REGAL – Software Radio Implementation

D. Avagnina, G. Cordara, A. Costa Laia, F. Dovis, A. Gramazio, P. Mulassano

Politecnico di Torino

ABSTRACT

The deployment of the GALILEO system in Europe and in general the interest arising toward the satellite based navigation applications based on GPS and GLONASS has caused a growing interest towards the deployment of effective receivers. In particular it is interesting to design reconfigurable receivers able to deal with the GPS signals and to be reconfigured at no cost for the GALILEO signals, when they will be present. The aim of the REGAL (REconfigurable Receiver for GALileo) project is the design of a reconfigurable navigation terminal based on software radio techniques, which easily allows for the design of multi-standard receivers, by means of high-speed signal processing. A hybrid architecture made of a FPGA Xilinx XC4010XL and of a Texas Instruments (TI™) TMS320C6711 has been chosen for the implementation of the baseband processing section of the receiver.

The (TI™) TMS320C6711 DSK board implements the core of the hardware platform. It has been selected among the floating point TI products, in light of its cutting-edge features in terms of speed, available memory, and number of serial interfaces.

“This document was an entry in the TI DSP Challenge 2001, an annual contest organized by TI to encourage students from around the world to find innovative ways to use DSPs. For more information on the TI DSP Challenge 2001, see TI’s World Wide Web site at www.ti.com/sc/dspchallenge.”
Contents

Introduction........................................................................................................................................................................2
REGAL architectural scheme........................................................................................................................................................3
Processing rate constraints......................................................................................................................................................4
The FPGA section....................................................................................................................................................................4
The DSP section.......................................................................................................................................................................5
DSP-FPGA interface...............................................................................................................................................................6
 McBSP0 configuration.......................................................................................................................................................6
 McBSP1 configuration.......................................................................................................................................................7
 DSP procedures.................................................................................................................................................................8

Figures

Figure 1. REGAL receiver scheme........................................................................................................................................3
Figure 2. Detailed Scheme representing the dual DPLL and signals exchanged between FPGA and DSP devices.............7
Figure 3. Channel management flow-chart.......................................................................................................................9
Figure 4. Hardware schematics of McBSP0 buffer: (a) standard DSK board realization, (b) enhanced REGAL application 10

Tables

Table 1. McBSP data format................................................................................................................................................7

Introduction

Since the Software Defined Radio (SDR) concept was introduced [1], much progress has been made in making it a reality, and many software radio based systems for communications have been designed. The main goal in SDR is the design of a flexible system that can transmit and receive signals at virtually any frequency using any protocol, any of which can be reprogrammed almost instantaneously: a sort of “universal receiver”. Such a system places great demands on the data converter and processor technologies, since they should be able to deal in real-time with tens to hundreds of megasamples of data produced by the analog to digital conversion of wireless signals. From a processing point of view, the challenge in SDR is to exploit the three basic processor types, fixed architecture processors, Field Programmable Gate Arrays (FPGAs) and Programmable Logic Devices (PLDs), and programmable DSPs/RISCs/CISCs, in such a way as to optimize the trade-off among speed, power dissipation and programmability. In particular, the issue of programmability also includes the choice of programming language and the related issues of portability and reprogramming speed.

Recently, great attention has been devoted to the applicability of SDR techniques in order to design reconfigurable personal communication terminals, leaving unexplored the potentialities of SDR for other terminals as navigation receiver, or integrated hybrid communications/navigation receivers.
The REGAL prototype receiver for navigation signals is a navigation receiver designed according to the SDR philosophy. The baseband processing section is based on a reconfigurable hardware (FPGA Xilinx XC4010XL and Texas Instruments (TI™) TMS320C6711). Generally speaking, the SDR approach requires having the A/D converter as close as possible to the RF front end and limiting as much as possible the hardware used in the receiver. The present technology does not allow for such a totally software receiver neither for the navigation terminals, which have less stringent processing constraints with respect to personal communications receivers. In fact, they have to deal with much lower data rates. In this project only the baseband processing section has been implemented with the hypothesis of having a sampled data stream of 25Msamples/s after the A/D converter. The functionalities of the receiver have been tested through simulated signals.

The choice of limiting the fixed hardware and the use of FPGAs and DSPs yielded a REGAL architecture able to deal with the GPS and GLONASS systems, uploading the suitable software during the initialization phase. In the future the same strategy will be expanded to the GALILEO presence. For the basics of Global Navigation Satellite systems, and navigation receivers we address the reader to [2].

In the following sections the REGAL hardware architecture is described, together with digital demodulation processing implemented in the baseline prototype.

**REGAL architectural scheme**

The conceptual scheme of the REGAL receiver is depicted in Figure 1. The designed prototype is a six channels receiver able to acquire all satellite navigation signals currently available (GPS and GLONASS) and in the future scenarios (GALILEO). The SDR approach to the receiver design requires a preliminary evaluation of the computational load given by each functional block, in order to identify the most suitable implementation strategy. In fact, as also in the conventional receiver design, the optimal system is given by a trade-off between off among speed, power dissipation and complexity.

*Figure 1. REGAL receiver scheme.*
Processing rate constraints

From the RF stage, after the A/D conversion, a data rate of 25 Msample per second, represented on 3 levels (2 bits), is expected according to realistic assumptions. Obviously a real-time processing is required and the data rate is too high for a direct interface with a DSP. For this reason hybrid architecture made of a FPGA Xilinx XC4010XL and of a Texas Instruments (TI™) TMS320C6711 has been chosen for the implementation of the baseband processing section of the receiver. Particular attention has been devoted to the design of the communication interface between the DSP and the FPGA. It has to be remarked that the constraints on the speed and complexity also depend on the methods chosen for the de-spreading process and for the codes alignment procedures.

The FPGA section

The demodulation and decoding operations of the CDMA signals incoming from the satellites require a high data processing rate. In a GPS receiver the demodulation of several satellite signals (channels) must be performed at the same time. These requirements for a parallel and onerous processing capability have induced an hardware implementation of the whole de-spreading process. It has to be remarked that the maximum number of channels that the receiver is able to manage, depends on the FPGA complexity (i.e. on the number of gates). In the REGAL prototype four channels have been implemented, but extension to a larger number of channels is straightforward.

The decoding section has been implemented on a FPGA Xilinx XC4010XL, programmed using the Xilinx Foundation Tool F3.1.

The implementation of each functional block has been optimized for the chosen FPGA, in order to limit the number of gates used to realize the functionalities briefly described in the following.

- **Doppler removal**: due to the satellite clock imperfections and to the Doppler effects, the signal sampled by the A/D converter is affected by a residual modulation. The entity of Doppler shift is about ±5 kHz at a frequency of 1.575 GHz (L1 GPS channel frequency, used as reference). The main task of this block is to remove such spurious modulation through a digital Single Side Band (SSB) modulator, in order to obtain the required phase rotation. This block returns in output a signal represented on 4 bits (3 amplitude bits and 1 sign bit).

- **Carrier NCO**: a frequency reference oscillator pilots the SSB modulator of the Doppler removal block. It has to be able to dynamically adapt to the variations given by the Doppler frequency shift. To this aim a numerically controlled oscillator (NCO) is used. At each clock pulse an accumulator register is incremented of a fixed value \(\Delta f\). When a carry is generated, the NCO output changes its state. A 30 bits register is used to allow an adequate frequency resolution.

- **Code Generator**: in order to demodulate the CDMA signals incoming from the satellites, a local replica of the spreading code has to be generated and variable delay between codes is realized with a resolution of 0.5 chip. This block generates the codes according to the
system specifications [3]. The codes are generated through shift-registers with proper feedbacks.

- **Code NCO**: because of the Doppler effect, not only the carrier phase but also the chip period of the CDMA codes dynamically changes on the basis of the relative speed between the satellite and the receiver. The Doppler shift causes a maximum chip period variation of \pm 2.76 chips at a frequency of 1.575 GHz. Thus, a variable frequency reference oscillator must pilot the PN (Pseudo Noise) sequence generator. The implementation of the Code NCO is conceptually identical to the realization of the Carrier NCO, but requires a lower frequency resolution. A 20 bits register is used.

- **Code removal**: this block is actually in charge of the CDMA demodulation removing the spreading effect of the PN codes, via a multiplication by the locally generated and aligned code. The block output is the navigation message, a low bit rate data stream.

- Since the phase relations between the locally generated and the received codes changes with the relative satellite-receiver speed, a closed loop system must be implemented. Through the DSP section a double phase lock loop system is realized, as will be detailed in the following sections.

**The DSP section**

The processing of the FPGAs outputs is demanded to a DSP TI TMS320C6711. The functionalities of the DSP section are described in the following.

- **Code alignment**: the Doppler shift leads to a change in the carrier frequency and also affects the code chip period. During the acquisition phase of a new satellite signal the locally generated code has to be aligned with the received one. To this aim a search for the “best” local code has to be performed both in the time domain (chip duration) and in the frequency domain (carrier central frequency). Several methods are usually implemented, as described in [4]. In the REGAL prototype the DSP realizes a fast parallel search in the frequency domain according to a method explained in [5]. The DSP performs an evaluation of the signal correlation in the frequency domain, by means of a FFT working on 2048 samples (2 samples per C/A code chip are used) as described in [5].

- **Code tracking**: once the phase relations have been evaluated it is necessary to maintain the code alignment through a tracking phase. To this aim the DSP realizes the double phase lock loop, jointly operating with the FPGA timing circuitry. In order to implement the lock in the time domain an Early/Late Delay Lock Loop (DLL) is implemented by the DSP. Delay between Early-Punctual-Late replicas is variable from 0 to 1 chip with steps of 1/16 chip. The DLL keeps the code locked. As far as the frequency domain is concerned, the DSP implements a Phase Lock Loop (PLL), which keeps the carrier synchronism, tracking the frequency shifts due to the Doppler effect with a resolution frequency of about 1 Hz. To realize this PLL a two-steps technique described in [5], is used:
  - At first step, a 128 samples FFT is performed in order to obtain an initial rough spectral estimation,
  - Then the estimated frequency value is passed to the digital PLL, which refines the frequency synchronism.
• **Channel Management:** since the navigation systems use Low or Medium Orbit satellites, the constellation scenario changes with time. The DSP acts as a supervisor of all the four channels, dynamically managing the acquisition and tracking phases of the available satellites. After having estimated its own location, the receiver is able to predict the future constellation in view, thanks to the interpretation of the almanac embedded in the navigation message.

• **Position Estimation:** the DSP actually estimates the receiver position, solving the navigation equation. With a rate definable by the user, the DSP starts the calculation of the navigation equations; the results are used to refresh the estimated receiver position, the satellite almanac and other parameters.

### DSP-FPGA interface

The TI processor TMS320C6711 presents two programmable serial ports McBSP, used by the DSP component to communicate with the FPGA hardware, as shown in Figure 1. One of these connection ports (McBSP1) is used to realize the dual phase locked loop, while the other one (McBSP0) transfers I-Q data, sampled by the A/D conversion stage, to DSP. Such data are used to evaluate the code alignment. Figure 2 shows signal connections between DSP and FPGA.

Two McBSP ports equip TI TMS320C6711 DSK version, but only one is available for an external development of daughter boards. In particular McBSP0 is used as timer for on-board Audio Codec and it is not re-configurable as external interface. The operation described is possible on older DSK boards.

In order to realize REGAL prototype, on-board Audio Codec has been shouted off, making available McBSP0, as depicted in Appendix A. Figure 4 shows signal connections between McBSP0 and FPGA. The DSK board has been enhanced with the possibility of excluding the Audio Codec and to make available the McBSP0 signals (CLKR0, FSR0, DR0) in receiving mode. U34 and U35 buffers have been turned in a three state configuration.

### McBSP0 configuration

Since code received by one generic navigation satellite is arbitrary delayed with respect to the locally generated code, the receiver must be able to estimate this delay in order to track satellite signal. McBSP0 serial port is used in receiving mode during the code alignment phase, to send data from FPGA to DSP; data, represented on 8 bits, are transferred with a rate of about 16Mbit/s (see Table 1), and the handshake is controlled by the FPGA device. In order to manage McBSP serial connection ports, TI processor TMS320C6711 is controlled by CPU interrupts. Data transfer for the FFT data takes from 5ms to 20ms, during which the CPU is busy. It has to be remarked that the FFT data transfer is required only when receiver loses satellite lock or when the signal of a new satellite is received.

---

1. TI web site offers a large documentation database about DSP boards, made of schematics, application notes and other technical references. Some problems have been met, using the EDMA of TI TMS320C6711; the file spra711.pdf reports a clear scheme about all the physical connections for McBSP0, but they do not exist on the hardware device.
 McBSP1 configuration

When the delay between the incoming code and the locally generated one is known, the dual DPLL (Digital Phase Locked Loop) must track the signals. TI DSP calculates needed correction for a correct running of two DPLLs, useful to recover the carrier and the code synchronization. This information is sent to the FPGA through McBSP1 serial port, used in transmission mode. At every C/A code cycle, FPGA device sends synchronization data to DSP, which calculates the correction feedback for both DPLLs, refreshing their parameters with 1ms clock rate. Data incoming from the FPGA is represented on 16 bits, while correction feedback on 32 bits (see Table 1). Transfer rate is about 16Mbit/s and FPGA hardware controls the timed handshake. The processing management of all data related to each channel is realized using a TDMA (Time Division Multiple Access) technique. McBSP1 data transfers are controlled by CPU interrupts and require about 10% of CPU time.

Table 1. McBSP data format.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Data Type</th>
<th>Acronym</th>
<th>Data Size</th>
<th>Data Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>McBSP0 RX</td>
<td>I-Q Samples for Code Alignment</td>
<td>IQDATA</td>
<td>8 bit</td>
<td>16.368 MHz</td>
</tr>
<tr>
<td>McBSP1 RX</td>
<td>I-Q Early/Late Correlation Values</td>
<td>IQCORR</td>
<td>16 bit</td>
<td>16.368 MHz</td>
</tr>
<tr>
<td>McBSP1 TX</td>
<td>DPLL Correction Values</td>
<td>DPLLERR</td>
<td>32 bit</td>
<td>16.368 MHz</td>
</tr>
</tbody>
</table>

Figure 2. Detailed Scheme representing the dual DPLL and signals exchanged between FPGA and DSP devices.
DSP procedures

The highest-level DSP task is channel management. The maximum number of channels (up to 12) is software definable and depends on hardware complexity of FPGA device. DSP tests if each receiver channel is correctly locked on a visible satellite and if the dual DPLL keeps the code tracking. The first task runs as main routine in cyclic mode, following the flow chart depicted in Figure 3. This procedure also checks each channel state. FPGA-DSP system realizes the dual DPLL and their parameters are continuously updated. Tracking procedure runs in background through CPU interrupts with a frequency of 1kHz (C/A code repetition frequency).

At each interrupt, DSP receives DPLL data from every channel in sequence using McBSP1 port and computes new correction factors to send to FPGA board. Usually when a locked satellite is lost a new code alignment is required. In this case FPGA device transmits needed data by McBSP0 to DSP that calculates phase delay between incoming code and locally generated one using the FFT operator. ISR (Interrupt Service Routines) allows realizing required transfer in background mode.

At regular intervals, the DSP calculates the actual receiver position, solving an equation system of fourth order. Known values are parameters extracted from navigation message and position estimation is ready at software definable intervals (up to 1 measurement per second) and is activated through Timer1. This routine works on a low priority interrupt, without interfering with ISR tracking routines, which are at high priority.
Figure 3. Channel management flow-chart.

References

Appendix A

Figure 4. Hardware schematics of McBSP0 buffer: (a) standard DSK board realization, (b) enhanced REGAL application.