Field Programmable Gate Array Realization of Microprogrammed Controller based Parallel Digital FIR Filter Architecture

Mohammed S. BenSaleh, *Member, IAENG*, Syed Manzoor Qasim, *Member, IAENG*
M. Bahaidarah, H. AlObaisi, T. AlSharif, M. AlZahrani, and H. AlOnazi

***Abstract*—This paper presents Field Programmable Gate Array (FPGA) realization of parallel architecture of microprogrammed controller based digital finite impulse response (FIR) filter. Digital FIR filter consists of a datapath and control unit. The datapath unit for the parallel FIR filter is a combination of bunch of registers, multipliers, adders and other digital building blocks. In this paper, we used the microprogrammed controller to control the operation of the datapath unit. The main advantage of the microprogrammed controller is its flexibility in modifying the microprogram stored in ROM based control memory. To demonstrate the proposed technique, we present a case study of third-order FIR filter. The parallel architecture is coded using VHDL based top-down hierarchical design methodology and realized in Spartan-3E FPGA using Xilinx ISE Webpack 12.2. Based on the FPGA implementation results, the maximum operating frequency of the third-order FIR filter is found to be 74.189 MHz and utilizing minimal FPGA resources. This leaves plenty of FPGA resources available for extending the design to realize higher order and high speed FIR filters which are commonly used in video and image processing applications.**

***Index Terms*—Digital Design, FPGA, Finite Impulse Response (FIR) Filter, Microprogrammed Controller, VHDL.**

I. Introduction

Finite impulse response (FIR) is a commonly used digital filter in many digital signal processing (DSP), image and video processing applications. FIR Filters are widely used because they have linear phase characteristics and guaranteed stability. Digital filters are mainly used for removing the undesirable parts of the input signal such as random noise or components of a given frequency content. FIR filters are commonly used in spectral shaping, motion estimation, noise reduction, channel equalization among many other applications. The simplest realization of an FIR filter is derived from (1).

Manuscript received July 23, 2012. This work was financially supported by National Center for Electronics, Communications and Photonics, King Abdulaziz City for Science and Technology (KACST) under grant no. 31/513 (Project Ejaz).

M. S. BenSaleh, S. M. Qasim, M. Bahaidarah and H. AlObaisi are with the National Center for Electronics, Communications and Photonics, King Abdulaziz City for Science and Technology, Riyadh, Saudi Arabia (emails: mbensaleh@kacst.edu.sa; mqasim@kacst.edu.sa).

T. AlSharif and M. AlZahrani are with the Electrical and Computer Engineering Department, College of Engineering, King Abdulaziz University, Jeddah, Saudi Arabia.

H. AlOnazi is with the Electrical Engineering Department, College of Engineering, King Saud University, Riyadh, Saudi Arabia.

The resulting architecture as shown in fig. 1 is called direct form realization because the multiplier coefficients are obtained directly from the filter transfer function

$$y(n) = \sum_{k=0}^{N-1} w(k) \cdot x(n-k) \quad (1)$$

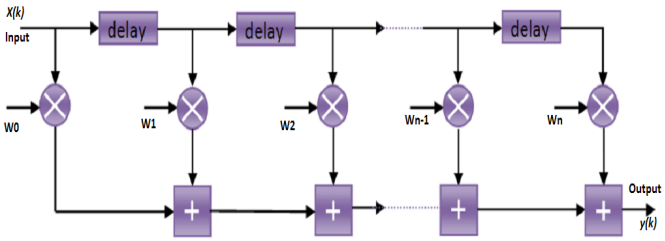

Fig. 1. Direct form FIR filter realization

Direct form FIR filters are also known as tapped delay line or transversal filters. The size of FIR filter is determined by the number of coefficients. A FIR filter of size N has N coefficients and N-1 delay elements to store the past values of the input. The size of the FIR filter is sometimes expressed in taps which is the number of delay elements+1 [1].

Different techniques for the realization of digital FIR filter using Field Programmable Gate Array (FPGA) have been reported and very well documented in the open literature [2], [3], [4]. However, the microprogrammed controller [5], [6], [7] based design of FIR filter and its hardware realization using FPGA has not been reported in the literature. The objective of this paper is to present the proposed technique using an example of parallel third-order FIR filter.

The rest of the paper is organized as follows. Section II presents the datapath architecture of FIR Filter. The microprogrammed controller design for parallel FIR filter is discussed in section III. FPGA implementation and simulation results are further presented in section IV and V respectively. The conclusions are presented in section VI.

II. Datapath Architecture

The proposed FIR filter architecture consists of two main building blocks which are datapath unit and control unit. The block diagram of third-order parallel FIR filter with the integrated datapath and control unit is shown in fig. 2. Fig. 3 illustrates the datapath architecture for third-order parallel FIR filter. The datapath architecture consists of the following sub modules: four 8-bit data registers, one 2-to-4 decoder, four 8-bit coefficient registers (w_o, w_1, w_2, w_3), four

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

multipliers, three 16-bit adders and one 16-bit register for latching the output [8]. Each sub modules are coded in VHDL and finally integrated to obtain the complete datapath. The control signals generated by the microprogrammed controller for this datapath are fed to different sub modules for proper operation of the FIR filter.

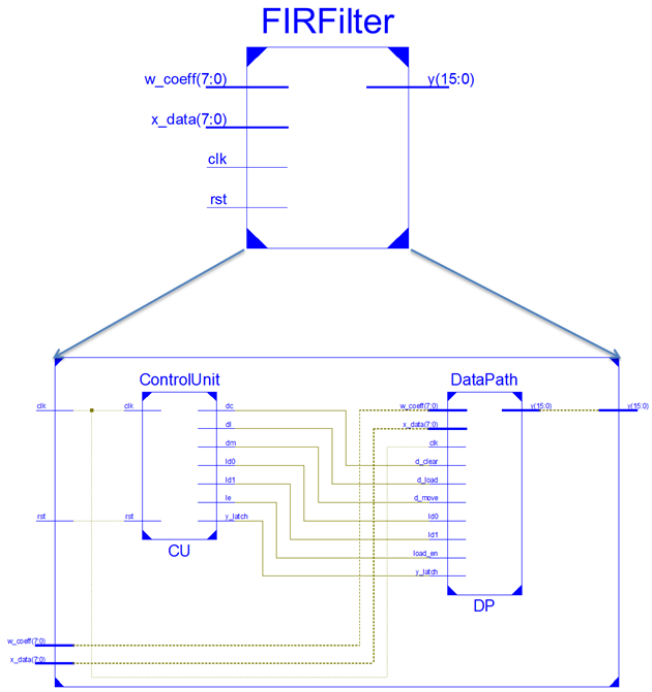

Fig. 2. Top level FIR filter module

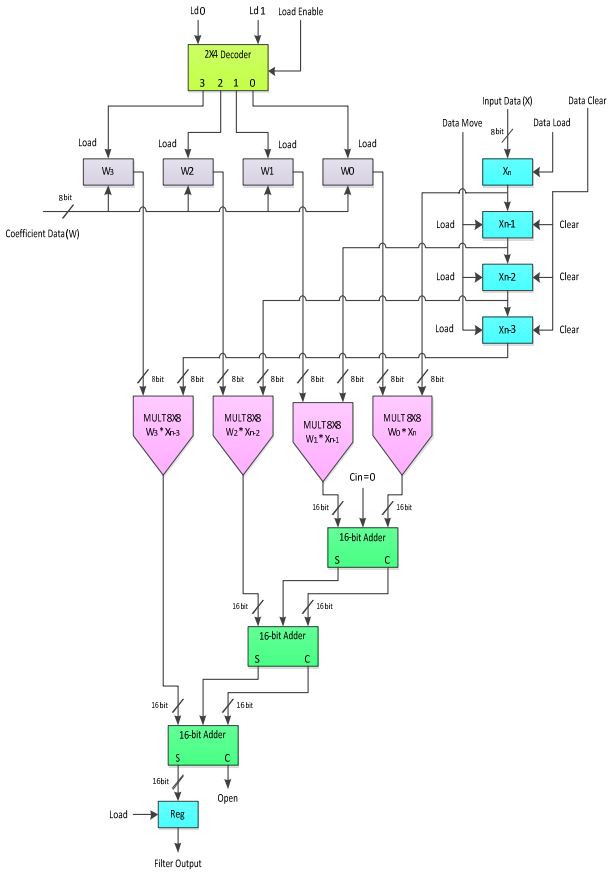

Fig. 3. Datapath architecture for third-order parallel FIR filter

III. Microprogrammed Controller

There are several methods to design the controller, such as hardwired controller and microprogrammed controller. In this paper, we used microprogrammed controller to implement the control logic of FIR filter [7], [8]. The main advantage of the microprogrammed controller is its flexibility to modify the microprogram in the EPROM based control memory [9], [10]. This makes the design of higher order FIR filter much easier.

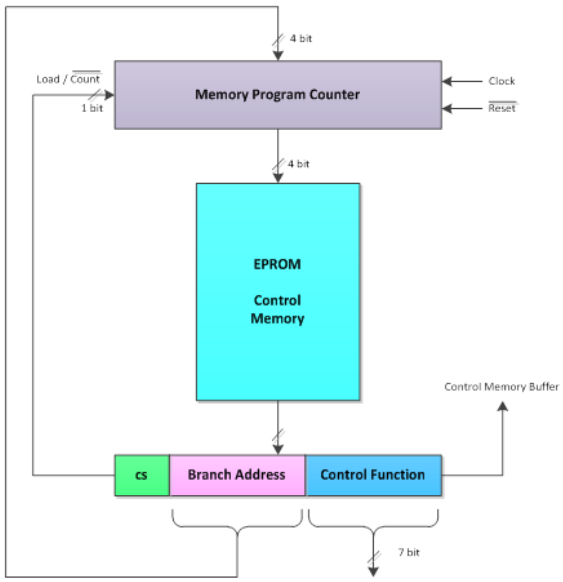

Fig. 4. Microprogrammed Controller for parallel architecture

As shown in fig. 4, microprogrammed controller consists of two main parts. The first part is responsible for addressing microinstructions kept in the control memory and the second part is used to hold and generate microinstructions for the datapath unit. The sequence of operations listed in table I is followed to generate the FIR filter output.

Table I presents the stored control information for the parallel architecture. The word stored in the control memory consist of three parts: single bit for signalling the counter either to count or to load external branch address, the next four bit represents the branch address and the rest of the bits represent the control signals for the datapath unit. In this paper, the microprogrammed controller generates seven control signals (12-bit microcode) for the FIR filter datapath. These control signals are then fed to different blocks of the datapath for proper operation.

As can be seen in table I, the FIR filter tap coefficient registers are loaded with data depending on load enable (Load_en) signal and the decoder output signals (Ld_1 and Ld_0). After loading the coefficient registers, all the input registers are cleared by making data clear (D_clear) signal high and then the input data to be filtered is entered into first data register after data load (D_load) is asserted high. The output (filtered data) is available only after the latch output (Y_L) signal is asserted high. The process is continued for the remaining registers only after the data move (D_move) signal is asserted high.

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I. CONTROL SIGNALS

#	Actions		Cs	Branch Address				Load_en	Ld1	Ld0	D_clear	D_Load	D_move	Y_L
1	Loading the Coefficients	w_0	0	0	0	0	0	1	0	0	0	0	0	0
2		w_1	0	0	0	0	0	1	0	1	0	0	0	0
3		w_2	0	0	0	0	0	1	1	0	0	0	0	0
4		w_3	0	0	0	0	0	1	1	1	1	0	0	0
5	Loading Input Data $x[n]$		0	0	0	0	0	0	0	0	0	1	0	0
6	Moving Input Data		0	0	0	0	0	0	0	0	0	0	1	0
7	Latch Output $y[n]$		0	0	0	0	0	0	0	0	0	0	0	1
8	Go to # 5		1	0	1	0	0	0	0	0	0	0	0	0

IV. FPGA REALIZATION RESULTS

The parallel FIR filter is designed and simulated using VHDL. To implement the proposed architecture, Spartan-3E (xc3s500e-4fg320) FPGA is used as the target device. ISE Webpack 12.2 is used for the synthesis, translation, mapping and place-and-route process. Different reports are generated by the tools. The FPGA resource utilization is listed in table II. The designed FIR filter operates at a maximum clock frequency of 74.189 MHz and consumes a small area out of the entire FPGA real estate leaving plenty of resources for implementing other parallel processors [11]. The RTL schematic generated by ISE 12.2 is shown in fig. 5, 6 and 7, which clearly illustrates the integration of the datapath with the control unit in the top level schematic, integration of program counter and ROM in the microprogrammed controller schematic and datapath unit respectively.

TABLE II. RESOURCE UTILIZATION

Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	88	9,312	1%
Number used as Flip Flops	84	-	-
Number used as Latches	4	-	-
Number of 4 input LUTs	87	9,312	1%
Number of occupied Slices	93	4,656	1%
Number of Slices containing only related logic	93	93	100%
Number of Slices containing unrelated logic	0	93	0%
Total Number of 4 input LUTs	150	9,312	1%
Number used as logic	87	-	-
Number used as a route-thru	63	-	-
Number of bonded IOBs	34	232	14%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	4	20	20%
Average Fanout of Non-Clock Nets	1.91	-	-
Maximum Frequency	74.189MHz	50MHz	-

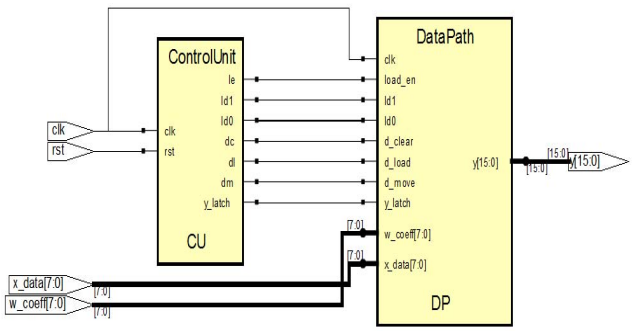

Fig. 5. Top level RTL schematic

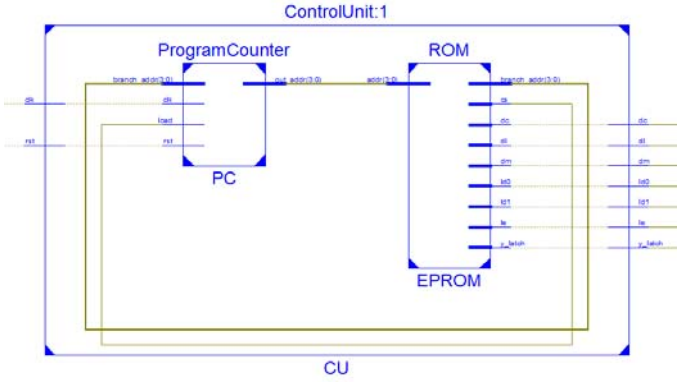

Fig. 6. RTL schematic of Microprogrammed Controller

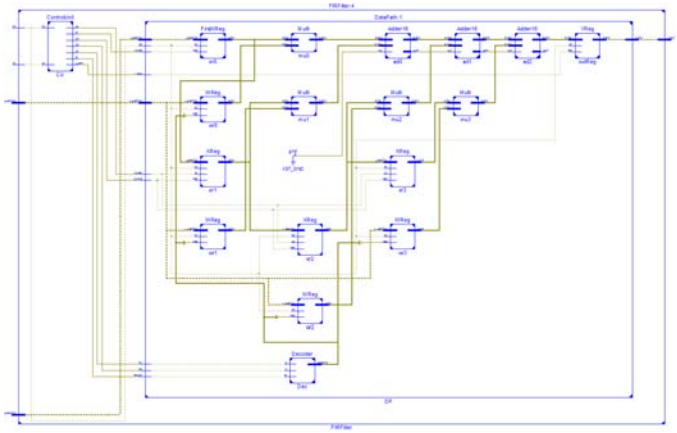

Fig. 7. RTL schematic of datapath unit

V. SIMULATION RESULTS

Three different test cases are used for testing the designed FIR filter circuit. The tap coefficients are chosen randomly with an objective to provide something that is observable at the output of FIR filter. These taps could be changed depending on the requirement of the application. The functionality of the parallel FIR filter is verified through simulation using Xilinx ISE built-in simulator.

Fig. 8 presents a snapshot of simulation results for the microprogrammed controller. Fig. 9, 10 and 11 presents the simulation waveform of the datapath unit for three different test cases as listed in table III. Finally, the datapath unit and microprogrammed controller are integrated together to demonstrate the simulation results of third-order FIR filter for each test case. The simulation waveforms of the top level FIR filter for each test case are presented in fig. 12, 13 and 14 respectively.

TABLE III. SIMULATION TEST CASES

Test Case	Tap Coefficients (W)	Input Data (X)	Output Data (Y)
1	{5, 4, 4, 1}	{3, 9, 7, 7}	{15, 57, 83, 102}
2	{3, 6, 6, 5}	{2, 10, 3, 3}	{6, 42, 81, 97}
3	{1, 2, 2, 1}	{1, 2, 3, 3}	{1, 4, 9, 14}

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

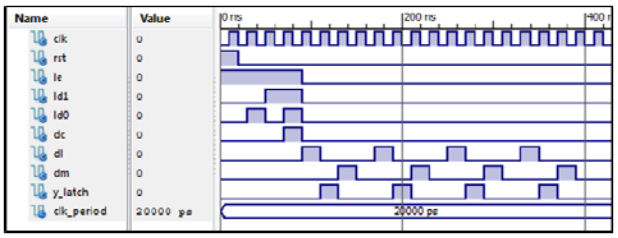

Fig. 8. Simulation waveform for microprogrammed controller

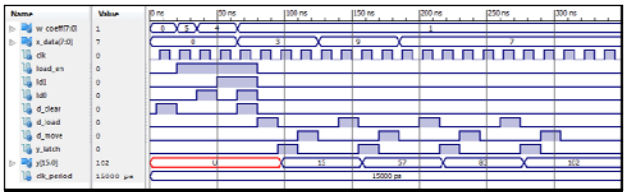

Fig. 9. Simulation waveform for datapath unit for test case no. 1

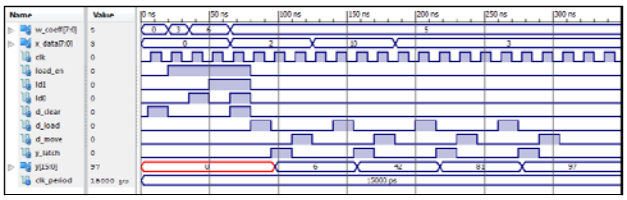

Fig. 10. Simulation waveform for datapath unit for test case no. 2

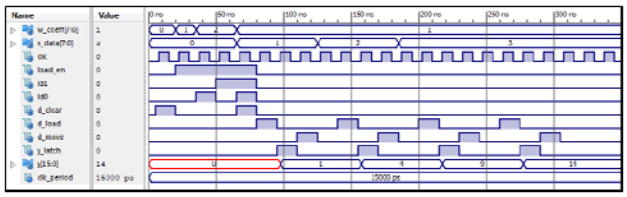

Fig. 11. Simulation waveform for datapath unit for test case no. 3

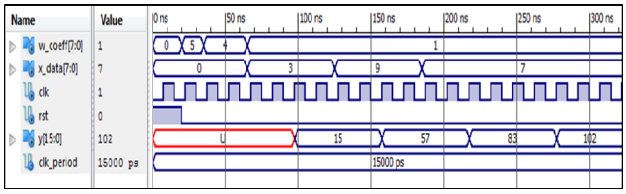

Fig. 12. Simulation waveform for FIR filter for test case no. 1

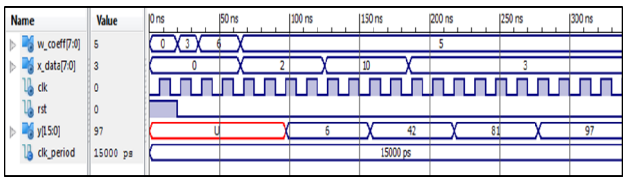

Fig. 13. Simulation waveform for FIR filter for test case no. 2

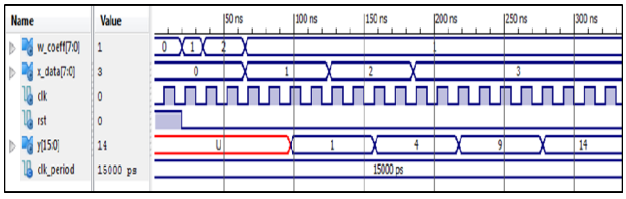

Fig. 14. Simulation waveform for FIR filter for test case no. 3

VI. Conclusions

In this paper, we have presented FPGA realization of third-order FIR filter using microprogrammed controller. A parallel architecture utilizing four multipliers and three adders along with other building blocks are used to demonstrate the proposed technique. Spartan-3E FPGA implementation results demonstrate that the design can operate at a maximum clock frequency of 74.189 MHz which is greater than the system clock frequency (50 MHz) of the used board and consumes a small area out of the entire FPGA real estate leaving plenty of FPGA resources for implementing other parallel processors on the same device. Since the size of the FIR filter presented in the paper is small, the results are not that significant, however, for higher order FIR filters, these results would be significant. Future efforts would focus on developing an intellectual property (IP) core of FIR filter based on the presented architecture. It is also envisioned to develop an equivalent sequential architecture of the FIR filter. Different optimization techniques such as pipelining will be applied and a comparison of parallel and sequential architecture for speed, area and power will be done.

Acknowledgment

The authors would like to thank Dr. Abdulfattah Obeid and the Engineering Staff of Micro Sensors and Devices Division, National Center for Electronics, Communications and Photonics, King Abdulaziz City for Science and Technology (KACST) for their support.

References

[1] A. Obeid, *Architectural synthesis of a coarse-grained runtime reconfigurable accelerator for DSP applications*, Ph.D Dissertation, Darmstadt University of Technology, Germany, 2006.

[2] F. Nekoei, Y. S. Kavian and O. Strobel, "Some schemes of realization digital FIR filters on FPGA for communication applications," in Proc. of 20th Intl. Crimean Conference on Microwave and Telecommunication Technology (CriMiCo), pp.616-619, 2010.

[3] Y. Zhou and P. Shi, "Distributed Arithmetic for FIR Filter implementation on FPGA," in Proc. of IEEE Intl. Conf. on Multimedia Technology (ICMT), pp. 294-297, 2011.

[4] U. Meyer-Baese, G. Botella, D. E. T. Romero and Martin Kumm, "Optimization of high speed pipelining in FPGA-based FIR filter design using genetic algorithm," in Proc. of SPIE, Vol. 8401, 2012.

[5] B. W. Bomar, "Implementation of microprogrammed control in FPGAs," *IEEE Transaction on Industrial Electronics*, Vol. 49, No. 2, pp. 415-422, 2002.

[6] A. Barkalov and L. Titarenko, *Logic synthesis for Compositonal Micro-program Control Units*, Springer, Berlin: Germany, 2008.

[7] R. Wiśniewski, A. Barkalov, L. Titarenko and W. Halang, "Design of microprogrammed controllers to be implemented in FPGAs," *Intl. Journal of Applied Mathematics and Computer Science*, Vol. 21, No. 2, pp. 401-412, 2011.

[8] M. Rafiquzzaman, *Fundamentals of Digital Logic and Microcomputer Design*, John Wiley and Sons Inc., 5th Edition, New Jersey: USA, 2005.

[9] A. A. Barkalov, L. A. Titarenko and K. N. Efimenko, "Optimization of circuits of compositional microprogram control units implemented on FPGA," *Cybernetics and Systems Analysis*,Vol. 47, No. 1, pp. 166-174, 2011.

[10] R. Wisniewski, M. Wisniewska, M. Wegrzyn and N. Marranghello, "Design of microprogrammed controllers with address converter implemented on programmable systems with embedded memories," in Proc. of 9th IEEE East-West Design and Test Symposium (EWDTS), pp.123-126, 2011.

[11] D. Amos, A. Lesea and R. Richter, *FPGA-based prototyping methodology manual: Best practices in design-for-prototyping*, Synopsys Press, USA, 2011.

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Date of modification: 17th September 2012

1. Page 1: Email address in the footnote on page 1 : ~~mqasim@kacst.edu.sa;~~ corrected to mqasim@kacst.edu.sa, just removed an extra semicolon at the end of the email id.

2. Page 3:The RTL schematic generated by ISE 12.2 is shown in fig. 5, 6 and 7, which clearly ~~illustrated~~ illustrates the integration of the datapath with the control unit in the top level schematic, integration of program counter and ROM in the microprogrammed controller schematic and datapath unit respectively.

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)