MICROCONTROLLER BASED MULTIPLE-INPUT MULTIPLE-OUTPUT TRANSMITTER

Balaji Madapuci Kanday (Student) and Kurt Kosbar (Advisor)

Telemetry Learning Center
Department of Electrical and Computer Engineering
Missouri University of Science and Technology

ABSTRACT

This paper describes how a microcontroller based system can be used to generate the signals needed in a multiple-input multiple-output (MIMO) system transmitter. The limited computational speed of the microcontroller, along with other tasks which the controller may need to handle, places limits on the throughput of the system, and the complexity of the MIMO signal design. However this can be a low cost design, and the microcontroller can be used to perform other operations in the system, which may make it attractive in some applications.

Keywords: MIMO, Microcontroller, Testbed, Low Complexity Design

INTRODUCTION

Multiple-input multiple-output (MIMO) wireless communication systems, use multiple antennas at both the transmitter and receiver. They are an important development in communication theory, because of their potential of increasing the data rates, and/or spectral efficiency of the communication system [1]. This benefit can be realized even in fading channels, because both temporal and special diversity techniques can be used simultaneously. It has been shown that MIMO systems can support higher data rates under the same transmit power budget and bit-error rate performance requirements as a single-input single-output (SISO) systems.

However these benefits come at the cost of increased complexity at both the transmitter and receiver. In addition to increased complexity MIMO systems can be sensitive to a number of practical imperfections, such as uncertainty regarding the channel parameters. Nonlinearity in power amplifiers and finite bit precision of analog-to-digital and digital to analog converters, to name a few. While some of these effects can be analyzed mathematically, or through simulation, others are best measured using a hardware test bed.

This paper investigates the high level design of such a test bed. In particular, we focus on the architectural design of the MIMO transmitter. The authors were aware of some test beds that were based on field programmable gate array (FPGA) technology. While these designs worked well, we had an application that called for a lower complexity, and less expensive approach. We are investigating the use of a microcontroller based MIMO transmitting system to achieve this cost and complexity reduction. This paper will discuss the benefits, and limitations, of a microcontroller based design.

SYSTEM ARCHITECTURE

A MIMO transmitter must generate complex waveforms for each of the transmitting antennas. While this can be done in a variety of ways, it is usually performed with some sort of digital signal processing device. Four candidate methods for this include: application specific integrated circuits (ASIC), field programmable gate arrays (FPGA), digital signal processing (DSP) devices, and microcontrollers. The ASIC approach has a substantial nonrecurring cost associated with it, and for this reason can be impractical in cost sensitive, low volume, applications – and is not investigated in this paper.

There are considerable resources available which describe the use of DSP chips to implement digital transmitters, so that will not be addressed here either. In the particular application being investigated, there are reasonably severe constraints on the power consumption, and cost, of the final design. This is another concern for the DSP chip approach, since DSP chips tend to consume more power, and often cost far more, than an approach using microcontrollers.

The concerns listed above limited the decision of which approach to use, to FPGAs and microcontroller based designs. The following paragraphs discuss the tradeoffs associated with these two approaches.

Field Programmable Gate Array Designs

Field programmable gate arrays are integrated circuits designed to be configured by the user. They contain logic cells which can be connected in a variety of ways using a set of on-board interconnects which are programmed at the time the chip is powered up. Each of these cells performs a variety of functions, which can also be programmed. The parallel hardware allows FPGAs to handle higher data rates than sequential processors, which can be a significant advantage in high throughput digital communication systems [2]. Figure 1 illustrates a high level block diagram for an FPGA based MIMO system.

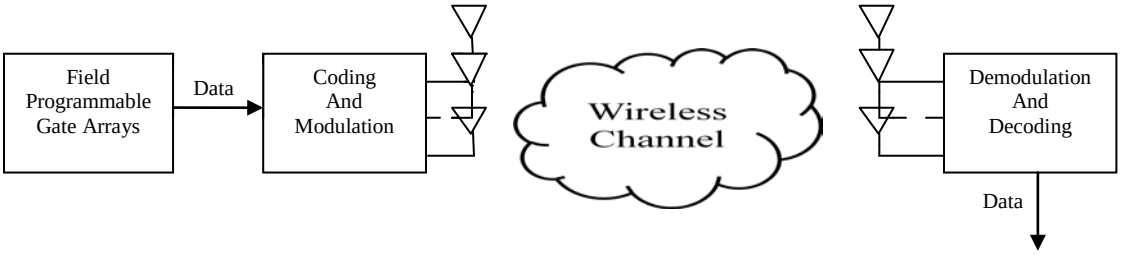

Figure 1 FPGA based MIMO system

In an FPGA-based MIMO system, the FPGA generates data which is coded and modulated; and transmitted across a wireless channel. At the other end the received signal is demodulated and decoded to recover the data.

While it is possible to implement a MIMO transmitter using FPGA, there are some significant hurdles which need to be overcome. The FPGA are normally programmed using a hardware description language, such as VHDL. While it is certainly possible to do this, many DSP algorithms can be coded and analyzed more quickly using a sequential process such as a DSP chip, or microcontroller. The sequential processors can also be easier to debug, than code written in a register-transfer level language.

Another concern is the amount of power required for an FPGA based design. Others have also observed [3] that for many applications FPGAs consume more power than can be justified in many DSP applications. Table.1 compares the maximum power consumption of various FPGAs and microcontrollers

FPGA	Family	Maximum Power Consumption(mW)
AT40KO5	Atmel	95
XQ4000E	Xilinx	28.8
AT6000(LV)	Atmel	48
AT40KAL	Atmel	33.6

Microcontroller	Family	Maximum Power Consumption(mW)
AT89C51	Atmel	7.6
LH7A404	NXP	5.2
87C52	Intel	8.32
AT32AP7000	Atmel	20.8

Table.1 Comparison between FPGA and Microcontroller power consumption

Based on these observations, we concluded that a FPGA MIMO based transmitter would be reasonable in situations where power consumption, and the associated heat dissipation issues, were not a major concern. Also, FPGAs would be a good choice where high throughput is required. But there are a number of problems where low power, and low data rates are both called for, which motivated us to look for alternative solutions.

Microcontroller Based Designs

Microcontrollers are widely used in embedded designs, and there is interest in using them for for DSP based applications also [4]. As the size and complexity of microcontrollers continue to increase, so does their ability to handle the complexities of DSP algorithms. Since many system designs already have provisions to include a microcontroller, we were interested in investigating the possibility of using the same processor to perform the calculations required in a MIMO based transmitter.

Many microcontroller based systems are designed to perform parallel processing, or multitasking. This allows the designer to compartmentalize the functionality of the design. This also has the advantage of making it easier to integrate a DSP based algorithm into existing microcontroller code [5].

Figure.2 represents a high-level block diagram of a typical microcontroller-based MIMO system.

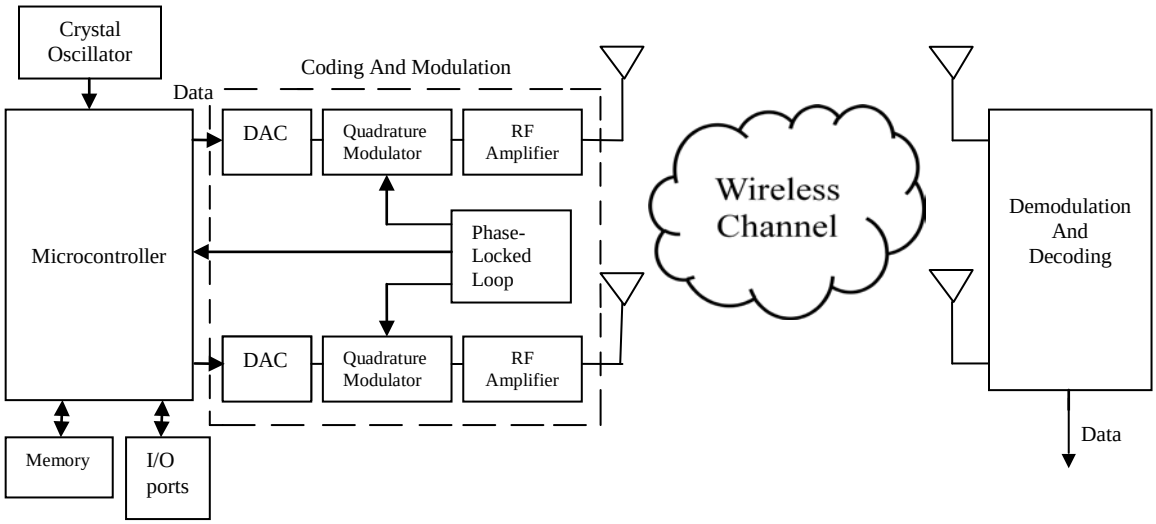

Figure 2 High-level Block Diagram of Microcontroller-based MIMO system

The microcontroller is connected to memory, peripheral ports and an oscillator. The peripheral ports are a way to connect to the outside world i.e. to perform external processes. In a microcontroller-based MIMO system, the microcontroller can be used to either generate the data based on measurements, can generate test signals to measure the performance of the system, or it can accept data from an outside source for transmission. The microcontroller must implement the coding and pulse shaping operations [6] needed for transmission before sending the data to the digital to analog converters (DAC). Figure 2 illustrates a 2-transmitter MIMO system, although it could be extended to more transmit antennas if the processor has the necessary computing power to keep up with both the modulation operations, and the other tasks which it must perform in the system.

At the receiver's end demodulation and decoding are carried out to extract the transmitted data. But as described in Table.1, this process of generating signals using a microcontroller as a source exhibits lower power consumption.

BLOCK DIAGRAM DESCRIPTION

Figure 3 gives a detailed block diagram of a microcontroller-based MIMO system Transmitter. The components of the design are described below

Microcontroller

The application requirement for low power consumption is fulfilled by the use of microcontroller. The particular controller intended to be used in this application is powered off of a 1.8 volt buss, and is intended for use in power constrained applications.

The design in Fig. 3 uses two microcontrollers. This may be necessary if the computing load is too heavy for a single process to keep pace. The input data is supplied to one of the controllers, which will pass it along to the second device through a general purpose I/O port (GPIO). The processors can then work in parallel to generate the necessary waveforms for transmission.

Crystal Oscillator

A crystal time based is needed for each of the processors. The signals generated by the processors need to be synchronized, however this synchronization requirement is not so severe that the processors need to have synchronized clocks. So for simplicity of design, it is assumed that each processor will have an individual time based dedicated to it, and the time bases will run asynchronously.

DMA controller:

One of the challenges of a microcontroller based DSP design, is moving data from the controller's memory to the digital to analog converter without consuming an inordinate amount of processor resources. This design accomplishes this task, by using processors which have direct memory access (DMA) capabilities. The DMA will allow the data to be transferred from processor memory to the DAC consuming a minimal amount of processor time.

However there are some challenges associated with a DMA based approach. There are a limited number of DMA ports on processors. This was part of the motivation for considering designs with multiple processors, since each processor will need to generate both in-phase (I) and quadrature phase (Q) digital signals – consuming two DMA ports for each transmitted signal in the MIMO system.

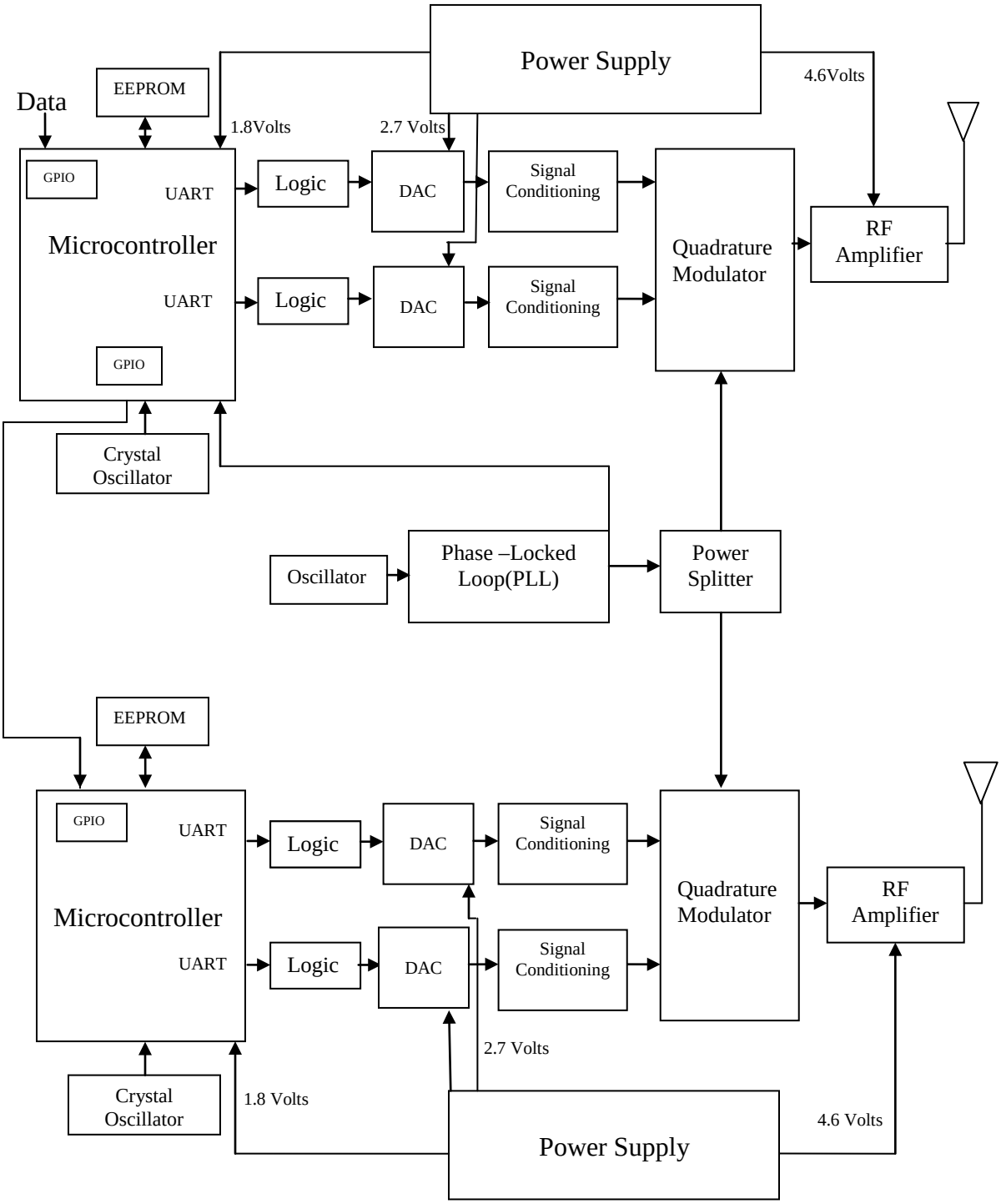

Figure 3 Microcontroller Based MIMO system Transmitter

Another challenge is that the DMA controller has a very specific format in which it generates the output signal, such as a universal asynchronous receiver/transmitter (UART) data bus. This does not always interface cleanly with the DAC used, and may require the addition of interface, or glue, logic.

Digital to Analog Converter

The generated data needs to be converted into an analog signal before transmission. There are two DAC required for each transmitted signal, to accommodate the I and Q baseband signals. To provide an easier interface to typical microcontroller DMA controllers, a serial digital interface DAC was selected. A typical DAC of his nature is described in Fig. 4.

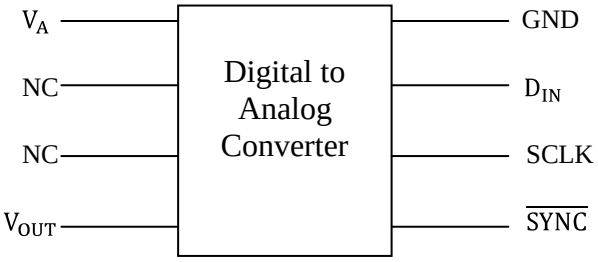

Fig.4 Pin Configuration of DAC

The data transmitted by the microcontroller is used as an input by the DAC via D_{IN}.The power supply V_A is provided by the power supply block. Some of the other inputs that must be driven by the processor include a serial clock, and a frame synchronization signal.

Glue Logic

The microcontroller is connected to the DAC via a UART. Unfortunately, the UART interface does not directly supply the signals necessary for a serial DAC interface. For this reason, a glue logic block was inserted, to perform the necessary signal conversion. This block will generate the serial clock (SCLK) and frame synchronization ($\overline{SYNC}$), signals required by the DAC serial interface. Figure 5 illustrates the glue logic block interface.

Figure 5. GLUE Logic Block Diagram

Quadrature Modulator

One can perform quadrature modulation on either digital or analog signals. In MIMO application investigated in this paper, the modulation is performed using analog signals, directly on the RF carrier. The analog signals generated by the DAC are passed to a quadrature modulator hybrid circuit, which mixes (multiplies) each of them by quadrature sinusoidal carriers, as shown in Figure 6.

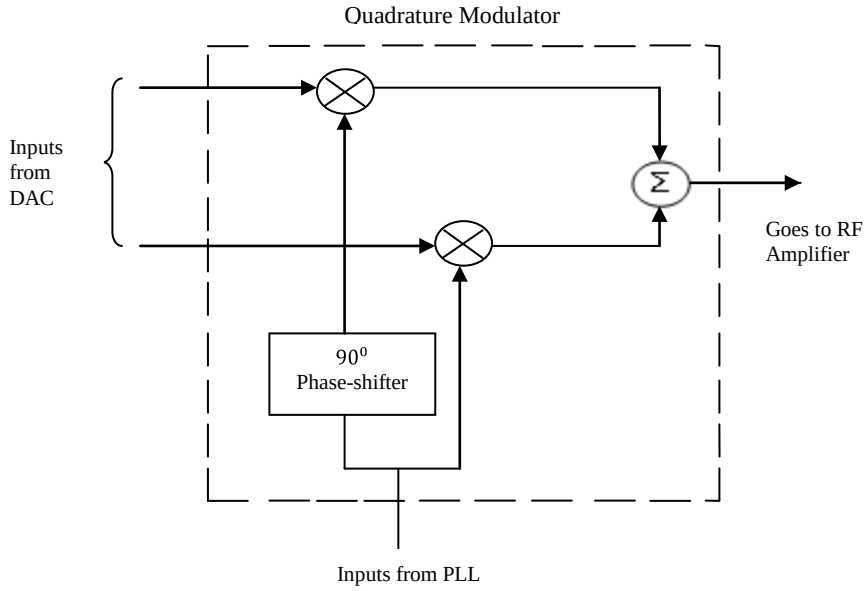

Figure 6. Block diagram of Quadrature Modulator

The carrier signals are provided as an input to the quadrature modulator by the PLL. One of the inputs from the PLL is directly used and the other signal is phase shifted by 90º.

Phase-Locked Loop (PLL)

A phase locked loop is used to generate the RF carrier in this system. Since it can be a challenge to stabilize a high frequency carrier, the PLL is locked to a lower frequency, crystal, time base. The microntroller will program the PLL upon power-up, to set the factor by which it will divide the RF carrier, before comparing it to the lower frequency crystal time base. A power splitter at the PLL output will couple the signal to the quadrature modulators.

RF Amplifier

The final step before transmission, will be to pass the modulated signal through an RF amplifier. The power level required for these amplifiers will depend on the data rate, required error rates, and distance the signal must travel. In the target design, reasonably

low power amplifiers are used since we are only interested in validating the functionality of the microcontroller based baseband processing state.

CONCLUSION

The architecture of a microcontroller based MIMO transmitter was discussed. This design is being investigated as an alternative to ASIC, FPGA and DSP processor based designs. The hope is that the microcontroller based design will provide a lower cost solution, and also a lower power solution, than the competing technologies. In addition, the microcontroller based approach may be easier to integrate with existing embedded systems which already incorporate a microcontroller in their design. One of the significant drawbacks of this approach, is that we anticipate a lower data rate, due to the limited throughput of the microcontroller. This is alleviated somewhat by the use of DMA techniques.

REFERENCES

[1] Thomas Kaiser, André Bourdoux, Holger Boche, Javier Rodraquez Fonollosa, Jorger Anderson and Wolgang Utschick, Smart Antennas: State of the Art, *Hindawi Publishing Corporation*, 2005 Page(s):811-821

[2] A.Burg, M.Rupp, M.Guillaud, E.Beck,D.Perels, N.Felber, and W.Fichtetner, FPGA Implementation of a MIMO Receiver Front-end for UMTS, *International Zurich Seminar on Broadband Communications*, Zurich, Switzerland, Feb. 2002

[3] Berkeley Design Technology, Inc. Staff, FPGAs for DSP, 2nd Edition, BDTi, 2007

[4] William A. Giovino**,** Microcontrollers and DSPs - Will the Two Worlds Intersect?, *EE Times Group*, 1998

[5] A.K.Rath and P.K.Meher; Design of a Merged DSP Microcontroller for Embedded Systems using Discrete Orthogonal Transform, *Journal of Computer Science* 2(5), 2006 Science Publications

[6] Wayne Stark, Coding, Modulation and Multiple-access, *International Technology Research Institute, Panel Report on Wireless Technologies and Information Networks*, Jul. 2000