

Fast Fourier Transform Implementation Using a Programmable Logic Device

Maria Amélia Cardoso Aquiles¹, Cristiano W. Fanchin¹, Sandra M. Riedo¹,
Gabriel Kovalhuk^{1,2}, Mauricio Kugler^{3,4}

¹Concurrent Engineering R&D Lab (NuPES)

²Electronics Department (DAELN)

³Graduate School in Electrical Engineering and Industrial Computer Science (CPGEI)

⁴Laboratory of Embedded Systems Innovation & Technology (LIT)

Federal Center for Education in Technology (CEFET-PR)

Av. Sete de Setembro, 3165 - Curitiba/PR, Brazil - CEP 80230-901

aquiles@nupes.cefetpr.br (author), cristiano@fanchin.net (author), sandra_riedo@hotmail.com (author),
kovalhuk@nupes.cefetpr.br (advisor), mauricio@kugler.com (collaborator)

Abstract – This article describes the modeling of a Fast Fourier Transform algorithm in a programmable logic device. The project was partially described using state-machine description tools, while other specific blocks were described directly in VHDL language. The system comprises an A/D converter interface block, a FIFO memory with 256 positions, a state-machine to control the operations and memories for calculus and coefficients. The FFT calculation is sequentially made using a combinational two points FFT block and a multiplier. The practical tests were run using a microprocessed signal generation block, which provides the input vectors in digital form. Statistics about logic cells utilization were made for the system blocks, which can be synthesized in only one device. This work will be part of a master dissertation about electroencephalographic signals analysis and pattern recognition.

Key words: FFT, VHDL, Programmable Logic Devices.

INTRODUCTION

One of the most used techniques in digital signal processing is the frequency spectrum calculation through the Fourier Transform.

The motivation for this work is the necessity to implement a very fast FFT calculation, using the speed, parallel processing capabilities and the very simple external circuitry of programmable logic devices. This system will be part of a master dissertation about electroencephalographic signals analysis and pattern recognition [2][3].

METHODOLOGY

The system's functional description was made in two different ways. Part of the project was described using the software Renoir, from Mentor Graphics, which generates the code in VHSIC

Hardware Description Language (VHDL) from the state machines. The remaining parts of the project were directly described in VHDL [4].

The system's block diagram is shown in Figure 2. The *ad_ctrl* block generates the necessary signals to the A/D converter interface. After the conversion, the data is received and written into a FIFO memory (First In, First Out) with 256 positions, which corresponds to the *fifo* block. The 256 values of this FIFO memory are read and stored in a RAM memory (Random Access Memory), composed by two columns of 256 lines each (*ram_fft* block).

The FFT processing, triggered by the *start_fft* signal, is managed by a state-machine, with its simplified diagram shown in Figure 1.

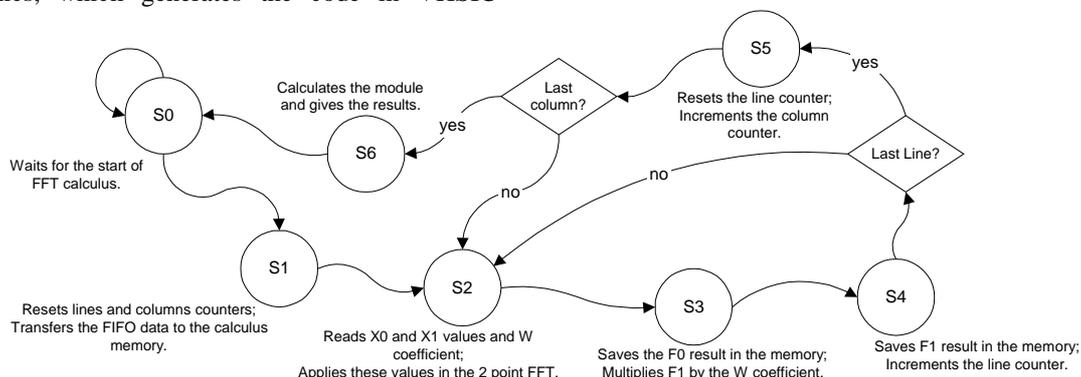


Figure 1 - FFT control state-machine

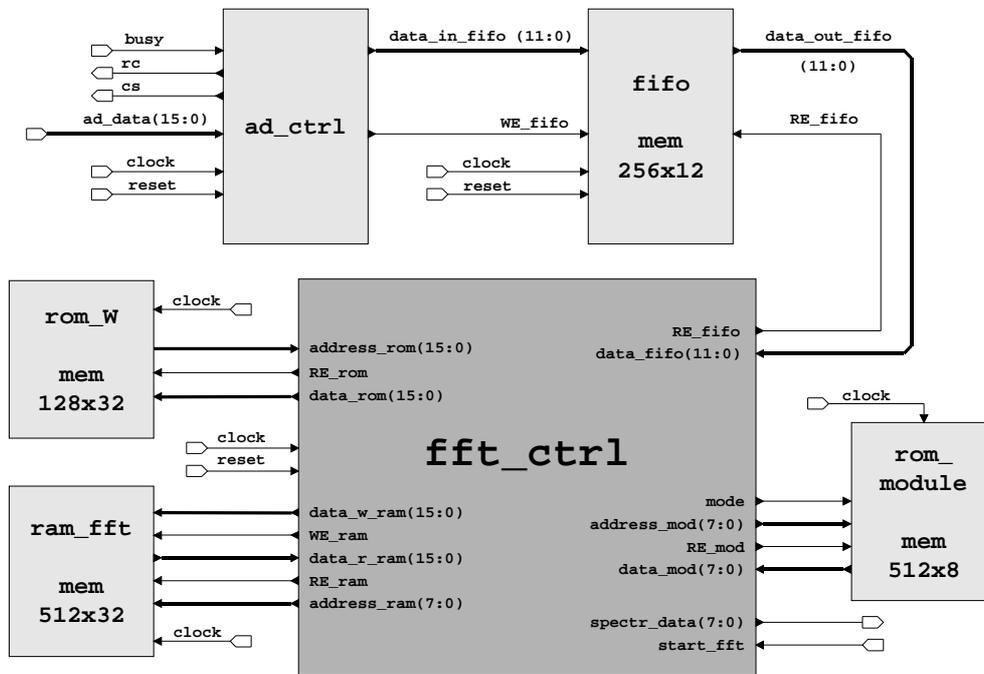


Figure 2 - FFT block diagram

The 256 points FFT calculation is sequentially made by a two points FFT block (Figure 3), which makes a sum and a subtraction in parallel, and then multiplies the subtraction result by its corresponding coefficient W_N^m [1]. The coefficients are stored in a ROM memory (Read Only Memory). The ROM, read once per iteration, corresponds to the *rom_w*. The data from each FFT stage are stored in one of the two *ram_fft* block columns, which are alternatingly used during the FFT calculation.

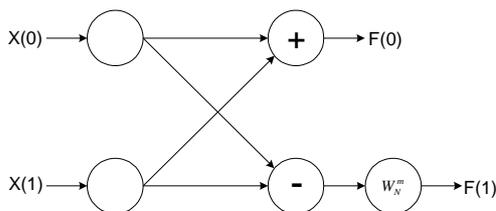


Figure 3 - Two-point FFT block diagram

The required result is the frequency spectrum module. As the FFT result is achieved in the rectangular complex form format, it is necessary to calculate the module of the coordinates. This is done via the square and square root tables, stored in the *rom_module* memory. The numeric representation used internally is 16 bit fixed-point, but the results are truncated to 8 bits for the module calculation.

The system's functional simulation, run using ModelSim software, from Model Technology, provided very good results comparing with the expected ones using equations. The logical synthesis and the optimization were done using the

Leonardo Spectrum software, from Exemplar, to an Altera's FPGA.

RESULTS

In order to validate the results, the system was loaded into an ACEX1k50 FPGA. The logic cells and memory bits utilization statistics were generated by Leonardo software (Table 1).

To provide a direct visualization of the generated values, a spectrum analyzer was implemented using the FFT core and, apart from the FFT calculation, VGA video signals are also generated. Thus, it's possible to connect a video monitor straight to the system and have a direct visualization of the frequency spectrum. In order to accomplish the test, two FPGA development kits were used, one of them containing a FLEX10k already connected to a VGA connector.

Table 1 - ACEX1k50 component utilization

BLOCKS	LOGIC CELLS	MEMORY BITS
<i>ad_ctrl</i>	32	-
<i>Fifo</i>	37	3072
<i>fft_ctrl</i>	784	-
<i>rom_module</i>	08	4096
<i>rom_w</i>	32	4096
<i>ram_fft</i>	01	*
FFT (total)	894	11264
acex1k (total)	2880	40960

* instantiate the *lpm_ram_dq* component, from Altera's library.

CONCLUSIONS

All the algorithms have been successfully tested. These tests were run using a microprocessed board to generate the test vectors that fed the spectrum analyzer inputs, directly in digital form.

The A/D converter interface hasn't been tested due to the noise generated by the FPGA. An ongoing new version is already taking the noise problems into consideration.

Due to the small number of used logic cells, this FFT core can be used in a lot of programmable logic systems that requires very fast digital signal processing. The VGA core that was implemented using another development kit could be fitted in the same FPGA of the FFT core. The second kit was used only because it already has a VGA connector.

The result module calculation, which is currently made using a memory table, will be made by a coordinate rotation algorithm (CORDIC) that is currently being implemented. This will improve the precision of the calculated values.

REFERENCES

- [1] AQUILES, M. A. C., Implementação de Multiplicador Digital em Dispositivos Lógicos Programáveis Utilizando VHDL. **VI Seminário de Iniciação Científica e Tecnológica – CEFET-PR**, p.169-172, 2001.
- [2] BANERJEE, A., DHAR, A. S., BANERJEE, S. FPGA realization of a CORDIC based FFT processor for biomedical signal processing. **Microprocessors and Microsystems**, v.25, n.3, p.131-142, 2001.
- [3] BOTROS, N., ZAKHEM, W., FFT Processor Using Programmable Gate Arrays. **Proceedings of Fifth Annual IEEE International ASIC Conference and Exhibit**, p.115–118, 1992.
- [4] PERRY, DOUGLAS. VHDL. 3rd ed. McGraw-Hill, 1998.
- [5] RABINER, L. R., **Theory and Application of Digital Signal Processing**, Prentice Hall, 1975.