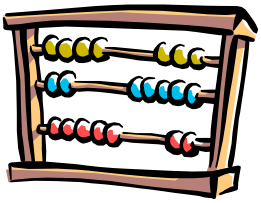
ECE 645

PROJECT SPECIFICATION

Design A
Microprocessor Functional Unit
Able To Perform Multiplication & Division



Professor: **KRIS GAJ**

Students: **LUU PHAM**
VARUN AGGARWAL

GMU – Mar. 2002

CONTENTS

1. ABSTRACT

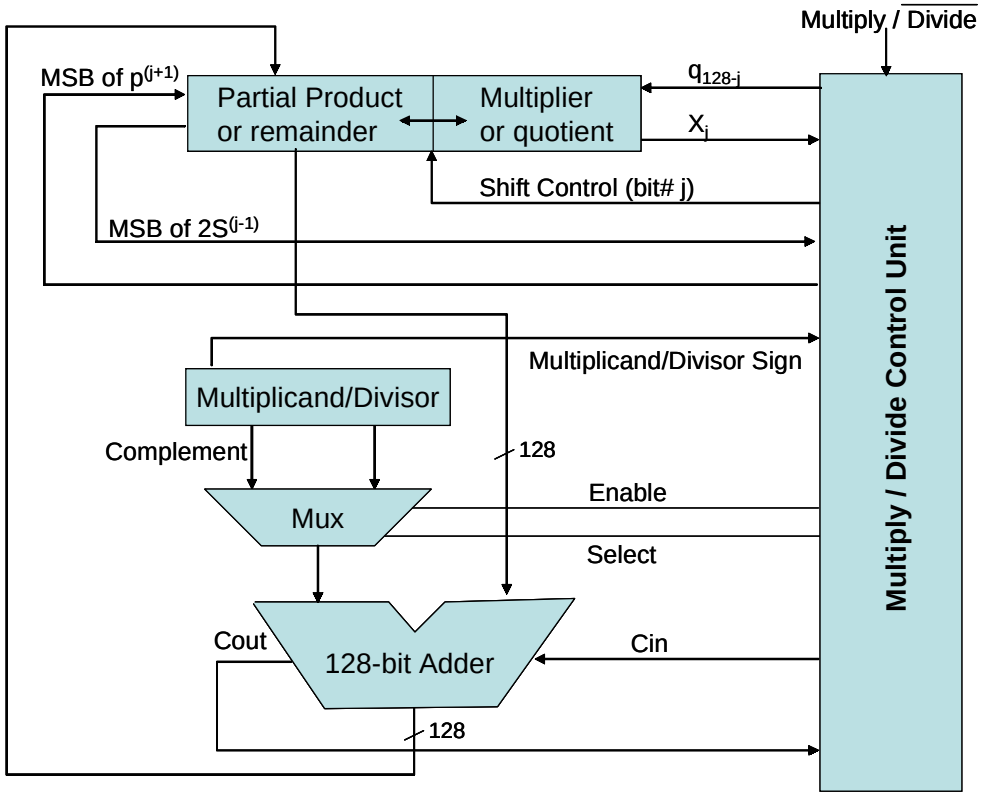
The purpose of this project is to design an arithmetic functional unit that is able to perform fast multiplication of 64-bit signed numbers and division of 128-bit signed numbers with the aspects of minimum latency and maximum throughput.

Throughout the design, some fast multiplication and division schemes, such as high-radix multipliers/dividers, tree and array multipliers/dividers, will be addressed and analyzed. An appropriate multiplier and divider method then will be selected and discussed in detail.

The designed functional unit is detailed in both hardware and software aspects including implementing and simulating using VHDL simulation. A detailed circuit diagram in basic-gate level will be produced; a VHDL code describing the behavior of the functional unit will be written and used to simulate and verify the correctness for the design.

2. INTRODUCTION

Current research shows that multiplication and division with minimum latency and maximum throughput can be designed and implemented in various schemes parallel, sequential, and array multiplier/divider for instance. For convenience in performing both multiplication and division the tree and array scheme is selected for the design. The block diagram of the functional unit is described as follows.



3. HARDWARE DESIGN SPECIFICATION

a. Design assumptions

The design is based on the following assumptions:

i- Components available:
The design is built on common logic gates available in market, such as the family of Quad 2-Input ports, Triple 3-Input ports, Dual 4-Input ports, and Hex Invert.

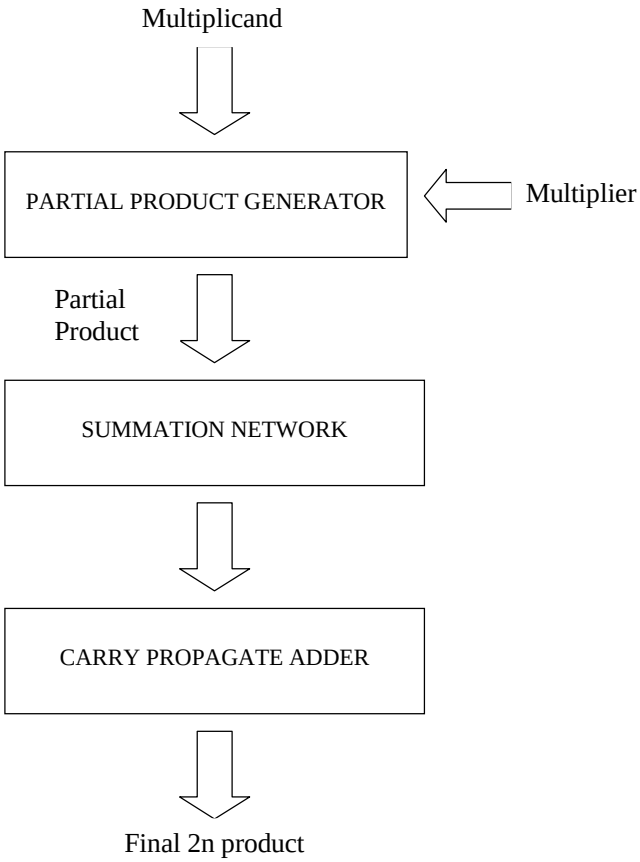
ii-. Gate delay and relative area of gate:
A one gate-delay unit and one gate-area unit, which equivalent to a delay and area of a simple two-input gate respectively are used to compare the delay for all adder schemes. Except for XOR and XNOR gates; they have the delay of two and also the gate-area of two.

Gates with more inputs are assumed to be composed of two-input gates using their fastest possible arrangement and summing up their gate-delays and gate-areas. Example 4-input OR/AND gate has gate-delay of 2 and gate-area of 3.

For some popular adder schemes, the gate-area and gate-delay are determined based on the calculated from [1], [2], [3], [4], and [5].

b. Hardware schematic

The division function is operated based on the multiplication unit whose the block diagram is described in figure below.



At the moment, the functional unit is planned to implement based on the following foundation.

The basic philosophy to perform high-speed multiplication/division is that all partial products are generated in parallel, and then their accumulations are obtained by using fast multi-operand adder.

The partial product gives the result of an AND operation of multiplicand bits by the multiplier bits shifted by the number of corresponding bits. The output of this operation goes to the summation network.

The multiplier uses the combination of a Booth bit pair encoding algorithm, a sign extension technique, and a carry lookahead adder with two levels of lookahead. The Booth encoding algorithm is a technique that will reduce the number of partial products generated. Using the booth encoding algorithm, fewer partial products will have to be added and therefore the overall speed of the multiplication will be faster.

The summation network uses a series of Carry Save Adder's (CSA) and Half Adder's (HA) to reduce the partial produce to two 128(2n) bit operands which are send to the Carry Propagate Adder Network. Here, Carry Propagate Adders (CPA) are used to add these two 128 bit numbers to get the final result.

Arithmetic Operations to be implemented:
Adders – Carry Propagate Adder, Carry Save Adder, Half Adder

Functions to be written:
Half Adder (HA)
- Takes two inputs X, Y and produces Sum S and Carry C.

Full Adder (FA)
- Takes two input X, Y, Cin produces Sum S and Carry Cout

Partial Product Generator
- Takes two 64 bit numbers and produces64, 64 bit numbers

Summation Network
- Takes 64, 64 bit numbers and reduces them to two 128 bit numbers using carry save adder or Full adder

Carry Propagate Adder
- Takes two 128 bit numbers and produces a 128 bit sum of these two numbers

c. Optimization criteria

The design is focusing on the following criteria:

- Minimum latency
- Maximum throughput

d. Design calculations

All the design calculations will be done in detail to design the multiplier and divider to obtain minimum latency and maximum throughput.

4. SOFTWARE SPECIFICATION

a. Flow chart – Algorithms

Flow chart for the program will be created using the favorite algorithms for best performance selected for the design.

b. Programming Codes

All the programming code will be implemented, debugged and executed to give the correct output as in accordance with the theoretical results.

c. Estimation of execution time

Memory Requirement: the VHDL will be converted to a bit-stream file using appropriate software. The size of the bit-stream file will determine the memory requirement of the VHDL code.

Execution Time: By synthesizing the code and performing the timing analysis of the code written we can determine the execution time of the code.

5. TESTING PLAN

Software:

Each of the files and the functional sub-units in the VHDL code will be tested separately first by writing an appropriate test bench and using test vectors. Then these sub-units will be put together in the final entity that will be tested, simulated, and verified with the hand calculations.

Hardware:

The functional correctness of the hardware will be verified through the VHDL simulation.

6. SIMULATION RESULT

All the theoretical results will be presented in form of a table and simulation results will be printed and attached with the final report.

7. DISCUSSION AND CONCLUSION

The results from simulation and theory will be carefully studied; discussions and conclusions will be made on the results.

8. REFERENCES

[1] J.L. Hennessy and D.A. Patterson, “Computer Architecture: A Quantitative Approach,” Morgan Kaufmann Publishers, 1990.

[2] R.Zimmermann and H. Kaeslin, “Cell-Based Multilevel Carry-Increment Adders with Minimal AT- and PT-product,” IEEE on Very Large Scale Integration (VLSI) system. March 11. 1996.

[3] Baugh, C. R., and B. A. Wooley, “A two’s Complement Parallel Array Multiplication Algorithm,” IEEE Trans. Computers, Vol.22, pp. 1045-1047, December 1977.

[4] B. Parhami, “Computer Arithmetic: Algorithms and Hardware Designs,” Oxford University Press, New York 2000.

[5] I. Koren, "Computer Arithmetic Algorithms," 2nd edition, A K Peters Ltd., Massachusetts 2000.

[6] B.E.Y. Wei and C.D. Thompson, "Area-Time Optimal Adder Design," IEEE Trans. Comput., Vol. 39, No.5, pp. 666-675, May 1990.

[7] Lecture Notes from Dr. Gaj for ECE645-GMU – Spring 2002.