Scientific Journal of Impact Factor(SJIF): 3.134
e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 3, March -2015

A NOVEL BASED METHOD TO DESIGN A 4G NETWORK AND TO IMPLEMENT IN REAL TIME USING DSP INTERFACE

J.C.GNANA SEKARAN[1], Asst Prof. Mrs. R.PONLAKSHMI[2]

[1]*Electronics and Communication, G.K.M College of Engineering and Technology Chennai-63*
[2] *Electronics and Communication, G.K.M College of Engineering and Technology Chennai-63*

Abstract *— Communication across the globe has been found increasing daily, so naturally the number of users is increased and there is a need for speed and capacity. Thus the existing technologies need to be improved with high modulation techniques to provide for the users. In this project we try to design a WiMAX physical layer and try to test the performance of the network in simulink using BER curves. We try to generate some machine codes preferably 'c' codes using Real Time Workshop (RTW) software, this part plays a huge role by converting the blocks in the simulink to a machine code this was previously done by manual effort, and any replacement in the block required alterations in the overall program and these codes can be implemented in a hardware (DSP KIT) TMS 320C6713DSK. This helps to study the output in a real time environment.*

Keywords*- WiMAX physical layer, Real time workshop (RTW), 'c 'code generation*

I. INTRODUCTION

Communication has reached its height to a great extent. As the new generation requiring the fast data transfer methods 3G, 4G has reached the peak importance,[1] so it is important to make 4g and forthcoming technologies much stronger and to design an apt model to get the highest data communication possible, so the basic aim in this model to make a 4G network and apply many changes to the network by changing the modulation technique and other coding methods and by comparing it with the BER curves and finding which system yields the best output, since we perform these operations in simulations we can make a cost effective, as only the best model will be implemented in the hardware sections [1]. An Real time workshop (RTW) embedded inside the simulation tool is used to generate 'c' code for the blocks designed , this mechanism helps us to eliminate the manual trouble to write a 'c' code for all the blocks designed, if any changes need to be done it impacts the whole program, so these difficulties are eliminated in this process The final stage of the output will be importing the model to a suitable hardware kit and then implementing a real time output.

II. RELATED WORK

A. DSP Based implementation of a Half duplex OFDM System

An implementation of a half-duplex OFDM System has been proposed by Michael Loughlin using DSP LAB 2000 kits by Texas Instruments

B. Implementation of a MIMO OFDM system based on TI C64x

MIMO-OFDM system based on the multi-core texas instrument. The system is employed by real time data exchange and serial rapid input/output techniques for communication between Personal Computer (PC) and DSP Kit.

III. SIMULINK

Simulink is a block diagram environment for multi-domain simulation and Model-Based Design. It supports simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modelling and simulating dynamic systems. It is integrated with MATLAB, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

.

3.1. Features

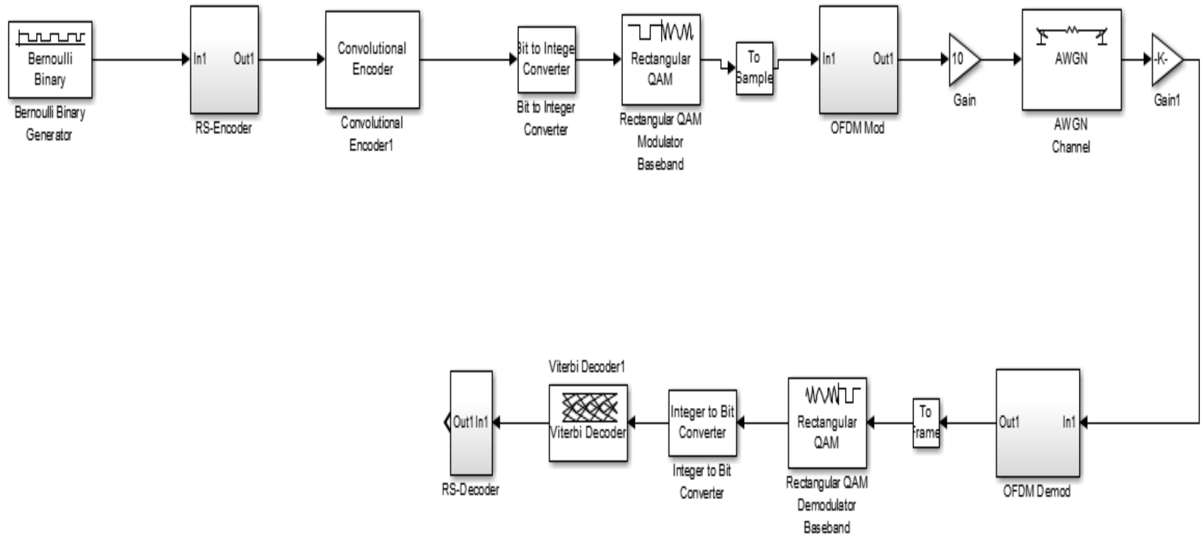
- Building the model
- Simulating the model
- Analyzing the simulations results
- Managing projects
- Connecting to hardwares

3.2. Functions

Simulink provides a set of predefined blocks that you can combine to create a detailed block diagram of your system. Tools for hierarchical modeling, data management, and subsystem customization enable you to represent even the most complex system concisely and accurately. Simulink Library browser has got all the communication block sets that are to be used in determining their performance. We can select our desired blocks that fits in WiMAX physical layer. Simulations can be run through Simulink editor or systematically from the command window. Different modes available in the simulation, are normal, accelerator and rapid accelerator. Mode that yield higher rate is preceded.

IV. WIMAX PHYSICAL LAYER

WiMAX physical layer is constructed using the communication block sets available in the simulink library block sets.



In the above block we have made lot of changes to the modulators and the coders until we got a good shape of signal, as the aim is to get a efficient signal reception with high speed. BER curves block can also be added to view the output in a graphical method.

4.1. Modulator Section

Bernoulli binary generator is the common source used in this physical layer, which can also be replaced by pulse generator or any pseudo noise generator. RS Encoder is used, which has sub blocks in it, latest version of the simulink software enables us to built this model without creating any sub blocks we can give the values for these blocks. Convolutional Encoder is used to redundancy matching, and this also limits some of the repeating words, thus reducing the space in the storage and in data transfer. Modulator section has the QAM and it has a 4 multiple combination, choosing the sixteen QAM yields the desired result making a good SNR in the BER curves. AWGN preferred channel is used in this section also. The AWGN Channel block adds white Gaussian noise to a real or complex input signal. When the input signal is real, this block adds real Gaussian noise and produces a real output signal. When the input signal is complex, this block adds complex Gaussian noise and produces a complex output signal. This block inherits its sample time from the input signal. This block accepts a scalar-valued, vector, or matrix input signal with a data type of type single or double. The output signal inherits port data types from the signals that drive the block. Alternate blocks with other latest features can be replaced. As the Real time workshop embedded in the simulink software converts the blocks into codes, it is not required to separately write codes for every alteration in the system layer.

4.2. De-Modulator Section

Respective blocks are used as in the modulator section, to properly decode the signals. OFDM Demodulator block demodulates an OFDM input signal. The block accepts a single input and has one or two output ports, depending on the status of Pilot output port. Viterbi Decoder block decodes input symbols to produce binary output symbols. This block can process several symbols at a time for faster performance. This block can output sequences that vary in length during simulation. For more information about sequences that vary in length, or variable-size signals .Like-wise the demodulator section decodes the input data, other stage is to convert these blocks into machine language.

4.3.Code Generation

Simulink models can be configured and made ready for code generation. By using Simulink with add-on code generation products, you can generate C and C++, HDL, or PLC code directly from your model.

V. DSP STARTER KIT

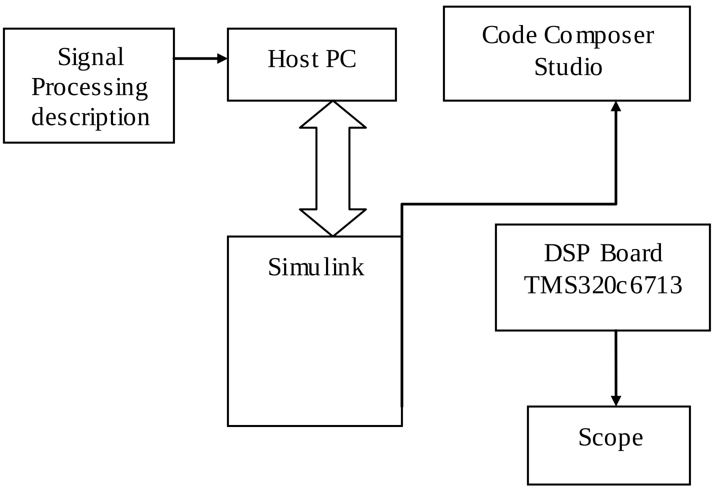
Digital signal processors such as the TMS320C6x (C6x) family of processors are like fast special-purpose microprocessors with a specialized type of architecture and an instruction set appropriate for signal processing [3]. The C6x notation is used to designate a member of Texas Instruments' (TI) TMS320C6000 family of digital signal processors. The architecture of the C6x digital signal processor is very well suited for numerically intensive calculations. Based on a very-long-instruction-word (VLIW) architecture, the C6x is considered to be TI's most powerful processor. The TMS320C6713 (C6713) is very well suited for numerically intensive algorithms. The internal program memory is structured so that a total of eight instructions can be fetched every cycle. For example, with a clock rate of 225MHz, the C6713 is capable of fetching eight 32-bit instructions every 1/(225 MHz) or 4.44 ns. The C67xx (such as the C6701, C6711, and C6713) belong to the family of the C6x floating-point processors, whereas the C62xx and C64xx belong to the family of the C6x fixed-point processors [3]. The C6713 is capable of both fixed - and floating point processing.

5.1.DSP Implementation

The interface of the Simulink model developed with MATLAB and CCS (Code Compose Studio) IDE along with the Target specified in the model. Here the Target we have used is the TMS320C6713 DSP kit as seen in the figure. RTW and Embedded Target Coder tools on Simulink generate ANSI C code which is automatically dumped onto the CCS IDE.

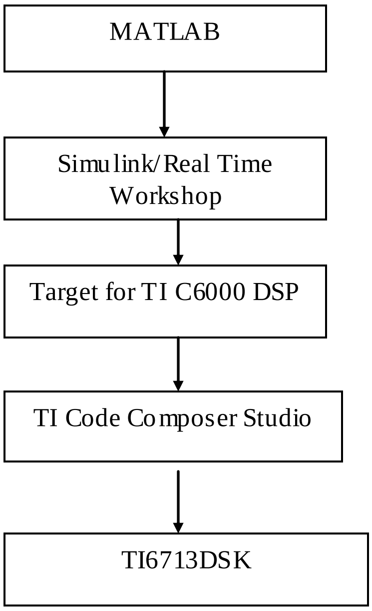
5.2.DSP Software

The TI C6713 DSK, like a personal computer, also needs to load software to establish its behaviour and function [3]. These software's can be designed in a variety of approaches. The designer will begin with a concept of what they want to program. The next step is to model the concept with blocks from Simulink's large collection of prewritten block-sets. Basically, a block diagram that models the concept is built using Simulink. If a specific block required is not included in Simulink's block-sets, the designer may choose to write their own blocks from scratch using Matlab. At this point the design is still not designed to operate on any specifically hardware. To do that, the designer uses the C6x Target and RTW to generate (or build) ANSI C code intended for the TI C6713 DSK. The C6x Target will then automatically take the generated ANSI C code and uses the TI CCS tools to compile specific machine code and finally loads the targeted machine code to the TI C6713 DSK hardware. For the experienced designer, they may choose to directly write the ANSI C code.



5.3.Code Composer Studio

CCS provides an IDE to incorporate the software tools. CCS includes tools for code generation, such as a C compiler,an assembler, and a linker. It has graphical capabilities and supports real-time debugging. It provides an easy-to-use software tool to build and debug programs. The C compiler compiles a C source program with extension .c to produce an assembly source file with extension*.asm*. The assembler assembles an *.asm* source file to produce a machine language object file with extension *.obj*. The linker combines object files and object libraries as input to produce an executable file with extension *.out*. This executable file represents a linked common object file format (COFF), popular in Unix-based systems and adopted by several makers of digital signal processors . This executable file can be loaded and run directly on the C6713 processor. The linear assembly source file with extension .sa, is a cross between C and assembly code. A linear optimizer optimizes this source file to create an assembly file with extension .asm (similar to the task of the C compiler). To create an application project, one can "add" the appropriate files to the project. Compiler/linker options can readily be specified. A number of debugging features are available, including setting breakpoints and watching variables; viewing memory, registers, and mixed C and assembly code; graphing results; and monitoring execution time. One can step through a program in different ways (step into, over, or out).



Real-time analysis can be performed using real-time data exchange (RTDX). RTDX allows for data exchange between the host PC and the target DSK, as well as analysis in real time without stopping the target. Key statistics and performance can be monitored in real time. Through the joint team action group (JTAG), communication with on-chip emulation support occurs to control and monitor program execution. The C6713 DSK board includes a JTAG interface through the USB port.

5.4.Software Implementations

There are several important extensions that enhance the real-time DSP programming features of MATLAB/Simulink tools. They include:

- Embedded Target for TI C6000 DSP which performs real-time prototyping and system deployment on C6000 processors.

- MATLAB Link for Code Composer which verifies and validates embedded software on Texas Instruments' C2000, C5000, C6000, OMAP, and TMS470 processors.

These latest toolboxes illustrate the real-time applications as they are being designed, and are powerful research platforms for verifying the DSP algorithms. However, for generating more efficient code, fine-tuning, low-level programming, and using optimized library functions from the chip manufacturer are still needed for embedded DSP applications. Another trend in the latest tool development is the integration of MATLAB environment with the code development software from the DSP chip vendors. For example, the MATLAB Link for Code Composer allows transfer of data between MATLAB and Texas Instruments' DSP processors. Data can also be acquired from the I/O channels of the DSP board to the MATLAB for further analysis. The model/code based design requires the following steps:

1) Simulink provides an exploratory and verification tool for both floating-point and fixed-point DSP systems and applications.
2) Simulink communicates with Code Composer Studio through Real-Time Workshop.
3) Code Composer Studio communicates with the DSK through an embedded JTAG emulator with a USB host interface.
4) Matlab communicates with CCS via Link for Code composer Studio.

VI. CONCLUSION

The Wimax physical layer is successfully implemented using the simulink model and thus the best model will be built and proceeded to be implemented in the hardware sections. And thus it reduces the complexity in writing codes for any changes in the block in design method.

REFERENCES

[1] Yangzi Li and Gengguo Cheng, "Fourth GenerationWireless Communication Network".

[2] Evangelos Pikasis Sotiris Karabetsos Thomas Nikas and Athanase Nassiopoulos "OFDM, SC-FDMA and MC-CDMA educational wireless transceivers using matlab and the TMS320c6713dsk".

[3] J. Glossner, J. Moreno, M. Moudgill, J. Derby, E. Hokenek, D. Meltzer, U. Shvadron, and M. Ware, "trends in compilable dsp architecture," proc. ieee workshop signal processing systems, pp. 181-199, 2000.

