UNIVERSITY OF CALIFORNIA
College of Engineering
Department of Electrical Engineering and Computer Sciences
Last modified on October 21, 2002.

Borivoje Nikolic ***FALL 2002 TERM PROJECT PHASE I*** *EECS 141*
Phase 1, due Tuesday, November 12, 5pm.

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

Arithmetic-logic units are the heart of any microprocessor. This semester, we will design the critical part of a 32-bit ALU.

1.1. High level structure

The high-level block diagram of a high-performance ALU is shown in Figure 1.

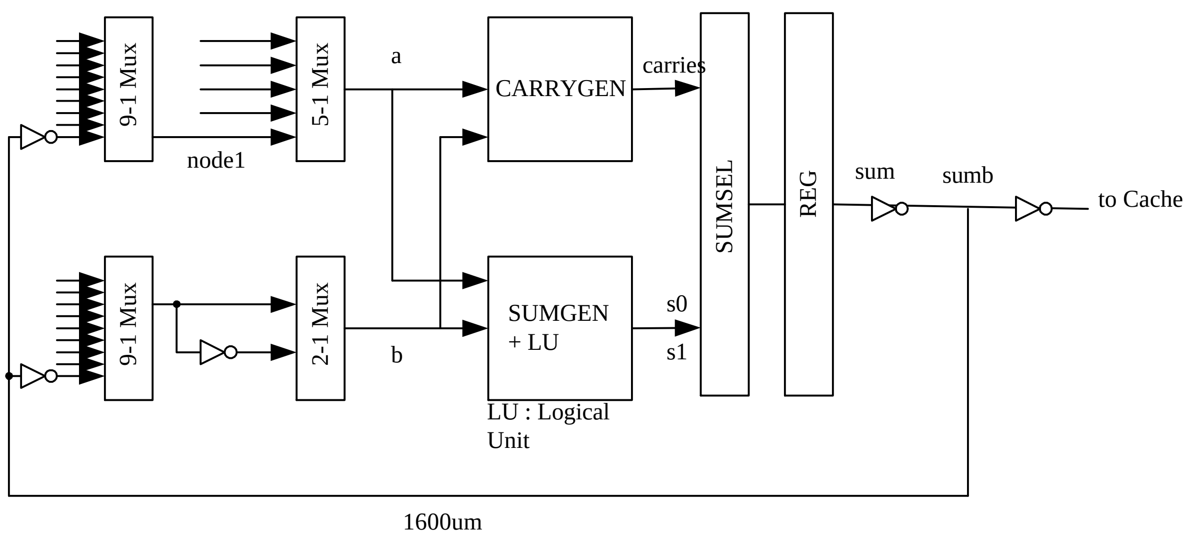

Figure 1. ALU high-level block diagram.

ALU's 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 9:1 MUX's). Sometimes these multiplexers are used to perform

some simple logic operations. The 5:1 MUX is a programmable shifter: its inputs contain 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 (though not always the case).

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 (1.6mm in this case) and therefore puts a significant load on the previous stages.

2. Implementation and Constraints

The goal of this project is to design the carry-lookahead adder for an ALU to be used in a high-performance or mobile microprocessor with a particular set of optimization criteria. The project will be completed in TWO phases.

PHASE 1, due Tuesday, November 12, at 5pm.

In the first phase of the project, you will choose a circuit style, design the logic, and lay out basic cells for a 32-bit adder. You will also have to do some pencil-and-paper optimization in order to meet the stated design goals and constraints. The complete adder will be assembled and simulated in PHASE 2.

Physical and electrical specifications:

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 any hand analysis.

2.3. PERFORMANCE METRIC: The propagation delay for static CMOS design 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. 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. Since the ALU must interface with the cache, all of the row-matched 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 area estimations based on the total transistor width and the wiring complexity. An expression for prediction of the area will be provided on the web page.

2.5. Each bit slice in the adder should accommodate 9 metal-5 busses and is 144λ (36 metal pitch) wide. Other circuits in the datapath set this constraint.

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 to use dynamic logic, you are permitted a precharge/evaluate clock, but the result must become available after ONE evaluate phase (no pipelined logic). Remember 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: Your adder is driving a 1.6mm long bus with 9 loads evenly distributed. Each capacitive load is equal to the adder input capacitance. Each wire in the bus is 4λ wide with 4λ spacing in M5.

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

The goal is to minimize the **delay**, **power** and **area** of the design. Your delay should not exceed **12 fanout-of-4 (FO4)** inverter delays.
In the first phase, you should make the following decisions:

- Circuit family (complementary static CMOS, pass-transistor, dynamic).
- Type of carry-lookahead tree to be used.

Remember that your decisions on the logic level significantly affect the final delay, area and power. You should discuss your designs with TAs.

In this first phase, your design will consist of:

- Pencil-and-paper design of the adder.
- Identification of the critical path in the adder.
- Hand optimization of the delay.
- Schematics and layouts of all different cells that will be used.
- Estimation of delay, area and power of your design, by simulating individual cells with proper loading.

3. Simulation

Analyze the circuit by using either SPICE or IRSIM to simulate the design and prove that it functions correctly. You will need to determine the input pattern that causes the worst-case propagation delay or energy consumption by analyzing your circuit schematic.

4. Report

The quality of your report is as important as the quality of your design. One must sell the design by justifying all design decisions. Be sure to provide all relevant information and 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 must 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 estimates of your results and describe how you arrived at them. Describe the sizing methodology used in your design. Include schematics and highlight important elements and the critical paths.

The total report should not contain more than four pages. You are not allowed to add any other sheets. The organization of the report should be based on the following outline:

- Cover page: Names, project title, decisions about logic family and the carry path, performance estimates.
- Page 1: Executive summary, overall design decisions: circuit style, drawing of the carry-lookahead tree, critical path, sizing, remarks, and motivations
- Page 2-3: Schematics and layouts of all circuit cells. Simulated delays, power and area.

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 two phases

- 50% Phase 1
- 50% Phase 2

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

- 30% Approach and correctness
- 30% Results
- 30% Report
- 10% Creativity