High Speed Computation using 8-bit Microcontroller and Floating Point Co-processor Unit

[1]Anant Narayan Tiwari,[2]Dr. Vinod Kapse,[3]Pankaj Sahu
[1]Research Scholar, [2]Professor, [3]Assistant Professor
[1]Department of Electronics and Communication,
[1]Gyan Ganga Institute of Science and Technology, Jabalpur - 482003,India

***Abstract*—This Paper presents design and implementation of high speed computation using 8051 microcontroller and floating point co-processor unit uM-FPU 3.1. Floating point operations are most widely used in scientific and engineering calculations. The 8-bit microcontroller takes a lot of time in computation of floating point operations. In this work a 32-bit floating point co-processor is used in conjunction with the 8051 microcontroller. The mathematical computation is performed by FPU that decreases overall computation time.**

***IndexTerms*—Embedded Systems,Floating point Coprocessor,8051 Microcontrollerand High speed computation.**

I. INTRODUCTION

In the field of science and engineering high speed computation is used in position monitoring system, object tracking system, sensor data processing, robotic control, DSP application etc. In embedded system, the 8-bit microcontroller is most widely used even after development of 16-bit and 32-bit microcontrollers. In this work 8051 core, 8-bit P89V51RD2 microcontroller is used along with the 32-bit floating point co-processor uM-FPU V3.1. The microcontroller sends data to the FPU and sends instruction about what operation to be performed on the operands. FPU sends back the result to microcontroller after computation is finished.The hardware model description of the system is given in section III of this paper and next section gives detailed analysis in software prospective and interfacing of FPU.

II.LITERATURE SURVEY

In recent past year'sfollowing work has been carried out - V. Patil et. al. [1] has presented out of order floating point co-processor for RISC V ISA. In which they have described - Floating point operations are important and essential part of many scientific and engineering applications. Floating point coprocessor (FPU) performs operations like addition, subtraction, division, square root, multiplication, fused multiply and accumulate and compare. Floating point operations are part of ARM, MIPS, and RISC-V etc. instruction sets. The FPU can be a part of hardware or be implemented in software. This paper details an architectural exploration for floating point coprocessor enabled with RISC-V floating point instructions. The Floating unit that has been designed with RISC-V floating point instructions is fully compatible with IEEE 754-2008 standard as well. The floating point coprocessor is capable of handling both single and double precision floating point data operands in out-of-order for execution and in-order commit. The front end of the floating point processor accepts three data operands, rounding mode and associated opcode fields for decoding. Each floating point operation is tagged with an instruction token for in-order completion and commit. The output of the floating point unit is tagged with either single or double precision results along with floating point exceptions, if any, based on the RISC-V instruction set. The coprocessor for floating point integrates with integer pipeline. The proposed architecture for floating point coprocessor with out-of-order execution, in-order commit and completion/retire has been synthesized, tested and verified on Xilinx Virtex FPGA. S. Franchini et. al. [2] has designed Conformal ALU: A Conformal geometric algebra coprocessor for medical image processing. In this work they described - Medical imaging involves important computational geometric problems, such as image segmentation and analysis, shape approximation, three-dimensional (3D) modeling, and registration of volumetric data. In the last few years, Conformal Geometric Algebra (CGA), based on five-dimensional (5D) Clifford Algebra, is emerging as a new paradigm that offers simple and universal operators for the representation and solution of complex geometric problems. However, the widespread use of CGA has been so far hindered by its high dimensionality and computational complexity. This paper proposes a simplified formulation of the conformal geometric operations (reflections, rotations, translations, and uniform scaling) aimed at a parallel hardware implementation. A specialized co-processing architecture (Conformal ALU) that offers direct hardware support to the new CGA operators is also presented. The Conformal ALU has been prototyped as a complete System-on-Programmable-Chip (SoPC) on the Xilinx ML507 FPGA board, containing a Virtex-5 FPGA device. Experimental results show average speedups of one order of magnitude for CGA rotations, translations, and dilations with respect to the geometric algebra software library Gaigen running on the general-purpose PowerPC processor embedded in the target FPGA device. A suite of medical imaging applications, including segmentation, 3D modeling and registration of medical data, has been used as testbench to evaluate the coprocessor effectiveness. R. Aneesh et. al. [3] has presented a hybrid mode floating point conversion co-processor. In this paper they have described an abstract- level hardware implementation of the conversion between various number formats for FPGAs in modular way. Replacing the floating point expressions with specialized integer or fixed point operations can greatly improve the system performance in several applications. The replacement requires several types of conversions from one format to another format. The proposed conversion co-processor accelerator can work in parallel with HOST machine to accept a large amount of input data and convert to another format and apply fixed point or integer arithmetic operations and the result is converted back to the floating point

or fixed point format. The floating point conversions unit designs are fully compliant with the IEEE 754-2008 standard. R. Gonzalez et. al. [4] has designed FPGA based floating point UD filter coprocessor for integrated navigation systems. In this paper, a UD filter coprocessor with single-precision floating-point format that runs in FPGA is presented. A comprehensive hardware/software exploration is carried out in order to find the combination of subsystems that achieves the best throughput. Such examination determines that the best solution is fully in hardware. In addition, it is demonstrated the convenience of using high-level development tools to synthesize in hardware algorithms that have been originally designed to run on a general purpose microprocessor. Then, a UD filter for a 21-states system is developed as a coprocessor for system-on-chip integration. The coprocessor is validated using real-world data sets. Measurements of area, power, and energy are provided. It is found that the coprocessor is suitable for mid-range FPGA. In conclusion, it is demonstrated that new hybrid FPGA are adequate devices to implement battery-operated navigation systems for robotics.

III. HARDWARE DESCRIPTION OFTHE SYSTEM

The proposed system design consists of P89V51RD2 microcontroller, uM FPU V3.1 and other circuitry required to run the system. The P89V51RD2 is an 8051 core microcontroller with advanced features. It has 64KB flash memory, 2KB Ram, SPI and enhanced UART, four 8-bit I/O ports [5]. The uM-FPU V3.1 is a 32-bit floating point coprocessor. It can be easily interfaced by any microcontroller using SPI or I^2C interface.Many microcontrollers used in embedded systems lack floating pointsupport,butawiderangeofsensorsavailabletodayrequire additional computations or data transformation to provideaccurateresults.These features like advancedoperationsandfastexecutionallowstheuM-FPUV3.1 chip to outperform comparable software math libraries.Italso provides Flash memory and EEPROM for storing userdefined functions and data, and 128, 32-bit registers for floatingpointandintegerdata.Softwaremathlibrariesoftenuselargeamountsofmemoryonmicrocontrollers,particularlyasmorecomplexlibraryfunctionsareused.TheuM-PUV3.1chipoffloadsthisoverhead,andprovidesacomprehensivesetoffloatingpointoperations,including advancedfunctionssuchasFFT,matrixoperationsandNMEAsentenceparsing.DevelopmentsupportisprovidedbytheuM-FPUV3IDEwhichtakestraditionalmathexpressionsandautomaticallyproducesuM-FPUcodetargetedforoneofthemanymicrocontrollers andcompilerssupported.TheIDEalsointeractswiththebuilt-indebuggerontheuM-FPUV3.1chiptoassist in debugging and testing the uM-FPU code. The features of uM-FPU includes 32-bit IEEE floating point support, 32-bit operations, GPS serial input, NMEA sentence parsing, FFT operations, matrix operations, string handling etc.

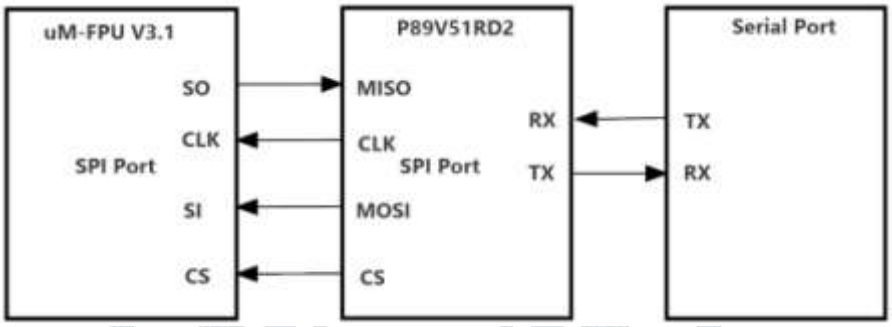

Figure 1. Block Diagram of System

IV. SOFTWARE DESCRIPTION OF THE SYSTEM

In this work FPU is interfaced with 8051 microcontroller using 3-wire SPI interface. The SPI connection uses separate data input (MISO) and data output (MOSI) pins on the microcontroller side. The CS (chip select) pin is tied to ground to enter SPI mode during Reset and it must remain low during operation [6].

IV.A FPU RESET OPERATION

The FPU is reset at the beginning of program to ensurethatthemicrocontrollerandtheFPUaresynchronized.TheFPUisreset, by sending nineconsecutive0xFFbytes for read, but it is recommended that ten 0xFF bytes should be sent by the microcontroller to ensure that at least nine0xFFbytesarerecognized even if the microcontroller and FPU are not synchronized. The minimum reset delay is set of 10 millisecond. After performing reset operation synchronization command is sent to ensure that FPU is synchronized with microcontroller [6].

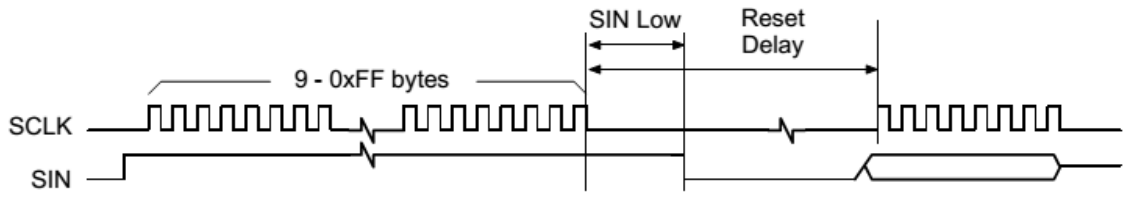

Figure 2. FPU Reset Timing Diagram [6]

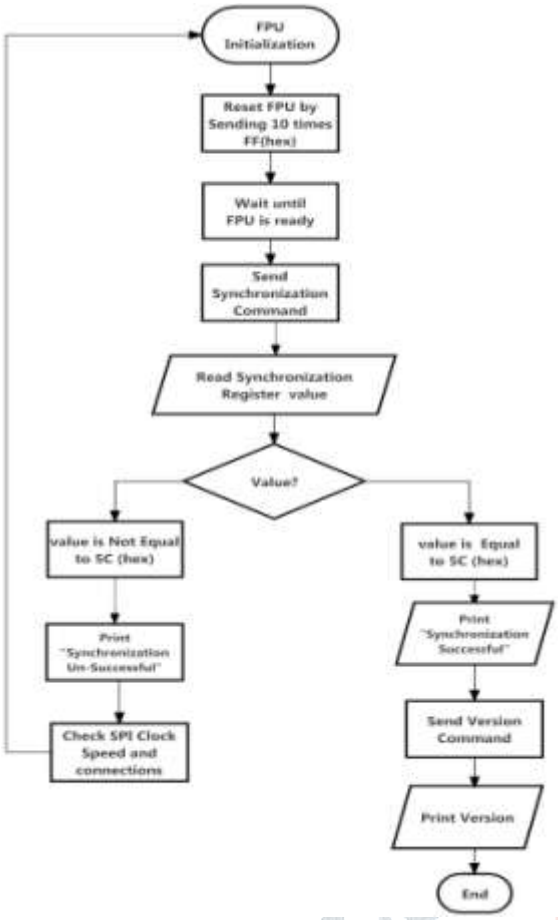

Figure 3. FPU Reset Flow Diagram

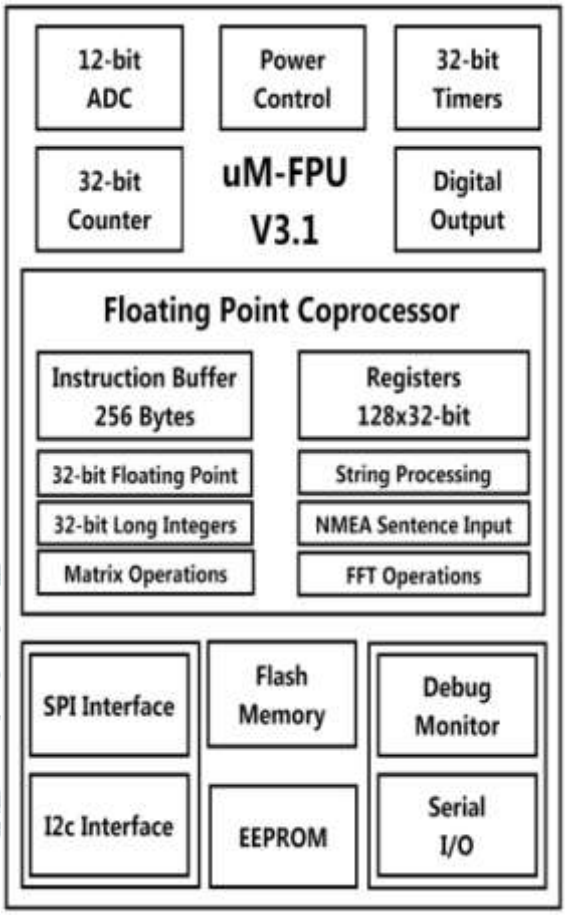

Figure 4. FPU Block Diagram

IV. B FPU Reading and Writing Data Operation

The microcontroller works as SPI master device whereas FPU works as a SPI slave device. Data is transmitted and received with the most significant bit (MSB) first using SPI mode '0', in which SCLK is active High (idle state is Low). Data latched on leading edgeofSCLKDatachangesontrailingedgeofSCLK.The maximum SCLK frequency supported by FPU is 15 MHz, but there must be a minimum data period between two consecutive bytes. The busy/ready status must be always checked to confirm the Ready status prior to any read operation.

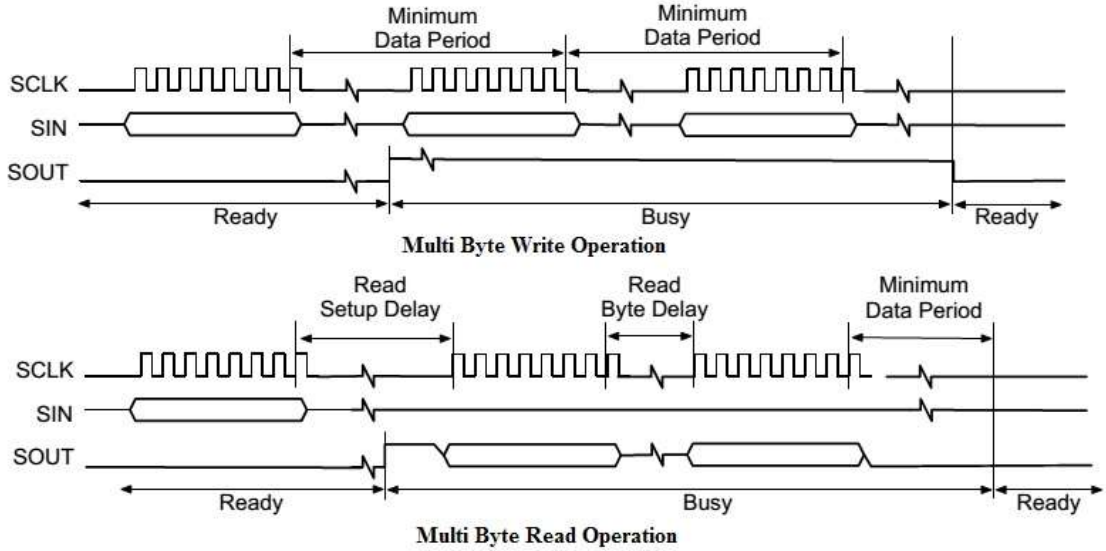

Figure 5. FPU Read and Write Data Timing Diagram [6]

V. RESULTS

The proposed system design is experimentally tested successfully.The experimental setup is shown in fig.6.The microcontroller and FPU both are running at 29.4912 MHz. The Floating Point Coprocessor performs operations like addition, subtraction, multiplication, division, square root etc. All these operations are performed by microcontroller individually and in conjunction with FPU and time taken in both the cases are compared. The Calculation time is calculated by starting timer before operation and stopping the timer after operation. The calculation time is determined by multiplying one machine cycle time with the timer register value.The comparison result is shown below in table 1. From the table it is clear that using FPU in conjunction with the microcontroller reduces computation time.

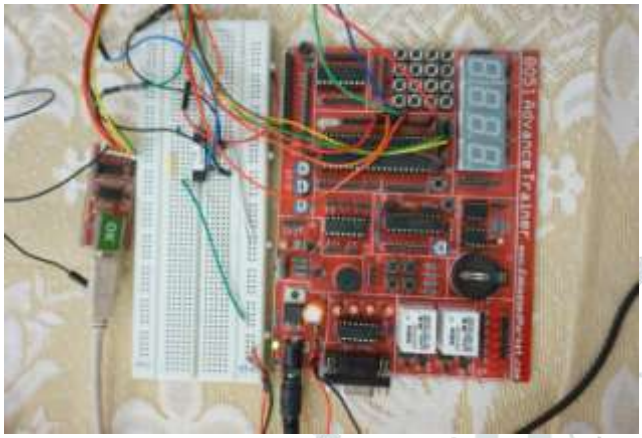

Figure 6. Experimental Setup

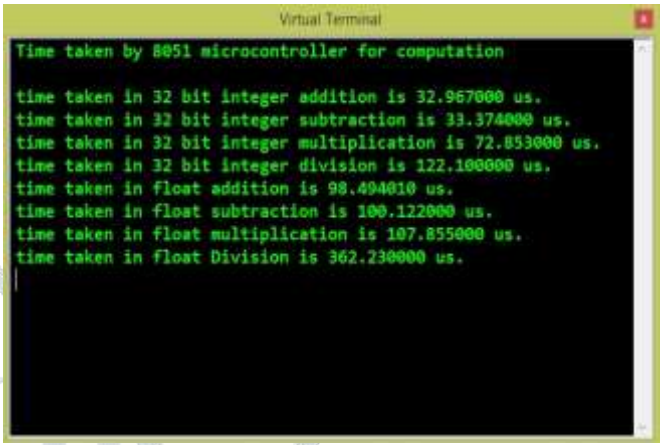

Figure 7. Calculation Time taken by Microcontroller alone

TABLE 1 CALCULATION TIME COMPARISON

S.No	Operation Performed	Time taken by microcontroller alone in µs	Time taken by Microcontroller and FPU in µs
1.	32-bit integer addition	33	18
2.	32-bit integer subtraction	34	18
3.	32-bit integer multiplication	73	18
4.	32-bit integer division	122	18
5.	Float addition	99	27
6.	Float subtraction	100	29
7.	Float multiplication	108	24
8.	float division	362	33

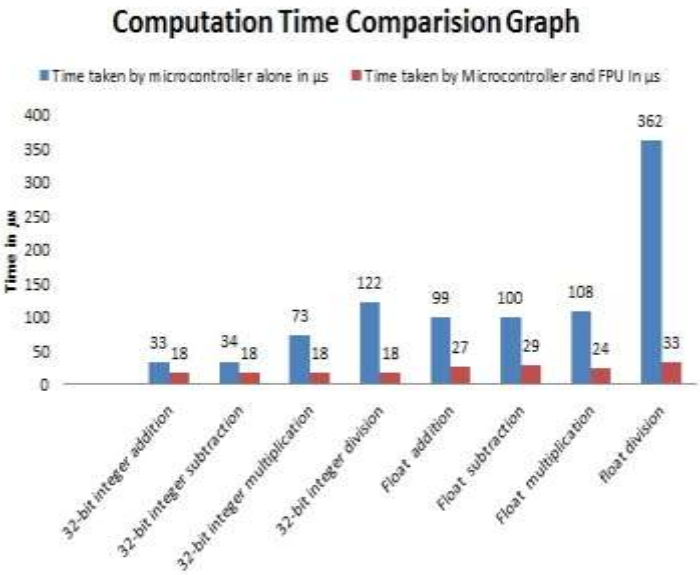

Figure 8. Computation Time Comparison Graph

VI. CONCLUSION

Inthispaper High Speed Computation using 8-bit Microcontroller and Floating Point Co-processor Unit is presented and implemented. By using FPU for computation work the overall computation time is reduced significantly and it is very useful for position tracking systems like GPS, robotics etc. and it extremely beneficial in DSP application. The developed system is fast and low cost device.

VII. ACKNOWLEDGEMENT

The primary requirements for a task to be successfully completed are constant inspiration, guidance, working environment and facility. I feel privileged to express my heartiest thanks and gratitude to guide Dr. Vinod Kapse, Mr. B.B. Shrivastava Scientific Officer-F, RRCAT and Mr.T.A. Puntambekar (Division Head, IOBDDD RRCAT, Indore) for providing me the time resource, guidance, encouragement and support throughout the work

REFERENCES

[1] V. Patil, A. Raveendran, P.M. Sobha, A. D. Selvakumar and D. Vivian, "*Out of order floating point co-processor for RISC V ISA*", IEEE 19thInternational Symposium on VLSI design and test (VDAT) 2015

[2] S. Franchini, A. Gentile, F. Sorbello, G. Vassallo and S. Vitabile*," ALU: A Conformal geometric algebra coprocessor for medical image processing*", IEEE Transactions on Computers volume : 64, Issue: 4, pp. 955-970,2014

[3] R. Aneesh, V. Patil, P.M. Sobha and A.D. Selvakumar,"*HMFPCC: - Hybrid mode floating conversion co-processor",*IEEE International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA), *2015.*

[4] R. Gonzalez, G. Sutter, C. Sistema and H.D. Patino," *FPGA based floating point UD filter coprocessor for integrated navigation systems",* IEEE Sixth Argentine Conference on Embedded Systems (CASE), 2015

[5] P89V51RD2 microcontroller datasheet http://www.keil.com/dd/chip/3711.htm.

[6] uM-FPU V3 datasheet http://micromegacorp.com/umfpu-v3.html.