Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-11, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Vedic Multiplication in AVR Micro Controller

Ravikant Tiwari[1], Er. Satbir Singh[2] & Dr. Balwinder Singh[3]
[1]M.Tech Student CDAC Mohali Punjab
[2] Project Engineer I CDAC Mohali Punjab
[3] Sr. Engineer & Coordinator (ACSD) CDAC Mohali Punjab

Abstract*: A high speed system depended greatly on multiplication algorithms which are used in coding. In embedded system there many micro controller, which not having multiplication Instruction in it, for example AVR 8515 micro controller. This paper presents new algorithm to speed up multiplication in AVR micro controller by Vedic mathematics concept, by this method we are able to reduce number of machine cycle. New algorithm is coded in assembly language; simulate using AVR Studio and ISS Lab software. Finally the results are compared with conventional multiplication algorithm to show the significant improvement in its speed.*

1. Introduction

Vedic mathematics is an ancient system of Indian mathematics. it is based on sixteen simple mathematical formulae form Vedas and give us unique method of calculation by which any mathematical problem can be solved ,it can be from any branch mathematics like trigonometry algebra ,geometry or arithmetic.

Bharati krishan Tirthaji , who was also the former shankracharya (major religious leader) at a puri temple in odisha India, Who translates all ancient Vedic texts and compile them into sixteen sutras and thirteen sub sutras. This text provides us unique and efficient method to speed up mathematical calculations. Vedic mathematics covers vast area in field of electronics research such as DSP "s, VLSI designs, FFT algorithms and embedded system design.[1,2]

In 21st century, the whole world nearly covered with 80% of embedded systems. The microcontroller capable to perform a very large number of different functions like addition, subtraction, transposed, multiplication, division etc. The performance of the microcontroller depends upon the complexity of mathematical computations which limits its speed. By using Digital signal processors (DSPs), we always having a option to improve the performance of system[3,4]. But due to the hardware complexity, the cost of project will be increase. The Vedic Mathematics is another option to speed up the mathematical computations by using the algorithm based upon the Vedic mathematics sutras. In our proposed work we demonstrate how to use these algorithms to program microcontrollers.[5,6,7]

If we are talking about the industries, these controllers have to compute the large number of calculations. Many processors are depends upon the result of these calculation. Therefore the time required for such type of calculation is very critical. We are considering microcontroller similar to humans in term calculation, human begin always try to find the logically way for these calculation to do in minimum time. Logics for micro controller are based on these sixteen formula, these logics can be programmed into the application

2. Conventional Multiplication Method

In conventional multiplication of two numbers in AVR 8515 micro controller we are mainly added multiplier times of multiplicand.

For example
99*68=99+99+99……64 Time +99.

1.1. Code for the multiplication

```
.INCLUDE "m8515def.inc"
.EQU NUM1 =68;
.EQU NUM2 =99;
.org 0x000
LDI R16, NUMBER1
LDI R17,NUMBER2
CLR R15
MOV R1,R16
L1:
ADD R16,R1
BRLO L2
L3:
DEC R17
BRNE L1
L4:
RJMP L4
L2:
INC R15
RJMP L3
```

Table 1. Machine cycle

Instruction set	Machine cycle required
LDI	1
CLR	1
MOV	1
ADD	1
BRLO	1
DEC	1
RJMP	1
BRNE	1/2

In the above code we are needed nearly 5 machine cycle for one time of loop,

68 x 99 time = 5+ (3+2)99=500 cycle

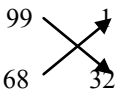
If we are using 10 MHz crystal frequency then period of the machine is = 0.1 µs
Then total time needed for calculation is
T = 500 x 0.1 µs = 50 µs

3. VEDIC MULTIPLICATION METHOD

3.1 Nikhilam Navatascaramam Dasatah Sutra

The formula simply means: "all from 9 and the last from 10.The formula can be very effectively applied in multiplication of number, which are closed to base like 10, 100, 1000 to the power of 10. Although it is applicable to all cases of multiplication, it is more efficient when the larger number will numbers multiple. Deviation is mainly the difference between the number and the base . Deviation may be positive or negative. Positive deviation is written without the positive. Since deviation between number is obtained by Nikhilam sutra we call the method as Nikhilam multiplication

Example
99 x 68
Nearest base = 100
99 = (100 - 1)
68= (100 – 32)
Column1 column2



67 32

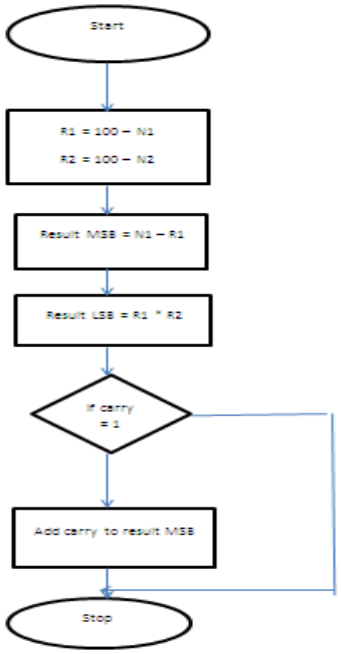
Common difference = 67
Difference multiplication = 32
Multiplication = 6732

3.2 Algebraic proof

Let the two number n1 and n2 be less than the selected base say x..n1 = (z-a), n2 = (z-b). Here a and b are corresponding deviation of the number n1 and n2 from the base z .Observer that x is a multiple of10[8].

Now n1xn2 = (z-a) (z-b) = z.z – z.b – a.z +ab
= z(z – a – b) + ab
= z [(z-a) – b] + ab = z (n1 – b) + ab.

Vedic multiplication flow chart
Where
1. R1 and R2 are any variable.
2. N1 and N2 are any which have to



If we coded the 8515 avr by above algorithm
Needed only 27 machine cycle for the calculation multiplication
Total time required = 27 x 0.1µs = 2.7 µs

4. Discussion and Conclusion

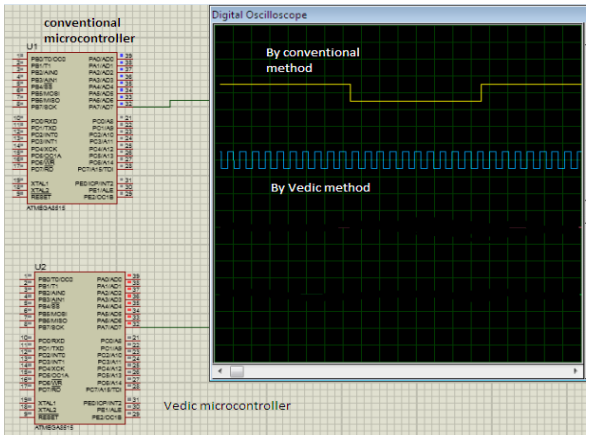
The new algorithm result obtained from proposed. Vedic multiplication algorithm is faster than conventional multiplication algorithm and reduces the number of machine cycle and power consumption needed for execution of multiplication. Table no 2 show the comparison of conventional multiplication algorithm with Vedic multiplication algorithm in terms of number of machine cycle and time needed for execution . The timing result show „s that Vedic multiplication algorithm has the greatest advantage as compared to convention al multiplication in terms of execution time .

Table No.2 comparison of conventional multiplication and Vedic multiplication (in ns)

Table 2

Parameter	Conventional multiplication	Vedic multiplication
Machine cycle	500	27
Time for execution	50µ	2.7µ

Assume the operating frequency in 10 MHz ":s



4. Vedic multiplication algorithm application

The idea can be adapted to various applications for making these faster and smoother.

1. Mobile applications.
2. Transformer winding machines.
3. To make the backup system of aircraft

This paper present a highly efficient algorithm for multiplication " Nikhilam Navatascaramam Dasatah Sutra" based on old India mathematic is called Vedic mathematics . by this algorithm can be very effectively speed up the mathematical computations it will take nearly 50% less time .

6. References

[1] P.-M. Seidel, L.D. McFearin and D.W. Matula, "Secondary radix recodings for higher radix multipliers", IEEE transc. on Computers, Vol 54, No. 2, pp. 111- 123, February 2005.

[2] P. Mehta, and D. Gawali, "Conventional versus Vedic mathematical method for Hardware implementation of a multiplier," Proc. IEEE ACT-2009, pp. 640-642, Dec. 28-29, 2009.

[3] H. D. Tiwari, G. Gankhuyag, C. M. Kim, and Y. B. Cho, "Multiplier design based on ancient Indian Vedic Mathematics," Proc. IEEE International SoC Design Conference, pp. 65-68, Nov. 24-25, 2008.

[4] Z. Huang, and M. D. Ercegovac, "High-Performance Low-Power Left-to-Right Array Multiplier Design," IEEE Transactions on Computers, vol 54, no. 3, pp 272-283, March 2005.
P. Saha, A. Banerjee, A. Dandapat, P. Bhattacharyya, Vedic Mathematics Based 32-Bit Multiplier Design for High Speed Low Power Processors

[5] A. Asati and Chandrashekhar, "An Improved High Speed fully pipelined 500 MHz 8×8 Baugh Wooley Multiplier design using 0.6 µm CMOS TSPC Logic Design Style", Proc. IEEE, ICIINFS 2008, pp. 1-6, Dec. 8-10, 2008.

[6] A. Chandrakasan and R. Brodersen, "Low-power CMOS digital design" , IEEE Journals on Solid-State Circuits, Vol. 27, No. 4, pp. 473–484, Apr. 1992.

[7] N.-Y. Shen and O. T.-C. Chen, "Low-power multipliers by minimizing switching activities of partial products", Proc. IEEE, ISCAS 2002, vol. 4, pp. 93–96, May 2002.

[8] B. Parhami, Computer Arithmetic Algorithms and Hardware Designs, 1st ed. Oxford, U.K.: Oxford Univ. Press, 2000.