Learning Objective
Learn the VLSI CAD tools and chip design concepts by designing 8-bit Ripple Carry Adder (RCA).

Instruction
Design 8-bit Ripple Carry Adder (RCA). Using the Cadence tool Virtuoso, design the 8-bit RCA schematic and layout. The design must be free from the DRC errors and pass the LVS checking. Then simulate both the schematic and the layout extracted circuit using Synopsys tool Hspice. Verify the correct functioning and measure the worst-case 8-bit add time. Calculate the parameter: (layout area)*(total adder delay time square) = AT**2.

1. Read Chapter 11, pages 560 to 567, of the textbook. Much of the needed RCA background is provided.

2. You may implement the mirror adder shown in Figure 11-6.

3. Use hierarchical design method to manage design complexity.

4. As an example of hierarchical design method, one can design 1-bit full adder with the carry circuit and the sum circuit. Each circuit is designed as separate function but combined together to form a 1-bit full adder. The 8-bit adder is then designed with the cascading of 8 instances of the 1-bit full adder.

5. Following sequence of figures show the design example.
Carry circuit layout

Sum circuit schematic
Sum circuit layout

1-bit full adder schematic
1-bit full adder layout

8-bit adder schematic
6. Be sure to **save** the files time to time as you do the design. Do run DRC on your layout designs. Also check LVS as you complete schematic and layout of each functional unit. Do verify the correct functioning by Hspice simulation. You may also obtain the timing information (signal propagation delay) if you do the Hspice simulation on the layout extracted circuit.

7. For an 8-bit adder, there are 9 outputs. Do put 10fF capacitors on each of the output when Hspice simulating.

8. Measure the worst-case 8-bit add time from the layout extracted circuit including the parasitic capacitors. Measure the total area of the 8-bit adder. Calculate the AT**2 for your 8-bit adder.

9. Homework 2 preparation: Create a directory hw2yourlastname under c411 directory (assuming you have c411 directory in your home for this class). Change the current directory to hw2yourlastname. Set up the directory for a new Cadence Virtuoso library. Then start the hw2 design project. For example, my hw2 directory will be hw2choi and I will be running the following unix commands right after I login to my account:

```
% cd c411
% mkdir hw2choi
% cd hw2choi
% runcds
% virtuoso &
```

You can follow the same except that you need to use your last name after 'hw2'.

10. Collect all the components in this directory, it will be 'tar' and zipped, turn-in to the instructor for grading. If you need previously designed cells copied to this project, you must use 'Copy Wizard ...' . This way, you can collect all the files within the hw2yourlastname directory and it will allow the grading. Otherwise, any missing files may cause...
your design verification for grade to fail. You will lose points if your turned-in hw2 design files cannot be verified by grader.

11. Create a HW2 report file hw2yourlastname.doc and include captured image of layout and the simulation results. Add your explanations and comments.

12. The HW2 report file must also include the answers to the questions below.

13. The HW2 report file can be in .doc or .pdf, must include a cover page for student information such as 'CMPEN 411', ‘Homework 1’, your name, etc. The sample format of the HW2 report file is posted: Sample HW2 Report

14. Create a tarred zip file of your hw2yourlastname directory in c411 directory. It will contain the schematics, symbols, layouts, .hsp, .sp files, and .doc report file. In your c411 directory, use the following unix commands

   ```
   % tar -czvf hw2yourlastname.tgz hw2yourlastname
   ```

to create a tarred zip file of hw2yourlastname library. For example,

   ```
   % tar -czvf hw2choi.tgz hw2choi
   ```

will archive the directory hw2choi and create a zipped file hw2choi.tgz in my c411 project directory. For the grading, the command ‘tar --xzvf hw2choi.tgz’ will be used to restore the project.

You must delete .tr0 files before zipping, for their sizes are very large.

15. Turn-in your project zip file through Penn State ANGEL. Deposit your zip file into the Homework 2 DropBox under CLASS tab in CMPEN 411 Course.

16. Make sure that you include all the files necessary into your project folder, in order to verify for grading. Turn-in your project before 11:30pm on the due date.

17. For your information, the grading sheet for the Homework 2 is posted: HW2 Grading Sheet

Questions:

1. What is the rise time and fall time of the output signal s7? Be sure to answer this question with the CL = 10 fF.

2. What is the worst propagation delay from any input ai, bi, ci to any output si? The speed of this circuit is computed as (1/delay). Be sure to answer this question with CL = 10 fF on each si.

3. What is the total area of the circuit? One can measure the size of the layout directly in the layout tool.

4. One may measure the chip design effectiveness by chip speed and chip size. Better design results in faster speed and smaller area. Calculate the $A T^2$ of your design, where $A$=total area in um$^2$ and $T$=propagation delay in uSec.

Adder Information

1. The Ripple-Carry-Adder is composed of eight full adders. Therefore, we should design the full adder first. A full adder is a logical circuit that performs an addition operation on three input binary digits (A, B and Cin). A and B represent the operators, and Cin represents the input carry digit. The full adder produces a sum (S) and a carry value (Cout). The logic expressions of the sum and carry are expressed as following:
\[ S = (A \oplus B) \oplus C_{in} \]
\[ C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B)) = (A \cdot B) + (C_{in} \cdot B) + (C_{in} \cdot A) \]

The gate level design is shown in the following figure:

![Sum and carry logic](image1)

Figure: Sum and carry logic.

2. You may design each logic gate separately and compose them together. The Figure below shows the transistor circuit using the following combination logic.

\[ S = A \overline{B} \overline{Cin} + \overline{A}BCin + \overline{A}BCin + ABCin \]
\[ C_{out} = \overline{A}BCin + A \overline{B}Cin + \overline{A}BCin + ABCin \]

There are many different implementations of 1-bit full adder, you may search on the internet to get more information, and choose the design you want.

![Transistor circuit for a 1-bit full adder](image2)

Figure: The transistor circuit for a 1-bit full adder (Mirror adder).

3. It is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is a ripple-carry- adder, since each carry bit "ripples" to the next full adder. Therefore, in our eight bits ripple-carry-adder, we have 16 bits (A0-A7 and B0-B7) of operators and one bit of input carry (C0). We have 8 bits of sum output (S0-S7) and one bit of carry output (C7). The structure is shown as follows (FA = full adder):
It is straightforward to design a full-adder first and cascade eight of them to make an 8-bit ripple-carry-adder.