A Proof-of-Concept Superregenerative QPSK Transceiver

Alexis López-Riera*, Pere Palà-Schönwälder*, Jordi Bonet-Dalmau*, F. Xavier Moncunill-Geniz†, Francisco del Águila-López* and Rosa Giralt-Mas*
* Dep. of Electronic System Design and Programming (DiPSE)
† Dep. of Signal Theory and Communications (TSC)
School of Engineering of Manresa (EPSEM), Universitat Politècnica de Catalunya (UPC)
Email: alexis@dipse.upc.edu

***Abstract*—In this paper we present a description and experimental verification of an HF-band proof-of-concept superregenerative transceiver for QPSK signals. We describe a simple implementation of an all-digital, FPGA-based, QPSK transmitter section. On the receiver side, the quench signal is generated in the same FPGA with a minimum of analog circuitry. As the main novelty, we present a simple synchronization scheme suitable for packetized transmissions.**

I. Introduction

In the context of radio-frequency communications, the superregenerative (SR) receiver [1] is a suitable architecture which is receiving renewed interest (e.g. [2], [3]) in applications where low-power and low-cost are the main driving forces.

Recently [4], a SR receiver structure suitable for QPSK detection was proposed and experimentally confirmed. The approach in [4] takes advantage of the fact that an SR oscillator (SRO) generates RF pulses (SRO pulses) which preserve the phase information contained in the incoming signal. Next, a number of 1-bit samples of the current SRO pulse are stored as a bit pattern in a shift register. This pattern is compared with the bit pattern obtained from the previous SRO pulse and a decision on the observed phase difference (and, therefore, on the received bits) is taken.

The approach in [4] assumes that there is a suitable synchronization mechanism which places the sensitivity periods of the SRO at the center of each transmitted symbol. Once this is achieved, it is expected that the drift between transmitter and receiver clocks is low enough, allowing a full packet to be successfully received.

In this paper we propose and demonstrate a novel approach that makes this synchronization possible. After processing a suitable preamble, the receiver is able to acquire the time reference necessary for successful detection. As another contribution, we also present a full QPSK SR transceiver, where the transmitter path and quench generation is implemented on an FPGA. Although presented for the QPSK case, the principle may be applied to the general M-PSK case with obvious changes.

Work supported by the Spanish Ministerio de Ciencia e Innovación and the Ministerio de Economía y Competitividad under Grant TEC2012-35571.

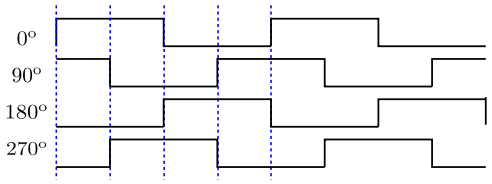

Fig. 1. The four basic carriers available at the register outputs.

The paper is structured as follows: In section II we present an all-FPGA implementation of a QPSK transmitter, exploiting the fact that the targeted carrier frequency is well within the capabilities of even low-cost FPGA devices. Section III is strongly based on [4] and describes the generation of the quench signal with the FPGA and, most importantly, presents a novel synchronization mechanism allowing for proper symbol synchronization in a packet transmission context. The transmitter and receiver have been integrated in a transceiver and section IV presents some results and discussion, while section V presents some conclusions.

II. QPSK Transmitter

Current low-cost FPGA devices, such as the one in [5], may operate at clock frequencies high enough so that direct RF signal generation in the HF band (the target of our proof-of-concept implementation) is clearly feasible.

Our objective is to transmit a differentially-encoded QPSK signal with constant phase during the whole symbol duration. This may be completely done inside the FPGA so that the only external analog circuitry additionally required is a filter to limit the frequency components at harmonics of the carrier frequency f_c. More flexible generation could be possible, for instance, with the approach in [6], at the expense of increased complexity.

The starting point is the generation of four carriers with phase shifts of 90° relative to each order. This is achieved with a shift register, initially loaded with the pattern `1100`, which is rotated at a frequency $4f_c$. A multiplexer selects between the four carriers according to the desired phase. Figure 1 shows the signals obtained at each one of the four register positions while Fig. 2 depicts the digital waveforms corresponding to a carrier phase change of 90°.

The data stream to be transmitted is split in pairs of bits to

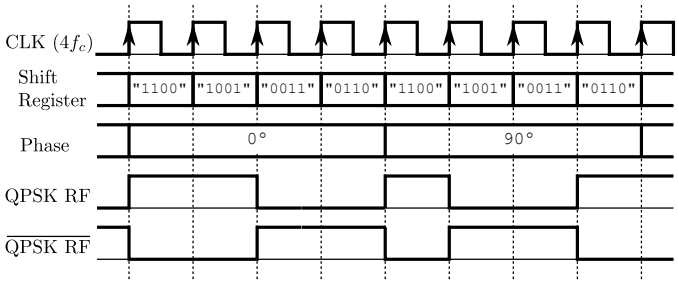

Fig. 2. Generated RF signal with QPSK modulation.

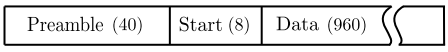

Fig. 3. Frame structure. Lengths are expressed in bits.

make symbols, which are then differentially encoded to avoid the need for an absolute phase reference at the receiver.

We have chosen a rate bit transmission of $f_{bit} = 20\,\text{kbps}$, i.e a symbol frequency of $f_{sym} = 10\,\text{kHz}$, and a carrier frequency of $f_c = 26.25\,\text{MHz}$ as in [4]. With this choice, each symbol period T_{sym} contains an integer number of carrier periods T_c.

In the current configuration, the transmitter sends a frame made of a 40-bit preamble (with carrier phase changes of 180° in each symbol), an 8-bit start delimiter and 960-bit of data (Fig. 3). For a given protocol, the data field length could be different (up to a maximum length, as shown in section III) and might contain other fields, but their content is irrelevant in the context of this paper. Also, the length of the preamble may be shortened trading preamble overhead for synchronization accuracy, as is also discussed in the next section.

III. SR-QPSK Receiver

The receiver module is based on the SR QPSK receiver designed in [4]. Essentially, this receiver has a conventional SRO core controlled by a suitable quench signal. A characteristic of SR receivers is that the signal generated by the SRO is only dependent on the received signal during a certain time window. The duration of the observation window is dependent on the quench signal (typically between 10-20% of the quench period) and is centered at the moment when the circuit transitions from being stable to unstable.

For QPSK detection, the SRO pulses are subsampled once they have achieved sufficient amplitude, taking N samples at a sampling frequency of $f_c N/(N+1)$. In our case, we take $N = 20$, as in [4]. For a carrier at 26.25 MHz carrier, this means taking samples at 25 MHz. The samples are quantized to 1 bit and stored in a shift register where they are circularly correlated with the previously received ones. The maximum of the correlation gives the phase difference [4].

In this paper, as a novelty, we have added the required circuitry to achieve proper synchronization and to generate the quench signal, allowing the system to be self-contained, without having to use an external generator. The underlying QPSK receiver operation has been kept unchanged.

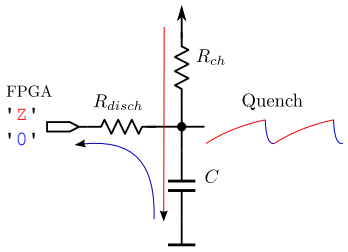

Fig. 4. Quench signal generation.

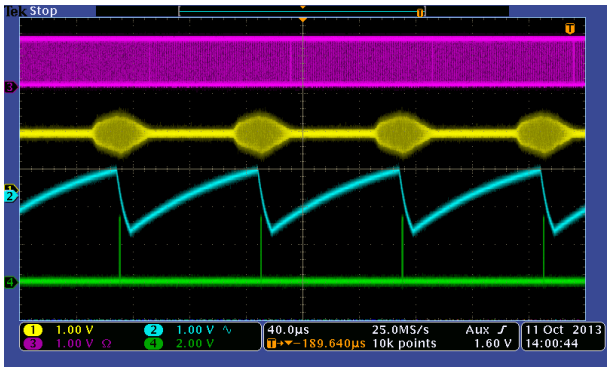

Fig. 5. SR receiver in operation with the sawtooth quench signal created by the FPGA. From top to bottom: trace #3 is the QPSK-modulated transmitter signal, trace #1 is the SRO signal, trace #2 is the quench signal and trace #4 depicts the 20 samples obtained in each quench period.

A. Quench signal

The shape, amplitude and dc component of the quench signal have a significant impact on the operation of a SR receiver [7]. We have found out that a suitable choice for our receiver is a sawtooth signal with $2\,V_{pp}$ amplitude. This signal has a relatively low slope at the sensitivity point where the oscillator becomes unstable and a high negative slope to quickly extinguish the oscillations. These properties contribute to reducing receiver bandwidth and limiting hangover [7].

The circuit in Fig. 4 allows a simple generation of this signal. An FPGA output pin toggles between a high-impedance state (`Z`) and ground. In the `Z` state, the capacitor is exponentially charged to `Vcc`. The first portion of the exponential waveform is sufficiently linear for our purposes. At a given time instant, the FPGA pin toggles to ground, quickly discharging the capacitor. Element values are $R_{ch} = 1.8\,\text{k}\Omega$, $R_{disch} = 47\,\Omega$, $C = 56\,\text{nF}$, which together with $V_{CC} = 3.3\,\text{V}$ provide the required waveform, with $\tau_{charge} = 38\tau_{discharge}$.

In order to have a degree of freedom to adjust the SR gain, another pin of the FPGA generates a PWM signal which, after being low-pass filtered, is added to the aforementioned sawtooth signal. Figure 5 depicts the generated quench signal and the RF pulses generated by the SRO in response to it. It also shows the samples of the SRO signal, taken when the SRO amplitude is high enough.

B. Synchronization

The synchronization problem appears in any wireless link where transmitter and receiver do not share a common time

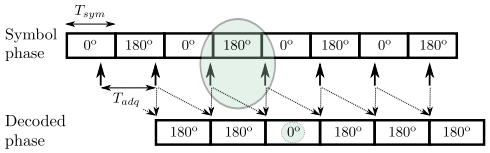

Fig. 6. Outline of the synchronization method. Sampling the preamble every $T_{sym} + \Delta T$, a symbol will be missed.

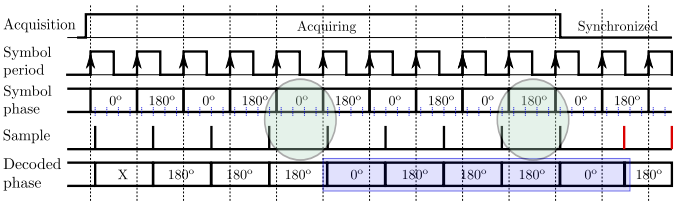

Fig. 7. The worst case synchronization with $m = 4$

reference. The fact of having two devices with two different clocks means that drifts among them are unavoidable.

For the QPSK receiver to operate correctly, the sensitivity period should be ideally centered in each symbol slot. However, in packetized wireless transmissions, there may be no need to have a constant control over synchronization: once synchronization is achieved (typically during a suitable preamble) we may rely on the stability of each (crystal-based) clock to keep synchronization during the whole packet length.

With this in mind, we have designed a preamble where the RF carrier has 180° changes in each symbol. Then, during acquisition, we try m different sensitivity points, spaced a period T_{adq} greater than the symbol period T_{sym}. Specifically,

$$T_{adq} = T_{sym} + \Delta T, \tag{1}$$

where ΔT is an integer number of carrier periods (to avoid introducing artificial phase changes due to the shift in time), i.e.

$$\Delta T = nT_c, \tag{2}$$

where $T_c = 1/f_c$. As the preamble is made of phase differences of 180°, if we observe two consecutive symbols, a phase difference of 180° will be observed. However, as $T_{adq} > T_{sym}$, occasionally we may observe a phase difference of 0°, indicating that a symbol was missed between observations (Fig. 6). This gives a direct indication that, with the available resolution, the best estimation of the optimum observation point lies in the middle of the last two points tried (Fig. 7), meaning we have to move $-T_{adq}/2$ relative to the last point. In this way, the error with respect to the true optimum is $\pm\Delta T/2$.

When a zero phase difference is detected, we may conclude that synchronization has been achieved. However, for robustness in the presence of noise, we wait until a second zero phase difference is obtained m time slots later. So, the synchronization process finishes when a complete sequence of two zero phase differences with $(m-1)$ 180° phase differences between them have been correctly detected. Otherwise the acquisition process is started again. This improves the robustness of the synchronization process against noise, which is the only component received when listening to the channel before the preamble is being transmitted. In the worst case, we will need a total of $2m + 3$ symbols to complete acquisition (Fig. 7).

TABLE I
MAXIMUM NUMBER OF DATA SYMBOLS FOR $r(ppm) = 20$

m	2	3	4	5	8	∞
N_{sym}	6200	8300	9300	10000	10900	12500

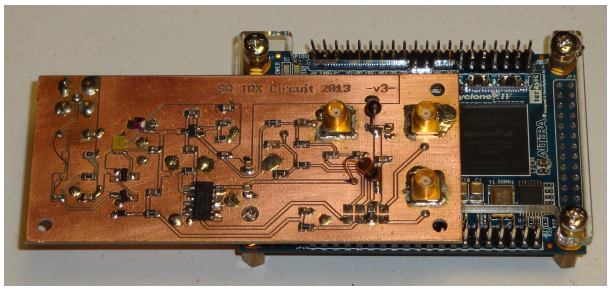

Fig. 8. A photograph of one of the implemented superregenerative QPSK transceivers.

Now, considering the uncertainty in t_{opt} and the relative clock drifts between transmitter and receiver, we may work out the maximum frame length that may be received successfully. Considering that both clocks have the same relative stability r (in ppm), in the worst case scenario, we get the number of symbols

$$N_{sym} = \frac{m-1}{m} \times \frac{1 - r10^{-6}}{4r10^{-6}} \simeq \frac{m-1}{m} \times \frac{1}{4r10^{-6}}. \tag{3}$$

Note that, from a practical implementation point of view, the required ΔT may not be realizable on the system clock available. For a given system clock period T_{CLK}, it is mandatory that

$$\Delta T = qT_{CLK} \tag{4}$$

with integer q. On the other hand, as (2) has to hold, we may have to relax the requirement of a constant $\Delta T = T_{sym}/m$. In this case, some of the m shifts may be slightly increased and others decreased, while ensuring that after m cycles we are observing the same time point. For instance, in our case, with a system clock of $f_{CLK} = 50\,\text{MHz}$, each symbol corresponds to 5000 clock periods. If we wish to take $m = 8$, we get $q = 5000/8 = 625$, which is not a multiple of 40, which is required from (2) and (4). In this case, we may count over the set $q = \{640, 640, 640, 640, 640, 600, 600, 600\}$ (or a permutation of it). A smaller ΔT, i.e a higher m in (3), allows a higher N_{sym}. Increasing m we quickly converge to the maximum value of N_{sym} as is shown in Table I, computed taking into account the previous practical implementation comments.

IV. Experimental results

We have built two transceivers using a bit rate of $f_{bit} = 20\,\text{kbps}$, i.e a symbol frequency of $f_{sym} = 10\,\text{kHz}$, a carrier frequency of $f_c = 26.25\,\text{MHz}$, and a clock frequency on the receiver of $f_{CLK} = 50\,\text{MHz}$. We have chosen $m = 8$, which means that the preamble must be longer than 19 symbols,

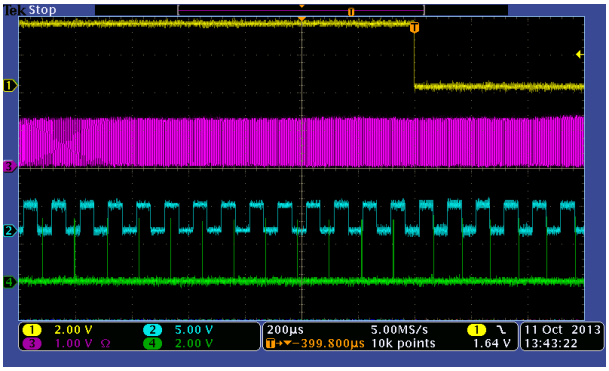

Fig. 9. Screenshot of the end of an acquisition process. From top to bottom: trace #1 is the acquisition signal, trace #3 is the QPSK-modulated signal, trace #2 is the receiver symbol clock and the trace #4 depicts the samples obtained and, indirectly, the time instants when they are obtained.

i.e. 38 bits, and the data field must be smaller than 10900 symbols for 20 ppm clocks. Both transceivers have been able to successfully exchange packets with frames of 40-bit preamble, 8-bit start delimiter and 960-bit of data (values which satisfy the previous requirements).

A proof-of-concept transceiver is shown in Fig. 8. It is composed of an analog PCB board which connects to a DE0-Nano prototyping board by means of an 8-pin header located on the bottom side. The analog board contains the SR front-end, similar to the one in [4], together with the circuitry related to the quench signal. The two MCX connectors on the right are the differential RF output while the MCX connector on the left (pointing towards the bottom) is the RF input. Another connector served as a test point. In this proof-of-concept implementation we have not included any filter to limit harmonic content of the signal. Of course, this would be required in a final implementation, together with means of switching a single antenna to the transmitter and receiver ports. For our across-the-lab test transmissions, we have just plugged in a short ($\sim$10 cm) length of cable into one of the transmitting MCX connectors. The SR receiver is sensitive enough to receive these signals without antenna (and without an input matching stage).

The generation of the quench signal by the circuit proposed herein has proven to be successful. Performance figures were almost indistinguishable from those using an external function generator -and comparable to the figures in [4] where sinusoidal quench was used. The proposed circuit is low-power and simple regarding component count (we used only one pin of the FPGA, two resistors and one capacitor), two points which are in accordance with the essentials of SR receivers.

The transceiver is able to switch between transmission and reception depending on a digital signal. In reception, the synchronization process has proven to work reliably. First, the receiver is placed into a synchronization-search state, which is exited after the synchronization has been achieved. After this, the start delimiter is detected and the frame is captured in a straightforward way.

Figure 7 showed a timing diagram of the synchronization process. Experimental waveforms of the same process are presented in Fig. 9. Here it may also be seen how each symbol is observed at a different position until synchronization is achieved.

Regarding complexity, including some debugging circuitry and without any special effort towards optimization, the digital part takes up 1750 logic elements, i.e. 8% of the resources in the FPGA on the low-end DE0-Nano development board [5].

The extension of this proof-of-concept to higher frequencies seems clearly feasible. The extension of the QPSK SR receiver is discussed in [4]. The synchronization principle is independent on the operating frequency and the only point that may require changes is the transmitter side. The direct generation described herein would have to be followed by an up-conversion stage. However, even in this case, the complexity is significantly lower than with a the conventional full-fledged IQ upconversion scheme.

V. Conclusion

In this paper, we have presented a working proof-of-concept QPSK SR transceiver operating in the HF band. We have shown how a QPSK transmitter can be efficiently implemented on an FPGA in an all-digital approach. Also, we have suggested a simple technique to generate the quench signal required for receiver operation. Next, and most importantly, we have described a technique to allow for proper synchronization in a packet-transmission context. To aid in synchronization, the transmitter emits a suitable preamble. In the receiver, the sampling operation inherent to SR receiver operation is performed at a rate lower than the symbol rate, which amounts to scanning between m possible synchronization points. The synchronization point is found when a symbol is missed. After achieving synchronization during the preamble, we rely on the stability of the transmitter and receiver clocks to allow the whole packet to be correctly detected. We have provided some expressions on the maximum data field length for given clock stabilities. Experimental prototypes have been built and tested, confirming the practical feasibility of the proposed approach.

References

[1] E. Armstrong, "Some recent developments of regenerative circuits," *Proceedings of the Institute of Radio Engineers*, vol. 10, no. 4, pp. 244–260, Aug. 1922.

[2] J. Ayers, K. Mayaram, and T. Fiez, "An ultralow-power receiver for wireless sensor networks," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 9, pp. 1759 –1769, sept. 2010.

[3] K. Kim, S. Yun, S. Lee, and S. Nam, "Low-power CMOS super-regenerative receiver with a digitally self-quenching loop," *Microwave and Wireless Components Letters, IEEE*, vol. 22, no. 9, pp. 486–488, September 2012.

[4] P. Pala-Schonwalder, F. Moncunill-Geniz, J. Bonet-Dalmau, F. del Aguila-Lopez, and R. Giralt-Mas, "A superregenerative QPSK receiver," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, pp. 1–8, Currently accessible online at IEEExplore. 2013.

[5] DE0-Nano development and education board. [Online]. Available: http://www.terasic.com.tw/cgi-bin/page/archive.pl?No=593

[6] J. Venkataraman and O. Collins, "An all-digital transmitter with a 1-bit DAC," *Communications, IEEE Transactions on*, vol. 55, no. 10, pp. 1951–1962, 2007.

[7] F. Moncunill-Geniz, P. Pala-Schonwalder, and O. Mas-Casals, "A generic approach to the theory of superregenerative reception," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 52, no. 1, pp. 54–70, Jan. 2005.