 PUBLICATIONS **694** CITATIONS
SEE PROFILE

Gokhan Memik
Northwestern University
193 PUBLICATIONS **5,392** CITATIONS
SEE PROFILE

An FPGA Based All-Digital Transmitter with Radio Frequency Output for Software Defined Radio

Zhuan Ye[1,2]
Wireless Solutions Research Center[1]
1301 E. Algonquin Rd.
Schaumburg, IL 60196, USA
+1-847-538-3847
Zhuan@labs.mot.com

John Grosspietsch[1]
Wireless Solutions Research Center
1301 E. Algonquin Rd.
Schaumburg, IL 60196, USA
+1-847-576-4846
johng@labs.mot.com

Gokhan Memik[2]
Department of EECS[2]
Northwestern University
Evanston, IL 60208, USA
+1-847-467-1168
memik@ece.northwestern.edu

ABSTRACT

In this paper, we present the architecture and implementation of an all-digital transmitter with radio frequency output targeting an FPGA device. FPGA devices have been widely adopted in the applications of digital signal processing (DSP) and digital communication. They are typically well suited for the evolving technology of software defined radios (SDR) due to their reconfigurability and programmability. However, FPGA devices are mostly used to implement digital baseband and intermediate frequency (IF) functionalities. Therefore, significant analog and RF components are still needed to fulfill the radio communication requirements. The all-digital transmitter presented in this paper directly synthesizes RF signal in the digital domain, therefore eliminates the need for most of the analog and RF components. The all-digital transmitter consists of one QAM modulator and one RF pulse width modulator (RFPWM). The binary output waveform from RFPWM is centered at 800MHz with 64QAM signaling format. The entire transmitter is implemented using Xilinx Virtex2pro device with on chip multi-gigabit transceiver (MGT). The adjacent channel leakage ratio (ACLR) measured in the 20 MHz passband is 45dB, and the measured error vector magnitude (EVM) is less than 1%. Our work extends the digital implementation of communication applications on an FPGA platform to radio frequency, therefore making a significant evolution towards an ideal SDR.

1. INTRODUCTION

There are many kinds of wireless communication systems currently in use. Software defined radio (SDR) technology is needed for multimode and multi-standard communication devices, allowing end users to move between domains and footprints and maintain serviceability [2, 11]. One definition of SDR is: an SDR is a radio in which the digitization is performed at some stage downstream from the antenna. Then the radio can use flexible and reconfigurable hardware such as FPGAs to implement the digital signal processing algorithms. As technology advances, for an ideal SDR, the digitization might be at, or very close to the antenna, such that almost all the radio communication functionalities can be realized using software based on high speed and reprogrammable digital signal processing engine [14].

However, current SDR implementations still require significant amount of analog components. An example of a current SDR based transmitter architecture [3] is shown in Figure 1. In this paper, we try to tackle this problem and show the feasibility of an all-digital transmitter illustrated in Figure 2. Almost all the transmitter's functionalities can be incorporated in the digital signal processing engine using FPGAs except the RF power amplification and simple filtering. The advantages of all-digital transmitters are: potential high efficiency power amplification ([8, 10]), the capabilities of digitally combining signals from multiple channels, and software programmability.

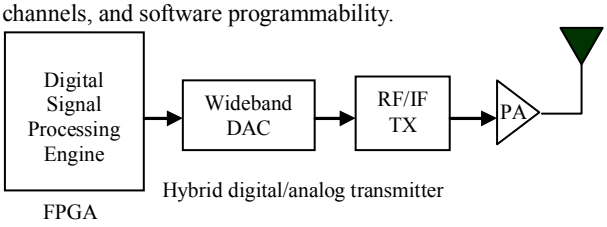


Figure 1. Current SDR based transmitter architecture

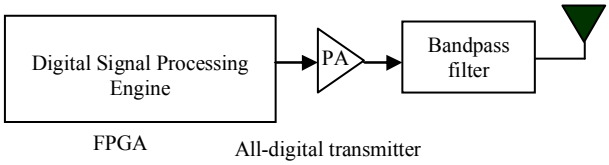


Figure 2. All-digital transmitter architecture

The generation of digital RF signals has drawn a lot of interest among researchers and engineers. However, only simulation results or non real-time test results have been presented in literature [8, 10, 16]. In these works, the digital RF signals were computed offline and stored in pattern generator for the purpose of measurement. In this paper, we present the architecture and implementation of a real-time system that demonstrates the feasibility of digital generation of RF signals.

Our all-digital transmitter consists of two major functional blocks: a universal QAM modulator and a radio frequency pulse width modulator (RFPWM). The entire transmitter is implemented on a Xilinx's Virtex2pro device. The binary output of the transmitter is centered at 800 MHz with 64QAM signaling format. Figure 3 shows the overall architecture of our all-digital transmitter.

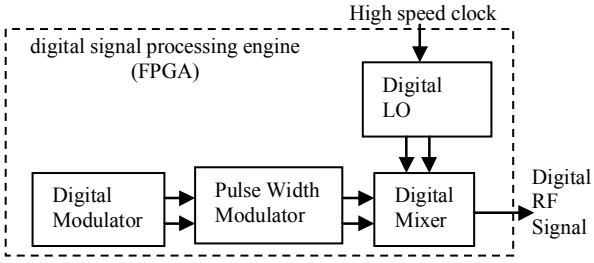


Figure 3. All-digital transmitter architecture

The remainder of the paper is organized as follows. Section 2 presents the architecture and implementation of the universal QAM modulator. Section 3 discusses the algorithm, architecture, and implementation of the RFPWM. The implementation results

978-3-9810801-2-4/DATE07 © 2007 EDAA

of our all-digital transmitter are presented in Section 4. Finally, we will discuss related work and draw conclusions in Sections 5 and 6, respectively.

2. UNIVERSAL QAM MODULATOR

In this section, we present the architecture and the FPGA implementation of our universal QAM modulator. The overall QAM modulator architecture is presented in Section 2.1. In Section 2.2, we present the details of the main functional blocks inside the modulator and their implementations.

2.1 Universal QAM Modulator Architecture

Figure 4 shows the block diagram of the QAM modulator. The double lines connecting different blocks indicate that there are two identical datapaths for both inphase path and quadrature path (I and Q). In a traditional modulator, a quadrature direct digital frequency synthesizer (QDDFS) will be used to modulate the signal to intermediate frequency (IF), and then the digital output is converted to analog signal using DAC [1, 15]. In the all-digital transmitter presented in this paper, an RFPWM will be used to directly synthesize RF signals in digital domain.

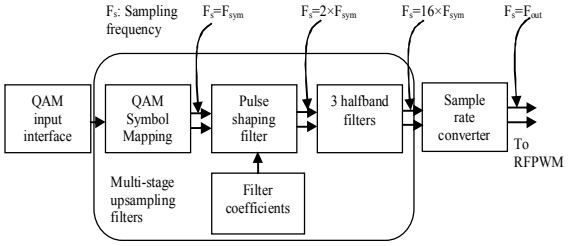


Figure 4. Block diagram of QAM modulator

The QAM modulator consists of several digital interpolation filters. The pulse shaping filter is used in almost all digital modulators to minimize inter-symbol interference (ISI). The pulse shaping filter is made programmable such that it is flexible to accommodate different standards and modulation formats. The three half-band filters further increase the sampling rate, which can reduce the complexity of the sample rate converter. The combination of pulse shaping filter and the subsequent halfband interpolation filters are sometimes referred to as multi-stage upsampling filters. The pulse shaping filter and half band filters all have integer interpolation ratios, therefore the sampling rate of the last half-band filter's output is an integer multiple of the input symbol rate. A sample rate converter is generally needed to convert this sample rate to another arbitrary sample rate. The reason for this conversion is that SDR based transmitter is required to support multiple air-interfaces, where each air-interface often specifies its unique symbol rate. In order to allow the digital to analog converter to have a constant sample rate regardless of the input symbol rate, a sample rate converter is needed. Although no DAC is needed in our all-digital transmitter, a sample rate converter (SRC) is still necessary to convert the sample rate to a fixed sample rate suitable for PWM generation.

2.2 FPGA Implementation of the Modulator

2.2.1 Multi-stage Upsampling Filters

The multi-stage upsampling filters consist of one pulse shaping filter and three stages of halfband interpolation filter. The pulse shaping filter is the most complex filtering operation in the QAM modulator. Finite impulse response (FIR) filters are normally used for pulse shaping due to their linear phase responses. The impulse response of a linear phase FIR is either symmetrical or anti-symmetrical. Design techniques of FIR filters have been well studied for ASIC implementations [7]. In this paper, we present the general interpolation FIR architecture specifically optimized for FPGAs.

There is an inherent interpolation operation in the pulse shaping filter. Interpolation filters can be efficiently implemented using polyphase structures by dividing the original filter into subfilters [7]. Efficient FIR implementations take advantage of the filter coefficients symmetry to minimize the number of multiplications. It has been shown that when the interpolation ratio is even and the length of the FIR filter is odd, we can have two linear phase subfilters after decomposing into polyphase structure [7]. Therefore we choose the interpolation ratios to be two for all of the interpolation filters, while designing them to have odd number of taps.

Our FIR architecture is shown in Figure 5. It consists of two linear phase FIR engines for the two subfilters respectively. Inside each FIR engine, the datapath takes advantage of the coefficients symmetry in the filter by performing the summation of two input samples before the multiply and accumulate (MAC) takes place. The additional input to the accumulator adder (acc_in) is needed when the input sample rate is too high for one FIR engine to complete all the calculation before the next input sample arrives. In that case we need to daisy chain several FIR engines together to pipeline the operation. The FIR output from one stage of the pipeline will be connected to the acc_in input of the next stage. We implemented both the input buffer and coefficient RAM using the distributed memory in Xilinx FPGA. Although the two subfilters are working on the same set of input, they require different ordering of the samples. Therefore a re-ordering block is designed to temporarily store the samples read from the buffer and then present them to the FIR engines at the appropriate time.

The halfband filter is a special class of FIR filter. Every other coefficient of the halfband filter is zero. Therefore by combining this special property together with the general coefficient symmetry of the linear phase FIR filter, the implementation of halfband filter can be very hardware efficient. In addition, the halfband filters in a digital modulator often have fixed coefficients. These coefficients can be formatted using canonic signed digits (CSD) [7]. By using CSD formatted coefficients, the multiplication and accumulation in a general FIR can be replaced by shift and add operation instead.

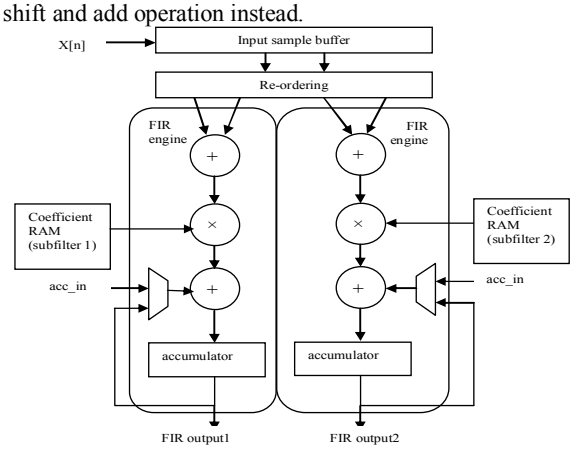


Figure 5. Interpolation FIR implementation architecture

2.2.2 Sample Rate Converter

As shown in Figure 4, sample rate converter (SRC) is needed to convert the digital signal from one sampling to another sampling rate while preserving the necessary information [4].

The underlying operation of an SRC is interpolation, where new sample values at arbitrary time instants in between the existing samples are computed. The sampling time offsets between the new sampling instants and original sampling instants are referred to as fractional intervals μ_k. The SRC can be possibly implemented using traditional FIR architecture, where the coefficients for this FIR filter vary with the fractional interval μ_k. Then pre-computed coefficients need to be stored in memory for each possible μ_k. When the number of elements in the set of $\{\mu_k\}$ becomes very large to achieve fine resolution on fractional interval, the memory requirement to store the coefficients can be high. Instead, in this paper, we use polynomial based interpolation, which does not require any storage of filter coefficients. In contrast, the coefficients can be computed based on the fractional interval value μ_k. Moreover, a hardware efficient architecture, Farrow structure, can only be applied to polynomial based SRC. The details about Farrow structure can be found in [4].

The architecture of the sample rate converter is shown in Figure 6. The SRC consists of: 1) the numerically controlled oscillator (NCO) that supplies the fractional interval μ_k and manages the input buffer of SRC; 2) the Farrow computation structure that calculates the SRC output.

The NCO is programmed by a frequency control word (FCW), determined by the ratio of the input sample rate and output sample rate (F_{in}/F_{out}). Different symbol rates can be supported by reprogramming the FCW. We assume the ratio is less than 1, therefore interpolation (sample rate increase) rather than decimation (sample rate decrease) is always performed.

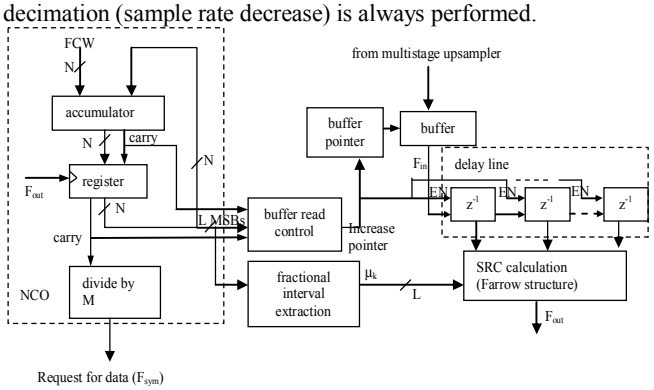


Figure 6. Complete sample rate converter architecture

Through the NCO, the SRC is able to maintain the input rate at F_{in}, while generating output at the rate of F_{out}. One important signal from the NCO is the accumulator overflow signal, which asserts at the rate of F_{in}. Whenever the NCO overflows, a new input sample is needed to calculate SRC output. We introduce a buffer between the output of the multi-stage upsamplers and the sample rate converter. Every time the NCO overflows, the buffer pointer will be incremented and a new sample is read from the buffer. The same signal is also used to enable the new sample to be shifted into the delay line. Also, the NCO register value represents the fractional interval. Only a number of MSBs from the register are needed for the SRC computation to meet our performance requirement, such as ACLR and EVM. Moreover, the NCO overflow indicator is divided by M in frequency to generate a "request for data" (RFD) signal to the multi-stage upsampling filters, where M is the aggregate interpolation ratio of all the filters consisting of the multistage upsampling filters. For example, in our QAM modulator architecture shown in Figure 4, the value of M should be set to 16. Upon receiving the RFD signal, the multi-stage upsampler reads one symbol through the input interface, and will produce M output samples to the SRC input buffer after performing the interpolations. Using this mechanism, the SRC buffer is guaranteed to maintain its fill level around some balanced state without experiencing any overflow or underflow.

A different control approach is to use the MSB of the NCO register as the reference clock for the delay line as shown in Figure 6. This reference clock is further divided down to provide sample rate clocks to the different interpolation filters inside the multi-stage upsampling filters [1, 15]. The novelty of our implementation only requires one system clock for all the filter stages. The rate control is accomplished by the request-and-respond between SRC and multi-stage upsamplers. By using such a handshake interface mechanism, it is possible to implement the multi-stage upsampling in an independent platform such as a digital signal processor.

3. RF PULSE WIDTH MODULATOR

RF pulse width modulator (RFPWM) is used in our all-digital transmitter to synthesize digital RF signals. In Section 3.1, we present the architectural consideration for the RFPWM. The algorithm transformation and implementation details are discussed in Section 3.2.

3.1 RFPWM Architecture

The digital generation of RF signals has been of interest in wireless communications. For example, Keyzer et al. [8] present a method of using band-pass delta-sigma modulation to generate binary signals at radio frequency. Binary signaling can be used together with switch mode power amplifier to achieve higher efficiency comparing with other types of PA technologies. The drawback of this architecture is that the band-pass delta-sigma (BPDS) modulator needs to be running at 4 times the output center frequency ($F_s=4f_0$), which can easily be in the multi-gigahertz range. To accomplish such high frequency operations, customized integrated circuits have to be carefully designed, instead of using off-the-shelf FPGAs.

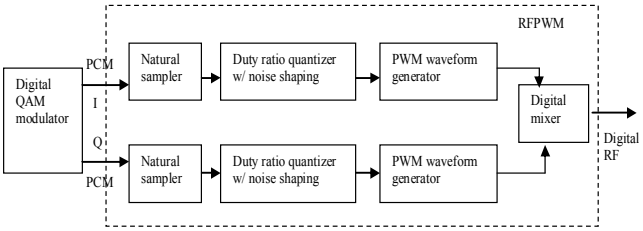


Figure 7. Block diagram of RFPWM

Another method using pulse width modulation (PWM) was presented in [13] to synthesize binary RF signals. Our architecture is based on this RFPWM concept. The architecture of our RFPWM based all-digital transmitter is shown in Figure 7. In our application, the entire signal processing is accomplished using 100 MHz clock rate. The low rate of the PWM modulator operation makes it feasible for an implementation on FPGA. The high speed operation is only needed for the final stage parallel to serial conversion, when the binary bit stream is generated.

3.2 RFPWM Signal Processing and Its Implementation

3.2.1 Architectural Transformation of RFPWM

The RFPWM consists of these main functional blocks: natural sampler, duty ratio quantizer with noise shaping, PWM waveform generator, and quadrature digital mixer.

In the RFPWM, we first convert the output from QAM modulator (PCM format) to pulse width modulation (PWM) format. A simplified diagram for the generation of double sided PWM (DPWM) waveform is shown in Figure 8 to illsutrate the overall processing. A good explanation of PWM can be found in [6]. The DPWM waveform has both the leading edge and trailing edge modulated by information source. The center of the DPWM waveform provides the reference where the pulse width is defined. The time difference between two consecutive reference points is defined as the pulse period. The pulse repetition frequency (PRF) is defined as the reciprocal of the pulse period. The PRF determines how frequently the pulses are generated, and is generally chosen to a sub-harmonic of the high speed clock frequency for digital PWM generation. In our application, the PRF is chosen to be 100 MHz, which is 1/32 of the high speed clock frequency (3.2 GHz).

In generating the DPWM waveform as shown in Figure 8, the leading edge and trailing edge are both modulated by different PCM samples. Therefore the PCM input should have the sample frequency equal to twice the PRF. With the PRF chosen to be 100 MHz, a 200 MHz effective sample rate is required for the signal processing operations in the RFPWM. After some experimental implementation on our target FPGA device, we found this clock frequency is too high to achieve. But it can be noticed from Figure 8 that the DPWM waveform can be viewed to consist of two PWM waveforms. One PWM to the left of the reference point has its leading edge modulated by one PCM sample, while the other PWM to the right of the reference point has its trailing edge modulated by another PCM sample. If we generate these two PWM waveforms independently and later combine them, the effective processing rate is halved to PRF. With this parallel transformation of DPWM generation, we are able to implement a real time RFPWM to demonstrate the all-digital transmitter.

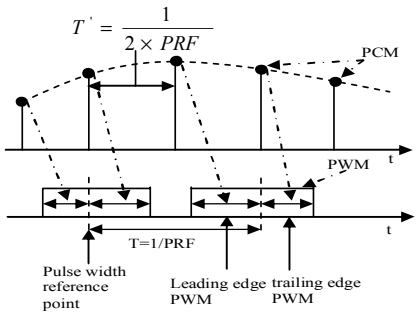


Figure 8. Generation of double sided PWM waveform

3.2.2 Natural Sampler

The natural sampler converts the uniformly sampled digital signal to naturally sampled signal values. After natural sampling, the pulse width modulated signal shows much less baseband distortion [5]. An ideal natural sampler calculates the intersection points between a linear ramp signal and the imaginary analog waveform. The intersection points define the transitions of the pulse. This process is shown in Figure 9, where a rising ramp is used to calculate the position of the leading edge, and a falling ramp is used to calculate the position of the falling edge. The edge positions are often referred to as pulse duty ratios, D_l and D_t. We use the δC algorithm [5] to approximate the natural sampling process.

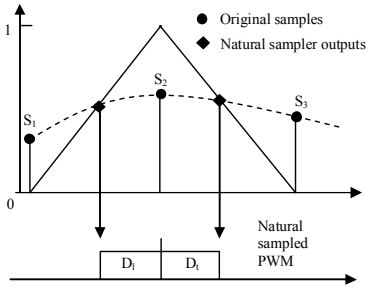


Figure 9. Natural sampling process

3.2.3 Duty Ratio Quantization with Noise Shaping

The digital PWM waveform can only transition at the edges of the reference high speed clock. Therefore the duty ratios calculated from the natural sampler need to be quantized to align with the closest high speed clock edges. After the parallel transformation described in 3.2.1, the output from the natural sampler has a sample rate of 100 MHz. With a 10-bit duty ratio representation, we will have to use a 102.4 GHz ($2^{10}\times$100 MHz) clock. In contrast, the reference clock frequency is 3.2 GHz if only 5-bit duty ratio values are used. However, the quantization in duty ratio introduces unwanted quantization noise. Like in over-sampled data converters, noise shaping can be performed in PWM to preserve the SNR performance in the frequency band of interest [6]. The noise shaping filter H(z) is an FIR designed using the principles described in [12], with some modifications to accommodate the inherent non-linearity in PWM modulation. In our application, the baseband signal bandwidth is designed to be 10 MHz.

3.2.4 RFPWM Signal Generation

The RFPWM generation includes two steps. Firstly the baseband PWM signal is synthesized using the quantized duty ratios. Two sets of baseband PWM waveforms are generated for inphase (PWM_I) and quadrature (PWM_Q) respectively. The baseband pulses are generated every 10 ns (1/100MHz) using the 3.2 GHz clock. Therefore there are 32 (3.2GHz/100MHz) possible transition edges in each pulse period. Secondly the baseband PWM signal will be mixed with a complex digital local oscillator (LO) to form the RFPWM signal at the carrier frequency. The carrier frequency is chosen to be 800 MHz, so there are four samples in each carrier cycle. We make the inphase LO (LO_I) take these ternary values {0, 1, 0, -1} in each cycle, while the quadrature LO (LO_Q) takes the same set of ternary values in the same cycle but with an offset, i.e., {-1, 0, +1, 0}. Then the mixer operation

$$RFPWM = PWM\_I \times LO\_I + PWM\_Q \times LO\_Q \quad (11)$$

should yield the interleaved, non-inverted or inverted PWM_I and PWM_Q value sequence {-PWM_Q, +PWM_I, +PWM_Q, -PWM_I}. The RFPWM generation is illustrated in Figure 10. Since both PWM_I and PWM_Q are binary signals, the final RFPWM output is of binary format too.

Normally, the generation of PWM waveform requires a digital counter using the high speed reference clock. However, because of the relatively small number of possible PWM waveforms in our

application, the PWM waveforms can be generated in parallel by table look-up and parallel vector combination. The only high speed operation is the final output stage of the serial RFPWM waveform. State of the art FPGA devices offers high speed serial interfaces [19], which provides us a convenient implementation of the parallel to serial conversion without any additional components.

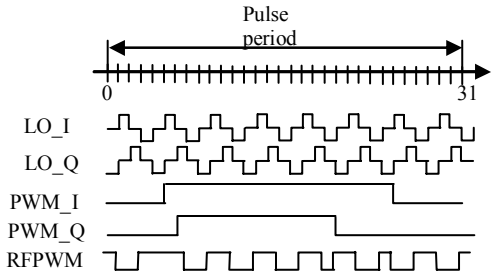


Figure 10. RFPWM generation

4. Implementation Results

The entire all-digital transmitter architecture is shown in Figure 11. Only the functional blocks for inphase path are shown in detail, while the quadrature path consists of almost identical functional blocks. In order to accommodate the parallel transformation of the PWM signal processing, the sample rate converter is modified to a parallel architecture. Two output samples are generated simultaneously, making the effective sample rate at the SRC output to be 200 MHz although the SRC is only clocked at 100 MHz. The natural sampler in turn calculates two duty ratio outputs for every two SRC outputs. The two duty ratios values, D_l and D_t, undergo their own quantization with noise shaping to form the quantized duty ratios. Then the quantized duty ratios are combined to look up the PWM waveform stored in the ROM, where D_l is used to modulate the leading edge and D_t is used to modulate the trailing edge. Finally, RFPWM waveform is synthesized using the two PWM waveforms from I and Q before it is converted to a serial bit stream at 3.2 Gbps.

The FPGA device chosen for the prototype of all-digital transmitter is from Xilinx's Virtex2pro family: XC2VPX20-FF896, speed grade -7. On chip multi-gigabit transceiver (MGT) is used as the high speed parallel to serial converter. Table 1 summarizes the system parameters and the measured performance for our all-digital transmitter. The screen shots for spectrum and EVM measurements are shown in Figure 12. In the 20 MHz passband, the measured ACLR is 45 dB, which can meet the requirement for WCDMA [15]. The noise shaping effects can clearly be seen from the spectrum plot where the noise rises outside the RF signal bandwidth. The EVM is measured to be less than 1% before and after RFPWM.

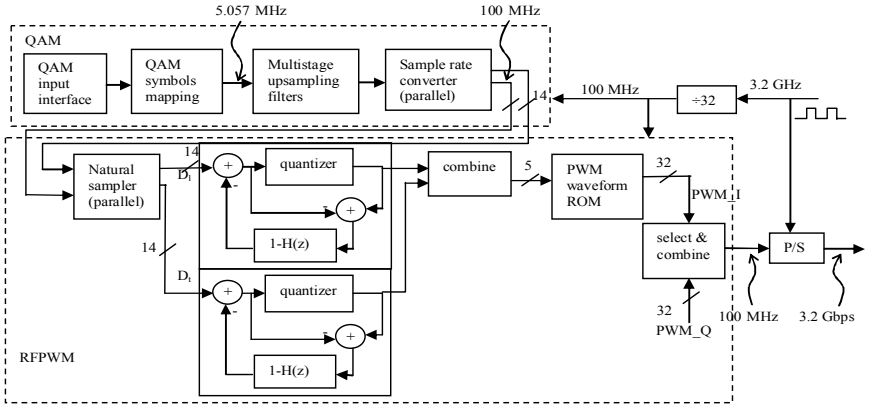
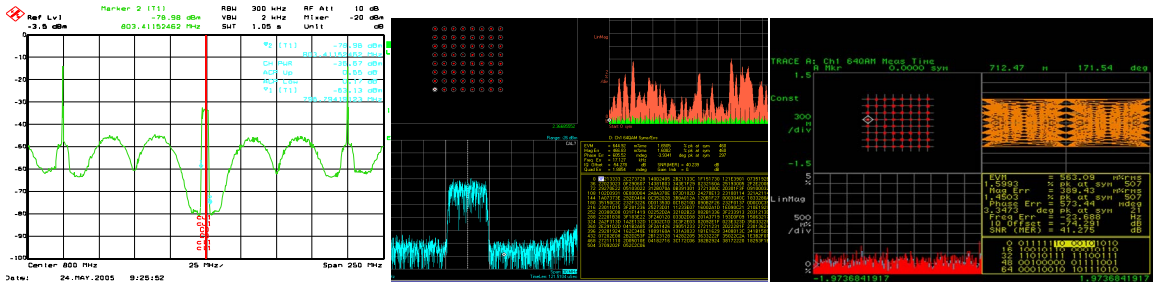


Figure 11. All-digital transmitter architecture implemented on FPGA



a **b** **c**

Figure 12. All-digital transmitter measurement: a. wideband spectrum (250 MHz), b. EVM (measured at RFPWM output), c. EVM (measured at SRC output)

Table 1. System parameters and measured performance

Symbol format and rate		64 QAM, 5.057 Msps
Pulse shaping filter		69-tap SRRC (α=0.15)
SRC output frequency		200 MHz
Pulse repetition frequency		100 MHz
Carrier frequency		800 MHz
High speed clock freq.		3.2 GHz
RF signal bandwidth		790 – 810 MHz
Measured passband ACLR		~45 dB
Measured EVM	SRC output	<1%
	RFPWM output	<1%

5. RELATED WORK

The related work presented in the literature can be grouped into two categories. The first category consists of architectural study and implementation of digital modulators and transmitters. All digital modulators implemented on ASIC were presented by Cho and Samueli [1] and Vankka et al. [15]. SDR based architecture using FPGAs is presented by Lee et al. [9] and Sheen et al. [13] for CDMA2000 and WCDMA, respectively. All of these transmitters are based on digital IF architecture, therefore require highly sensitive analog and RF components such as wideband DAC. The all-digital transmitter architecture presented in this paper eliminates the IF stage by directly synthesizing RF signal in digital domain. The other category consists of algorithm study or partial implementation of digital RF generation. Keyzer et al. [8] describe a digital transmitter using BPDS architecture. The high rate of processing makes it very difficult for implementation. RFPWM algorithm and simulation results were presented in Midya et al. [11]. Wagh et al. [16] present a digital RF transmitter based on digital PWM, however the duty ratios were computed offline and then fed to the PWM generator for testing. This paper presents the implementation of a complete all-digital transmitter with RF output.

6. CONCLUSION

Software defined radio (SDR) technology enables wireless infrastructures and devices to support multiple air-interfaces, by using reconfigurable hardware platform. From the transmitter point of view, the current SDR architecture still contains highly sensitive analog and RF components which are hardly software definable. In this paper, we extend the software defined functionalities to radio frequency with the novel all-digital transmitter. The all-digital transmitter uses pulse width modulation (PWM) method such that the RF signal with binary format can be directly synthesized in digital domain. The low rate of the signal processing of PWM makes it suitable for implementation on off-the-shelf FPGA devices. By combining a universal QAM modulator and an RF pulse width modulator, we demonstrated a real time transmitter system with digital RF output using realistic signaling format, i.e., 64QAM. The on chip multi-gigabit transceiver (MGT) is used to generate 3.2Gbps binary RF signals, with 800 MHz carrier frequency. The measured passband ACLR is about 45 dB. The measured EVM is less than 1%. With the advancement of FPGA technology, the all-digital transmitter presented in this paper will help to make a feasible ideal SDR in the future.

7. REFERENCES

[1] Cho, K. and Samueli, H. A Frequency-Agile Single-Chip QAM Modulator with Beamforming Diversity. *IEEE Journal of Solid State Circuits*, Vol. 36, No. 3, March 2001, 398-407

[2] Cummings, M. and Haruyama, C. FPGA in the Software Radio. *IEEE communications Magazine*, Feb. 1999, 108-112

[3] DoCosta, K. Virtex-II DSP Engines Enable Software Defined Radio. *Xcell Journal*, Fall/Winter 2001, 82-83

[4] Erup, L., Gardner, F., and Harris, R. Interpolation in Digital Modems – Part II: Implementation and Performance. *IEEE Trans. on Communications*, Vol. 41, No. 6, June 1993, 998-1008

[5] Gwee, B., Chang, J., and Li, H. A Micropower Low-Distortion Digital Pulsewidth Modulator for a Digital Class D Amplifier. *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 49, No.4, 245-256

[6] Goldberg, J. and Sandler, M. New High Accuracy Pulse Width Modulation based Digital-to-Analogue Converter/Power Amplifier, *IEE Proc. Circuits Devices Syst.*, Vol. 141, No. 4, Aug. 1994, 315-324

[7] Hawley, R. et al. Design Techniques for Silicon Compiler Implementations of High-Speed FIR Digital Filters. *IEEE Journal of Solid State Circuits*, Vol. 31, No. 5, May1996, 656-667

[8] Keyzer, K. et. al. Digital Generation of RF Signals for Wireless Communications with Band-pass Delta-Sigma Modulation. In *Microwave Symposium Digest, 2001 IEEE MTT-S International*, 2127-2130

[9] Lee, W., Park, W., Jung, J., and Lee, K. SDR-based digital IF for Multi-Band W-CDMA Transceiver. In *Proceedings of 4th International Conference on Information, Communications and Signal Processing*, Dec. 2003, Vol. 3, 1737-1741

[10] Midya, P., Wagh, P., and Rakers, P. Quadrature Integral Noise Shaping for Generation of Modulated RF Signals. In *Proceedings of the 45th Midwest Symposium on Circuits and Systems*, Vol. 2, 537-540

[11] Mitola, J. The Software Radio Architecture, IEEE Communications Magazine, May 1995, 26-38

[12] Norsworthy S. R., Optimal Nonrecursive Noise Shaping Filters for Oversampling Data Converters Part I: Theory. In *Proceedings of ICASSP 1993*, 1353-1356

[13] Sheen, M., Lee, S., and Kim J. Design and Implementation of Re-configurable transceiver for cdma2000. In *60th Vehicular Technology Conference*, Sept. 2004

[14] Tuttlebee, W. (editor), *Software Defined Radio: Enabling Technologies*. John Wiley and Sons, England, 2003

[15] Vankka, J. et al. A GSM/EDGE/WCDMA Modulator with On-Chip D/A Converter for Base Stations. *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 49, No. 10, Oct. 2002, 645-655

[16] Wagh, P. et. al. An All Digital Universal RF Transmitter. In *Proceedings of IEEE CICC 2004*, 549-553

[17] Xilinx, RocketIO X Transceiver Guide,2004