A Reconfigurable Quadrature Oscillator Based on a Direct Digital Synthesis System

João Gonçalves(1), Jorge R. Fernandes(2,3), Manuel M. Silva(2,3)

(1) INOV Lisboa, Portugal
(2) Instituto Superior Técnico / (3) INESC-ID Lisboa, Portugal

Abstract—We propose an architecture to realize an RF quadrature oscillator, in which a frequency generated by a Direct Digital Synthesis (DDS) system is added to (or subtracted from) the frequency generated by a Phase-Locked Loop (PLL). The DDS system is easily reconfigurable to change the channel spacing and bandwidth, and allows the implementation of several digital modulation schemes.

A computer program was developed to calculate the parameters of the DDS system, based on the specifications supplied by the user, and to generate the VHDL code of the digital part of the system. The DDS is designed to obtain outputs in quadrature with a minimum ROM area.

The DDS is implemented in a FPGA and has excellent quadrature relation throughout the frequency band of the system.

I. INTRODUCTION

Most wireless systems use different channels within a certain frequency band. Some systems have now to be adaptable to different communication standards, using different frequency bands, and with different channel spacing and bandwidth.

The local oscillator is usually implemented by a Phase-Locked Loop (PLL). PLLs can work in the GHz range, exhibiting good stability; however, changing the frequency is slow [1], and the channel spacing is unchangeable, once designed.

Direct Digital Synthesis (DDS) [2,3] generates analog signals with low frequency, in the tens of MHz range, but they are very flexible, can change frequency fast and, due to their digital nature, can be reconfigurable.

The combination of a PLL with a DDS to realize a local oscillator can be very advantageous: it can operate in the GHz range (due to the PLL), and channel changes of up to tens of MHz can be done with the DDS; in addition, there is the possibility of implementing digital modulation schemes.

The proposed oscillator is well suited, for instance, to wideband, multi-band, or frequency hopping applications.

We have developed a computer program to obtain the parameters of DDS systems, which generates the VHDL code of the digital part of the system.

DDS systems with quadrature outputs, can be designed using ROM compression techniques combined with address multiplexing, in order to keep the clock frequency at reasonable values and reduce the area.

An example of a DDS system designed with the proposed techniques is implemented in the ISM bands; accurate quadrature throughout the entire output band has been observed.

In Section 2 we describe the hybrid system obtained by the combination of a PLL and DDS. In Section 3 we present the DDS design tool and in Section 4 we explain the generation of quadrature outputs. In Section 5 we present experimental results from a prototype implementation in a FPGA. In Section 6 we draw some conclusions.

II. OSCILLATOR ARCHITECTURE

A PLL used as a frequency synthesizer in the analog domain is represented in Fig. 1, where a phase detector (PD, usually implemented by means of a mixer) compares the input reference frequency (f_{ref}) with the output frequency (f_{OUT}) or a multiple of it. The output of the PD is an error signal proportional to the phase difference (\theta_e). This is averaged by a low-pass filter and controls the frequency of a voltage controlled oscillator (VCO). The divider by N allows the generation of frequencies that are multiples of f_{ref}.

This work was supported by the Portuguese Foundation for Science and Technology (project POCTI/38533/ESE/2001).
PLLs can synthesize stable high frequency signals, but the frequency tuning is limited to a small number of discrete frequencies. PLLs require a time of the order of ten clock cycles to stabilize after a frequency change.

In a DDS system the frequency reference is the clock signal that controls the digital blocks. The most common DDS system is represented in Fig. 2, where sine values are stored in a read only memory (ROM) that is addressed by a circular accumulator. The accumulator step defines the frequency.

![Fig. 2 DDS block diagram.](image)

The ROM output is converted into the analog domain by a digital to analog converter (DAC) followed by a low-pass filter (LPF). The DDS system is easily reconfigurable, and allows a fast change of the output frequency, but the output frequency is limited (of the order of tens of MHz).

There are other techniques to implement a DDS [4,5]. We use the ROM based DDS, since it is simple to implement, and the ROM area can be minimized with appropriate techniques [6].

The combination of a PLL with DDS, can inherit the best features of the two systems. The wanted frequency can be obtained by single sideband mixing of the two frequencies. A high frequency is obtained by the PLL, and the DDS is used to perform small frequency changes (e.g., to change channels).

The combination of a PLL and DDS has been presented in [7,8], but in [7] the architecture is quite different, and in [8] two DDS blocks are used, and the implementation described has a single (not quadrature) output. We propose the Hartley architecture, which does not require a band-pass filter (Fig. 3). The quadrature oscillator/mixers in [9] can be combined with the DDS described here.

![Fig. 3 Hybrid Synthesizer based on the Hartley architecture.](image)

It can be shown that either the lower side band or the upper side band is obtained at the output of Fig. 3, depending on whether a sum or subtraction is performed. Thus, the channel frequency can be to either side of the PLL frequency, for a given DDS frequency: thus, the resulting frequency can change within a range that is twice the DDS frequency.

In the remaining of this paper we will focus on the design of the quadrature DDS system, which is a key portion of the new architecture.

III. DDS DESIGN TOOL

To design the DDS system, a program was developed that incorporates an algorithm to reduce spurious frequencies [10], and which generates the VHDL code. This software also allows simulation of the system and the observation of the expected outputs in the time and frequency domains.

The input data is: minimum channel spacing, clock frequency, DAC resolution, and minimum SFDR (Spurious Free Dynamic Range). The three main design parameters are determined: number of bits of the accumulator, number of bits to address the ROM, and number of bits of the quantifier. Interaction with the user is possible using a graphic interface (Fig. 4) in order to obtain the best compromise between the specifications, or to warn against impossibilities to meet them: for instance, it can warn that, with a specific DAC, the specification on the SFDR cannot be met.

![Fig. 4: Graphic interface of the DDS synthesis tool.](image)

Although the objective is to synthesize the digital part of the system, the DAC and the LPF are also implemented at behavioural level for simulation purposes.

IV. GENERATION OF QUADRATURE OUTPUTS

To reduce the ROM area we take into account the symmetry of the sine function, and we obtain quadrature outputs by using the same ROM, since $\cos(x) = \sin(x+\pi/2)$. The digital part of the designed quadrature DDS system is represented in Fig. 5.

The accumulator block contains a circular power of 2 accumulator for generating the sequence for addressing the memory and, it is also responsible for generating the signals that indicate the quadrant and sign of the samples.
A more detailed diagram of the memory block is represented in Fig. 6. It is composed of a ROM with $2^w$ sine samples and a register with a singular sample, since there is the need to have a full quadrant, which corresponds to $2^{w+1}$ words.

The Change Sign block (in Fig. 5) changes the sign of the sample when it is in the negative half-cycle. The Multiplexer block addresses sequentially the ROM for the in-phase and quadrature signals. This allows a significant reduction in area with respect to the alternative of using another ROM for the quadrature signal. The Demultiplexer block (Fig. 5) is used to select the output (in-phase or in quadrature).

In Fig. 7 we represent the main signals concerning the DDS operation, for accumulator number of bits $n=5$, number of bits of address $w=3$, and accumulator increment $\Delta r=1$. Fig. 8 shows the simulated outputs for two different frequencies, based on the generated VHDL. The two signals are accurately in quadrature.

V. EXPERIMENTAL RESULTS

We have used a development board Digilent D2-SB, with a FPGA Spartan II-E XC2S200E-PQ208 to implement the digital part of the DDS system, and we have built a board with discrete components to implement the analog part. The DDS should be able to deal with several frequency standards, as indicated in Table 1. The bandwidth is 25 MHz for all bands, except for the DECT, where it is 20 MHz. We will design the DDS to operate in the range [1,12.5] MHz, with a 2 kHz resolution (the channel spacing can be twice the DDS frequency, as pointed out above).
frequency of 50MHz, which means 25MHz for each of the quadrature components. Due to Nyquist theorem, this limits the DDS output frequency to 12.5MHz.

The system specifications are summarized in Table 2, and the implementation of the digital part in the FPGA uses the resources indicated in Table 3.

![Figure 8: IQ outputs with $f_{CLK} = 50$ MHz](a), (b) $f_{OUT} = 30$ kHz , (b) $f_{OUT} = 2.08$ MHz.

### TABLE 1: TELECOMMUNICATION STANDARDS FREQUENCY SPECIFICATIONS

<table>
<thead>
<tr>
<th>Standard</th>
<th>Transmission Band</th>
<th>Reception Band</th>
<th>Ch.Spacing</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS-54</td>
<td>824 – 849 MHz</td>
<td>869 – 894 MHz</td>
<td>30 kHz</td>
</tr>
<tr>
<td>IS-95</td>
<td>824 – 849 MHz</td>
<td>869 – 894 MHz</td>
<td>1.25 MHz</td>
</tr>
<tr>
<td>GSM</td>
<td>890 – 915 MHz</td>
<td>935 – 960 MHz</td>
<td>200 kHz</td>
</tr>
<tr>
<td>DECT</td>
<td>1.88 – 1.9 GHz</td>
<td>1.88 – 1.9 GHz</td>
<td>1.728 MHz</td>
</tr>
</tbody>
</table>

### TABLE 2: QUADRATURE DDS SPECIFICATIONS.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Clock</td>
<td>27.8MHz</td>
</tr>
<tr>
<td>Output Freq. band</td>
<td>[0, 12.5]MHz @ $f_{CLK}=25$MHz</td>
</tr>
<tr>
<td>Freq. resolution</td>
<td>$\Delta f=5.96$Hz@$f_{CLK}=25$MHz</td>
</tr>
<tr>
<td>Latency</td>
<td>4 clock cycles</td>
</tr>
<tr>
<td>Accumulator increment</td>
<td>22 bit (6 bits truncated)</td>
</tr>
<tr>
<td>ROM address</td>
<td>14 bit</td>
</tr>
<tr>
<td>DAC resolution</td>
<td>12 bit – I, 12 bit – Q</td>
</tr>
<tr>
<td>ROM size</td>
<td>192 kbit</td>
</tr>
<tr>
<td>SFDR</td>
<td>64 dBc</td>
</tr>
</tbody>
</table>

### TABLE 3: PERCENTAGE OF FPGA SLICES USED IN THE QUADRATURE DDS AND IN A SINGLE OUTPUT SYSTEM.

<table>
<thead>
<tr>
<th>Block</th>
<th>Slices used (% of area used)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accumulator(IQ)</td>
<td>39 (2.4%) 28 (1.80%)</td>
</tr>
<tr>
<td>Memory</td>
<td>1496 (92.5%) 1496 (97.4%)</td>
</tr>
<tr>
<td>Inverter</td>
<td>2x12 (1.5%) 12 (0.8%)</td>
</tr>
<tr>
<td>Mux IQ</td>
<td>9 (0.6%)</td>
</tr>
<tr>
<td>Demux IQ</td>
<td>14 (0.9%)</td>
</tr>
<tr>
<td>Control IQ</td>
<td>24 (1.5%)</td>
</tr>
<tr>
<td>Total</td>
<td>1618 (69% of 2352) 1542 (65% of 2352)</td>
</tr>
</tbody>
</table>

From Table 3 we see that the memory uses most of the FPGA resources, and that with the techniques proposed here, the generation of quadrature outputs represents a small increase of 4% in the number of slices. This dominance of the memory is, to some extent, due to the FPGA organization, and might be reduced in an ASIC implementation.

Fig. 9, shows an accurate quadrature relation. This accuracy deteriorates above about 9.4MHz, due mainly to the low number of samples per period. This would be overcome in an ASIC implementation, in which a higher clock frequency and faster digital circuits can be used.

### VI. CONCLUSIONS

In this paper we propose a new architecture for the realization of a quadrature oscillator, which combines a PLL with a DDS to obtain a digitally reconfigurable system that can operate with different communication standards.

The emphasis of the paper is on a new DDS implementation with quadrature outputs, which minimizes the area by storing one sine quadrant and by addressing the same ROM for the sine and cosine using address multiplexing. We developed a program that generates the VHDL for the DDS implementation.

Experimental results using an FPGA based implementation of the DDS confirm the advantages of the techniques proposed here.

### REFERENCES


