Design and Implementation of A Divider Using Non-restoring Division Algorithm

Tin Nilar Maung
University of Computer Studies (Mandalay)
tinnilarmaung@gmail.com

Abstract

A complex system, such as a computer system, is seen to be composed with several levels of abstractions. In general, a computer system is seen to be composed of three levels; the gate level, the register level and the processor level. The function of the register level or register-transfer level is to accomplish a single instruction in the Instruction Set Architecture (ISA).The register level plays an important role to define the proper sequence of micro-operations for every ISA instructions. The major objective of this paper is intended to present design and implementation of a divider. This system is based on non-restoring algorithm for unsigned integers. A register level implementation of unsigned division based on the non-restoring algorithm is carried out to highlight the important role of the register level implementation. Parallel Interfacing and Turbo C++ Programming language is used to implement this paper.

1. Introduction

In general, a computer system is seen to be composed of three levels; the gate level, the register level and the processor level. The gate level is primarily responsible to construct the basic data storage and data processing units. The processor level makes up of the entire computer system including the basic input/output devices.

In this system, register level components are used to design and to implement the algorithm. The function of the register level or register-transfer level is to accomplish a single instruction in the Instruction Set Architecture (ISA) seen at the processor level by issuing right sequence of micro-operation to the gate level underneath. To reduce the number of components within a CPU. In computer arithmetic, all the arithmetic operations such as add, subtract, multiply and divide operations are implemented only with adder. In this system, design and implementation based on non-restoring division algorithm is separated in datapath and control unit. The non-restoring algorithm is founded upon the repeated add/subtract and shift method.

This paper is organized with eight sections. Section one deals with Introduction. Section two describes Relate work. Section three is Non-restoring Algorithm. Section four is Overview and Components of the System. Section five presents System Design and Implementation. Comparison between Restoring and Non-restoring includes in section six and next section is Conclusion. Final section is references.

2. Related Work

A new nonrestoring square root algorithm that is very efficient to implement is presented. Yamin Li and Wanming Chu illustrate two VLSI implementation of the new algorithm. One is a fully pipelined high-performance implementation that can accept a new square root instruction each clock cycle with each pipeline stage requiring a minium number of gate counts. The other is a low-cost implementation that uses only a single adder/subtractor for iterative operation [1]. R.SESHASAYANAN and Dr S.K.SRIVATSA propose to analyze a Very Long Instruction Word (VLIW) processor which exploits Instruction Level Parallelism (ILP) to increase the speed of execution. Simulation analysis is done on a prototype VLIW processor under GALS multiprocessor environment to reduce power with same performance. An architecture, which is a hybrid between a fully synchronous and an Asynchronous processor, termed as Globally Asynchronous and Locally Synchronous processor (GALS) is being incorporated in this design [2].

3. Non-restoring Algorithm

Figure 1, presents a non-restoring division algorithm for unsigned integers. The divisor V and dividend D are unsigned integers. The divisor *V* and quotient *Q* are *n* bits long while the dividend *V* is up to 2*n*-1 bits long, which is the maximum length of the product of two *n*-bit integers. The flip flop S is appended to the accumulator A to record the sign of the result of an addition or subtraction and to determine the quotient bit. Each new quotient bit is placed in *Q* [0], and the final values of the quotient *Q* and the remainder *R* are in the Q and A registers, respectively [3].

```
NRdivider            (in:INBUS;out:OUTBUS);
                     register S, A[n-1:0], M[n-1:0]
                     Q[n-1:0], COUNT[[$\log_2 n$]:0];
           bus INBUS[n-1:0], OUTBUS[n-1:0];
BEGIN:               COUNT := 0, S := 0,
INPUT:               A := INBUS;
                     Q := INBUS;
                     M := INBUS;
SUBTRACT:            S.A := S.A – M;
TEST:                if S = 0 then
                       begin Q[0] := 1;
                       if COUNT = n-1 then
                 go to CORRECTION; else
               begin COUNT:= COUNT +1,
                    S.A.Q[n-1:1] := A.Q; end
           S.A:=S.A-M; then go to TEST; end
                 else{if S=1}
                    begin Q[0] := 0;
                    if COUNT=n-1
               then go to CORRECTION; else
                    beginCOUNT:=COUNT+1,
                      S.A.Q[n-1:1]:= A.Q; end
                     S.A := S.A + M; go to TEST;
                     end
CORRECTION:          if S =1then S.A:= S.A + M;
OUTPUT:              OUTBUS := Q;
                     OUTBUS := A;
End NRdivider;
```

Figure 1. Non-restoring division algorithm for unsigned integers

4. Overview and Components of the System

4.1. Overview of the System

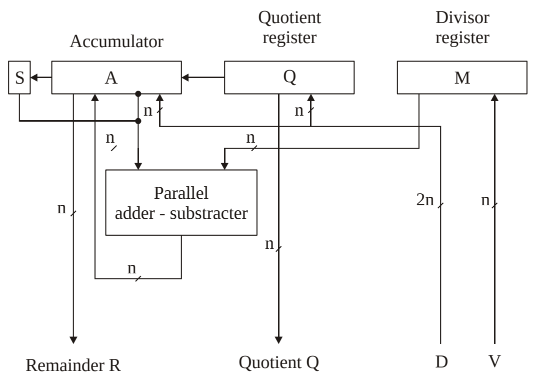

Figure 2. Overview of the system

This system is intended for unsigned integers. There are three registers (74LS273) dividend, divisor and quotient values, namely register A, Q and M as shown in Figure 2. At the start of the algorithm, register A and Q together holds the dividend and the register M holds the divisor value. At the end of the algorithm, register A and Q holds the remainder and quotient.

4.2. Components of the System

The electronic device applied in this system include multiplexer, register, parallel adder and isolation.

4.2.1. Multiplexer. A multiplexer is a device that allows digital information from several sources to be routed onto a single line for transmission over that line to a common destination. The basic multiplexer has several data-input lines and a single output line. Multiplexers are also known as data selectors. In this system, multiplexer 74LS157 IC is applied. It contains, there are two data inputs, a single bit data-select input, enable and one data output. To operate the IC, the enable, $\overline{EN}$, must be grounded. If the data-select input, S, is 0, the input0 data values are output at the output pins. If the data-select input, S, is 1, the input1 data values are output at the output pins [4].

4.2.2. Register. Register is to store several bits of data from parallel lines simultaneously, in a group of flip-flops. In this system 74LS273 IC, which contains octal positive edge triggered D flip-flops, is utilized to hold the operand values. Each flip-flop contains D input and Q output. There contains clock input, master reset, eight data input lines and eight data output lines [5].

4.2.3. Parallel Adder. Parallel adder is using two's complement arithmetic. The 4-bit parallel adder 74LS283IC is implemented with four full-adder stages. The LSBs ($a_1$and b_1) in each number being added go into the right-most full-adder; the higher order bits are applied to the successively higher-order adders. This type of adder is known as ripple carry adder. The carry output of each adder is connected to the carry input of next higher-order adder. To generate a two's completement binary numbers, invert every o to 1 and every 1 to 0, then add one to the result [6].

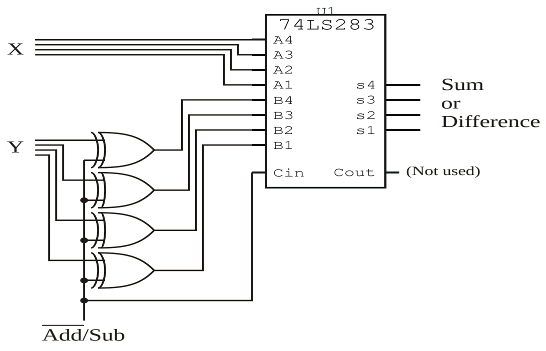

Figure 3. Adder/subtractor circuit

All four of the basic arithmetic functions involving positive and negative numbers. Subtraction is done by adding the two twos complement numbers. Thus, the same digital circuitry can be used for additions and subtractions. In adder/subtractor circuit, IC for XOR gates is 74LS86, which contains the quad XOR gates. In Figure 3, 74LS283 IC performs a regular binary addition. If the complementing switch is up, the number on the B inputs will be subtracted from the number on the A inputs. If it is logic 0, the sum is taken. The C_{out} of the MSB is ignored. X value is applied to A input and Y value is applied to B input [6].

4.2.4. Isolation. Isolating the computer interfaces from the external circuits is required to prevent the computer damages due to accidental faults. But, the signaling must be accomplished even though the computer and external circuits are electrically isolated. Amongst many electronic devices which operate optically, opto-coupler's are used to electrically isolate the computer and external circuits. In Figure 4, if the PC asserts one of its output pins. It's diode forward bias and emits the infrared light, which stimulates the base of the transistor. The transistor is ON and the collector and emitter are shorted together, giving logic 0 to the external circuit. If the PC's output pin is logic 0, there is no emitted light of the LED and so, the transistor is OFF, giving logic 1 to the external circuit. Therefore, the PC can be electrically isolated from the external circuits even though the signal transfer is accomplished [7].

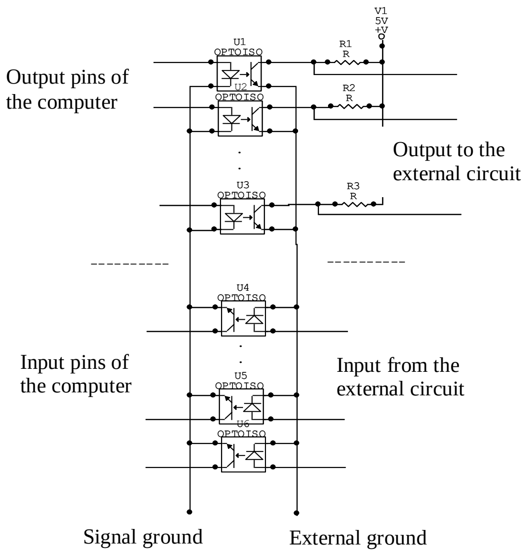
Output pins of the computer
Output to the external circuit
Input pins of the computer
Input from the external circuit
Signal ground
External ground
V1 5V +V
U1 OPTOISO
U2 OPTOISO
U3 OPTOISO
U4 OPTOISO
U5 OPTOISO
U6 OPTOISO
R1 R
R2 R
R3 R

Figure 4. Isolating the PC from the external circuits

5. Design and Implementation

In this system, the datapath unit is implemented by electronic circuit and control unit is implemented by computer. In datapath portion includes 4 types of subsystem, data routing, data storage, data processing and computer interface. In control portion, necessary control signals are transferred from the computer system via parallel port.

Data routing of the system is used to three multiplexers 74LS157 IC. In this system, there are three sources to update the value of S and A; the adder/subtractor output, the left shifted value of A and the left four bits of the external data for dividend. Therefore, two 2-input four bit multiplexers have to be fitted at the input of register A as shown in Figure 5. The computer system has to control the multiplexer select bits S1 and S2. For the register Q, there are two sources to update the content of register Q; the external data for right four bits of dividend and the left shifted of Q and Quotient bit combination. Therefore, another multiplexer has to be fitted at the input of register Q as shown in Figure 5. The computer system has to control the data select bit S3. To accomplish the shift operation, Q(1:0) and Quotient bits have to be inserted into Q(2:1).

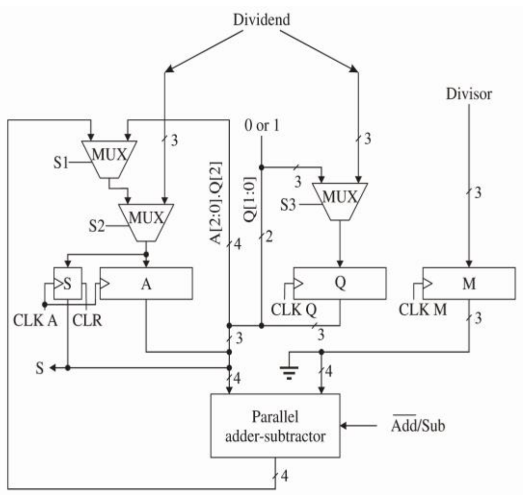

Figure 5. Design of the system

Data Storage of the System is used to four registers (74LS273IC) and to hold the divisor and dividend; register S is used to flip-flop, registers A and Q for dividend and register M for divisor as shown in Figure 5. At the initialization stage, flip-flop S must be cleared and so the CLR input of the flip-flop S must be controlled by the computer system. To strobe in the corresponding values into the registers CLK1, CLK2 and CLK 3 of the registers A, Q and M are also controlled by the computer system.

The data processing unit of the system is the adder/subtractor using two's complement arithmetic. According to the non-restoring algorithm, the five bit operation must be performed in adder/subtractor. To subtract register M from S and A as shown in Figure

6, $\overline{\text{Add}}$/Sub must be controlled. When this control bit is 0, addition is performed and this control bit is 1, subtraction is performed. This control bit is also controlled by the computer system.

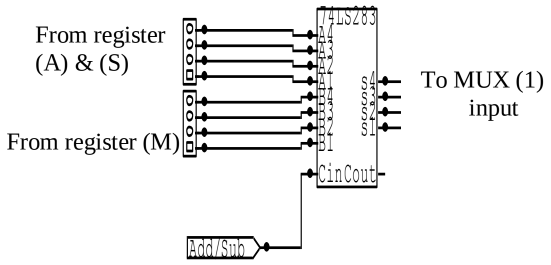

Figure 6. Data processing of the system

The sign bit S of the register A is applied into the computer by means of ERR status pin. Three multiplexer select bits, clear input of register A, three clock inputs of three registers and $\overline{\text{Add}}$/Sub pins are respectively controlled by the data bits of the parallel port. The 0or1 control pins is handled by the INI status bit of the control register as output. Figure 7 shows photo of the complete circuit. More detailed diagram of the complete circuit is shown in Appendix.

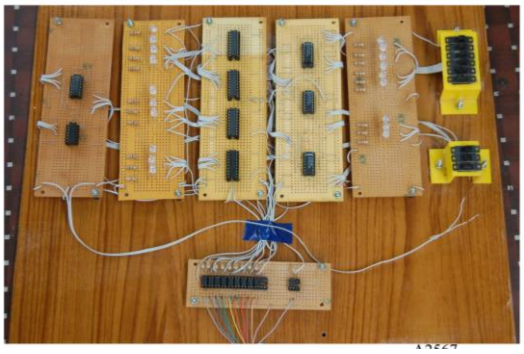

Figure 7. Photo of the complete circuit

6. Comparison between Restoring and Non-restoring Algorithm

The system is implemented for hardware division algorithm based on non-restoring. Division is implemented by add, subtract and shift method. The result of subtraction is positive, restoring and non-restoring algorithm is the same function. In process, the result of subtraction is negative, in restoring algorithm, the dividend needs be first restored by adding the divisor back before left shift dividend registers and the next subtraction. It takes three clock cycles per iteration. In non-restoring algorithm, the dividend needs be restored by adding the divisor after left shift dividend registers and the next subtraction. It takes two clock cycles per iteration. Comparison result for using restoring and non-restoring is shown in Table 1.

Table 1. Restoring and non-restoring for three iterations.

	No. of Cycle			No. of 4 bit parallel adder/subtractor		
Data Bits	3	4	8	3	4	8
Restoring	9	12	24	1	2	3
Non-restoring	6	8	16	1	2	3

7. Conclusion

This system is based on micro-architecture level. It is a register level design and implementation of unsigned division based on the non-restoring algorithm that is carried out to highlight the important role of the register level implementation. The input enters from circuit manually and is intended for unsigned numbers. The non-restoring algorithm is founded upon the repeated add/subtract and shift method giving the quotient and remainder of the two unsigned numbers of dividend and divisor for the output. Control unit is implemented by the computer. It can be replaced by hardwired control and PIC control for future. The advantages of this system is more faster the processing time than using the restoring algorithm. Transistor Transistor Logic (TTL) IC family is only applied in this system without using types of digital IC families. Complementary Metal Oxide Semiconductor CMOS IC family can be used in further extension of this system.

8. References

[1] Yamin Li and Wanming Chu. A New Non-Restoring Square Root Alogrithm and Its VLSI Implementations.

[2] R.SESHASAYANAN and Dr S.K.SRIVATSA. Implementation of Novel Pipeline VLIW Archittecture In FPGA.

[3] John P. Hayes , *Computer Architecture and Organization,* Third Edition, McGraw-Hill International, Inc.

[4] Thomas L. Floyd, *Digital Fundamental,* Eight Edition, The Prentice-Hall International, Inc, 1994.

[5] Andrew S. Tanenbaum, *Structure Computer Organization*, Fourth Edition.

[6] David A, Patterson John. Hennessy L, *Computer Organization And Design,* Third Edition, The Hardware Software Interface.

[7] Thomas L. Floyd, *Electronic Devices*, Fifth Edition Prentice-Hall International, Inc.

APPENDIX

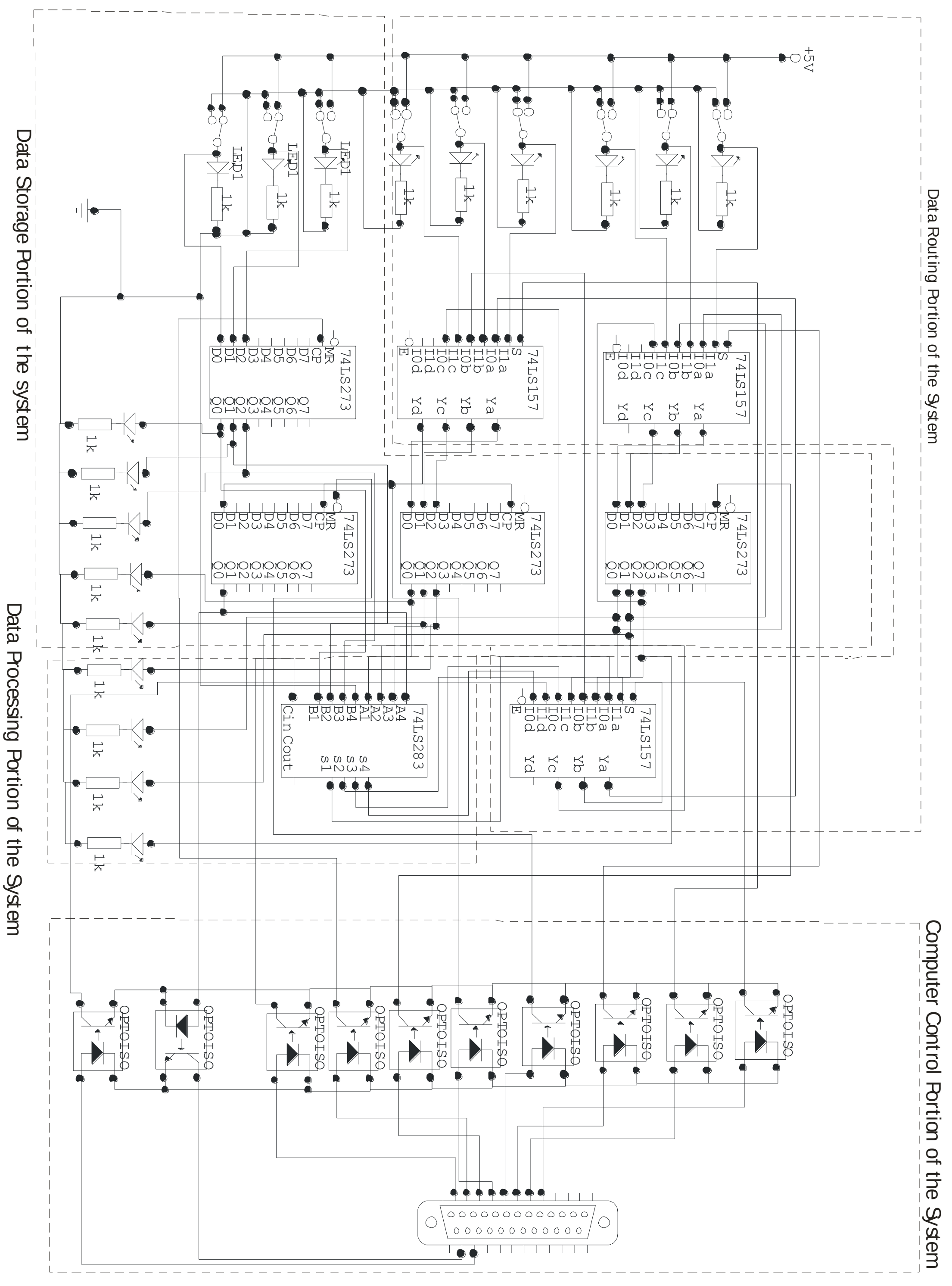
Data Storage Portion of the system
Data Routing Portion of the System
Data Processing Portion of the System
Computer Control Portion of the System
+5V
LED1
1k
74LS273
74LS157
74LS283
OPTOISO

Figure A. Complete circuit diagram of the system