DESIGN, IMPLEMENTATION AND OPTIMISATION OF 4x4 MIMO-OFDM RECEIVER FOR COMMUNICATION SYSTEMS

S. T. Manju, *Sanket Dessai

Department of Computer Engineering,
M.S. Ramaiah School of Advanced Studies, Bangalore-560058

*Contact Author e-mail: sanketdessai@msrsas.org

Abstract

Multiple Input Multiple Output (MIMO) systems use more than one antenna at both ends of the communication link. For more reliability the number of antennas at receiver should be more than at transmitter side. Over the past decade, the use of MIMO system has rapidly gained popularity due to its enhanced performance capabilities of improved Reliability, Spatial Diversity Gain and Spatial Multiplexing Gain. Orthogonal Frequency Division Multiplexing (OFDM) is one of the best digital modulation schemes, where signal is divided into number of narrow band signals to obtain spectrum efficiency and minimizing the Inter Symbol Interference (ISI). Thus, combining MIMO and OFDM technologies will improve spectral efficiency, Link reliability, spectral gain and data rate.

In this paper, MIMO and OFDM techniques are combined to improve the receiver performance. Software reference model for 4x4-MIMO-OFDM Transmitter for wireless communication system is designed and implemented in MATLAB and the same design is also implemented in C for embedded platforms. The evaluation of Bit Error Rate (BER), Signal to Noise Ratio (SNR), Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) performance of the MIMO-OFDM technique based on comparative analysis of modulation schemes namely QPSK over AWGN channels is carried out. Optimization of the system implemented in C has been carried out using Real View Development Suite (RVDS) by considering different ARM cores namely, ARM 1176, and ARM Cortex A8. Serialised image data was used to analyse the system performance. MSE and PSNR are computed for 4x4 MIMO-OFDM and compared with QPSK modulation.

For a color image, QPSK has PSNR of 58.6713 dB at SNR 5dB and for a binary image 78.195 dB at SNR 3 dB. Profiling results of the system developed in C is analysed in RVDS. ARM Cortex A8 gives better performance in terms of number of cycles estimated to execution of code and code coverage when compared with lower versions of it. After optimisation in ARM 1176, the number of cycles reduced is 39,561. And in ARM Cortex A8 the number of cycles reduced is 31,616. The performance of the system can be improvised by using the higher methods of sub- blocks like higher QAMs, increase the number of sub carriers in FFT and using advanced MIMO techniques.

Keywords: 4G, Alamouti STBC, MIMO-OFDM, FFT, 16-QAM, QPSK, ARM1136, Cortex A9, RVDS

Nomenclature

Data	bit, b
Date rate	bits per second, bps
Cycle	Cycle, s
f	Frequency, Hz
Kbits	Kilo bits, Kb=2^{20}=1024b
Mbits	Mega bits, Mb= 2^{20} = 1,048,576b
IC	Instruction Count
IPS	Instructions per second

Abbreviations

ASK	Amplitude Shift Keying
ARM	Advanced RISC Machine
BER	Bit Error Rate
CC	Code Coverage
CP	Cyclic Prefix
DFT	Discrete Fourier Transform
FSK	Frequency Shift Keying
FEC	Forward Error Correction
FT	Fourier Transform
FFT	Fast Fourier Transform
IC	Instruction Count
IPS	Instructions Per Second
ISI	Inter Symbol Interference
IFFT	Inverse Fast Fourier Transform
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
MIPS	Million Instructions Per Second
OFDM	Orthogonal Frequency Division Multiplexing
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RVDS	Real View Development Suite
RISC	Reduced Instruction Set Computer
STC	Space Time Codes
STBC	Space Time Block Codes
STTC	Space Time Trellis Code
SNR	Signal to Noise Ratio
ZP	Zero Padding

1. INTRODUCTION

Wireless communication is one of the thrust area of research and it is developing extremely faster in current generation and is promising in future. Currently 3G standards are deployed in mobile communication system and 4G is evolving worldwide. This comparative study between 3G & 4G tells about the background and the vision for the 4G. 3G is at this time the world's most excellent connection method when it comes to mobile phones particularly meant for mobile Internet. This paper deals with the Receiver design of MIMO-OFDM, which uses multiple antennas at receiver for fast communication. A 4G framework has the basic keys like multiplicity and flexibility.

The growing demand for high system capacity, high transmission rate and broadband access is main reason for development of the wireless system which will satisfy all the above conditions. One of the solutions for all these conditions is Multiple Input Multiple Output (MIMO) Orthogonal Frequency Division Multiplexing (OFDM) system, which is the combination of MIMO and OFDM. The combination of MIMO with orthogonal frequency OFDM is the important solution for enhancing the data rates and to improve spectral efficiency of next generation wireless communication systems. Improvements in data rates can be achieved by using multiple antennas both at the transmitter and the receiver side.

2. PROBLEM STATEMENT

The available spectrum is fixed but the demand for high data rate and high reliability is growing day by day to suite the more improved and attractive applications. In this paper, we design, implement and optimize the 4x4 MIMO-OFDM receiver for different embedded platform. The idea behind developing the MIMO-OFDM transmitter is to get the advantage of both MIMO and the OFDM technology in achieving higher data rate, spectral efficiency and reliability.

3. SYSTEM REQUIREMENTS

The following are the system requirement specifications arrived for different subsystems of the system

- MIMO system typically consists of *m* transmitting antennas and *n* receiving antennas
- In this system channel is assumed to be a static channel and that the channel is known perfectly at the receiver for all the systems being developed
- OFDM system generally consists of different sub blocks namely FFT block, Cyclic prefix and parallel to serial conversion
- Both spatial diversity and spatial multiplexing techniques should be supported

In this research work channel is assumed to be a static channel and is known perfectly at the receiver for all the systems being developed.

4. METHODOLOGY

- Design specifications of the implementation issues of the 4x4 MIMO-OFDM receiver was arrived based on the reviewed literature and study
- Design and Modelling of the 4x4 MIMO-OFDM receiver using different algorithms/techniques was been carried out using the MATLAB tool based on the literature review and the study carried out
- Performance estimation of the 4x4 MIMO-OFDM receiver and its different algorithms/techniques has been analyzed based on the modelled design
- Implementation of the 4x4 MIMO-OFDM receiver has been carried out using C and Assembly coding for embedded platforms using keil/RVDS/CCS for ARM cores and DSP
- Performance estimation of the 4X4 MIMO-ODFM receiver has been carried out based on the designed model
- Optimization techniques has been applied to optimize the implementation of 4x4 MIMO-OFDM receiver for the embedded platforms based on the design and implementation performance estimations
- Functionalities test has been carried out on the optimized 4x4 MIMO-OFDM receiver subsystems based on the design specifications, implementations specifications

5. DESIGN AND SOLUTION PROCEDURES

Integrating the MIMO and OFDM which consists of blocks of demodulator, de-interleaver and viterbi decoder FFT and cyclic prefix results a block diagram 4x4 MIMO OFDM receiver shown in Figure 1. The receiver combines the 4x4 data transmitted with noise added in the channel. The combined data is converted to parallel to remove the guard interval inserted during transmission. After removing guard interval fed to FFT to remove the carrier frequency to form a continuous wave. Output of FFT is converted to serial. The serial output is demodulated to the corresponding constellation points (QAM or QPSK) to obtain binary output bits. The demodulated output is de-interleaved by changing the row and column matrix. Trellis is calculated before decoding to assume the next coming bits, then decoded using viterbi decoding. Finally reconstructing of the decoded bits had carried out to get the original output.

Error correcting in digital communication is the major challenge to eliminate noise produced in the channel due to noise occurred for various conditions. Using this error correction codes can achieve a good performance even in bad carrier or channel due to noise condition. Present communication uses Forward Error Correction (FEC) codes. In FEC coding, during transmission a extra redundant bits are added to the input given. This allows correcting the errors to a certain extent at the receiver.

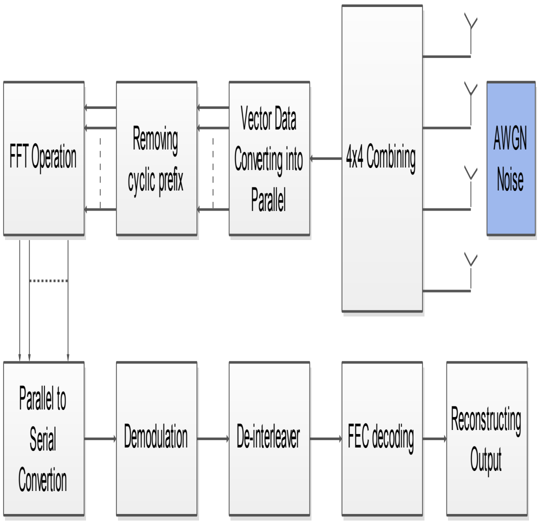


Fig. 1 Block Diagram of MIMO-OFDM Receiver

The modulation technique used is digital modulation. The digital modulation provides a better error correction capability; transmission takes place in security and avoids multipath fading. The digital data provided by any application is to be modulated onto a carrier for transmission. The data is transmitted by adding noise. The receiver receives the 4 elements at a time along with its complex conjugates. So at a time four elements are received with its four multiple copies. In C, real elements are received in one array and imaginary elements are received in other array. The matrix multiplication is done for each element, which contains both real and imaginary terms. To perform matrix multiplication, a function is created (Receiver_Amouti). A matrix multiplication function is created for this. The output obtained in C is compared with the MATLAB. The 4x4 combined output is input to cyclic prefix.

6. RESULTS AND DISCUSSIONS

This section deals with the results of the implemented 4x4 MIMO-OFDM receiver. The results are obtained with their output file that is generated after successful simulation in MATLAB and C. The performance of the system is estimated in MATLAB for image input and checking the performance using MSE and PSNR and in C the performance is evaluated on different embedded platforms.

Figure 2 shows the SNR versus BER plot for QPSK and QPSK integrated with 4x4 MIMO-OFDM. As shown in the graph, the bit error is reduced for QPSK integrated with 4x4 MIMO-OFDM receiver. It proves that the MIMO system is reliable and reduces the bit error rate. The error is very low using QPSK integrated with MIMO-OFDM that the noise level completely reduces at 2dB.

Implemented C code is used in RVDS to run on different ARM cores and performance analysis of the different cores is studied. Profiling in RVDS shows the performance of the codes, like bottle necks in code, code coverage by the processor. Stack utilization of the codes and cycle per instruction. Average time taken to execute a instructions. By running on the different cores and analysing it, the best core for the application of the 4x4-MIMO-OFDM Transmitter can select.

In MATLAB, the code development is made taking the color image and the binary image as shown in the Figure 3 and Figure 4. After transmission noise is added to the transmitted data. After increasing the SNR rate to a sufficient level, a reconstructed image equivalent to input is obtained.

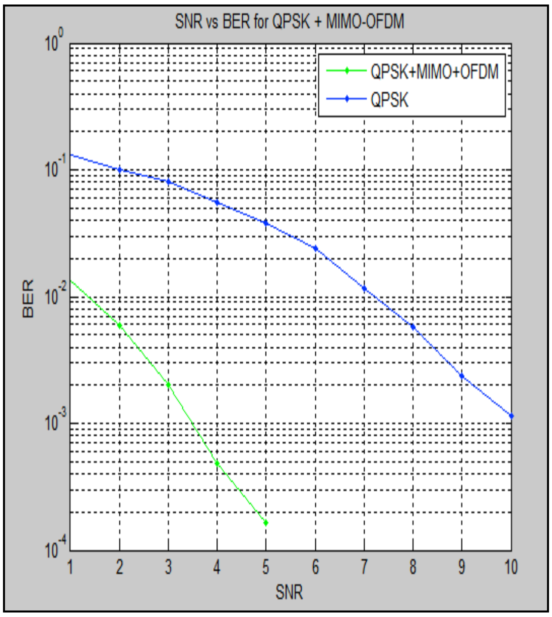
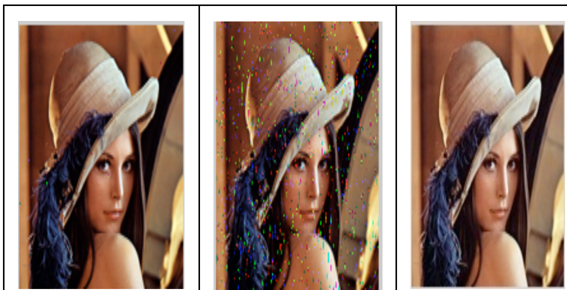


Fig. 2 SNR V/s BER of QPSK Integrated with MIMO-OFDM



Input Image	Image with noise at SNR=1db	Noise reduced at SNR = 5db

Fig. 3 SNR V/s BER of MIMO-OFDM for Color Images

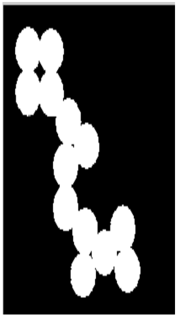
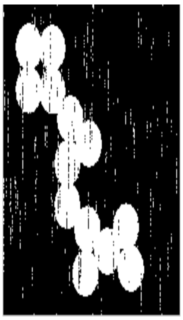
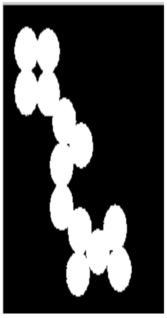
 Input Image	 Image with noise at SNR=1db	 Noise reduced at SNR = 5db

Fig. 4 SNR V/s BER of MIMO-OFDM for Binary Images

Figure 5 shows the profiling graph for the ARM Cortex A8 core. Left hand side top shows the over analysis section. The overall analysis section gives the information about the, total time, execution count and sampling information, number of estimated cycles is 2,75,845 and average instructions per second is 74,244. The code coverage pie charts are located in the bottom left and it gives a look at percentage of instruction and functions executed in the code during the execution in ARM Cortex A8 core. The code coverage by instruction pie chart shows graphically the percentage of executed assembly instructions in ARM Cortex A8 core. In that graph green represents the fraction of entirely executed instructions, red represents uncovered instructions, and yellow represents partially covered assembly instructions. The 2nd pie chart is like that of the code coverage by instruction chart, but shows code coverage function by function. For ARM Cortex A8, its 72.14% percent instruction is completed fully that is shown in green and 3.04% in yellow which is partially used and 24.82% in red color which is not used at all. And code coverage by function is 32.26 % fully executed so in green color, and 41.29% yellow and 26.45% in red respectively. The ARM Cortex A8 took 3, 89, 702 cycles to execute the code.

6.1 Optimised Results

Figure 6 shows the optimised profiler window of RVDS for ARM Cortex A8. The numbers of instructions are 252,201 average instructions per second 110,087 and number of cycles estimated are 359,086.The process involve in the tool of RVDS of 5 threads and its consumes it's on space of memory and execution time are reduced compared to profiled shown in Figure 5.

The first pie chart shown in Figure 6 shows the code coverage by instruction, where green colour indicates full usage of the code, yellow colour indicates partially used and red indicates some of the instructions not used at all. The profiler shows the code coverage by instructions of full coverage by 70.39%, partial coverage by 3.27% and the uncovered code is 26.34%. The code coverage by functions is as follows full coverage is 32.90%, partial coverage by 40.65% and the no coverage is 26.45%

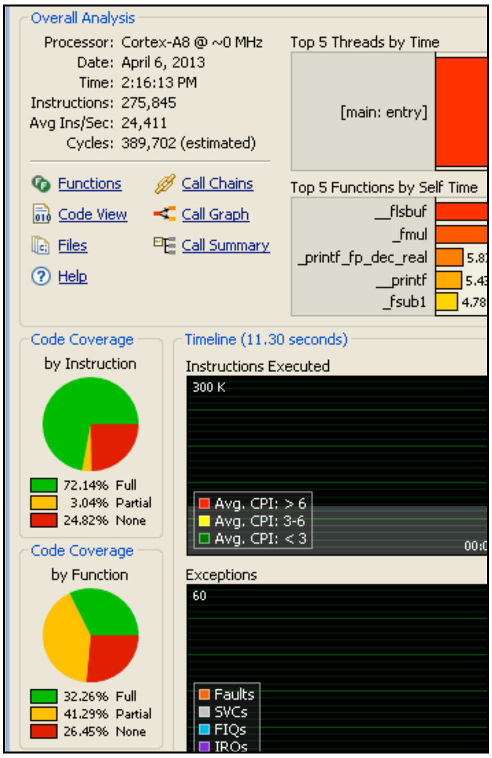


Fig. 5 RVDS Profiling for ARM Cortex A8

Figure 7 shows the plot of the ARM CortexA8 and its optimized coverage based on instructions, average instructions and the cycles. Blue bars for the ARM CortexA8 and the red bar for the optimised code on the ARM CortexA8. Also the various analysis studies were also carried out using the ARM1176 and its observations that the blue bar presents the ARM1176 and red bar presents the optimised code on the ARM1176.The observed results form the Figure 8 say 300000 instructions, 140000 average instructions and the estimated cycles obtained are 370000.In case of the optimised code for the ARM1176 it is observed 275000 instructions, 130000 average instructions and the estimated cycles of obtained are 130000.300000 instructions, 140000 average instructions and the estimated cycles of 355000.In comparison between all ARM CortexA8 and ARM1176 As shown in the Figure 7 and 8 the number of cycles taken to execute the code on ARM CortexA8 is less compared with ARM1176.Hence it is said that the ARM Cortex A8 processor is the better for MIMO-OFDM applications as embedded platforms.

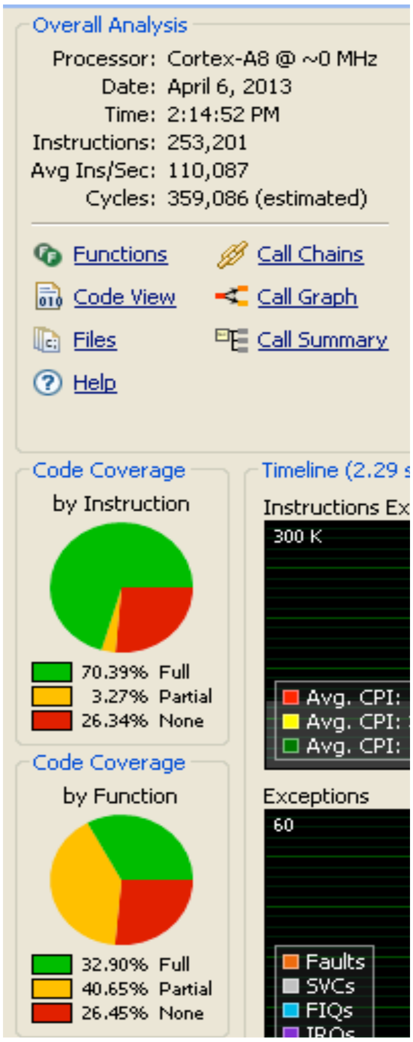


Fig. 6 RVDS Profiling for ARM Cortex A8 with Optimisation

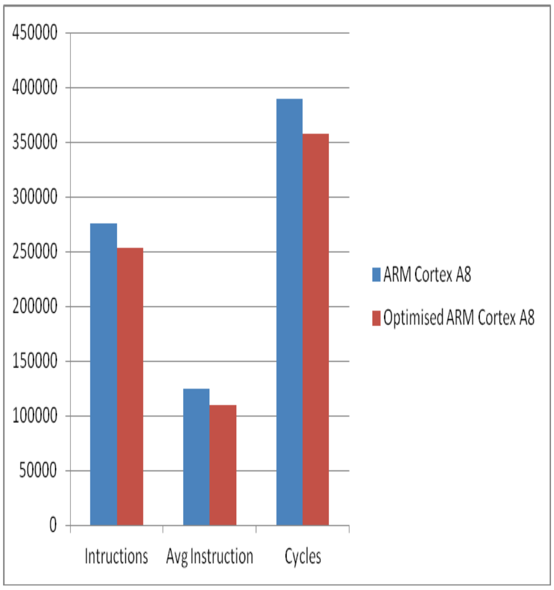


Fig. 7 Bar Graph for the ARM CortexA8 Performance

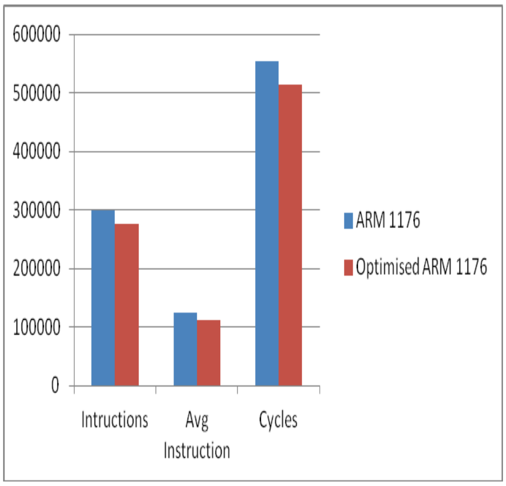


Fig. 8 Bar Graph for the ARM1176 Performance

7. CONCLUSIONS

The following are the conclusions drawn in the analysis and studies carried in this paper:

- MIMO-OFDM provides higher data rate whereas MIMO with QAM reduces the BER
- ARM Cortex A8 is a suitable platform to develop MIMO-OFDM receiver sub-system

Future work involves improving the overall performance of the system and the sub-blocks can be improved as follows:

- Using the higher order FFT to increase the number of subcarriers, thereby increasing the data rates
- Using the higher order QAMs to improve the spectral efficiency and improving the data rate by using proper channel estimation
- The MIMO used in this project gives a reasonable data rates, improves diversity and reliability. Using the advanced MIMO techniques gives a better performance