

MicroTCA Compliant WiMAX BS Split Architecture with MIMO Capabilities Support Based on OBSAI RP3-01 Interfaces

Cristian Anghel and Remus Cacoveanu
*University Politehnica of Bucharest,
Romania*

1. Introduction

Modern mobile communication systems must fulfill more and more requirements received from the customers. This leads to an increase of complexity. The control part of the system becomes very important, a multi-level approach being needed. With respect to this, all BS (Base Stations) from a system are synchronized using GPS (Global Positioning System) or IEEE 1588 [1] standard, high speed synchronous interfaces are used between the BBM (Baseband Modules) and the RRU (Remote Radio Units), for example OBSAI (Open Base Station Architecture Initiative) [2, 3] or CPRI (Common Public Radio Interface) [4], and standard communication methods are provided between the control parts placed in different levels of the system.

This chapter describes the management and synchronization procedures for a WiMAX BS architecture compliant with MicroTCA standard (Micro Telecommunications Computing Architecture) [5]. The block scheme of such a BS for the case of a 3 sectors cell is presented. One can observe the main parts of the MicroTCA standard, i.e. the MCH (MicroTCA Carrier Hub) modules and the AMC (Advanced Mezzanine Card) [6] modules.

Referring now to the OBSAI RP3-01 interface, this represents an extension of the RP3 (Reference Point 3) protocol for remote radio unit use. The BS can support multiple RRUs connected in chain, ring and tree-and-branch topologies, which makes the interface very flexible. Also, in order to minimize the number of connections to RRUs, the RP1 management plan, which includes Ethernet and frame clock bursts, is mapped into RP3 messages. This solution is an alternative to the design in which the radio module collocates with the BBM. Although in such a case the interface between the radio unit and the BBM becomes less complex, the transmitter power should be increased in order to compensate the feeder loss. For the proposed WiMAX BS block scheme, some improvements can be done starting from the proprieties of OBSAI RP3-01 interface. In this proposed BS split architecture, a BBM is connected to the two RRUs in order to have multiple transmit/receive antennas for MIMO capabilities. The connection between the two RRUs is realized using a chain topology. In order to obtain a single point failure redundancy scheme, a second BBM connected to the two RRUs is required. Only one BBM will be active at the

time. There are also described the OBSAI RP3-01 Interfaces required for blocks interconnection. Note that on OBSAI RP3-01 interface of each RRU the same Transport and Application Layers serve the both Physical and Data Link Layers.

This chapter is organized as following: Section 2 describes briefly the MicroTCA standard and the most important elements of such an architecture. Section 3 proposes a simple and efficient method of synchronizing a WiMAX BS using the GPS signals. There are provided synchronization signals for the air interface, in order to avoid interferences with other BSs. Also there are obtained, based on this proposed method, synchronization signals used inside BS with the scope of aligning all the modules of the architecture, which is very important when split solution is adopted, i.e. not all the units are co-located in the same physical element. Finally, Section 4 proposes a new way of using OBSAI RP3-01 Interface in a WiMAX BS, this new implementation solution providing support for MIMO techniques and redundancy.

2. MicroTCA standard – Overview

The MicroTCA standard is created by PICMG (PCI Industrial Computer Manufacturers Group) and it defines the requirements of chassis hardware system. Such a system uses AMC (Advanced Mezzanine Card) modules interconnected by a board having a high speed interface on the backplane of the chassis. The standard defines the mechanical, electrical and management specific characteristics needed for supporting AMC standard compliant modules.

The described structure is a modular one. By the configuration and the interconnection of the AMC modules inside the chassis, a high variety of applications can be obtained. Besides this, the standard doesn't impose a certain physical configuration of the chassis and neither a mandatory signaling protocol for the backplane high speed interface. Instead, a set of communication and interconnection requirements is defined. This set of requirements should be available for any structure, providing this way a high compatibility between the equipments compliant with the standard.

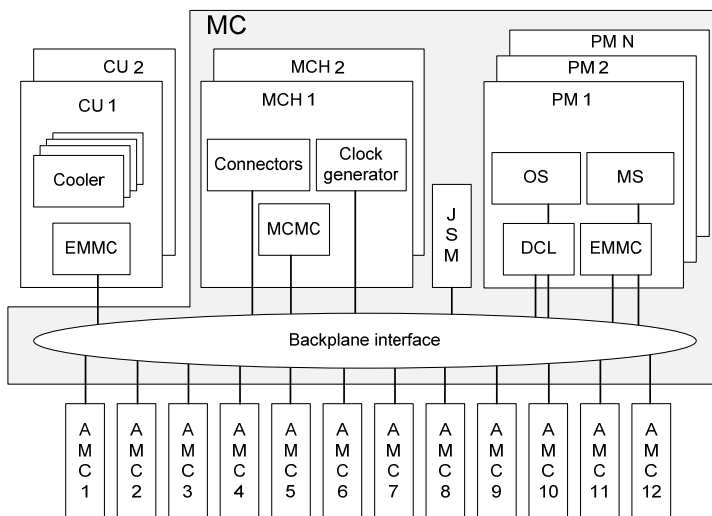


Fig. 2.1 MicroTCA – Block scheme

The proposed architecture is a modular one, as one can see from Figure 2.1. It results in a very flexible solution, allowing a high diversity in AMC modules implementation and in obtained functions.

A MicroTCA chassis is made of 1 up to 12 AMC modules, which will realize together the system functionality. Then there is a MCH (MicroTCA Carrier Hub) module which represents the support for the implementation of the main system management functions. On the chassis there can be found also PM (Power Modules) modules used for power supplying, CU (Cooling Unit) modules used for temperature control at system level, interconnection elements between the modules or with the external inputs / outputs (Backplane, Faceplate) plus other mechanical elements and redundant modules. A second MCH module and a second CU module can also be present in a chassis, from redundancy reasons.

2.1 AMC modules

The AMC modules are the main components of a MicroTCA chassis, containing the elements which will provide the system processing functions. There can be listed here the microcontrollers, the digital signal processors, the routers, the memory blocks, the I/O interface controllers, the base band and RF processing modules.

Initially, in AdvancedMC specifications, the AMC modules were defined as additional boards used for CB (Carrier Board) functionality extension. In MicroTCA, the AMC modules totally perform the system processing, while CB will be distributed between different architecture elements, having only a support role.

Due to the signal processing functions, the management tasks implemented on the AMC module are to be reduced as much as possible, in order to provide the maximum of the resources to the main process. This is the reason why these modules are controlled by a low level functionality entity called MMC (Module Management Controller). The set of functions this entity is performing is very simple so that it can be implemented on a low cost processor. The communication between MMC and a dedicated management entity at chassis level is made through IPMB-L (Intelligent Platform Management Bus - Local) interface, using a reduced set of requests/ confirmations specified in IPMI v2.0 (Intelligent Platform Management Interface) [7] standard.

The IPMB-L connections are isolated between each others in order to avoid the case when module issue is blocking the complete system, as in the case of bus topology.

The most important advantage introduced by this standard is the possibility of introducing/ switching in the system any module without being required to stop the power supply (Hot Swap feature) or to make any other hardware/ software modification (Plug and Play feature).

2.2 MicroTCA Carrier

MC (MicroTCA Carrier) is the novelty proposed by MicroTCA and it represents the main board, as defined by AdvancedTCA. It is responsible for the power distribution, the interconnection and the IPMI management for the 12 AMC modules. MC components are distributed in the chassis and are described in Figure 2.1.

- The power distribution infrastructure

It provides and controls the power distribution for each AMC module. The standard indicates the existence of 3 functional aspects:

- the operational supply OS providing 12V to each AMC
- the management supply MS providing 3.3V to each AMC. OS and MS are separated sub-systems in order to ensure the isolation between the processing processes and the management ones.
- the distribution control logic DCL being responsible of the protection, isolation and validation functions for each network branch

The PM units include also system surveillance functions required for the management part. The PM circuits should detect the units in the chassis, should monitor the parameters quality on each branch and should provide protection for overload. Part of these functions is made locally by a low level intelligence entity called EMMC (Enhanced Module Management Controller), but the system will be controlled by the Carrier Manager, which will compute the power budget based on the FRUI (Field Replaceable Unit Information) tables before validating the power distribution for AMC units. This process is similar with the AdvancedTCA one, and it is described in Figure 2.2.

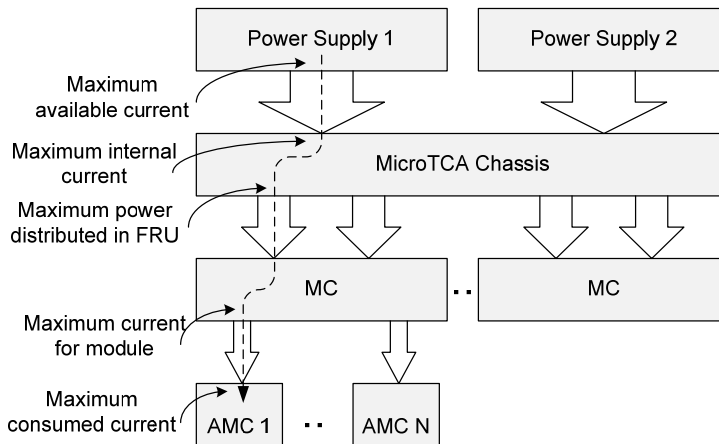


Fig. 2.2 Power distribution infrastructure

- Test infrastructure

It is represented by a JTAG switch called JSM (JTAG Switch Module). This allows the verification of the chassis, together with the modules inside, both in production period and in normal functioning period. As an alternative solution, serial asynchronous UART interfaces are implemented for the same testing scope.

2.3 MicroTCA Carrier Hub

This module combines the control and management infrastructure with the interconnection one, as depicted in Figure 2.3. It provides support for the 12 AMC units. Also, it is available

for all the other modules in the chassis. Having this important role, the MCH module is a critical point of the MicroTCA architecture, and for this reason it is required to have another MCH module in the chassis for redundancy.

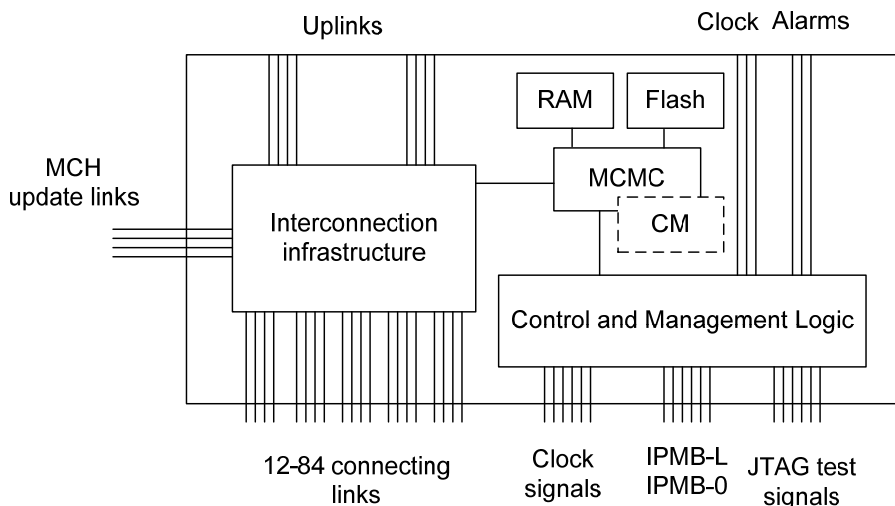


Fig. 2.3 MCH - Block scheme

The MCH module represents the physical support for a set of control and management functions called CM (Carrier Manager), which is the main authority in MicroTCA Carrier. It has to deal with the power control, the AMC network connecting management, the IPMI control and management, the E-Keying functions, the Hot Swap functions and the synchronization at system level.

The MCH structure and functionalities are based on the following modules:

- MCMC (MicroTCA Carrier Management Controller). Using the IPMB-L interface, the AMC detection signals and the validation signals provided by the PM units, the MCMC is monitoring and controlling the AMC units through CM. On the same time, MCMC acts as MMC for the MCH.
- Control and Management Logic block. It generates and distributes the clock signals to all the AMC units. It also provides the system level synchronization, using a reference clock received from an external module with universal synchronization capability (for example a GPS receiver) and generating a set of clock signals for the system components, with a required precision.
- Interconnection Infrastructure. It represents the main communication path between the AMC units. It includes a switch and several high speed serial interfaces (1..10 Gb/s) which create a star network connecting the modules based on PCI Express or Ethernet protocols.

The logical link between the modules connected like this is made by the E-Keying function provided by the Carrier Manager. This function verifies that all the AMC units from a chassis are electrically compatible before giving the authorization to enter in the network.

2.4 Base station components

Based on the above description, the MicroTCA architecture totally fulfills the requirements of a mobile communication base station implementation. Properties like modularity, flexibility, high cooling capacity and low cost are supported by the standard. In addition, the MCH unit ensures an efficient control of the system elements and provides information required by the network high layers. The AMC modules have also an important role, being responsible of base band processing and RF processing that are required by a mobile communication system. A WiMAX base station example is described below based on the described MicroTCA architecture.

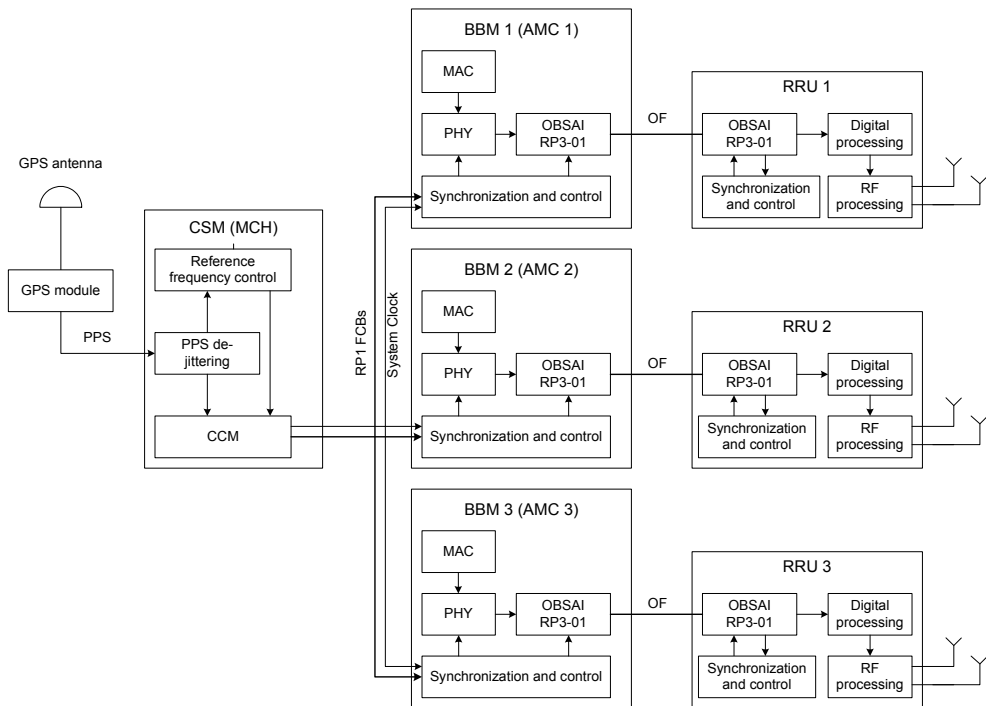


Fig. 2.4 WiMAX BS for a cell with 3 sectors

Figure 2.4 describes the WiMAX base station main components for the case of a cell with 3 sectors, each sector providing support for a certain level of diversity at transmission and reception. The main processing components are:

- the GPS module [8] equipped with a GPS antenna: this unit is used for generating a signal called PPS (Pulse Per Second) which is synchronized with the universal reference extracted from the GPS system
- the CSM module (Control, Synchronization and Management): it contains the algorithm used for reducing the PPS signal jitter. The new generated PPS signal controls the

system reference frequency generator. On the same time there are build in CCM (Control and Clock Module) the RP1 (Reference Point 1) synchronization signals used for OBSAI Interface and for Radio Interface. These FCBs (Frame Clock Burst) are sent time multiplexed to each BBM (Base Band Module) on a serial interface. On another serial interface all BBMs are receiving the system clock which will be used directly or to control the frequency generated by a local quart.

- the BBM module: it makes all the base band processing required for transmitting and receiving data in the system. There are included here the MAC and the PHY levels, as they are described by IEEE 802.16e standard. A block called Control and Synchronization is also part of the BBM. It is responsible of extracting the FCBs used for OBSAI interface synchronization, the FCBs used for radio interface synchronization being at this point encapsulated over the RP3 data stream inside the OBSAI RP3-01 interface and then being sent to RRUs (Remote Radio Unit) on the optic fiber.
- the RRU module: it is the external unit which, besides a possible digital processing (the decimation/ interpolation filters used for the selected channel, for example), performs all the radio domain tasks. This module may have multiple transmission/ reception paths and so, multiple antennas. The way the OBSAI interface is used for fulfilling this kind of requirements will be presented next in this chapter.

Having on one hand the WiMAX base station modules characteristics and on the other hand the MicroTCA architecture, one can identify that the CSM module has MCH specific functions, while the BBMs can be considered as being AMCs. Of course that besides these main modules, the power suppliers and the cooling units have to be added as WiMAX base station components.

3. WiMAX base station GPS based synchronization

3.1 Introduction

In communications systems using TDD (Time Division Duplex), appropriate time synchronization is critically important. In order to avoid inter-cell interference, all base stations must use the same timing reference. One solution to this problem is the Global Positioning System (GPS). The users can receive accurate time from atomic clocks and can generate themselves synchronization signals. Commonly the GPS receiver generates a Pulse per Second (PPS) signal and, optionally, a 10 MHz signal, phase synchronized with the PPS.

All the transmission over the radio channel, both on downlink and uplink, should be synchronized with the PPS signal. The RP1 synchronization burst generator, called Clock Control Manager (CCM) shall provide frame timing and time stamping for each of the air interface systems independently. The quality of the PPS signal will dictate the periodicity of these synchronization bursts. Also, algorithms for maintaining the stability of the clock reference, which can be affected by the temperature variance or by aging, can be developed based on the PPS signal. It is obvious why the PPS jitter level is a critical parameter in obtaining high synchronization performances [9].

This document will describe the digital method used for PPS de-jittering and the VCXO (Voltage Control Crystal Oscillator) oscillating frequency controlling algorithm.

3.2 Clock reference controlling scheme

The controlling scheme is a hybrid one, using both analog and digital elements. The scheme is depicted in Figure 3.1. In this application, the PPS input is sourced by a low cost GPS receiver called Resolution T, produced by Trimble.

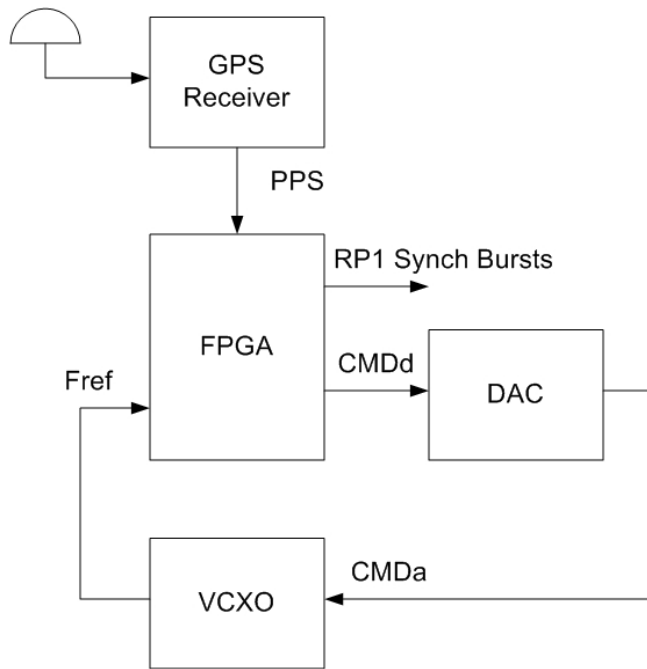


Fig. 3.1 Controlling scheme

The scheme works as follows: the Field Programmable Gate Array (FPGA), which is a XC3S500E chip representing a Spartan 3E family member produced by Xilinx, increments a counter value on every F_{ref} rising edge and resets this counter on every PPS pulse. Let's consider the nominal frequency F_{ref}^n and the counter value at a time instant $count_val$. When a new PPS pulse is received from the GPS module, before the counter reset, his value is stored and compared with F_{ref}^n . If the two values are not equal, the digital block computes a digital command CMD_d that is converted into a voltage level by a Digital to Analog Converter (DAC). The analog command CMD_a controls the VCXO and the value of F_{ref} is changed accordingly.

As it was mentioned before, the PPS jitter can produce VCXO commands that are unnecessary or imprecise. This is the reason way the PPS signal from the GPS receiver is passed through a digital de-jittering block before it is used by the controlling algorithm and by the CCM. The bloc scheme of the digital part of the structure described in Figure 3.1 is depicted in Figure 3.2.

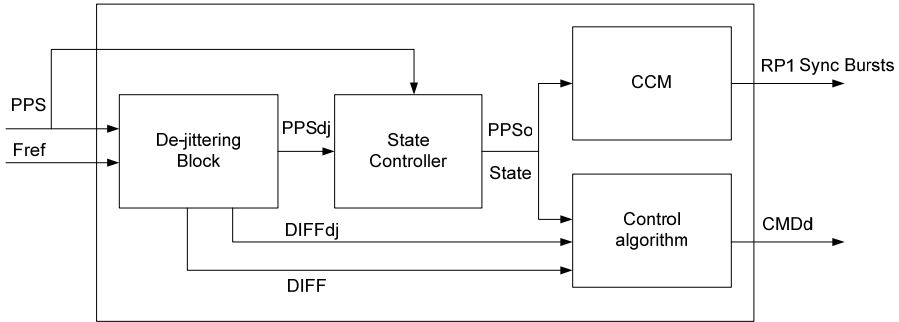


Fig. 3.2 The bloc scheme of the digital part

A. The de-jittering block

The PPS jitter characteristics are to be presented now. Figure 3.3 depicts the time instant value of the jitter. One can see from this figure that the PPS jitter is in the range $\pm 20ns$ for the selected GPS receiver. Also is easy to observe that it does not have uniform distribution. For this reason a simple mean will not eliminate the jitter problem (see Figure 3.4).

Figure 3.5 depicts the Allan deviation. For all of these measurements, it is assumed that the function $e(t)$, representing the time error (the deviation from 1 second value), is sampled with N equally spaced samples, $e_i = e(i\tau_0)$, for $i = (1, 2, \dots, N)$, and with a sampling interval τ_0 of 1 second. The observation interval, τ , is given by $\tau = n\tau_0$. The Allan deviation is computed using equation 3.1:

$$ADEV(n\tau_0) = \sqrt{\frac{1}{2n^2\tau_0^2(N-2n)} \sum_{i=1}^{N-2n} (e_{i+2n} - 2e_{i+n} + e_i)^2} \tag{3.1}$$

where $n \in \left(1.. \left\lceil \frac{N-1}{2} \right\rceil\right)$

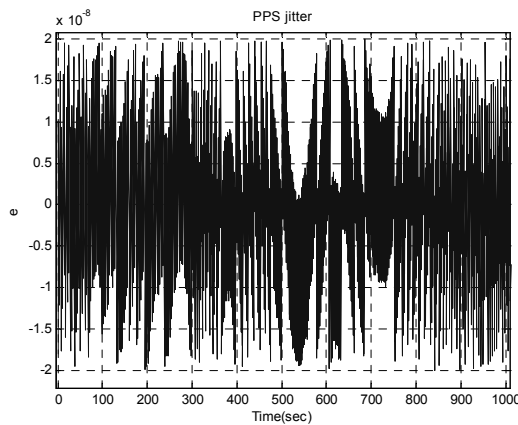


Fig. 3.3 PPS jitter

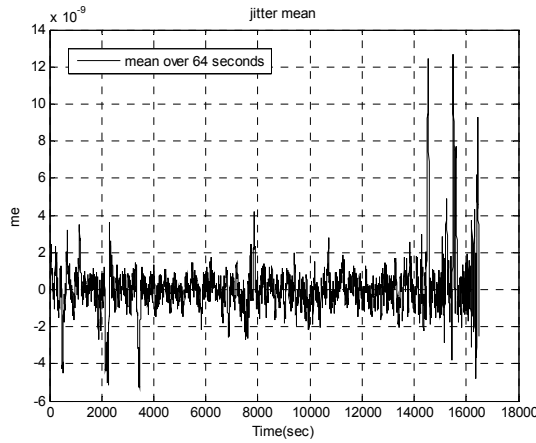


Fig. 3.4 PPS jitter mean

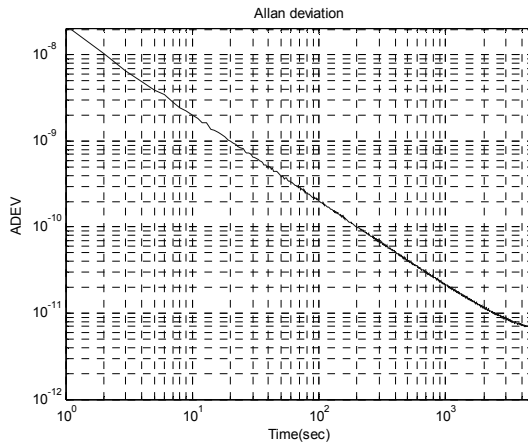


Fig. 3.5 PPS jitter Allan deviation

The slope of $ADEV(\tau)$ is τ^{-1} , which corresponds to white noise phase modulation and flicker phase modulation [1].

The de-jittering block contains a discrete-time Kalman filter. We will consider a particular algorithm of one-dimension Kalman filter intended for frequency estimation only in oscillators if GPS timing signals are used as the reference ones [10]. As it was mentioned before, on every PPS pulse we compute:

$$DIFF(n) = count_val(n) - F_{ref}^n \tag{3.2}$$

If the oscillator frequency is F_{ref}^n then $DIFF(n)$ will reflect only the PPS jitter. If not, the $DIFF(n)$ will contain the frequency deviation also. These values, computed every second, are used as the Kalman filter input.

Using the notations Q for process variance and R for estimate of measurement variance, the de-jittering algorithm is as follows:

$$\begin{aligned}
 & \text{Initialization} \\
 & \tilde{x}(1) = 0; \\
 & P(1) = 1; \\
 & \text{for } n = 1 : N \\
 & \quad \text{(Time update)} \\
 & \quad \tilde{x}_-(n+1) = \tilde{x}(n) \\
 & \quad P_-(n+1) = P(n) + Q \\
 & \quad \text{(Measurement update)} \\
 & \quad K(n+1) = P_-(n+1) / (P_-(n+1) + R) \\
 & \quad \tilde{x}(n+1) = \tilde{x}_-(n+1) + \\
 & \quad \quad K(n+1)(DIFF(n) - \tilde{x}_-(n+1)) \\
 & \quad P(n+1) = (1 - K(n+1))P_-(n+1) \\
 & \text{end}
 \end{aligned} \tag{3.3}$$

The $DIFF_{dj}$ signal from Figure 3.2 is the filter output, i.e. $\tilde{x}(n)$, and it is used to compute the digital command for the VCXO. Also, the de-jittering block provides a de-jittered PPS pulse, denoted PPS_{dj} which should have a 1 second period.

B. State Controller

Some times, due to the lack of visibility, the GPS receiver might not transmit the PPS pulse. This situation should be detected by the State Controller by expecting the PPS pulse within a time window. This window depends on the oscillator stability. If the oscillator has a $\pm 25ppm$ variation within the temperature range and a nominal frequency of 153.6 MHz, then the maximum delay of the PPS pulse can be $153.6e6 \times 25e-6 = 3840$ clock periods, i.e. the PPS pulse can be found after the previous one at $153.600.000 \pm 3840$ clock periods. If it is not so, the State Controller block confirms the absence of the PPS pulse.

The Finite State Machine (FSM) of the synchronization block is depicted in Figure 3.6. The four possible states are:

- IDLE: when the synchronization block waits for the first PPS
- TRAINING: when the synchronization block starts the Kalman filtering and waits T_{TR} seconds in order to obtain a stable output
- NORMAL: when the synchronization block works based on PPS pulses received from the GPS module

- HOLD OVER: when the PPS pulse is not received from the GPS module and the synchronization block works based on local PPS pulse.

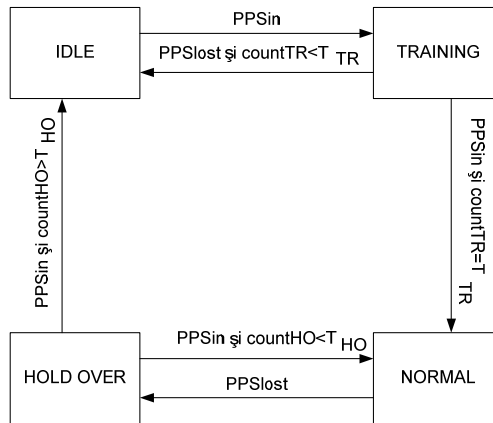


Fig. 3.6 FSM for synchronization block

After the first PPS received, the synchronization block switches from IDLE to TRAINING. In TRAINING, a counter called $count_{TR}$ is incremented on every PPS pulse. If the counter value equals the T_{TR} parameter, then a transition is made in NORMAL state. Else, if the block declares the absence of the PPS pulse and the counter value is less than T_{TR} then the new state becomes IDLE.

When the FSM is in NORMAL state and the PPS is declared to be absent, a transition to HOLD OVER state is made. In this state, from the last PPS pulse received, a counter is started in order to generate an internal PPS pulse. Also, a counter called $count_{HO}$ is incremented on every local PPS pulse, counting the number of successive absent external PPS pulses. If this counter reaches T_{HO} parameter the synchronization bloc state becomes IDLE. Else, if a new PPS pulse is detected before the counter reaches the T_{HO} value, the synchronization block returns in NORMAL state.

The values of the FSM parameters are given in Table 3.1.

Parameter	Value	Unit
T_{TR}	192	sec
T_{HO}	Depending on VCXO	sec

Table 3.1 FSM parameters

C. Control Algorithm

The Control Algorithm block receives the Kalman output and the state of the synchronization block. It also receives the Kalman input, as one can see from Figure 3.2. The control algorithm should compute the VCXO command. The DAC has a 16-bit input, so 2^{16} values are used to control the range of the VCXO. For a measured frequency deviation of $\pm 16ppm$, result a control step of:

$$\Delta f = \frac{153.6 \times 10^6 \times 32 \times 10^{-6}}{2^{16}} = 0.075 \text{ Hz} \approx \frac{1}{13} \text{ Hz} \tag{3.5}$$

The CMD_i signal used as feedback for the controlling loop is selected as described in Figure 3.7. The $DIFF$ and $DIFF_{dj}$ are expressed in clock periods per second and so the DAC value is computed as:

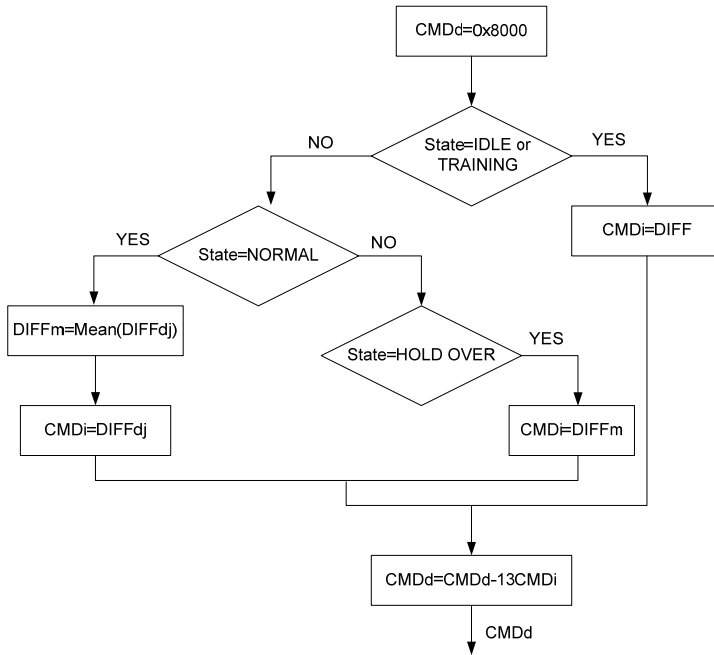


Fig. 3.7 Control Algorithm

$$CMD_d(k) = CMD_d(k-1) - 13CMD_i \tag{3.6}$$

The starting value is the central level of the DAC range, i.e. 2^{15} . In order to obtain a faster convergence, the starting value might be a DAC value saved when the synchronization block was in NORMAL state.

When the State Controller indicates IDLE or TRAINING the oscillator is controlled directly with the measured frequency deviation, in order to achieve fast convergence. In NORMAL state, the Kalman output is used for jitter reduction. The $DIFF_{dj}$ signal has a floating point format, so that frequency corrections less than 1 Hz can be produced. Also the mean of the N_m last values of $DIFF_{dj}$ signal is computed. The mean value is used when the State Controller is in HOLD OVER state and no valid $DIFF_{dj}$ values are received. Also this value is maintained for T_{HO-N} seconds when the synchronizations state returns from HOLD OVER state to NORMAL state, in order not to produce de-synchronization due to the new position of the PPS pulse.

The values of the Control Algorithm parameters are given in Table 3.2.

<i>Parameter</i>	<i>Value</i>	<i>Unit</i>
N_m	128	
T_{HO-N}	2	sec

Table 3.2 Control Algorithm parameters

3.3 Experimental results

The VCXO oscillating frequency is affected by the temperature variations. The controlling algorithm should provide commands fast enough not to accumulate frequency deviations. This problem is observed at the start up too. Even if the temperature is stable, the oscillator is not in a stable thermal state and the algorithm should provide frequency corrections. Figure 3.8 depicts the mean of the DAC commands, considering 0x8000 value as the reference. One can see that at start-up the oscillator has a significant frequency drift and although the temperature variation, depicted in Figure 3.9, is not important, the frequency deviation is about 10Hz per 9000 seconds.

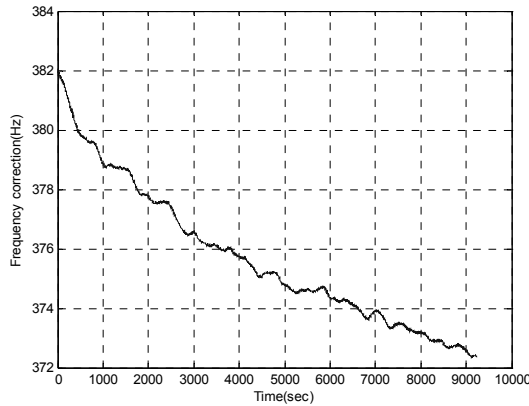


Fig. 3.8 Mean of frequency deviation

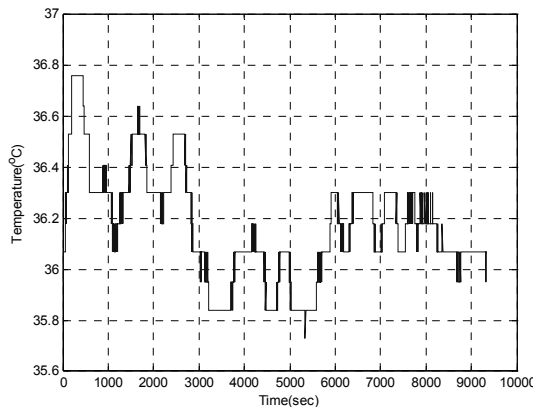


Fig. 3.9 Temperature variation

Figure 3.10 depicts the deviation measured at the input and of the Kalman filter, while Figure 3.11 depicts the deviation measured at the output of the Kalman filter. One can see that the initial deviation is the jitter level ($\pm 20ns \times F_{ref} \approx \pm 3$ clock periods) plus some temperature randomly added deviation. At the output of the de-jittering structure, the level of the jitter is much lower.

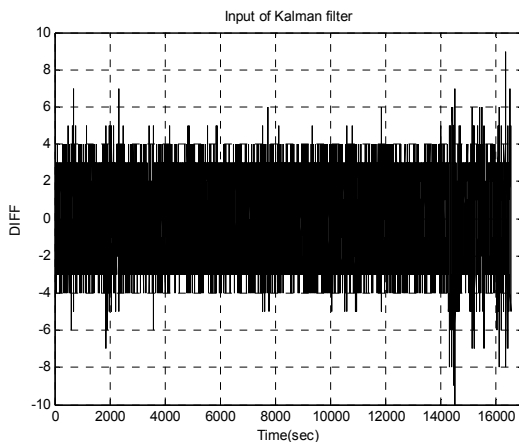


Fig. 3.10 DIFF values in clock periods

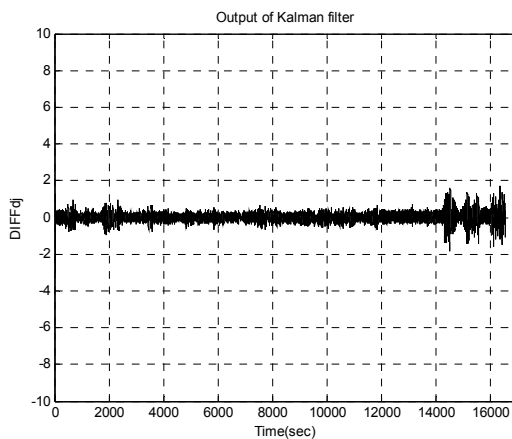


Fig. 3.11 DIFFdj values in clock periods

Figure 3.12 and Figure 3.13 depicts the deviation measured at the input and output of the Kalman filter while a frequency deviation of 1 Hz per second was added for 500 successive seconds and subtracted for the next 500 seconds. One can observe that the $DIFFdj$ signal indicates the need of about 1 Hz reduction of the oscillating frequency for the first 500 seconds and then the need of about 1 Hz adding for the next 500 seconds.

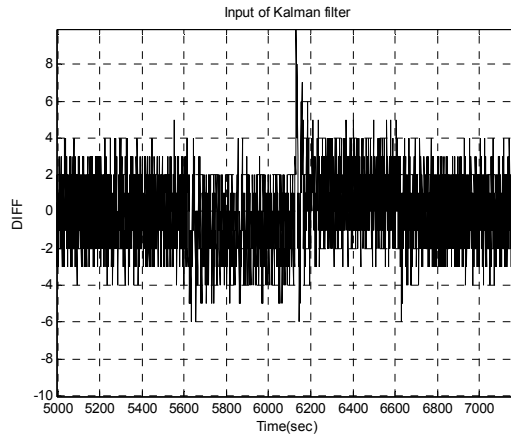


Fig. 3.12 DIFF values in clock periods

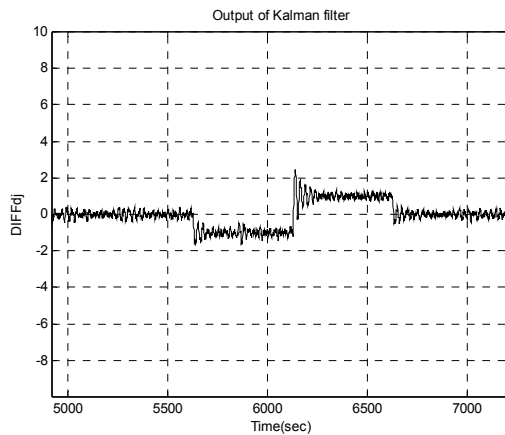


Fig. 3.13 DIFFdj values in clock periods

The results presented in this section were obtained using a hardware platform compliant with Figure 3.1. The inputs and the outputs of the algorithm were transferred to the PC using a UART interface. The pictures were obtained using Matlab.

3.4 Conclusions

This section presented a digital method of reducing the jitter level of the PPS signal generated by a GPS receiver. Also a controlling algorithm of a VCXO oscillating frequency was described. The results indicated that the frequency correction was applied only when the thermal state of the oscillator was not stable. False corrections due to the PPS jitter were almost completely eliminated.

4. WiMAX base station architecture with MIMO capabilities support based on OBSAI RP3-01 Interface

4.1 Introduction

More and more companies try to provide full solutions when it comes to wireless telecommunications systems. But unfortunately this concept, called ecosystem, is not always an easy task to realize. The costs are quite large and, due to system high complexity, the development time is also very long. That means it is very possible that parts of a system can be made by different vendors. The interconnection between different parts should be made by standard interfaces. The usage of third party intellectual property solution reduces the compatibility area. This is the reason why standards as Common Public Radio Interface (CPRI) and Open Base Station Architecture Initiative (OBSAI) were developed. The OBSAI RP3-01 interface permits the transport of data corresponding to different communications standards, such as WCDMA, GSM/EDGE, CDMA2000 and 802.16.

The OBSAI RP3-01 interface represents an extension of the Reference Point 3 protocol for remote radio unit use. The BS can support multiple RRUs connected in chain, ring and tree-and-branch topologies, which makes the interface very flexible. Also, in order to minimize the number of connections to RRUs, the RP1 management plan [3], which includes Ethernet and frame clock bursts, is mapped into RP3 messages. This solution is an alternative to the design in which the radio module collocates with the BBM. Although in such a case the interface between the radio unit and the BBM becomes less complex, the transmitter power should be increased in order to compensate the feeder loss.

This section describes the RP3-01 protocol stack and the corresponding parameters, presents the synchronization procedure. Then a BS split architecture is proposed and the implementation solutions for the most important interface layers components are provided, together with the implementation results obtained when a XC4VFX60 FPGA is targeted.

4.2 RP3-01 protocol stack

The RP3-01 interface is a high speed serial interface for both up link and down link data and control transfer. The protocol stack is based on a packet concept using a layered protocol with fixed length messages.

A. Physical Layer

The transmitter Physical Layer is responsible for the 8b10b encoding, which provides a mechanism for clock recovery, and data serialization. At receiver, the mirrored functions are applied. The Physical Layer can be implemented by a dedicated device, such as an XGMII transceiver [11], or by an internal FPGA component, such as RocketIO transceiver from Xilinx Virtex 5 family [12], when a hardware implementation is considered. The supported rates are 768 Mbps, 1536 Mbps, 3072 Mbps and 6144Mbps. The 6144 Mbps rate does not concern this study.

B. Data Link Layer

The Data Link Layer contains the frame builder and the link synchronization unit. The frame builder receives from superior layers the data and control messages and generates, according to the transmitter rate, the RP3 frame. The data or control message has a fixed 19

bytes length. The message format contains 4 fields. The first one is the Address field on 13 bits, the second one is the Type field on 5 bits, the third one is the Time Stamp field on 6 bits and the last one is the Payload field on 128 bits.

The duration of RP3-01 Master Frame (MF) is fixed to 10 ms. This length corresponds to $i \times N_MG$ Master Groups (MG), where i is selected accordingly to the transfer rate, i.e. 1 for 768 Mbps, 2 for 1536 Mbps and 4 for 3072 Mbps. A MG consists of M_MG messages and K_MG idle bytes (special characters). Figure 4.1 presents an example of MF corresponding to 802.16 standard [13], when 768 Mbps rate is used. Also one can observe the values of the specified parameters.

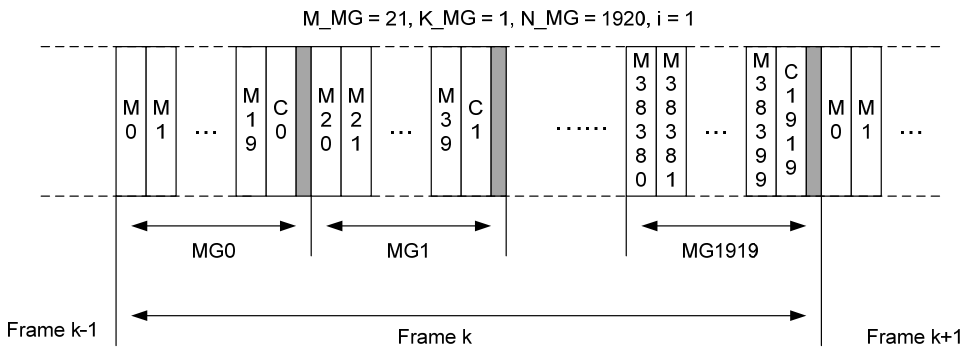


Fig. 4.1 MF for 802.16 air interface standard for the 768Mbps line rate

For the example described in Figure 4.1, the transfer rate can be computed as follows: first we compute the maximum number of bytes per frame, and then, having in mind the 8b10b line encoder and the MF duration, we calculate the transfer rate, as in (4.1). In order to generate the MF, the frame builder uses two counters: for data messages and for control messages.

$$Rate = i \cdot N_MG(M_MG \cdot 19 + K_MG \cdot 1) \cdot 1e3 \tag{4.1}$$

The link synchronization unit contains transmit and receive Finite State Machine (FSM). Both the transmitter and receiver FSMs contribute to physical and logical link synchronization. The physical synchronization is based on special characters, K28.5 and K28.7, which mark the end of message groups and MF, while the logical synchronization is based on fixed MF structure.

C. Transport Layer

The Transport Layer is responsible for the end-to-end delivery of the messages, which could be simply routing of messages. The routing is based on the first 13 bits of the message, which represent the address field. The first 8 bits of the address represent the Node address and the other 5 bits represent the Sub-node address. These fields are used in a hierarchical addressing scheme where the first field identifies the bus nod and the second one selects the corresponding module from the device. The message routing is made based on a Routing Table which indicates the correspondence between the used addresses and the node ports. The table content is defined by the initial configuration procedure of the interface.

Another block of the Transport Layer is the Message Multiplexer/ Demultiplexer. It performs time interleaving/ deinterleaving of messages from N RP3 input links into one RP3 output link. Several multiplexing/ demultiplexing tables are defined as functions of the number of input links and their corresponding rates and the rate of the output link.

D. Application Layer

The Application Layer builds the data messages. It maps data and control information into the message payload and attaches the message header. The payload is represented by concatenation of signal samples in the baseband. For the case of the 802.16 air interface, the format of a data message payload field is shown in Figure 4.2.

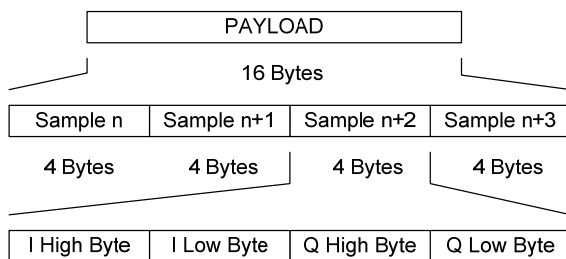


Fig. 4.2 802.16 payload mapping

All transfers over the bus are performed over paths. A path represents a connection between a source node and a destination node. The connection is made by a set of bus links defined by the routing tables. Physically, a path consists of a set of message slots per MF. Paths are defined before bus initialization and they remain fixed during operation, i.e. the transfer between two nodes is made on the same bus links using the same message slots. For each path, a message transmission rule is applied. There are two types of rules: mandatory low level rules, using modulo computation over message slot counters and optionally high level rules, using the Bit Map (BM) concept. In the second case each MG is formed as groups of messages from baseband channels. Each group may contain Ethernet frames as well.

4.3 RP3-01 synchronization procedure

In frame based communications systems, appropriate frequency/ time synchronization is critically important. In order to avoid inter-cell interference, all base stations must use the same frequency/ time reference. The transmission over the radio channel, both on downlink and uplink, should be synchronized with the same reference. The RP1 synchronization burst generator, located at Clock Control Manager (CCM) level, shall provide frame timing and time stamping for each of the air interface standard independently, based on that reference. The PR1 Frame Clock Bursts (FCB) is mapped into RP3 messages.

The synchronization algorithm uses several inputs and is realized based on information collected both at BBM and RRU. The first one is the propagation delay (PD) between BBM and RRU. This time interval is measured on BBM side using special message transmission called Round Trip Time (RTT) measurement message. The second input is the receiving time of the RP1 FCB. This interval is measured from the beginning of the last MF until the last bit of the FCB using a counter called $C1$. After measuring this time interval, the Frame

Clock Synchronization (FCS) message is generated. FCS contains information from FCB and the $C1$ value. The last input is the detection time of a FCS inside a MF. This time interval is measured by RRU using a counter called $C2$. Having all this information, RRU computes the buffering time ($Brru$). Figure 4.3 describes such an example. The time intervals are not at real scale. They are expressed in Time Units (TU). One can observe that at RRU side, the end of recovered FCB corresponds to the beginning of $RF(k+4)$.

Using the formulas from [2], we obtain the $Brru$ as in (4.2), where k equals to 2 from some computing conditions.

$$Brru = k \cdot Rfd + (C1 - PD) - FCBd = 11TU \tag{4.2}$$

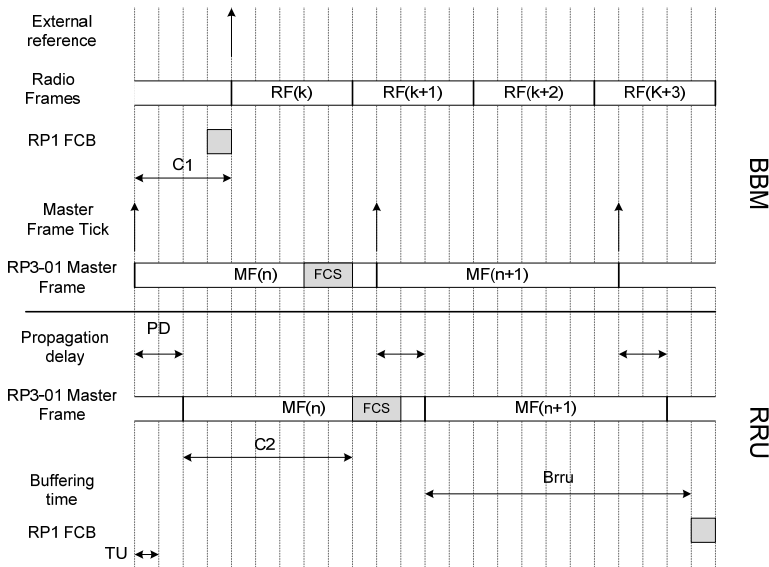


Fig. 4.3 Timing principle in RP1 frame clock burst transfer

4.4 Proposed implementation scheme

The proposed BS split architecture is presented in Figure 4.4. A BBM is connected to the two RRUs in order to have multiple transmit/ receive antennas for MIMO capabilities. The connection between the two RRUs is realized using a chain topology. In order to obtain a single point failure redundancy scheme, a second BBM connected to the two RRUs is required. Only one BBM will be active at the time. Figure 4.4 depicts also the OBSAI RP3-01 Interfaces required for blocks interconnection. Note that on OBSAI RP3-01 interface of each RRU the same Transport and Application Layers serves the both Physical and Data Link Layers.

The RP3-01 connections between each BBM and RRU or between RRUs are bidirectional. On downlink (DL) direction (from BBM to RRU), the data stream from a BBM can contain multiple data streams interleaved/ multiplexed for the two Transceivers (e. g. in order to provide Space Time Coding or MIMO) or can contain data streams only for one Transceiver.

By selecting the right node/ sub-node address, the Application Layer from BBM OBSAI RP3-01 Interface selects the desired RRU. The Transport Layer from RRU1 OBSAI RP3-01 Interface directs the data streams to its own Application Layer when RRU1's address is used, otherwise forwards the data streams to the second Data Link Layer from the OBSAI RP3-01 Interface. In uplink (UL) direction (from RRU to BBM) the procedure is similar. Both receivers can be used (e. g. receive diversity or collaborative MIMO) or only one receiver can be active. In addition to these data and control streams that should be treated by the OBSAI RP3-01 Interface as RP3 streams, an Ethernet stream will be also transmitted between BBM and RRUs in order to connect the corresponding Control & Management (CM) modules to RRU. This stream should be treated by the OBSAI RP3-01 Interface as RP1 stream.

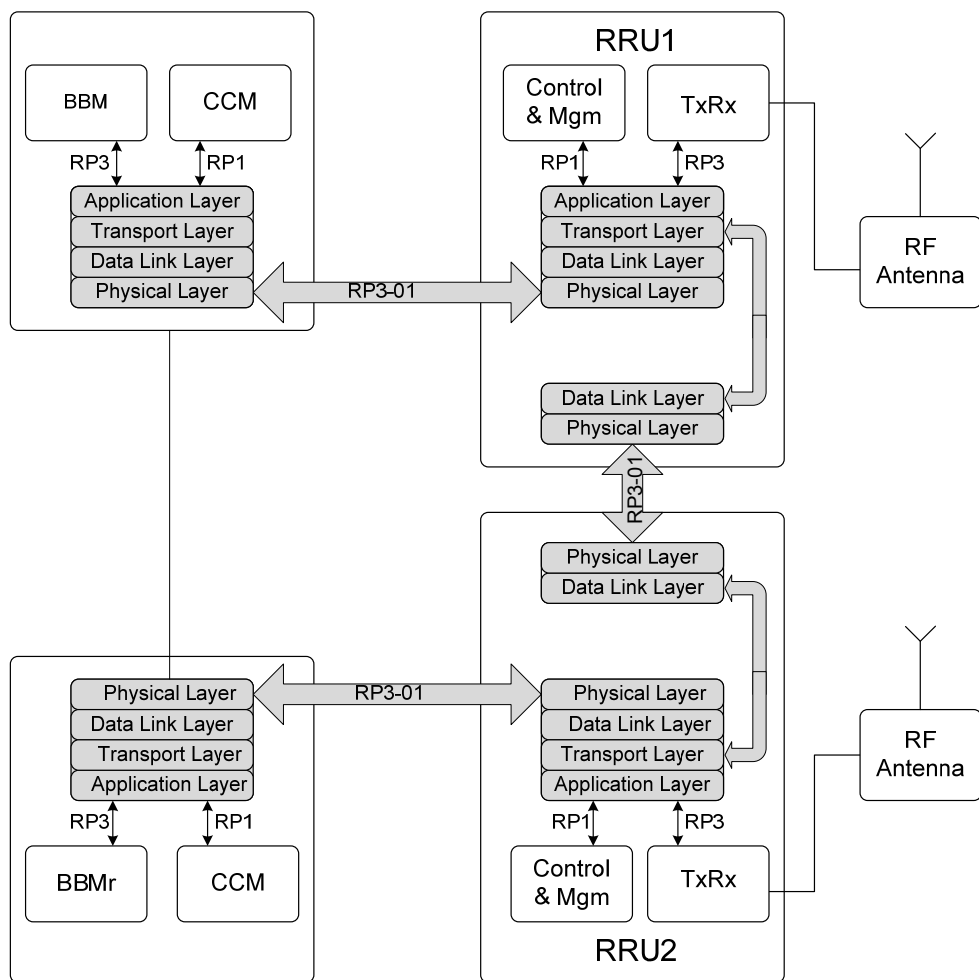


Fig. 4.4 Proposed BS split architecture with interface protocol stack

Figure 4.5 presents the generation of RP3-01 stream at BBM side, while Figure 4.6 corresponds for RRU1 side. Each RP3-01 node has its own address known from the initial configuration procedure. The DL node addresses is represented with normal fonts and with bold italic fonts the UL node addresses. For the proposed system architecture, in DL direction the RP3-01 link is used to connect a source node with two destination nodes, so two paths exist over DL connection. Even if only one RRU transmitter is used at the time, the two paths will exist and the Transport Layer from BBM OBSAI RP3-01 Interface will place its messages over the message slots corresponding to selected RRU (consider twice the rate on link between BBM and RRU1 comparing with the one between RRU1 and RRU2). In the downlink direction, a point-to-point message transfer is applied, while in uplink direction, the same message is multicast to the two BBMs. Only the active BBM uses the received information.

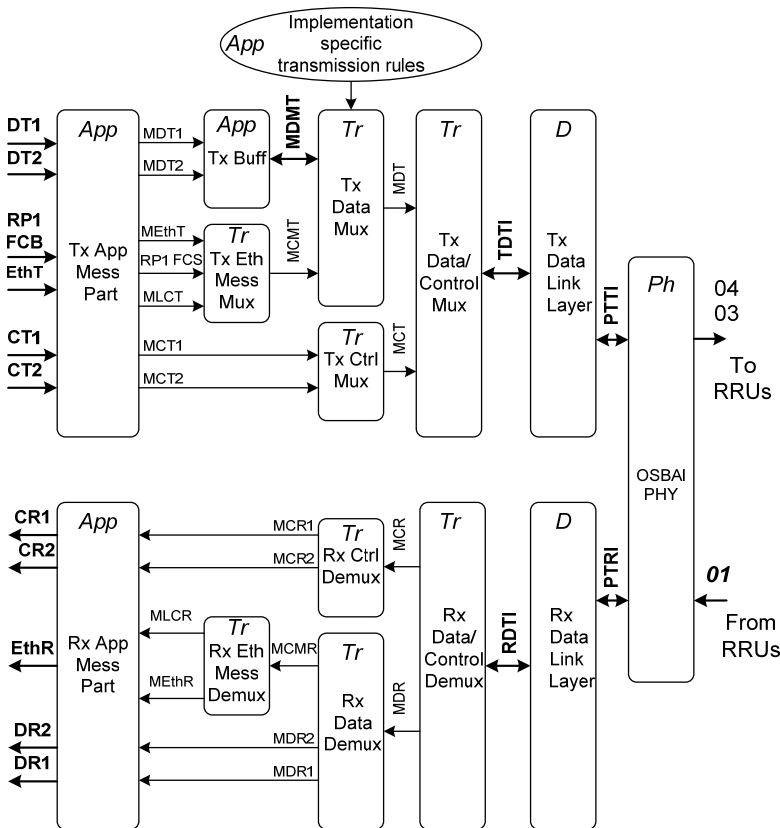


Fig. 4.5 Block scheme for BBM RP3-01 interface

Each path will be considered having a data path and a control path. Bus manager will provide separate message transmission rules for the paths utilizing data and control message slots. We will explain the notations used in Figure 4.5 and Figure 4.6 and we will describe the steps made for RP3-01 interface generation.

In DL direction, on BBM side, the Application Layer receives two data streams for the two RRUs, DT1 and DT2 and two control streams, CT1 and CT2. Also, an Ethernet stream for management called EthT and the RP1 FCBs are received. The Application Layer generates the corresponding messages streams, i.e. MDT1, MDT2, MCT1, MCT2, MEthT and RP1 FCS. Also specific RP3-01 link control messages are generated. This stream is called MLCT. Beside the message generator function, Application Layer is responsible for buffering the data paths. A buffer is required for each 802.16 signal (antenna-carrier) in order to compensate the jitter caused by message transmission. Finally, the Application Layer has to provide to Transport Layer the implementation specific message transmission rules. These rules could include the lower layer message transmission rules and/ or extra rules for mapping the RP1 traffic to RP3 data message slots.

The Transport Layer has four blocks with interleaving/ multiplexing function. First, the management messages, including RP1 FCS, MLCT and MEthT are put on the same flow called MCMT, based on a priority list. The two control messages streams MCT1 and MCT2 are multiplexed in the MCT flow. The data flow, called MDT is obtained by interleaving/ multiplexing the data messages from buffers and the management messages from MCMT stream. Finally, the Transport Layer multiplexes the MDT and the MCT streams based on the TDTI interface with the Transport Layer. Using this interface, the frame builder from Transport Layer requires data or control messages and increments the corresponding counters for each successful transfer. The generated RP3-01 frame, including also the special characters, is transferred to Physical Layer on PTTI interface. The DL continues on RRU side with the receiving chain from Figure 4.6. The Transport Layer works out the RP3-01 flow, first on data stream MDR and control stream MCR, and then the data path is split into data streams MDR1 and MDR2, respectively management messages RP1 FCS, MEthR and MLCR, while the control path is split into MCR1 stream and MCR2 stream. The Application Layer receives all these messages and extracts the corresponding payload. One interesting observation is that at RRU, the Transport Layer has also the message router function, as one can see from Figure 4.6.

In UL direction the procedure follows the same steps as the one described for DL.

For implementation we considered a XC4VFX60 device from Xilinx Virtex 5 family. The functional tests were made using ModelSIM 6.2g and the synthesis results were obtained using Xilinx ISE 9.2i. From the proposed architecture depicted in Figure 4.4 one can see that the RRU Interface contains two Data Link and two Physical Layers. The implementation results obtained for the Data Link Layer are critical for the global resources, while the Physical Layer implementation cost reflects in the number of used RocketIO Transceivers.

For this reason we start our implementation with the Data Link Layer. Figure 4.7 depicts the main blocks of Data Link Layer on the transmit chain, respective on the receive chain of the interface. The area and speed reports are presented in Table 4.1.

<i>Component</i>	<i>No. of slices (from 25280)</i>	<i>Speed (MHz)</i>
FSM Sync Tx	8	407
Framer Tx	197	194
FSM Sync Rx	8	407
Framer Rx	274	185

Table 4.1 Area and speed reports

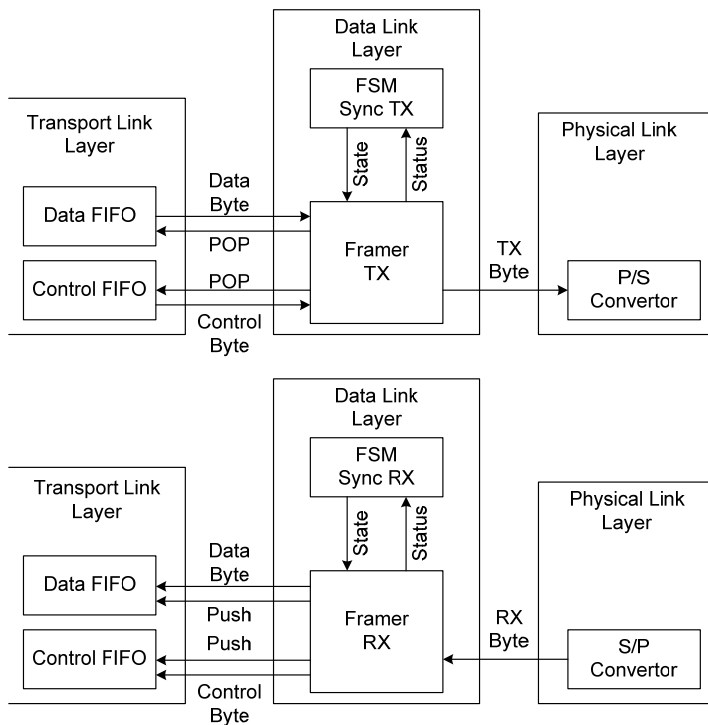


Fig. 4.7 Interface Tx and Rx chains

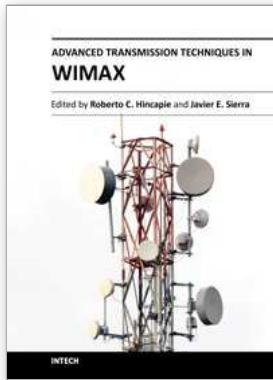
4.5 Conclusions

This section presented an overview of an OBSAI RP3-01 Interface implementation. It was described the main functions of the interface layers and there were presented interface block schemes for both BBM and RRU sides based on the protocol stack. Some examples were made for 802.16 air interface standard without reducing the generality of presentation. Base station split architecture was proposed, with support for redundancy and multiple transmit and receive antennas.

5. References

- [1] John C. Eidson, „Measurement, control and communication using IEEE 1588”, Springer 2006
- [2] “OBSAI Reference Point 3 Specification”, version 4.1, July 2008
- [3] “OBSAI Reference Point 1 Specification”, version 1.0, October 2003
- [4] “CPRI Specification v4.0”, June 2008
- [5] “PICMG MicroTCA Base Specification R1.0”, July 2006
- [6] “PICMG Advanced Mezzanine Card Base Specification R 2.0”, November 2006
- [7] “Intelligent Platform Management Interface Specification – v2.0”, February 2006
- [8] “Trimble Resolution T system designer reference manual”, www.trimble.com

- [9] L. Gasparini, O. Zadedyurina, G. Fontana, D. Macii, A. Boni, Y. Ofek, "A Digital Circuit for Jitter Reduction of GPS-disciplined 1-pps Synchronization Signals", AMUEM 2007 - International Workshop on Advanced Methods for Uncertainty Estimation in Measurement Sardagna, Trento, Italy, 16-18 July 2007
- [10] Yu. S. Shmaliy, A. V. Marienko and A.V. Savchuk, "GPS-Based Optimal Kalman Estimation of Time Error, Frequency Offset and Aging", 31st Annual Precise Time and Time Interval Meeting
- [11] "PMC Sierra PM8358 QuadPHY 10GX" datasheet, March 2005
- [12] "Virtex 5 FPGA RocketIO GTP Transceiver. User Guide",
http://www.xilinx.com/support/documentation/user_guides/ug196.pdf
- [13] IEEE Standard 802.16e, 2005



Advanced Transmission Techniques in WiMAX

Edited by Dr. Roberto Hincapie

ISBN 978-953-307-965-3

Hard cover, 336 pages

Publisher InTech

Published online 18, January, 2012

Published in print edition January, 2012

This book has been prepared to present the state of the art on WiMAX Technology. The focus of the book is the physical layer, and it collects the contributions of many important researchers around the world. So many different works on WiMAX show the great worldwide importance of WiMAX as a wireless broadband access technology. This book is intended for readers interested in the transmission process under WiMAX. All chapters include both theoretical and technical information, which provides an in-depth review of the most recent advances in the field, for engineers and researchers, and other readers interested in WiMAX.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Cristian Anghel and Remus Cacoveanu (2012). MicroTCA Compliant WiMAX BS Split Architecture with MIMO Capabilities Support Based on OBSAI RP3-01 Interfaces, *Advanced Transmission Techniques in WiMAX*, Dr. Roberto Hincapie (Ed.), ISBN: 978-953-307-965-3, InTech, Available from:
[http://www.intechopen.com/books/advanced-transmission-techniques-in-wimax/microtca-compliant-wimax-bs-split-architecture-with-mimo-capabilities-support-based-on-obsai-rp3-01-](http://www.intechopen.com/books/advanced-transmission-techniques-in-wimax/microtca-compliant-wimax-bs-split-architecture-with-mimo-capabilities-support-based-on-obsai-rp3-01)

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.