

**CSCE 613 – Design Project Guidelines  
Fall 2005**

1. Topics & Background

#	Project Topic	Reference	Design Description
1	Ripple-Carry (RPC) Adder.	Rabaey, Ch 11, sections 11.3.1, 11.3.2	pp. 561-566.
2	Carry-Look Ahead (CLA) Adder	Rabaey, Ch 11, sections 11.3.1, 11.3.3	pp. 578-579.
3	Carry Select (CSA) Adder (linear version)	Rabaey, Ch 11, sections 11.3.1, 11.3.3	pp. 574-575.
4	Carry Bypass Adder	Rabaey, Ch 11, sections 11.3.1, 11.3.3	pp. 571-573.
5	Tree Multiplier (Graduate students).	Rabaey, Ch 11, sections 11.4.1, 11.4.2, 11.4.3	pp. 592-594.

2. Project Objectives

- ❖ To have the teams work through an entire design project—from concept to silicon layout, including parameter analysis, verification, and performance and power characterization. The scenario is that your team is part of a fables IC design team creating a model library for a collection of arithmetic circuits that would be used by system designers to build high-speed arithmetic processing components in the design of a custom CPU.
- ❖ The design sizes will be **4-bits** in width for these arithmetic circuits; this is a reasonable size for a project and is comparable for the base level design unit for inclusion in a high-level, characterized “hard macro” library. Although this bit-width may not be sufficient to show tradeoffs in the architectures your team is exploring, it will nonetheless allow you to complete a reasonable design unit comparable to what you would do if you were assigned such a project in industry.
- ❖ We will attempt to use the set of objective design criteria defined by Rabaey et al., Chapter 11. However, he is using different tools and technology library. These are listed as follows:
- ❖ *Technology* - We’ll use a 4-Metal layer CMOS process using the ADK ‘ami05’ .05 micron technology library. Our design style will be to use *complementary, static CMOS* designs using the Pull-up/Pull-down matched PMOS/NMOS circuit patterns we have been discussing in the class.

- ❖ *Power Supply* - We'll use a 5-volt power supply, to be consistent with current industry standard and to roughly match the ADK library.
- ❖ *Clocking* – Most of the arithmetic circuits should not require an explicit clock signal, as we have not included output storage registers in the design specifications. We will employ the most basic of clocking layouts to achieve design functionality and performance, in places where we need to incorporate clocking. We'll assume that we would like our combinational circuit to operate within a tolerance of 2 ns, meaning that we're clocking at 500 MHz.
- ❖ *Area* – The designs should take up as small an area as possible. Since we will be doing a “block” based design, the cells you design should be as compact as possible, and yet still pass the DRC (layout design rule check). Overall area is defined as the smallest rectangular box that can be drawn around the design (Rabaey et al.). For schematic design, you should make area estimations based on total transistor width and the wiring complexity. (You can probably make some initial assumptions about transistor sizing based on the text material in Chapters 5 & 6, for inverters and NAND/NOR gates.)
- ❖ *Delay* – The propagation delay through static CMOS complementary logic structures (PMOS/NMOS) is defined as the time interval between the 50% transition point of the inputs and the 50% point of the worst-case output signal (Rabaey et al.). You will need to determine your worst-case delay path through your CMOS cells, and you will need to state in your analysis what path is the worst-case one. We'll use manual methods to estimate this critical-path delay parameter. It basically constrains the speed of your design unit. Global, pin-to-pin critical-path delay will dictate how fast you can clock your circuit (i.e., clocking the storage registers on either side of your arithmetic circuit). So, if we have a 2 ns cycle time target, you'll need to insure that your design is fast enough to meet this critical constraint. The Hodges et al. materials will be a good reference for how the manual methods are used to obtain a quick assessment of the performance (first order model approximation).
- ❖ *Noise Margins* – We will only make cursory analysis of the noise margins, using the materials provided in the text, Chapters 5 & 6. However, they should be at least 10% of the voltage swing. You can test this point by computing the VTC (voltage transfer curve) between one of the inputs and the output signals--with other outputs set to appropriate values—for a static CMOS design (Chapter 6).
- ❖ *Rise/Fall Times* – We'll assume that all input signals have rise and fall times of 50 picoseconds. The rise and fall times of the output signals (10% to 90%) should not exceed 200 picoseconds—a design constraint, and subject to conformance by your design activity and use of the Mentor design library.

- ❖ *Load Capacitance* – We'll have to assess this parameter for our specific designs and Mentor ADK library elements. However, the parameters of concern are: (1) length of the interconnect signal path, (2) number of fan-out loads being driven, (3) assumption that the circuit's capacitive load is equal to its driving input capacitance (see Chapter 5 on this point), and (4) wire width and wire spacing for each data line of the  $n$ -bit arithmetic input buses and the  $n$ ,  $n+1$ , or  $n+2$  bit output buses for the Adders.
- ❖ *Input Capacitance* – Each input of the arithmetic circuit sub-units should load the previous stage with no more than 50 picofarads (pF). We'll see whether this is possible given our available Mentor ADK technology library selections. If it is not possible, then you'll show me the analysis that says why it is not possible.
- ❖ *Global optimization* – The overall optimization goal is to minimize delay (for high-performance) while also keeping as small of a design area as possible, and as minimal power consumption as possible. These are often conflicting objectives. However, the assumption is that the application will be for high-performance computing—therefore, area and power consumption are secondary objectives. However, if your design project has two different optimization problems, you will create one as discussed for high-performance. However, the second will have minimal power consumption as its primary objective, with area followed by speed as the secondary objectives. This is because, in many low-power, mobile applications, battery life is of primary concern; often, low-power design techniques require extra area for redundant logic.

### 3. Project Procedure

- ❖ The project will be completed according to the staged sequence of project deliverables. The primary deliverables will consist of the following: (1) functional and architecture analysis, (2) gate-level schematic design, (3) CMOS schematic design, (4) cell-level layout and parameter analysis, (5) global circuit layout and parameter analysis, (6) comparison of different architectures on which your team is working (each team has two), and (7) reporting via Project Report and PowerPoint® presentation to the class.

### 4. Project Grading

- ❖ Organization and presentation of the design artifacts and the analysis data for area, delay and power consumption, including: (1) estimation of these at the gate and CMOS switch levels, (2) characterization of these for the layout cells, (3) characterization of these for the global design including inter-cell wire routing interconnect.

- ❖ Completeness of the design results, in terms of meeting the target deliverables, where all the artifacts have been completely specified and delivered.
- ❖ Correctness of the design artifacts to meet the functionality of the specified circuits, and realize the high-level architecture in terms of high-speed, low-power CMOS designs.

## 5. Team Organization

- ❖ I'd recommend that your team organize itself as follows: (1) if you are doing a project comparing two different arithmetic architectures, then one team member can focus on one, while the other team member focuses on the other. (2) If your project involves a single circuit, then you are required to carry this circuit through two optimization paths—one for speed and one for minimal power consumption. In this case, one team member can focus on the high-performance version, while the other team member focuses on the low-power version. These should/will be different—even at the gate-level schematic.
- ❖ When it comes to preparing deliverables each week, I'd recommend that you carry out your team verification in tandem. As such, one team member reviews the work of the other team member, critiquing and making suggestions. When verifying circuit functionality, you should select the same input data patterns (4 patterns should be sufficient) for each pair of architectures on which your team is working. These architectures will either be (1) using two different Boolean descriptions for arithmetic circuits for addition or multiplication, or (2) a single architecture for which you are designing with two different set of optimization criteria (speed versus power consumption).
- ❖ For each deliverable, the team members should work together to prepare this, as I am interested in comparison of your circuits at each stage of the design's evolution. This comparison should be in terms of (1) hardware resources used, (2) number of gates in longest path through the design (input pin to output pin). At each design level, these quantities will be in terms of the elements at that level of the design (logic gates, CMOS transistors, interconnect length, etc.). These will be constrained by the design objectives noted earlier in this document.
- ❖ Note that, since we are interested in the system-level characteristics of the resultant designs, you team will need to also look at the overall throughput of your pair of design submissions. For the arithmetic circuits, this should be in terms of how many arithmetic operations can be performed per second, comparing either between two arithmetic architectures, or between high-performance and low-power versions of a single architecture. This is derived

from the overall time, in nanoseconds, it takes to perform a single calculation (the “latency” of getting data through the circuit from input to output).

- ❖ Each team must do a comparison of your candidate designs on which you have been working. This comparison will be part of the final report and the presentation we’ll have for each project team at the end of the semester. Once you have completed the designs and the accompanying analysis tasks, I’d recommend taking the relevant data from each design and plotting it in Excel® for inclusion into your PowerPoint® presentation.

## 6. Project Deliverables

<b>Deliverable</b>	<b>Description</b>	<b>Comments</b>
Gate Level Schematics	This is the gate-level description of the circuit. Given a register-level architecture diagram and a functional understanding of your circuit, you should detail the schematic structure of each component of your design, and include any Boolean expressions or state machine descriptions that are appropriate. Note: going through the gate-level design stage is an important point we must pass through to get to the CMOS circuit. Since we want to carry out high-performance/low power design, we are not using VHDL for this.	Most of the projects consist of a number of decomposed blocks, many of which are recurring (given we are targeting multi-bit designs for the arithmetic circuits). The basic arithmetic elements should be designed, and the control logic (or carry logic network) should be designed.
Switch-Level Schematics	Given a gate-level schematic, you’ll need to convert the gates into NAND, NOT and NOT structures for realization as Static Complementary CMOS structures. This should also include an inventory list of the set of basic cells you will be designing. Note: this depends on characteristics of the available CMOS technology libraries with the Mentor tool set.	You should consider doing schematics for the basic cells first, placing them in a library, and then reusing them for the complete schematic, where you complete the connections between cells.
Cell Layouts and Cell-level circuit analysis	Given the CMOS circuit structure, layout the basic cells for the design. The layout may be done by migration from schematic, but rule checking and cell-level simulation should also be done. If we can’t get simulation, then manual analysis techniques will do.	These should be verified and added to your library for reuse when creating the whole circuit layout. The analyses will ref. Rabaey, Chs 3-6, 11.
Routed Circuit Layout and Global analysis	This will be the incorporation of your cells, along with interconnect, to realize the function of your circuit according to your architecture	This will be the complete circuit deliverable with analysis results.
Final Report and Presentation	The Final report is the collected final deliverables, according to the format.	Template is posted on web page.