UNIVERSITY OF CALIFORNIA
College of Engineering
Department of Electrical Engineering and Computer Sciences
Last modified on October 16, 2001.

Borivoje Nikolić ***FALL 2001 TERM PROJECT PHASE I*** *EECS 141*
Phase 1, due Friday, November 2, 5pm.

1. Designing a 32-bit atithmetic-logic unit – Background

Arithmetic-logic units are in a heart of any microprocessor. For example Intel Pentium 4 microprocessor has 6 integer execution units, and each of them has a 32-bit ALU.

In this semester's project we will design a critical part of a 32-bit ALU, under different design constraints.

1.1. High level structure

The high-level block diagram of a high-performance ALU (as used in Pentium 4) is shown in Figure 1.

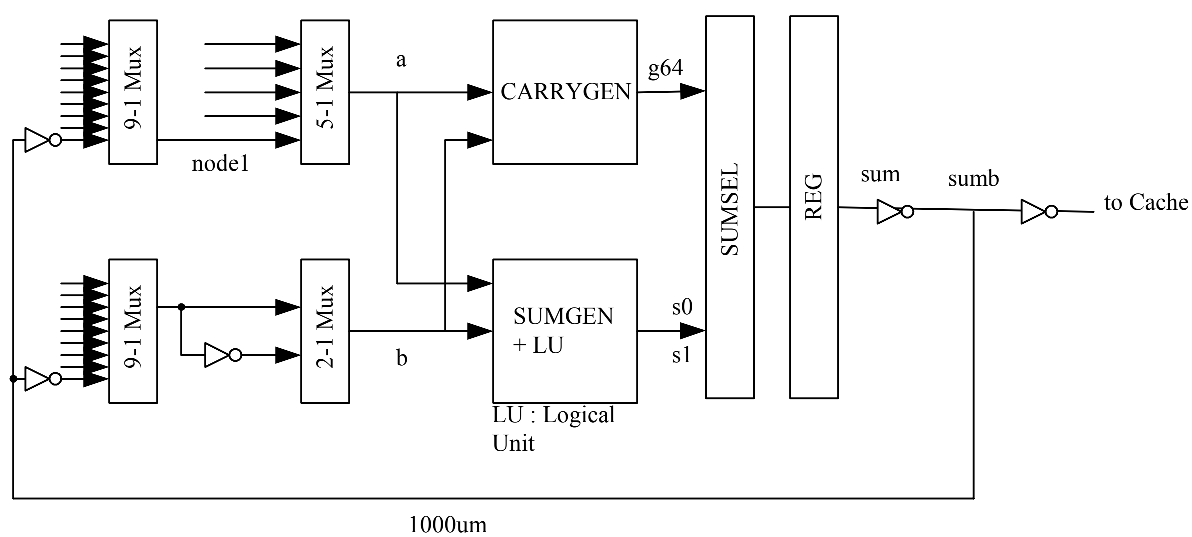

Figure 1. ALU high-level block diagram.

ALUs have four major parts:

- Arithmetic block: this block is used to perform arithmetic operations such as addition, subtraction and comparison. The core of the arithmetic block is an adder. In the architecture presented in Figure 1 the adder uses carry look-ahead and sum-select techniques (the blocks labeled CARRYGEN, SUMGEN and SUMSEL).

- Logic block: this block is used to perform simple bitwise logic operations such as AND (masking), OR and XOR (the block labeled LU in Figure 1)
- Multiplexers: these blocks are used to select the appropriate inputs for the arithmetic and logic blocks. Usually more than two buses arrive at the inputs of the ALU (9 buses in Figure 1, selected by the 9:1 MUXes). Sometimes these multiplexers are used to perform some simple logic operations. The 5:1 MUX is a programmable shifter: its inputs contains shifted versions of the 9:1 MUX output. The 2:1 MUX can be programmed to invert one of the operands (this can be used to execute a subtraction using just an adder).
- Registers: these blocks are used to store the operands and the results. Usually these registers are not part of the microprocessor's register file (but this is not a rule, though).

Note that there is a bus looping from the output back to the input of the ALU, allowing it to use the newly computed results as operands in the next cycle. This is usually a very long wire (longer than 1mm in this situation) and therefore it puts a significant load on the previous stages, that is approximately 0.5pF.

2. Implementation and Constraints

The goal is to design the ALU to be used in a high-performance or mobile microprocessor with a particular set of optimization criteria. The project will be completed in THREE phases.

PHASE 1, due Friday, November 2, at 5pm.

The goal of the first phase is to perform the logic optimization, circuit style selection, and first-order COMBINATIONAL circuit optimization of a 32-bit adder to meet the stated design goals and constraints. The fine-tuning of the design and the actual physical layout of the adder will be performed in PHASE 2.

You should select one of the following design cases:

a) High-performance operation: The goal is to minimize the worst-case propagation delay through the adder. Constraints: in very high performance systems, power is a major concern and usually there are limits imposed on it. The worst-case energy per transition of your design should not exceed 1nJ.

b) Low power for mobile. The goal is to minimize the power consumption of the adder while maintaining reasonable performance. Constraints: the worst-case propagation delay of the adder should not exceed 3ns.

The project is to be done in pairs. You should sign up in teams of two students and choose one of the two design goals by FRIDAY, October 18 at 5pm – A sign-up sheet is posted on the door of Professor Nikolić's office in 570 Cory.

You are free to choose any logic family for the implementation of the project: complementary CMOS, pseudo-NMOS, pass-transistor logic, dynamic logic, etc.

2.1. TECHNOLOGY: The design is to be implemented in a 0.25 μm CMOS process with 4 metal layers. The SPICE technology is in the g25.mod file.

2.2. POWER SUPPLY: You are free to choose any supply voltage and logic swing up to 2.5 V. Make sure that you use the appropriate model when you perform hand analysis.

2.3. PERFORMANCE METRIC: The propagation delays for static designs is defined as the time interval between the 50% transition point of the inputs and the 50% point of the worst-case output signal. Make sure you pick the worst-case condition and state EXPLICITLY in your report what that condition is. Note that for dynamic designs the propagation delay is defined in this case as the delays of the evaluate phase ONLY (at least in this phase of the project)!

2.4. AREA: The area is defined as the smallest rectangular box that can be drawn around the design. Note that since the ALU must interface with the cache, all of the row-match inputs must be accessible from the left side of the design, in row-address order. In the first phase of the project, you should make the area estimations based on the total transistor width and the wiring complexity. An expression on how to predict the area will be provided shortly on the web page.

2.5. NAMING CONVENTIONS: The input operands of the adder are named A<31:0> and B<31:0>. The output is SUM<32:0>, where SUM<32> is the carry out bit.

2.6. REGISTERS: In this phase of the design you do not need registers. The data flow from input to output should be combinational logic.

2.7. CLOCKS: There should be no global clock, since the design is combinational. If you choose dynamic logic, you are permitted a precharge/evaluate clock, but the result must become available after ONE evaluate phase (no pipelined logic). Observe that the load capacitance of the clock should be included in the energy analysis.

2.8. V_{OH}, V_{OL}, NOISE MARGINS: You are free to choose your logic swing. The noise margins should be at least 10% of the voltage swing. Test this by computing the VTC between one of the inputs and the output signals (with the other outputs set to the appropriate values) for a static design. For a dynamic circuit, apply an input signal with a 10% noise value added to the input and observe the outputs.

2.9. RISE AND FALL TIMES: All input signals have rise and fall times of 50 ps. The rise and fall times of the output signals (10% to 90%) should not exceed 200ps.

2.10. LOAD CAPACITANCE: Each output bit of the adder should have a load of 0.5pF, appropriately buffered.

2.11. INPUT CAPACITANCE: Each input of the adder should not load the previous stage with more than 50fF (less is OK).

3. Simulation

Analyze the circuit by using either SPICE or IRSIM to simulate the design. Prove the functional operation of your circuit using either SPICE or IRSIM. The input patterns to be used to determine functionality and energy dissipation will be provided on the web page. The pattern that causes the worst-case propagation delay or energy consumption should be determined by yourself, based on an analysis of your circuit schematic. Make sure that you define your circuit in a hierarchical fashion in order to keep it easy to understand and debug.

4. Report

The quality of your report is as important as the quality of your design. One must sell the design by justifying the design decisions and by providing all the vital information. Be sure to emphasize relevant information and to eliminate unnecessary material. **Organization, conciseness, and completeness are of paramount importance.** Do not repeat information we already know. Use the templates provided on the web page (Word and PDF formats). Mail an electronic version of your report as a Word or PDF file to ee141-project@bwrc.eecs.berkeley.edu. Make sure to fill in the cover-page and use the correct units. A report has to be submitted at the end of each phase of the project.

4.1 Report 1

Your report should discuss your overall design philosophy and the important design decisions made at the logic and circuit level. Discuss why your approach increases the operating speed or helps to reduce energy or area, while meeting the performance specifications. Provide your current estimates of the results and describe how you arrived at them. Describe the sizing methodology used in your design. Include schematics and highlight the important elements, and critical paths.

Prove that your results are accurate by providing the crucial plots (don't forget to mention the input patterns used to obtain those plots). The total report should not contain more than three pages. You are not allowed to add any other sheets, except for important plots. The organization of the report should be based on the following outline:

- Page 1: Executive summary, overall design decisions, remarks, and motivations
- Page 2: Logic and transistor diagrams annotated with transistor sizes and worst-case timing path. Plot showing the functional operation of the cell. Comments.
- Page 3: Timing and energy simulations – derive the value of the worst-case path and average energy. For the latter, a set of test patterns will be provided on the web page.

Lastly, you are required to e-mail the SPICE INPUT DECK used to analyze the energy to mailto:ee141-project@bwrc.eecs.berkeley.edu. Remember, a good report is like a good layout: it should perform its function (convey information) in the smallest possible area with the least delay and energy (to the reader) possible.

The quality of the report is an important (major) part of the grade!

The total project grade is divided into the three phases

- 40% Phase 1
- 30% Phase 2
- 30% Phase 3

For each phase, the grade will be divided as follows:

30%	Results
20%	Approach and correctness
40%	Report
10%	Creativity