External Memory System Optimization for FPGA Based Implementation of Speech Signal Processing

J. Srinonchat, Member, IAENG

Abstract—A Field Programmable Gate Array is an integrated circuit designed to be configured and implemented any logical function. Recently, the use of an FPGA becomes a suited way to implement real time signal processing systems because of the increasing number of logic elements and clock frequencies. However the real time processing requires high speed, high precision and wide dynamic range which is necessary to use RAM for store and computation. This article presents an external SDRAM embedded broad for memory optimization system to implement the speech signal based on FPGA. The external 128 Mb SDRAM embedded broad is designed on two layers PCB to work with Xilinx FPGA module.

Index Terms—FPGA, Memory optimization, SDRAM, Real time speech signal.

I. INTRODUCTION

Recently, there has been a lot of research done to use speech signal as biometric feature in security systems and also in telecommunication applications which is mostly required a real-time computation and processing. There is often a trade-off between ease of use and system complexity. Therefore the Digital Signal Processing (DSP) or Field Programmable Gate Array (FPGA) chips are recently investigated to use in those applications. DSP normally uses a special processing equipment to collect, transform, filter, evaluate, enhance, compress and identify the signal, to obtain the signal form met the needs. The essential of the DSP technology is to convert the analog signal or some signals into digital signals and carry out corresponding processing for the converted digital signals. In general, FPGA is composed by six components, which separately are programmable input and output unit, basic programmable logic unit, embedded block RAM, abundant routing resources, underlying embedded function unit and embedded specific hard-core [1-4].

Real-time speech signal processing is usually required high-performance microprocessors. This is because of the computation complexity of applied algorithms. General purpose processors contain floating-point units able to carry out millions of operations per second at frequencies in the GHz range, what allows resolving the complex algorithms in just a few hundred of milliseconds. The use of an FPGA becomes a suited way to implement systems that require a high computational capability. Moreover the FPGA allows dividing and implementing algorithm as parallel parts. Modern FPGA from Xilinx and Altera have a large number of embedded broad which has been used in the research. There are some publications dealing with FPGA implementations of speech system. In [5], a FPGA implementation of the matching stage using a kernel based on an exponential function. In [6] the implementation of the speech system based on dedicated hardware is presented. The system consists of several stages dedicated to calculate the feature vectors, based on mel-frequency cepstral coefficients. In [7] and [8] there are presented only hardware implementations of some specific part of algorithms for speech recognition or speaker identification that allow a significant acceleration of the processing time. All of those publications are based on the modern FPGA which produces from the Xilinx and Altera.

Modern FPGA from Xilinx and Altera have a large number of embedded broad which has an embedded memory block for operation. In a Xilinx FPGA, these embedded memory blocks are known as block RAMs and in Altera FPGA is called embedded array block. This research is focus on memory optimization system for FPGA broad which is applied to use in speech processing system. Also the experiment is focused on a Xilinx FPGA. Block RAMs are dual-ported and can be accessed independently in a single cycle, typically 400 to 500 MHz. in the Xilinx Virtex FPGA. A memory virtualization technique called folding is applied that allows messages from different submatrices of quasi-cyclic code to share the physical block RAM.

II. REAL TIME SIGNAL PROCESSING FOR FPGA

FPGAs represent an excellent choice as regards high performance applications and low production level. FPGA becomes a suited way to implement real time signal processing. However, a real time signal processing imposes three basic requirements on arithmetic hardware: 1) high speed, to accommodate wide signal bandwidth 2) high precision, to minimize rounding errors, 3) wide dynamic range, to handle various signal to noise levels without overflow or underflow. These requirement typically conflict with each other creating a basis of tradeoffs.

One of the most common implementation tradeoffs is choosing between fixed point and floating point arithmetic. Fixed point arithmetic is easier to implement and has a lower level of computational complexity, therefore requires less logic to implement [9]. It, however, suffers from limited
FIFO buffer is normally always available and thus can be read at any time. The rd signal actually acts like a “remove” control signal, for write and read operation.

The overall process of software is following which uses FIFO (first-in-first-out) to be the input buffer. FIFO buffer is an “elastic” storage between two subsystems. It has two control signals, wr and rd, for write and read operation. When wr asserted, the input data is written into the buffer. While reading process is somewhat misleading. The head of the FIFO buffer is normally always available and thus can be read at any time. The rd signal actually acts like a “remove” signal.

III. SOFTWARE DESIGN

The overall process of software is following which uses FIFO (first-in-first-out) to be the input buffer. FIFO buffer is an “elastic” storage between two subsystems. It has two control signals, wr and rd, for write and read operation. When wr asserted, the input data is written into the buffer. While reading process is somewhat misleading. The head of the FIFO buffer is normally always available and thus can be read at any time. The rd signal actually acts like a “remove” signal.

The process of writing data in this experiment is following:
1) Firstly, the status of the VBLK signal is changed from “1” to “0”. This means the data is pass through the INPUT_FIFO in order to the wr_en HBLK is defined.
2) At the FIFO, rd_data_count signal will pass the signal to the Crt_SDRAM to indicate and notice that INPUT_FIFO get and store the input data.
3) When Crt_SDRAM gets the rd_data_Count signal, the rd_data_Count will provide the rd_en 1 pulse to that INPUT_FIFO which does synchronous with rd_clk (50MHz).
4) At this point, INPUT_FIFO will also provide the valid signal to indentify the status itself that “reading” or “writing” which this experiment uses the valid signal to make a “command writing” at SDRAM_Core by WRENQ valid (1 pulse per 1 address). Also Crt_SDRAM provides the address for SDRAM_Core to wait writing process to SDRAM.

The process of reading data in this experiment is following:
1) Firstly, the VBLK signal is changed from “1” to “0”. This means the data is pass through the INPUT_FIFO in order to the wr_en HBLK is defined.
2) At the FIFO, rd_data_count signal will pass the signal to the Crt_SDRAM to indicate and notice that INPUT_FIFO get and store the input data.
3) When Crt_SDRAM gets the rd_data_Count signal, the rd_data_Count will provide the rd_en 1 pulse to that INPUT_FIFO which does synchronous with rd_clk (50MHz).
4) While reading process is operated, the counter is also count the reading data to check the error.
5) While SDRAM is read, the system also provides the sda_dir signal. This signal is used to command the write_buffer of UART writing data from SDRAM to UART_TX.
6) The rd Eug signal also checks the buffer_full status. If the buffer_full status is operated, this means the data is full at UART_TX.

When writing process is operated to SDRAM, the counter is required to count the all data of each period of VBLK for checking the data with UARTTX. Therefore the VBLK status is defined as “1” for writing and “0” for reading.
IV. HARDWARE DESIGN

UART (Universal asynchronous receiver and transmitter) is a circuit that sends parallel data through a serial line. UARTs are frequently used in conjunction with EIA (Electronic Industries Alliance) RS-232 standard. Because the voltage level defined in RS-232 is different from that of FPGA I/O, a voltage converter chip is needed between a serial port and an FPGA’s I/O pins.

A UART includes a transmitter and a receiver. The transmitter is essentially a special shift register that loads data in parallel and then shifts it out bit by bit at a specific rate. The receiver, on the other hand, shifts in data bit by bit and then reassembles the data. The serial line is “1” when it is idle. The transmission starts with a start bit, which is “0”, followed by data bits and an optional parity bit, and ends with stop bits, which are “1”.

The overall process of hardware is following:

V. EXPERIMENT

This experiment uses the two layers PCB to design an external 128 Mb SDRAM embedded board for speech signal. This broad is design to work with Xilinx FPGA module. The results of each step are shown in the Fig.6-9.
VI. CONCLUSION

This paper presents the external SDRAM embedded broad for memory optimization system to implement the speech signal based on FPGA. This technique exploits the external SDRAM to improve the capable of floating and fixed point in FPGA broad. This results show that the external 128 Mb SDRAM embedded broad, which is designed on two layers PCB, can work with XilinxFPGA module as the digital speech data storage.

For real time processing, the digitized speech signals were compressed immediately after each sample was available. Then the compressed speech was written into the SDRAM. The whole compressing process was done in real time. The compressed speech signals were read back, up sampled by 8, and then played in the external speakers. The resulting compressed speech could be heard clearly with no audible background noise.

ACKNOWLEDGMENT

J. Srinonchat would like to express my deepest gratitude towards Prof. Dr. Sean Danaher, University of Northumbrai at Newcastle, for his guidance and excellent academic advice. Also I would like thanks the XilinxFPGA technical staff to advise this research.

REFERENCES