FPGA Implementation of Simple 8-Bit Signal Processor

Dharam Shah[1] Arpeet Kapadia[2] Mohin Bhavsar[3] Manoj Patel[4] Eagle Tarapara[5]
[1, 2, 3, 4, 5] Department of Electronics & Communication Engineering,
[1, 4, 5] V.G.E.C., Chandkheda, Ahmedabad, Gujarat.
[2, 3] S.C.E.T., Surat, Gujarat.

Abstract— Digital Signal Processor, or DSP, is a device used for processing signals digitally. A DSP is very similar to a microprocessor. In fact, it is regarded by many as a special microprocessor created particularly to process signals. The fundamental difference between a DSP and a microprocessor is what their built -in processing capabilities were designed for. Digital Signal Processor is used in Digital sound and image processing, Digital communications, Consumer electronics, Medical electronics Industrial and automation electronics. Signal processing encompasses a large variety of actions performed on signals like filtering, encoding/decoding, amplification, compression/decompression, modulation, pattern matching, mathematical / logical operations, level detection and much more. In this paper, We have made 8-bit signal processor on Spartan3 XC3S400 FPGA. It has Data bus (8-bit), Address bus (8-bit),8 bit ALU (8-bit adder/ subtractor, 8-bit logic blocks),Register file (8 registers R0-R7 each of 8-bits),Program Counter (8-bit), Instructions register (16-bit), Program memory (256x16 ROM),Data memory (256x8 RAM),MAC Unit (8x8 multiply accumulate),FIR filters (4th order Tap filters with rectangular window),Shifter (left/ right shift of maximum 8-bit shift),one input port & one output port.

Keywords: - ALU;MAC; Shifter; Filter; Control Unit

I. INTRODUCTION

Digital Signal processing (DSP) is the mathematical manipulation of an information signal to improve or modify it in different way. It can be represented by discrete time/frequency, or other discrete domain signals by a sequence of symbols or numbers and the processing of these signals.The goal of DSP is usually to filter, measure and/or compress continuous real-world analog signals. First step is to convert the signal from an analog to a digital form, by sampling and analog-to-digital converter (ADC), which turns the analog signal into a stream of numbers or symbols. Mostly the required output signal is also analog, which uses a digital-to-analog converter (DAC). Even if this process is more complex than analog signal processing and has a discrete value range, many advantages over analog processing in many applications, such as error detection and correction & in data transmission as well as data compression makes it more powerful [1].

The main applications of DSP are digital image processing, audio signal processing, audio/video compression, speech processing, speech recognition, digital communications, RADAR, Financial signal processing, SONAR, seismology and biomedicine. Specific examples are speech compression and transmission in digital mobile phones, hi-fi loudspeaker crossovers and equalization, computer graphics, image manipulation, weather forecasting, medical imaging such as CAT scans and MRI, economic forecasting, room correction of sound in hi-fi and sound reinforcement applications, seismic data processing, analysis and control of industrial processes, MP3 compression and audio effects for use with electric guitar amplifiers [2].

Conventional MPU (Micro-Processor Unit), such as X85 or X86, applies Von Neuman architecture. Data and program memory is same and there is only one address bus and one data bus in the conventional MPU. Different with common MPU, Digital Signal Processor uses Harvard architecture and has separated program memory and data memory with separated program data/address bus and data data/address bus. DSP have the property of pipe line/ parallel operation and even support multi-processor task. Compared with conventional MPU, DSP is more popular in embedded application for its small size, powerful calculation ability and saving power. However, more complex structure and limited on-chip resource cause the DSP application very difficult.

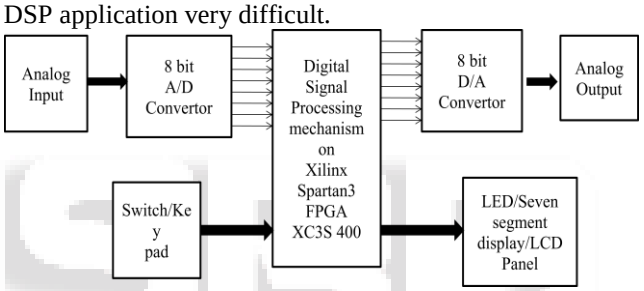

Fig. 1: Main Block Diagram

Figure 1 shows main implementation diagram. We had made digital signal processing mechanism on Spartan3 XC3S400 FPGA. It can have digital or analog input/output. Digital input can be given to FPGA by on-board Switch/Key pad. Digital output can be shown on LED/Seven Segment display/LCD panel on-board. If signal is analog then first it will be converted into digital by ADC. If we want final output in analog then at the output side we have to use DAC.

In this Paper, Section I introduces the Signal processor along with its application. Section II gives brief Implementation with full architecture. Section III covers all results. Conclusion is mentioned in Section IV.

II. IMPLEMENTATION

A. Main Architecture

Figure 2 shows simple 8-bit digital signal processor block diagram. As this is Digital signal processor, We have used Harvard structure for it. Data and Program memory are different as shown in figure. Control unit has mainly two blocks Program memory & Instruction decoder. Instructions are fetched into instruction register. Instruction decoder decodes the instruction and generates various control signals that will be applied to various parts. It has 20 pins. 8-pins are for Port-1 input port. 8-pins are for Port-2 output port. And other 4-pins are Reset, VCC, Ground and Clock. Register File consists of Eight 8-bit registers. Execute Unit consists of mainly four units ALU (Arithmetic and Logical Unit), MAC (Multiply & ACcumulate), Shifter, FIR Filter

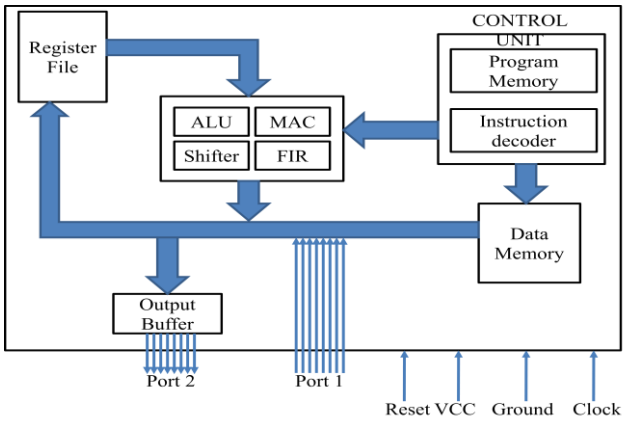

Fig. 2: Main Architecture of 8-bit Signal processor

B. Details of Architecture

Entire architecture can be divided into three units according to their operation: Fetch, Decode, Execute unit. Fetch unit fetches instructions from program memory. Decode unit decodes the instruction which have been fetched. It also generates control signals. Execute Unit execute the instruction according to control signals generated.

1) Fetch & Decode Unit

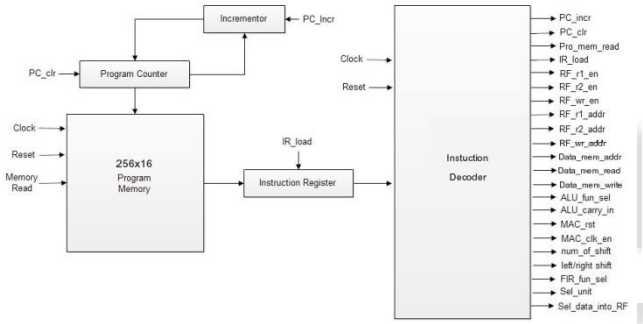

Fig. 3: Fetch & Decode Unit

It consists of Incrementor, Program counter, Program memory, Instruction register, Instruction decoder. At reset, Program counter is set to zero address. After each execution incrementor increments program counter vale. Instruction is first fetched into instruction register. Then it is loaded into instruction decoder. Instruction decoder generates various control signals as shown in figure 3.

2) Execute Unit

Execute unit consists of Eight registers. Two de-mux sends register values to one of the four units (ALU, MAC, Shifter, FIR Filter). 8*1 mux is used to send one value to given register. Data can be computed from Data memory and to Data memory.

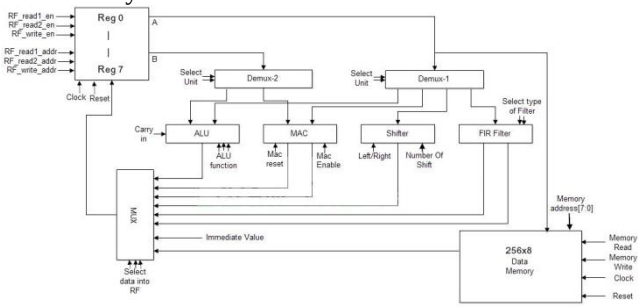

Fig. 4: Execute Unit

3) State Diagram

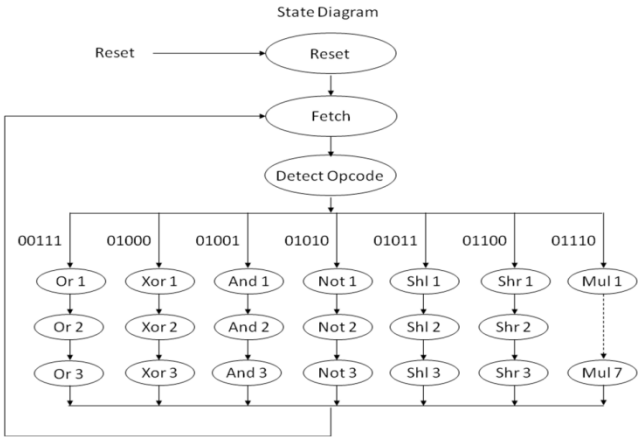

Fig. 5: State Diagram

VHDL implementation of Signal processor is done with State diagram. At reset or power on, Processor is in Reset state. Then it goes to the Fetch state. And then to Detect Op-code state. After getting op-code it will execute any instruction state as shown in figure 5. After executing one instruction, it will again go to fetch state. This process continues.

C. Details of Different unit

1) Control Unit

We had implemented two types of control unit: Hardwired [7] & micro-programmed [8]. Hardwired control unit requires more LUTs, Cells and more Gates. Micro-program control unit give more delay but require less hardware. So we select Micro-program control unit from the point of Hardware. There may be Because of the ease with which new instructions can be added and updated and for using pipelining with two separate memories of Data and program, we have chosen to use micro-programmed type of control unit design.

2) MAC

Dedicated MAC consisting of a multiplier implemented in combinational logic followed by an adder and an accumulator register that stores the result of output. The register output is fed back to one input of the adder, so that the output of the multiplier is added to the register per cycle. Combinational multipliers require more logic gates, but can compute a product quickly than the method of shifting and adding typical of earlier computers. MAC operation is very important for signal processing operations like Auto-correlation, Cross-correlation, Filtering, Convolution etc.

3) Shifter

Shifter is used for Rotate & shifting instruction. It is Barrel type shifter A barrel shifter is a digital circuit that can shift a data word by a specified number of bits in one clock cycle.

4) ALU

The different functions of the ALU(taken from reference [3]) have been shown in the table below. A and B are two 8-bit inputs and S2, S1, S0 and Cin together decide the type of operation as shown in the results. Cin also acts as the input carry for the operations. Adder is used in making of ALU. Adder is implemented with four methods(Ripple Carry Adder [5], Carry Look-ahead Adder [5], Carry Select Adder [6], Manchester Carry Adder [6]) & compared. After comparing, we have selected Carry Look-ahead Adder because it has high speed advantage.

Table 1: ALU Functions

S_2	S_1	S_0	C_{in}	Output	Function
0	0	0	0	F=A	Transfer A
0	0	0	1	F=A+1	Increment A
0	0	1	0	F=A+B	Addition
0	0	1	1	F=A+B+1	Add with carry
0	1	0	0	F=A-B-1	Subtract with borrow
0	1	0	1	F=A-B	Subtraction
0	1	1	0	F=A-1	Decrement A
0	1	1	1	F=A	Transfer A
1	0	0	D	F=A or B	OR
1	0	1	D	F=A xor B	XOR
1	1	0	D	F=A and B	AND
1	1	1	D	F=not A	Complement A

5) FIR Filter

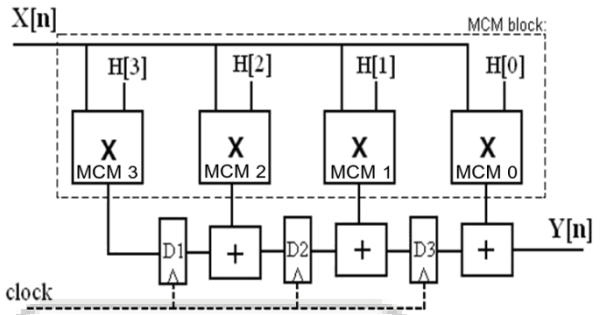

Fig. 6. FIR Filter

FIR filter [4] is implemented with MCM (Multiple Constant Multiplication) block. X[n] is input data. Y[n] is filtered output. H[n] is coefficients. We have implemented Low/High/Band-pass/Band-stop. Co-efficient are generated with MATLAB EDA tool. We have used 4th order filters by using rectangular window technique and tap filter design for FIR filters as shown in figure 6.

III. RESULTS

A. Hardware & Software used for Implementation :

Family:-Xilinx Spartan3
Device:-XC3S400
System gates = 400k
Simulator=ISE
Package type =Plastic Quad
Package pins =208
Speed Grade:-4(Standard Performance)
Temperature Range:-C [Commercial (0°C to 85°C)]

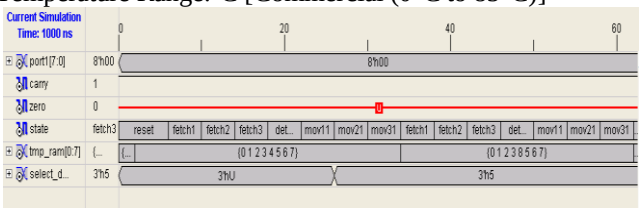

Fig. 7: MOV r4, [08h] instruction

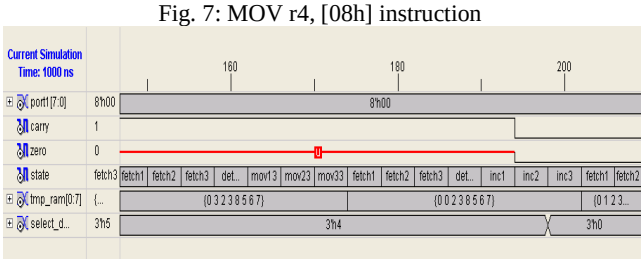

Fig. 8: MOV r1, port1

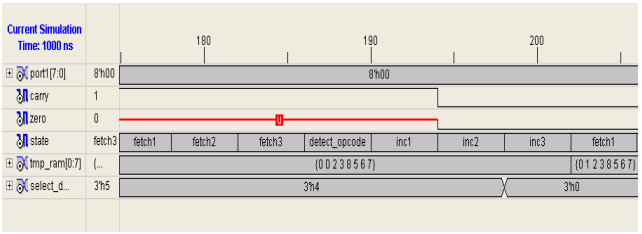

Fig. 9: INC r1

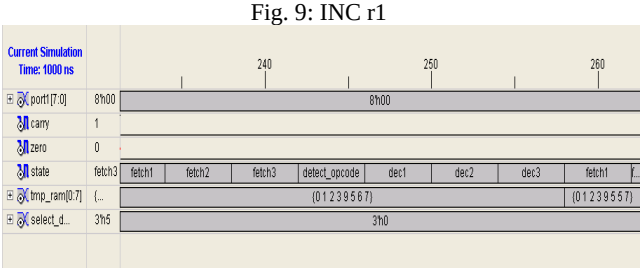

Fig. 10: DEC r6

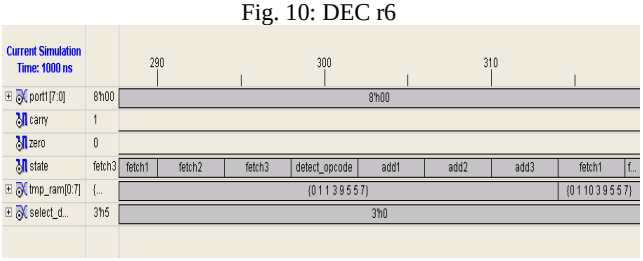

Fig. 11: ADD r2, r1, r4

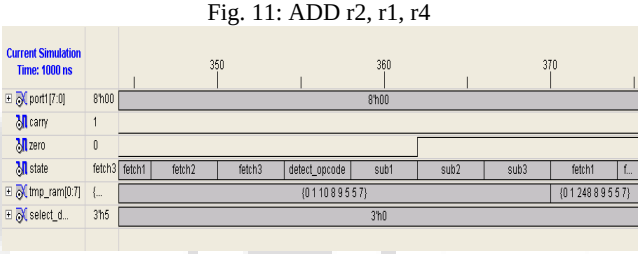

Fig. 12: SUB r2, r1, r4

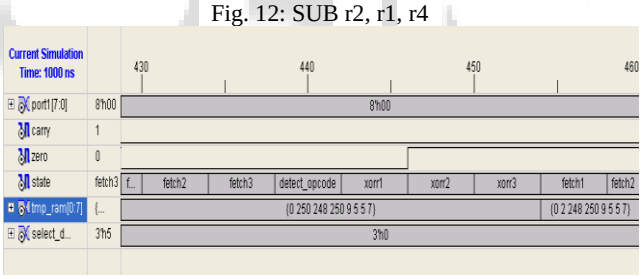

Fig. 13: XOR r1, r2, r3

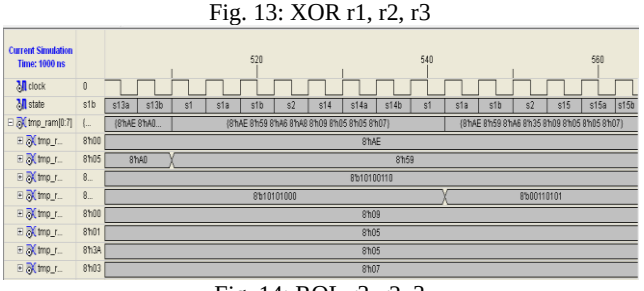

Fig. 14: ROL r3, r2, 3

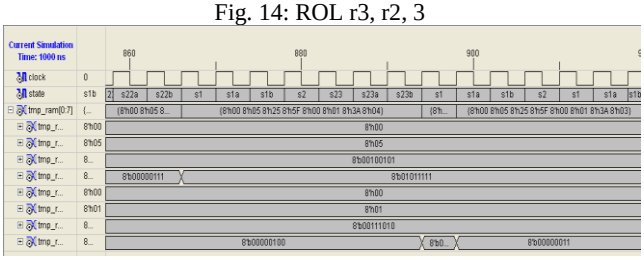

Fig. 15: SAR r7, r6, 4

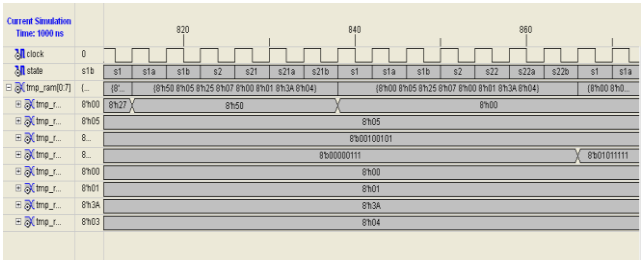

Fig. 16. SHR r0, r1, 4

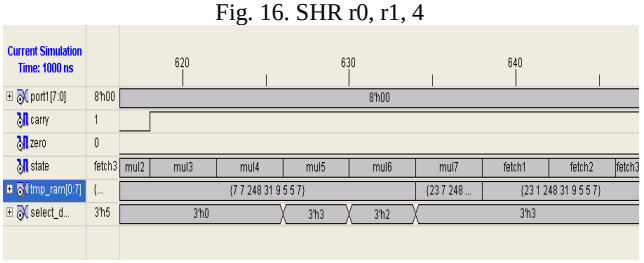

Fig. 17: R1:R0=MUL R3, R4

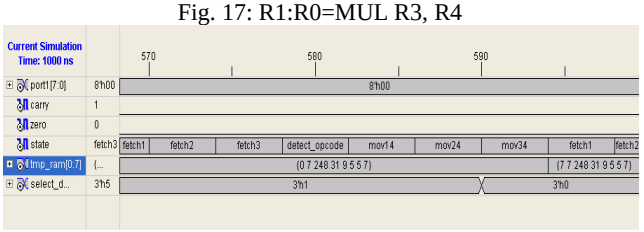

Fig. 18: MOV R0, R7

Table 2 : Device Utilization

Parameter	Used
Number of LUTs used	25%
Number of occupied slices	56%
Number of Slice flip-flops	32%
Total Equivalent Gate Count	52,925

Table-2 shows device utilization in terms of LUTs, slices, Flip-flops, Total equivalent Gate count. Some instructions which have been implmented in VHDL is shown in figures(7 to 18). Figure 7 moves data at memory lacation 08h to register r4. Figure 8 moves data at port1 to r1. Figure 9 gives r1=r1+1. Figure 10 gives r6=r6-1. Figure 11 gives r2=r1+r4. Figure 12 gives r2=r1-r4. Figure 13 gives xoring of r2 & r3 registers and puts it into r1. Figure 14 shows rotating r2 three times left side and put results in r3. Figure 15 shows shifting arithmentic right r6 four times and put into r7 register. Figure 16 shows shifting logically right r1 four times and put into r0 register. Figure 17 shows multiplication of register r3 and r4 and puts 16-bit output in R1:R0. Figure 18 shows moving of data from r7 to r0.

IV. CONCLUSION

We conclude that Simple 8-bit Signal processor implemented can be useful for many signal processing applications. We can also conclude that Signal Processor with micro-programmed control unit is better than signal processor with hardwired control unit in terms of hardware requirement. We have implemented each unit (like adder, multiplier, shifter, memory etc.) with different Techniques and have analyzed each unit in terms of Maximum Path Delay, Number of Slice used, Gate Count. After analysis, we have selected best technique for each unit.

REFERENCES

[1] Alan V. Oppenheim, Ronald W. Schafer, John R. Buck : Discrete-Time Signal Processing, Prentice Hall, ISBN 0-13-754920-2
[2] Boaz Porat: A Course in Digital Signal Processing, Wiley, ISBN 0-471-14961-6
[3] Morris Mano, Digital Logic & Computer Design, 3rd Edition. Pearson Prentice hall, ISBN 971-81-317-0070-9
[4] Oppenheim, Alan V., Willsky, Alan S., and Young, Ian T.,1983: Signals and Systems (Englewood Cliffs, New Jersey: Prentice-Hall, Inc.) ISBN 0-13-809731-3
[5] Vinay kumar, Tanvir Singh, Amit Kumar, "Comparative Study and Designing of Look Ahead Carry Adder and Ripple Carry Adder ", IJRECE,VOL 1 ISSUE 1 OCT-DEC 2013, ISSN: 2348-2281 (ONLINE).
[6] J. Rabaey, Digital Integrated Circuits: A Design Perspective, Prentice Hall India, *ISBN*: 9780130909961
[7] Englander, Irv (2009). The Architecture of Computer Hardware, Systems Software, & Networking, 4th Ed. Hoboken, NJ: John Wiley & Sons, Inc. p. 200. ISBN 978-0471-71542-9.
[8] Wiśniewski, Remigiusz (2009). Synthesis of compositional microprogram control units for programmable devices. Zielona Góra: University of Zielona Góra. p. 153. ISBN 978-83-7481-293-1

