Design of A Multiplier for Two's Complement Numbers Using Robertson's Method

Ar Kar Baung
University of Computer Studies (Mandalay)
arkarbaung20@gmail.com

Abstract

To design a CPU, its instruction set architecture is first developed, including its instruction set and its internal registers. A CPU contains three primary sections: the register section, the ALU, and the control unit. The micro-operations needed to fetch, decode, and execute every instruction in its instruction set are created together with the Register Transfer Level (RTL) specifications. This paper proposes a two's complement multiplier based on the Robertson's add and shift multiplication algorithm. The registers and execution units needed to carry out the micro-operations of the system are designed and constructed. The computer system generates the RTL code which specifies the functions to be performed by the registers and execution units. Based on simple add and shift operations, with the small amount of additional logic, multiplication of signed numbers can be accomplished by the Robertson's algorithm.

1. Introduction

This system is aimed to design a multiplier for two's complement numbers using Robertson's method. In this system, there are divided two main portions. They are datapath and control portion. In datapath portion, to construct Robertson's multiplier system, there contains

- Registers
- Parallel adder
- Xor gate
- Multiplier

In control portion, the sequence of control signals to accomplish the required sequence of micro-operations for the datapath is generated by Turbo C++ programming language via parallel interfacing. Applying the proper control signals in proper sequence gets the product of two two's complement numbers. The system for this paper has based on microarchitectural level. This algorithm is founded upon the repeated add/ subtract and shift method.

This paper is organized with sixth sections. Section one deals with Introduction of the System. Section two explains Background Theory of the System. Section three presents System Design and Implementation. System Evaluation includes in section four and next section is Conclusion. The final section is references.

2. Background Theory

2.1. Robertson's Algorithm

This algorithm presents Robertson's algorithm designed for the circuit of Figure 1 with two's com - plement numbers. When multiplying by the sign bit, perform subtraction rather than addition in the final step if a minus sign x_{n-1} =1 is encountered. This observation is the basic of a twos complement algorithm developed by James E. Robertson, which has been widely used in computer design [1].

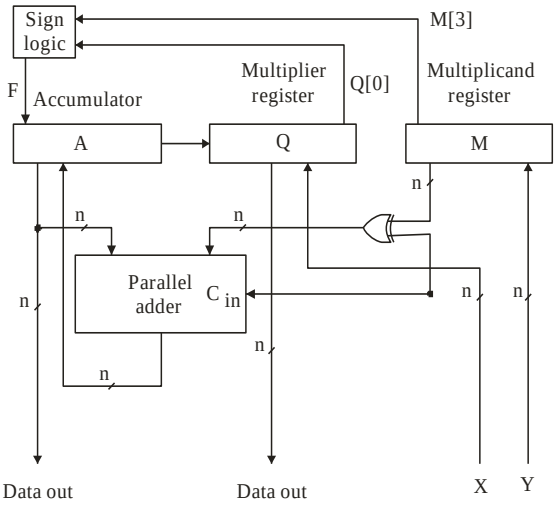

Figure 1. The datapath of the twos complement multiplier

```
2Cmultiplier    (in: INBUS; out: OUTBUS);
                register  A[3:0],M[3:0],Q[3:0],
                COUNT[1:0], F;
                busINBUS[3:0],
                OUTBUS[3:0];
BEGIN:          A := 0, COUNT := 0, F := 0;
INPUT:          M := INBUS;
                Q := INBUS;
```

```
ADD:        A[3:0] := A[3:0] + M[3:0] × Q[0];
            F := (M[3] and Q[0]) or F;
RSHIFT:     A[3]:=F;A[2:0].Q:=A.Q[3:1],
            COUNT := COUNT + 1;
TEST:       if COUNT ≠ 3 then go to ADD;
SUBTRACT:   A[3:0] := A[3:0] – M[3:0] × Q[0];
            A[3]:=A[3],A[2:0].Q:= A.Q[3:1];
OUTPUT:     OUTBUS := Q;
            OUTBUS := A;
End 2Cmultiplier;
```

2.2. Components of the System

In this section, the electronic devices are used in the system. These are multiplexers, registers and parallel adders.

2.2.1. 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. It also has data-select inputs, which permit digital data on any one of the inputs to be switched to the output line. It also has data-select inputs, which permit digital data on any one of the inputs to be switched to the output line. Multiplexers are also known as data selectors.

Multiplexer IC, 74LS157, which is a 2-way (input) 4-bit multiplexer is applied in this system.

The 74LS157 IC contains

- Two data inputs (each is 4-bit in length)
- A single bit data-select input
- Enable (active LOW)
- One data output (4-bit in length)

To operate the IC, the enable (active LOW), $\overline{EN}$, 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 [2].

2.2.2. Register. Register (74LS273IC) contains clock input, master reset, eight data input lines and eight data output lines. If there is logic 0 on master reset , the flip-flops are cleared and it must be at logic 1 to store the data values in the flip-flops. At positive edge triggered, the value of D input is stored into the flip-flop at the positive edge of the clock. All D flip-flops are synchronously tied together [3].

2.2.3. Parallel Adder. Parallel adder is using two's complement arithmetic. The 4-bit parallel adder (74LS283) IC 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, with the MSBs ($a_4$and b_4) in each number being applied to the left-most full-adder. The carry output of each adder is connected to the carry input of next higher-order adder. These are called internal carries. To generate a two's complement binary numbers, invert every o to 1 and every 1 to 0, then add one to the result. For example, to do Negate 2_{ten} the following process is needed. 2_{ten}=0000 0000 0000 0010$_{two.}$ Negating this number by inverting the bits and adding one [4].

$$\begin{array}{rl} & 1111\ 1111\ 1111\ 1101_{two} \\ + & \qquad\qquad\qquad 1_{two} \\ \hline = & 1111\ 1111\ 1111\ 1110_{two} \\ = & -2_{ten} \end{array}$$

2.2.4. The Parallel Port I/O Addresses and its Register. The 25 pins of the parallel port can be classified as the data, status and control. The control signal controls circuits which are fetched from the system bus and stored in the data register. Therefore, the contents of the data registers are actually outputs to the parallel port. The status of the parallel port, such as busy, error, paper out, etc., are fed into the system bus via a status register. The control signals, some inputs to the system bus and some outputs from the system bus to the parallel port are latched in a control register. The address of data, status and control registers are 0378h, 0379h and 037Ah [5].

3. System Design and Implementation

3.1. System Design

This system is only intended to two's complement numbers. There are three registers to hold the multiplicand, multiplier and product values, namely register A, Q and M as shown in Figure 2. At the start of the algorithm, register A and M holds the multiplicand and multiplier respectively and at the end of the algorithm, register A and Q together hold the product values. This system is composed of three portions .These are data routine, data storage and data processing.

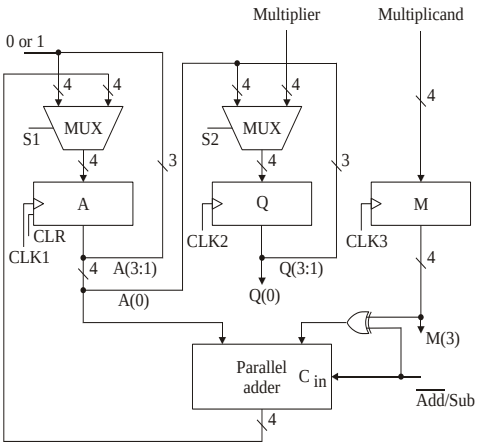

Figure 2. System design

3.1.1. Data Routine. In this portion, two multiplexer (74LS157) ICs are used for Data Routine. According to the Robertson algorithm, there are two sources to update the value of A; the adder/ subtractor output and the right shifted value of A. Therefore, a 2-input four bit multiplexer has to be fitted at the input of register A. The computer system has to control the multiplexer select bit S1. According to the algorithm, the vacated bit of the shifted value has to be filled with 0 or 1 according to the sign bit of the original content value of A. Therefore, the computer system has to control this bit. To accomplish shift operation, this bit and A(3:1) bits must be inserted in A(3:0). For the register Q, there are two sources to update the content of register Q; the external data for the multiplier value and the right shifted bits of A and Q combination. Therefore, another multiplexer has to be fitted at the input of Q. The computer system has to control the select bit S2 of the multiplexer. To accomplish shift operation, A (0) and Q (3:1) bits have to be inserted into Q (3:0) bits.

3.1.2. Data Storage. Three register (74LS273) ICs are used for Data Storage; multiplier, multiplicand and product bits. At the initialization stage, the register A must be cleared and so the CLR input of the register A must be controlled by the computer system. To strobe in the corresponding values into the registers, the CLK1, CLK2 and CLK3 of the registers A, Q and M are also controlled by the computer system.

3.1.3. Data Processing. The (74LS283) IC is used for Data Processing. To subtract register M from A, the content values of register M is XORed with a control bit, $\overline{\text{Add}}$/Sub as shown in Figure 3. When this control bit is 0, addition is performed and when this control bit is 1, subtraction is performed. This control bit is also controlled by the computer system.

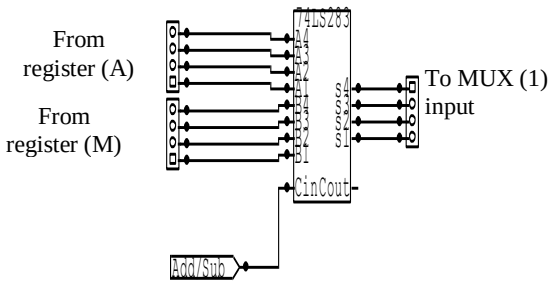

Figure 3. Logic diagram of data processing portion

3.1.4. Two's Complement Adder/Subtractor. When adding numbers in the twos complement form, simply perform a regular binary addition to get the result. When subtracting numbers in the twos complement form, convert the number being subtracted to a negative twos complement number and perform a regular binary addition.

Adding 18 + (-9):

twos complement of 18=0001 0010
+ twos complement of –9=1111 0111
0000 1001 = + 9_{10} answer

The 74LS283IC 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 as shown in Figure 4. X value is applied to A input and Y value is applied to B input. Therefore, the X-Y operation can be accomplished.

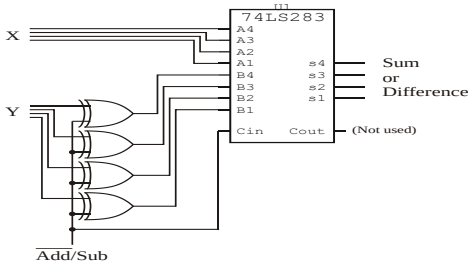

Figure 4. Adder/subtractor circuit

3.2. The Parallel Port's Register Utilization of the System

In this system, registers Q(0) and register M(3) are applied into the computer by means of ON/OFF and ERR status pins as shown in Figure 5. The computer system monitors these bits according to the algorithm. Two multiplexer select bits, clear input of register A, three clock inputs of three registers and $\overline{\text{Add}}$/Sub, 0orl pins are controlled by data bits of parallel port.

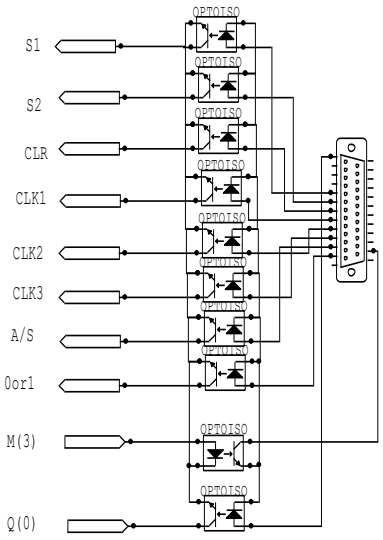

Figure 5. Computer control portion of the system

3.3. System Implementation

Figure 6 is top view of the complete circuit. More detailed diagram of the complete circuit is

shown in Appendix. In this circuit includes datapath and control. A datapath consists of data-processing as well as data storage units. In data-processing unit, has used 4-bit parallel adder/subtractor (74LS283IC), to perform the necessary add and subtract operations. The data-storage units used to three registers (74LS273IC), to store the Accumulator, Multiplier and Multiplicand. To select the necessary operands, two 2-input 4-bit multiplexer IC 74LS157 is applied. In order to be able to control the operations of the system, the computer system has to be monitored the sign bit of the multiplier and multiplicand operands. At the last stage of add-shift operations, the sign bit of the multiplier is at the least significant bit position of register Q and the sign bit of the multiplicand is at the most significant bit position of register M.

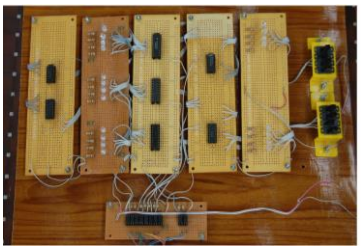

Figure 6. Photo of the complete circuit

3.4. Flowchart of the System

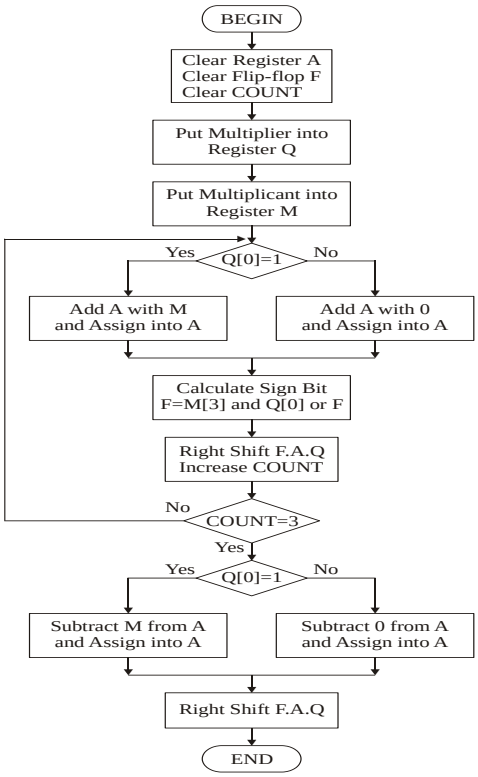

Figure 7. Flowchart of the system

This Figure 7 shows the flowchart how to multiply two two's complement numbers using Robertson's Algorithm. In this multiplier M and multiplicand Q is accepted as input data in registers.

The sign bit of the multiplier is at the least significant bit position of register Q[0] and the sign bit of the multiplicand is at the most significant bit position of register M[3] are applied into the computer system. The computer system monitors these bits and determine the necessary operations according to the Robertson's algorithm. When the algorithm is accomplished, the product of the two two's complement numbers are required.

4. Evaluation of the System

To design and construct a complete logic circuit that can generate the product of two signed numbers is very complicate and heuristic. By means of this proposed system, even though there contains sequence of micro-operations, multiplication of two signed numbers can be accomplished with a small amount of logic circuit using only one adder. When developing this system, using Transistor-transistor logic (TTL) ICs without the other types of digital IC families makes the system simple and unique.

5. Conclusion

In this system, the register transfer level implementation of multiplication instruction for two's complement operands is carried out, specifying the micro-operations based on the Robertson's add and shit multiplication algorithm.To get product of the two two's numbers, use only with adder. Datapath portion is designed with electronic devices and control signal which need to design are generated by the computer system. It can be replaced by hardware control and PIC control. This system, Transistor-Transistor-Logic (TTL) ICs are only used. Complementary Metal Oxide Semiconductor (CMOS) IC family can also be used in further extensions of the system.

6. References

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

[2] Thomas L. Floyd, *Digital Fandamentals*, Eight Edition, 1994 by The Prentice-Hall International, Inc.

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

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

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

APPENDIX

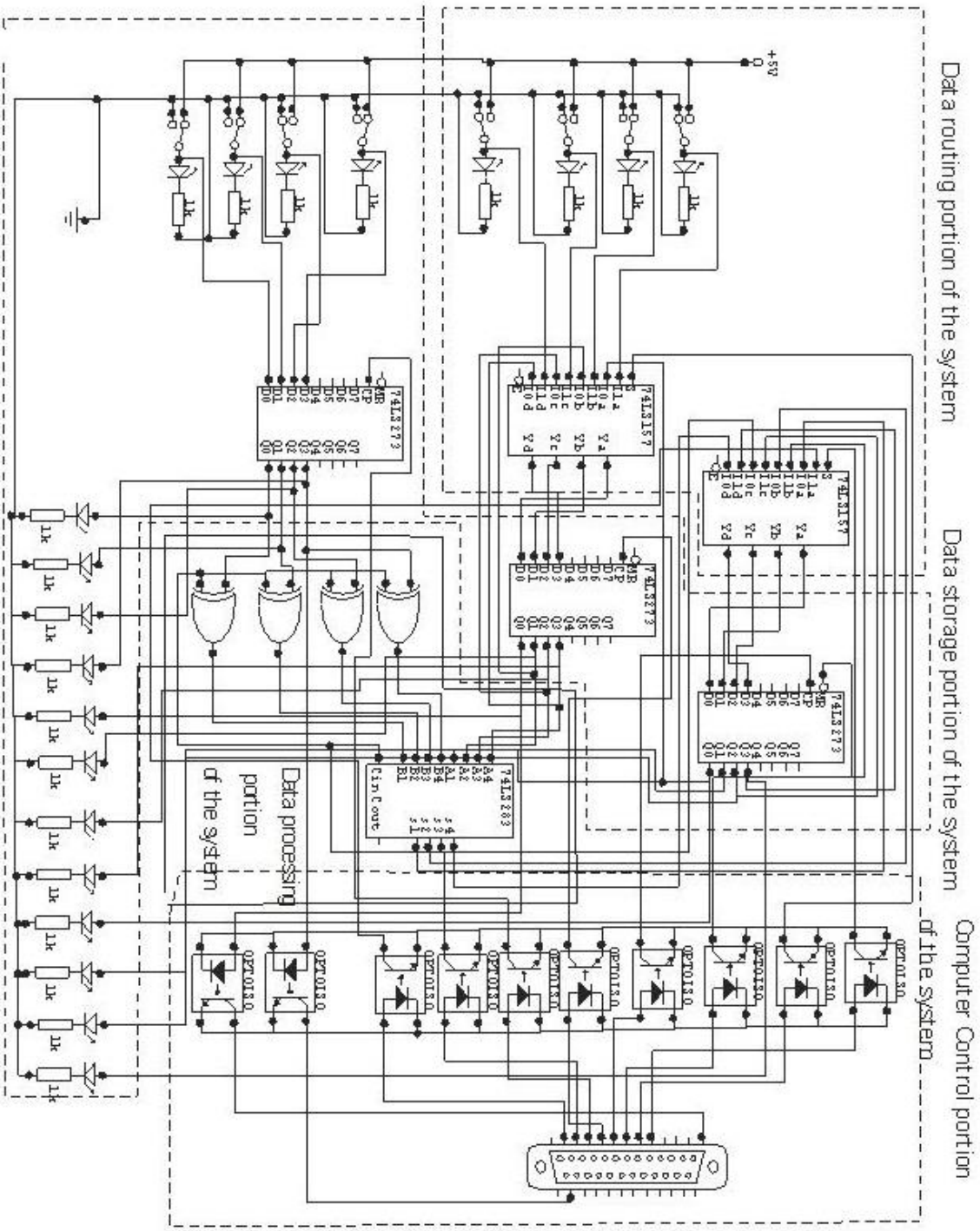

Figure A. Complete circuit diagram of the system