# Hardware Implementation of Greatest Common Divisor using subtractor in Euclid Algorithm

Darshana Upadhyay
Institute of Technology,
Nirma University
Ahmedabad, Gujarat, India.

Harshit Patel
Institute of Technology,
Nirma University
Ahmedabad, Gujarat, India.

## ABSTRACT

This paper proposed an efficient implementation of digital circuit based on the Euclidean Algorithm with modular arithmetic to find Greatest Common Divisor (GCD) of two Binary Numbers given as input to the circuit. Output of the circuit is the GCD of the given inputs. In this paper subtraction-based narrative defined by Euclid is described, the remainder calculation replaced by repeated subtraction. The selection of the Division Method using subtractor is due to ease of implementation and less complexity in connection with reduced hardware. The circuit is built using basic digital electronic components like Multiplexers & comparator (A<B) as control function and Registers, Full subtractor as Register transfer components. Although the circuit is developed to handle 4-bit of data, it can be easily extended to handle any number of bits just by increasing capacity of basic components (Multiplexer, Registers, Full Subtractor and comparator).

## General Terms

GCD – Greatest Common Divisor

## Keywords

Greatest Common Divisor ,Magnitude Comparator, Multiplexer, Full Subtractor, Euclidean Algorithm.

## 1. BACKGROUND

For two nonzero integers a and b, their greatest common divisor is the largest integer which is a factor of both of them [1]. It is denoted (a, b). For instance, (12, 18) = 6 and (-9, 15) = 3. Do not confuse our usage of parentheses in (a, b) with the notation for open interval in calculus, The number 1 is always a common divisor, and it is the greatest common divisor exactly when a and b are relatively prime[1]. The naïve method of finding the greatest common divisor of two integers is to factorize each into prime and extract the greatest common divisor from the prime power factors that appear. Factorizing is hard (even on a computer when the integer has several hundred digits), so this method of computing (a, b) is not good when a and b are large. There is a method of computing greatest common divisor, going back to Euclid, which avoids the need to factor at all[2]. Instead of factoring, we will do successive division with remainder in such a way that the remainder keeps dropping. The last nonzero remainder will turn out to be the greatest common divisor. Let a and b be nonzero integers ($|a|<|b|$). Divide a by b and carry out further division according to the following method, where the old remainder becomes the new divisor[2]:

$a = b*(q1) + r1;\ 0 < r1 < |b|$

$b = r1*(q2) + r2;\ 0 < r2 < r1$

$r1 = r2*(q3) + r3;\ 0 < r3 < r2$ ……

The non-negative remainders r1, r2… Are strictly decreasing, and thus must eventually become 0. The last nonzero remainder is the greatest common divisor. The key idea that makes Euclid's algorithm[3] work is this: if a = b + mk for some k in Z, then (a, m) = (b, m). That is, two numbers whose difference is a multiple of m have the same GCD with m. Indeed, any common divisor of a and m is a divisor of b = a – mk, and therefore is a common divisor of b and m. This tells us (a, m) <= (b, m). Similarly, any common divisor of b and m is a divisor of a = b + mk, and therefore is a common divisor of a and m. Thus (b, m) <= (a, m) too, so (a, m) = (b, m).

## 2. INTRODUCTION

Finding Greatest Common Divisor is one of the most basic tasks of Mathematics and Algebra[4]. There are several methods available to find GCD of given two numbers. Some of them are 1) Method of Factorization and 2) Division Method. Algorithm Used to find Greatest Common Divisor (GCD) in this the paper is the Division Method (Euclidean Algorithm). Design of circuit is carried out in the simulator software named Logisim 2.7.1. For digital electronic implementation of circuit to find GCD, the Division Method is more convenient then the Method of Factorization which can be justified by the following subsections.

### *2.1.1 Method of Factorization*

According to the method of factorization, express each given number as product of primes and take product of common factors.

**EXAMPLE:** GCD of 136 and 144.

136 = 2 x 2 x 2 x 17 and 144 = 2 x 2 x 2 x 2 x 3 x 3

Therefore, product of common factors = 2 x 2 x 2 = 8. And hence, GCD = 8

The procedure described above requires finding prime factors of each numbers which is difficult to find using and combinational and sequential logic of digital electronics. So this method is not preferred to find GCD of given two numbers via digital electronic circuit.

### *2.1.2 Division Method*

For two given numbers, divide greater number by the lesser one; divide lesser by remainder, divide the first remainder by the new remainder, and so on till there is no remainder. The last divisor is the required GCD**.**

**EXAMPLE:** GCD of 12 and 15 in binary representation.

```
1100)1111(1
     -1100
    ------------
      0011)1100(0100
           -1100
          ------------
            0000
```

Hence, GCD = Last divisor = 0011

There is also a subtraction based version of the same method defined by Euclid (Given in Algorithm Section with minor changes according to the design of the circuit). This version based on subtraction is the convenient method to design a sequential circuit to generate GCD.

## 3. ALGORITHM

Given the two input positive numbers A and B, the algorithm will calculate the GCD of A and B using Euclid's method. There is a little modification in Euclid's method to match algorithm with the circuit designed. A', B', SUB and RESULT are numbers which is defined in the algorithm. At the end of the algorithm RESULT will show the GCD of A and B. The algorithm shows the operations performed by the circuit. At each assignment operation to A' or B', the value is stored in registers named REG A and REG B and retained until another assignment operation is performed.

1 ) Set A' as max{A,B} and B' by another number. If A = B then A'← A and B'← B.

2 ) SUB ← A' – B'. (SUB will be initialized)

3 ) Repeat steps 4 to 9 while SUB != 0.

4 ) SUB ← A' – B'.

5 ) Repeat steps 6 to 8 while !(SUB < B' ).

6 ) A' ← SUB.

7 ) SUB ← A' – B'.

8 ) B'← B'. (Value in Register B is retained.)

9) If SUB != 0
 Then A' ← B'
 B' ← SUB.

10 ) A' ← A'. (Value in Register A is retained.)

11 ) B' ← B'. (Value in Register B is retained.)

12 ) RESULT ← B'.

13 ) Set RESULT at OUTPUT.

Circuit based on the above algorithm will give constant output until the circuit is reset.

## 4. CIRCUIT COMPONENTS

The circuit for 4-bit data operation is designed using following digital electronic components.

1) 4-bit Register
2) 4-bit 2x1 MUX
3) 4-bit Comparator(A<B)
4) 4-nit Full Subtractor
5) Inverter
6) And gate

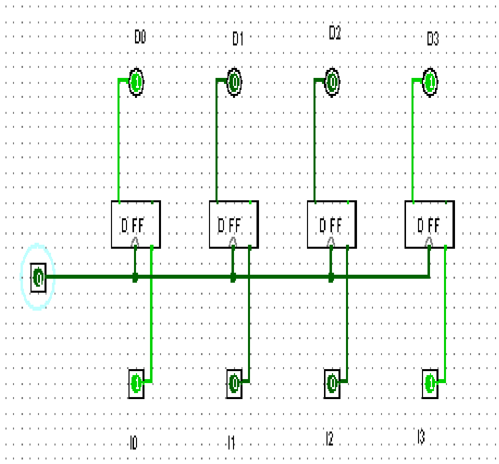


**Fig 1: 4-BIT Register**

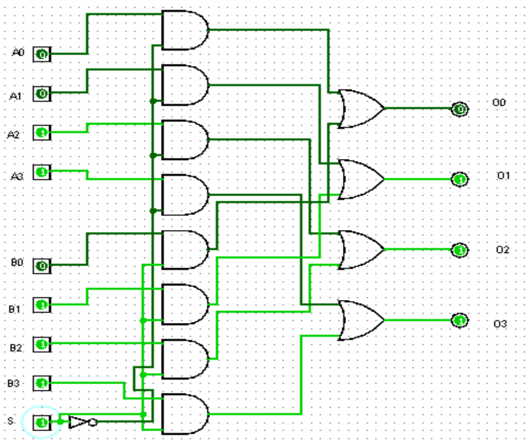


**Fig 2: 4-BIT 2x1 Multiplexer**

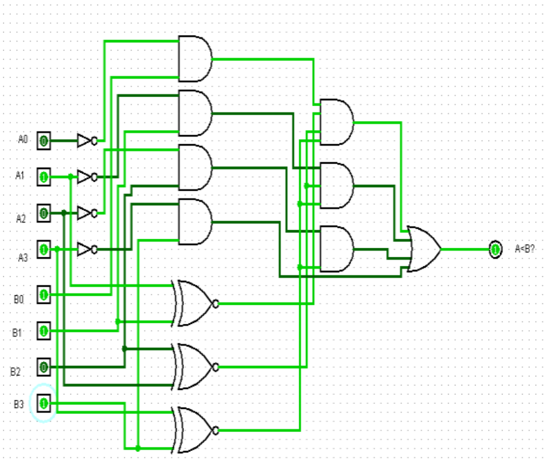


**Fig 3: 4-BIT Comparator (A<B?)**

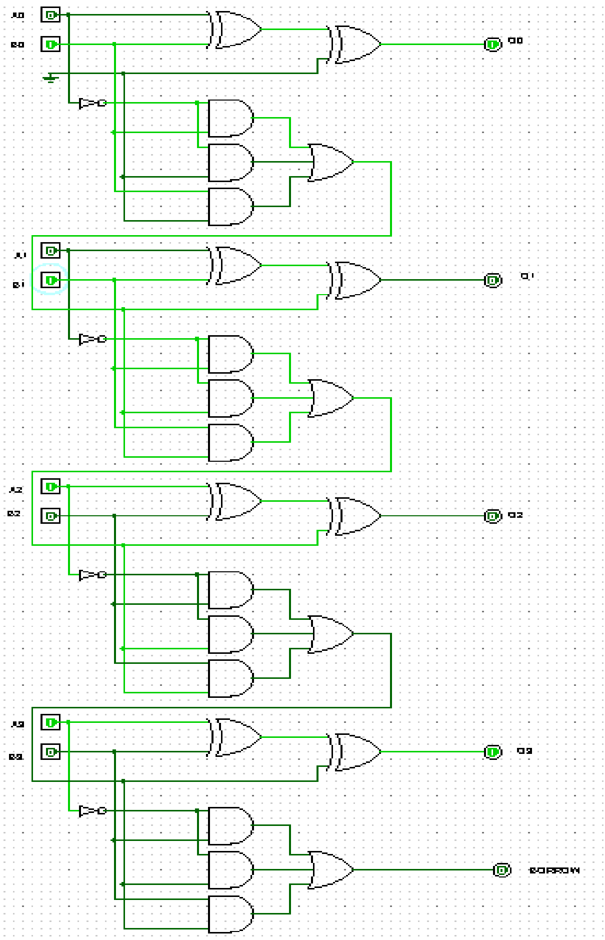

**Fig 4: 4-BIT Full Subtractor**

## 5. MAIN CIRCUIT SPECIFICATIONS

### *5.1.1 Inputs:*

1) 4-BITS binary number A.
2) 4-BITS binary number B.
3) RESET/GENERATE input to switch between input and output mode. ( bit 0: input mode, bit 1: Output mode )
4) Clock Pulse input.

### *5.1.2 Outputs:*

1) 4-BITS binary output GCD as RESULT.

## 6. WORKING

### *6.1.1 Input Mode*

Circuit switches to input mode when RESET/GENERATE bit is set to 0 (zero). In this mode two non-negative binary input numbers A and B can be given as input to the circuit for calculation of GCD of the same.

### *6.1.2 Output Mode*

Circuit switches to output mode when RESET/GENERATE bit is set to 1. In this mode two input binary numbers A and B will be taken in the circuit for calculation of GCD in the following way,

1. Circuit will perform subtraction of B from A until A-B < B or A-B=0

2. if A-B < B then A will be replaced by contents of B and B by contents of A-B and again step 1 will be carried out.

3. if A-B = 0 then B will be displayed at the output.

Checking of various conditions like whether A-B = 0, A<B etc is carried out by combinations of AND gates, INVERTERS and COMPARATOR. Accordingly control signals are sent to the multiplexers to replace content of register (REG A & REG B) with appropriate contents according to the algorithm mentioned.

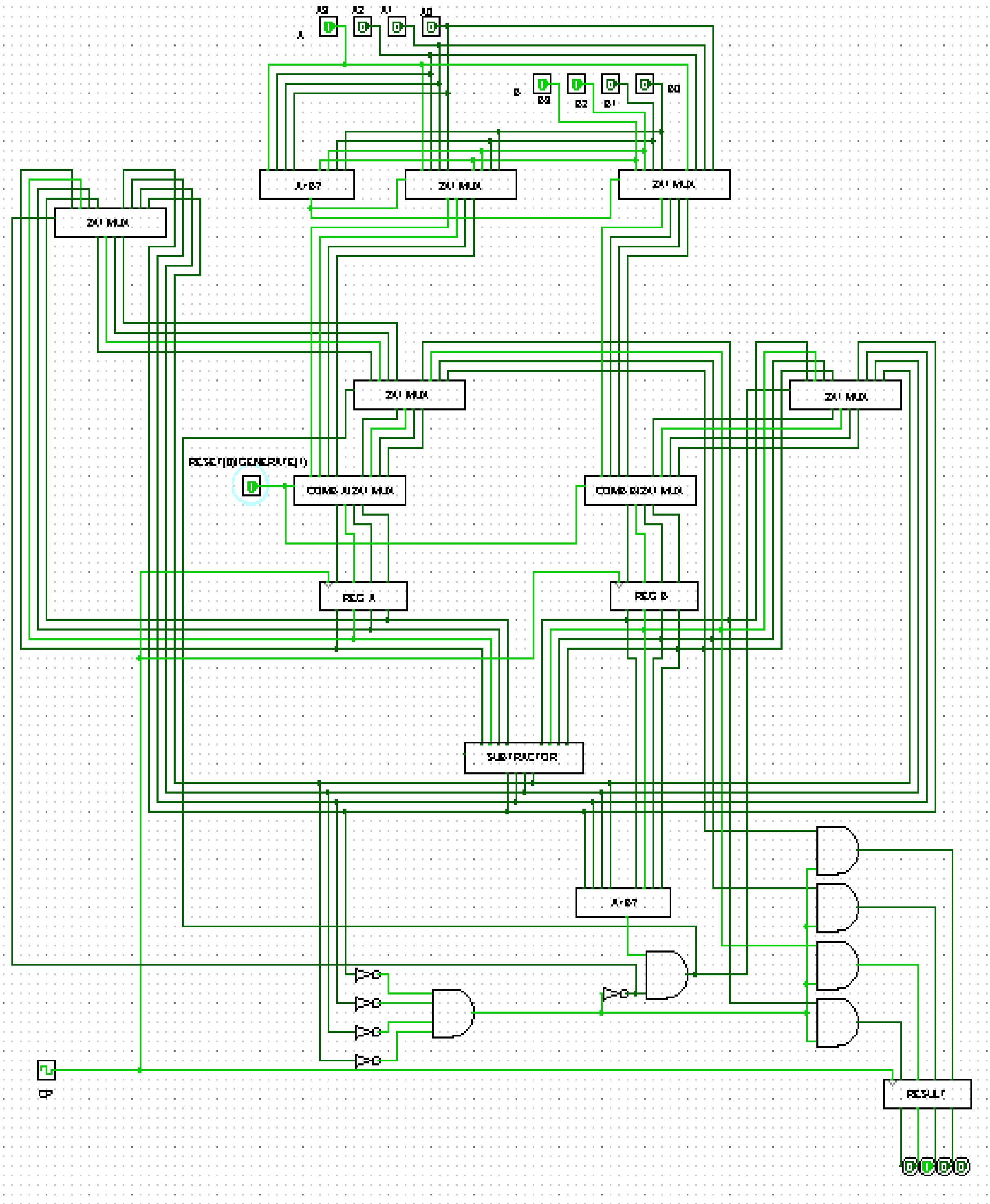

**Fig 5: Main GCD Calculator Circuit**

## 7. CONCLUSION

The digital circuit for calculating GCD of given two numbers can be designed using Euclid's algorithm with little bit modification in the same for convenience of design. It provides less complexity of design with minimum hardware requirement and takes less clock pulses to execute the overall procedure. So, circuit uses reasonable amount of hardware and time to calculate GCD of two numbers. This implementation can work with general purpose Arithmetical and Logical unit as it uses subtraction which can be design using adder by 2's complement method. Also, the circuit can be developed for any number of data bits by extending components accordingly.

## 8. REFERENCES

[1] M. Morris Mano "Digital Logic and Computer Design", Published by Pearson Education, Inc. and Dorling Kindersley Publishing Inc., Printed in India by Gopsons Papers Ltd, 2011.

[2] R. P. Brent, H. T. Kung, "A Systolic Algorithm for GCD Computation" Proc. 7th IEEE Symp. On Comp. Arith., pp 118-125, 1985.

[3] M. J. Foster and H. T. Kung, "The Design of Special Purpose VLSI Chips" IEEE Computer , Vol.13, pp.26-40, Jan. 1980.

[4] B. Vallee "Dynamical analysis of a class of Euclidean Algorithms." The Computer Science, 297(1-3):447-486, 2003.

[5] R. M. Corless, S. M. Watt, and L. Zhi QR factoring to compute the GCD of univariate approximate polynomials", 2003

[6] P. Emeliyanenko. "A complete modular resultant algorithm targeted for realization on graphics hardware".In PASCO '10, pages 35–43, New York, NY, USA,2010. ACM.

[7] P. Emeliyanenko. "Accelerating Symbolic Computations on NVIDIA Fermi". In GTC '10, 2010. Poster presentation.

[8] P. Emeliyanenko. "Modular Resultant Algorithm for Graphics Processors". In ICA3PP '10, pages 427– 440, Berlin, Heidelberg, 2010. Springer-Verlag.

[9] Haroon Altarawneh, A Comparison of Several Greatest Common Divisor (GCD) Algorithms,International Journal of Computer Applications (0975 ・ 8887), Volume 26・ No.5, July2011.

[10] T. Jebelean,Comparing Several GCD Algorithms " Che Wun Chiou, Fu Hua Chou , Yun-Chi Yeh , Speeding up Euclid's GCD algorithm with no magnitude comparisons・ , International Journal of Information and Computer Security table of contents archive Volume 4 Issue 1, February 2010.

[11] Jonanthan Sorenson, Two fast GCD Algorithms, Journal Of Algorithms16,110-114(1994).

[12] Jeffrey Shallit and Jonathan Sorenson, 鄭nalysis Of Left-Shift GCD Binary Algorithm・, J.Symbolic Computation (1994) 17, 473-486.