

# ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation

---

We will now begin the design by implementing the logic design of the 8-bit Pipelined adder. The following sections introduce you to the procedures to use Cadence for schematic capture and simulation which you will use to implement the required logic design.

## 1. Introduction to Cadence Tool

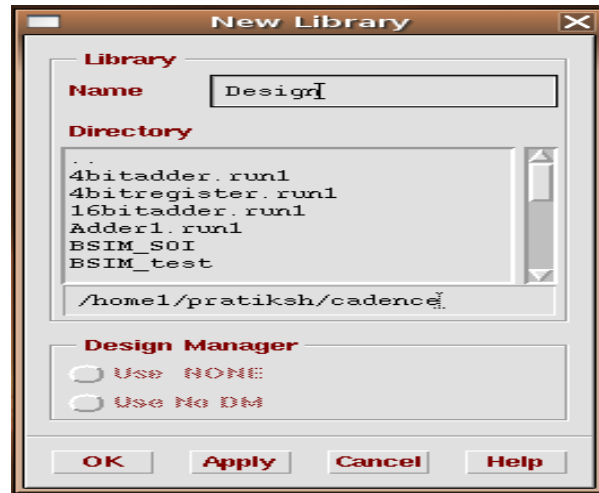
- By now you should have created a directory called "cadence" in your home directory and changed your working directory to cadence. If not refer to Lab0 before you begin.
- To start Cadence type "icfb &". If you see "\*\*\*/xlsfonts: not found", disregard it. You can also see two pop-up windows telling you what's new about this version of Cadence. Click ok. At the same time, a small rectangular window will pop up. This is called Command Interpreter Window (CIW). This is where icfb will display all the messages about your commands. Do not close this window (except when quitting!) and always keep this window visible. CIW displays all the error and warning messages, so please look into this whenever possible. Also it will guide you what to do next. A window called Library Manager will also pop up.
- The heading of the CIW is icfb - Log: /homex/xxxx/CDS.log and has three menus File, Tools and Options. The lower left corner has the icfb prompt which displays ">", when it awaits next action.
- We can now create designs in icfb. It stores all data in a library. A library can contain one or more designs. Designs in a library use a technology file, which describes the rules associated with the fabrication processes. So when we create a new library, the first thing which should be done is to attach a technology file to a new library. To create a new library, go to Library Manager > File > New. You will see three items when you do "left click" on New: Library, Cellview, and Category. Library is the top concept in hierarchical structure. And below are cells with several different properties. For physical layout design, the Cellview name is layout. If it is a gate level design, the Cellview name is schematic. For transistor level design, we use cmos.sch. We have also "extracted", and "symbol" which will be used in the following labs.
- Icfb uses Library Manager for browsing through all the libraries. The Library Manager has several libraries including any designs created by you. If you cannot see "NCSU\_Analog\_Parts" or "NCSU\_Digital\_Parts", you need to add these libraries. To add them to your libraries, go to Library Manager > Edit > Library Path". A window called "Library Path Editor" pops up. Type "NCSU\_Analog\_Parts" in Library and "/baby/cadence/ic50/local/lib/NCSU\_Analog\_Parts" in Path. And press Return. You also should have "basic" library. It is also in the same directory. Type "basic" and the directory path. After ending a typing a library, press enter and it will give you another line to add library. **If you do not find the TSMC 0.2um library(NCSU\_TechLib\_tsmc02), type NCSU\_TechLib\_tsmc02 in Library and "/baby/cadence/ic50/local/lib/NCSU\_TechLib\_tsmc02" in path ).** Close and save the

# ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation

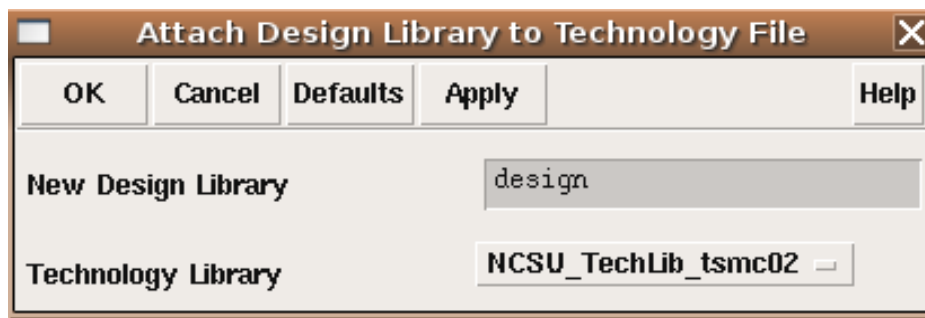
---

“Library Path Editor” window.

- Click new->library and type in name “Design” as shown below. Click OK.



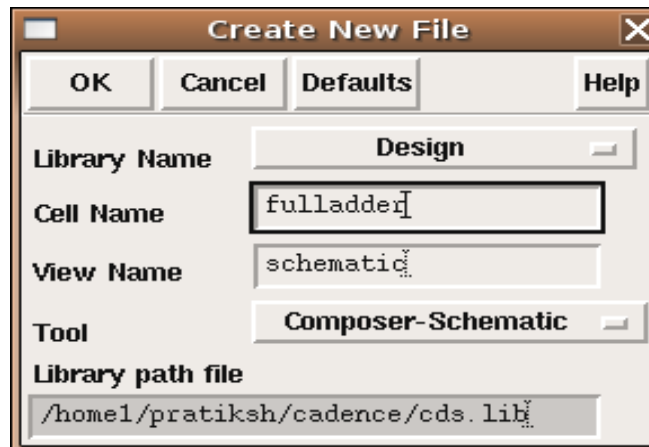
Next, you will find a menu for technology file. Choose attach an existing file and click “NCSU\_TechLib\_tsmc02” as shown below. It means from now when you design a device, you will do it according to the rules related to this technology



- A library contains more than one design. Icfb calls each design a cell. For example there can be one cell for ALU, one cell for a flip-flop, and so on. In this lab we will design a one-bit adder. So let’s create a new cell called "fulladder". For this click on the design library in the Library Manager and choose File New > Cell view. Type cell name and view name. (Figures are just reference only. Do type as it shows)

# ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation

---



- In the Cell Name field, type "fulladder". As mentioned, above there can be several designs in a library and cadence stores each design as a cell. Also each cell can be represented in many ways for example as gate level or transistor level schematics, layouts, SPICE netlists or verilog etc. Icfb calls these representations as cellviews. Since a design can be represented in many ways i.e. there can be many cellviews for a cell.
- For the adder design we will create a gate level schematic. This is called a "schematic view". In the view name field, type "schematic". Click OK on the form and a schematic view is created. The CIW reads created cell view (fulladder schematic).
- Click on the fulladder cell in the Library Manager and you can see the schematic cell view ("fulladder schematic"). By now you should be clear with cells and cell views, if not ask your TA. Also look back and see the procedure for creating cells and cell views. This is the general procedure and you must remember it.

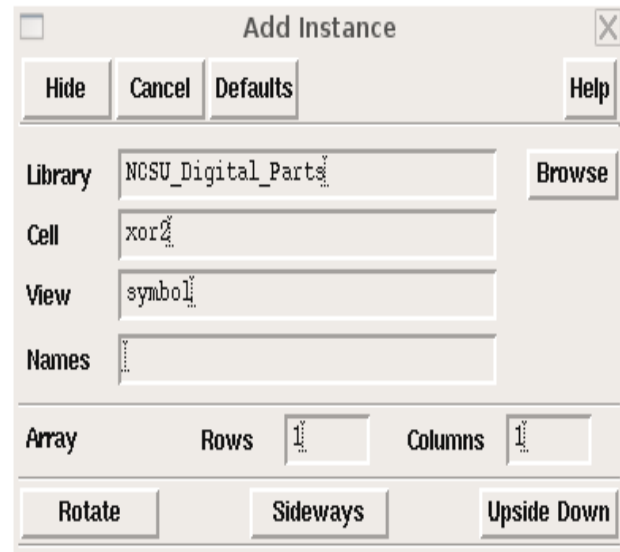
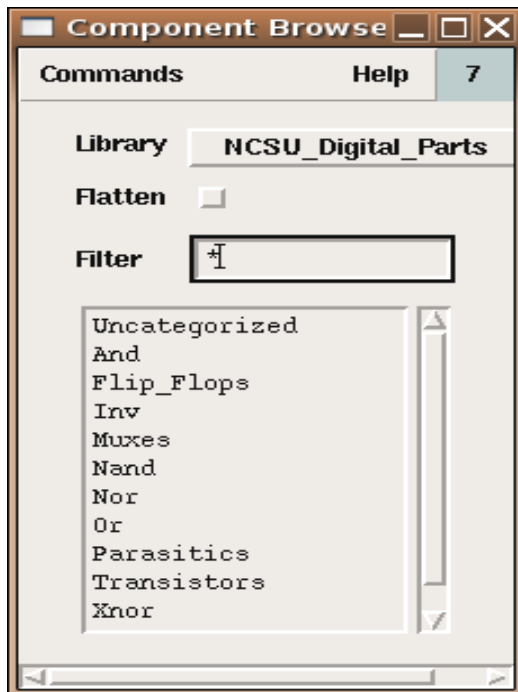
## 2. Schematic of the Fulladder Circuit.

The full adder is the basic building block of the 4-bit adder block that we need to design which we will eventually use to create the 8-bit Pipelined adder. The method is to create the design (schematic in this case) of the lowest building block (fulladder) and create a “symbol” of this design which can be reused in some other design (we would be using it in a higher level design to create a 4-bit adder)

- We can now create the first design - a gate-level schematic of 1-bit adder. The adder will take three inputs: A, B and C (carry-in). The output will be two bits: SUM and CARRY. If you are not aware of the truth table for the adder, ask your TA.
- From the truth table the SUM and the CARRY functions can be generated. This can be simplified to  $CARRY = AB + (A \text{ XOR } B) C$  and  $SUM = (A \text{ XOR } B) \text{ XOR } C$ .

# ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation

- To start the Cadence schematic editor, called Virtuoso, (if it is not already open), hold the middle mouse button on the schematic cellview of adder1 in the library Manager, scroll down and choose "Open". At most of time the editor will pop up automatically when you add the new cell. This is where you design your schematic. To create an instance of a gate, choose Add → Instance from the editor menu. Component Browser (CB) and Add Instance window pops up.



- Choose library “NCSU\_Digital\_Parts” in CB. You will see lots of items like And, Flip-flops, etc. You can choose any gate you want from this list. If you want OR gate, click “OR” in CB. You will see OR2, OR3... OR6. Choose OR2 if you want 2 inputs OR gate. If you click OR2, the Add Instance window will be automatically filled. Move your mouse to Virtuoso window and click at the point where you want to place the OR gate. One more click you will have one more OR gate. Press ESC if you do not want OR gate any more.

**NOTE: Use only XOR and NAND2 (you will have to figure out how to replace the AND and OR gates with only NAND2 gates) for the full adder. This is to maintain consistency among all students and it also means you have to implement only the XOR and NAND2 layouts. (Remember: step 2 in section 2 of Lab0 will ask you to implement layouts of all the logic gates you used in the schematic design. Not only will you have to design fewer layouts (NAND2 instead of AND and OR) but also the implementation of a NAND2 layout can be easier than that of an AND or OR layout as you will find in the following weeks).**

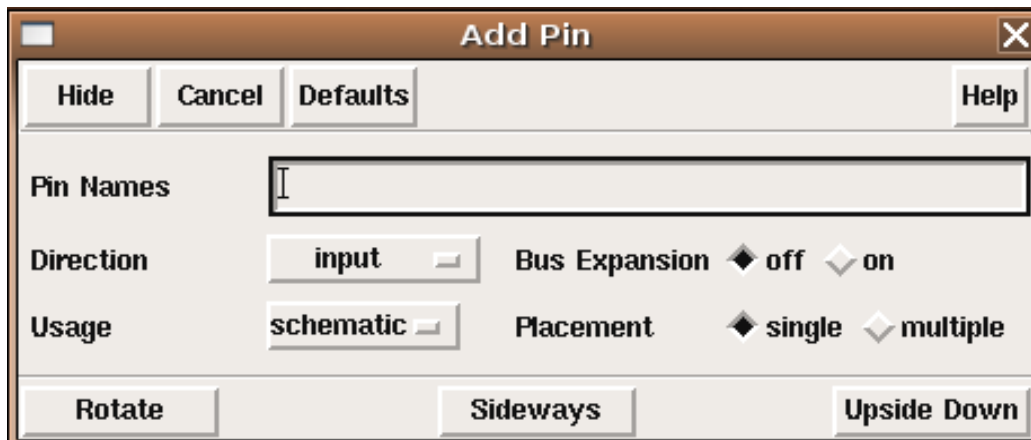
- Place as many instances of this gate as required. When you finish press ESC key. If you

## ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation

---

have placed some extra instances then go to Edit → Delete in the Composer menu and click on the instance to delete. To exit from the Delete mode press ESC. To save the design click on Design → Save. Always save your work when you finish some important operations.

- Place instances of all the gates required and make you familiar with Add Instance and Delete Instance functions. Make sure you have plenty of space left for wiring your schematic. You can Zoom In and out to fit a part or whole of your design using Window → Zoom → Zoom In and Zoom Out respectively.
- We now have to wire the design. Before that we need to place input and output pins. The pins are for the input output connection of this design and also help for the simulation. We need five pins, three input pins A, B, C and two output pins SUM and CARRY. Select Add → Pin in the Virtuoso menu. Add Pin form pops up. Type “A B C ” in the Pin Names field. Make sure that the Direction of the pin is “input”. Go to the schematic window and place the three input pins wherever you feel appropriate. (You might have to zoom).



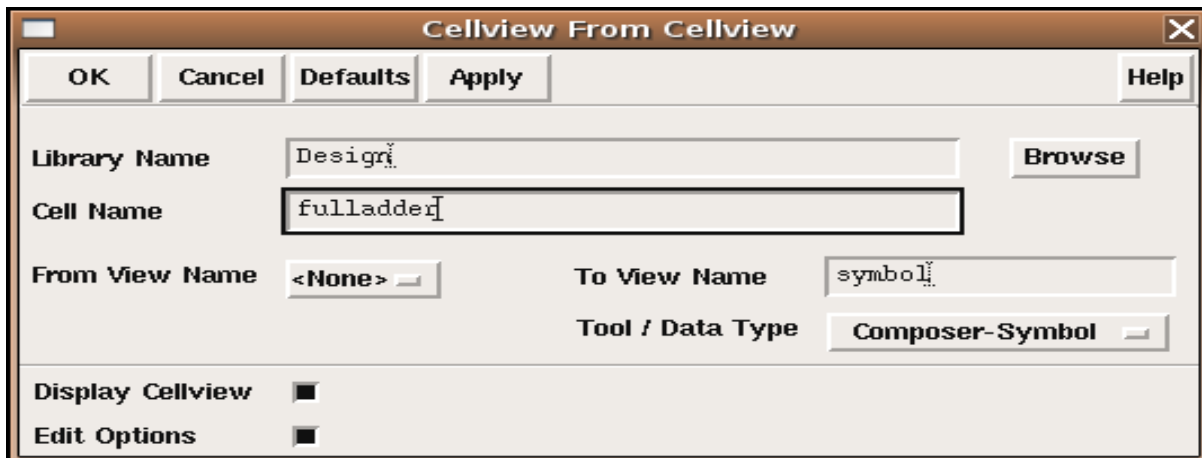
- Note that the pins A, B and C are different from the A, B and C you see on the instances of the gates. These (the later) are for the gates themselves, one level below the hierarchy. In the same way, create the output pins “SUM and CARRY” but remember to change the direction field in the Add Pin form to “output”.
- Before wiring let’s see some Composer functions. You can move and copy the instances using Edit → Move and Edit → Copy functions. Also you can undo the changes made using Edit → Undo. The CIW and the lower corner of the Composer will guide you.
- Finally let’s wire the schematic. Choose “Add → Wire (narrow)” in the window. Click on the A pin, a wire follows you. Take it to the point you want it to go and notice that the editor does automatic routing for you. Place it at the output of the proper gate. Similarly wire the rest of the circuit. Note that the wires can overlap without making contacts unless you explicitly do so. To make sure you wire to the exact place you want,

# ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation

---

notice that there is a diamond appearing when you wire to some place.

- Wire the rest of the circuit on your own. To save the design click on Design > Check and Save. Look in the icfb for error messages. If there were errors, there would also be markers blinking on the schematic window. Try to correct the errors and ask your TA if you need help.
- We next create a symbol view of the schematic. Choose Design > Create Cellview > From Cellview.

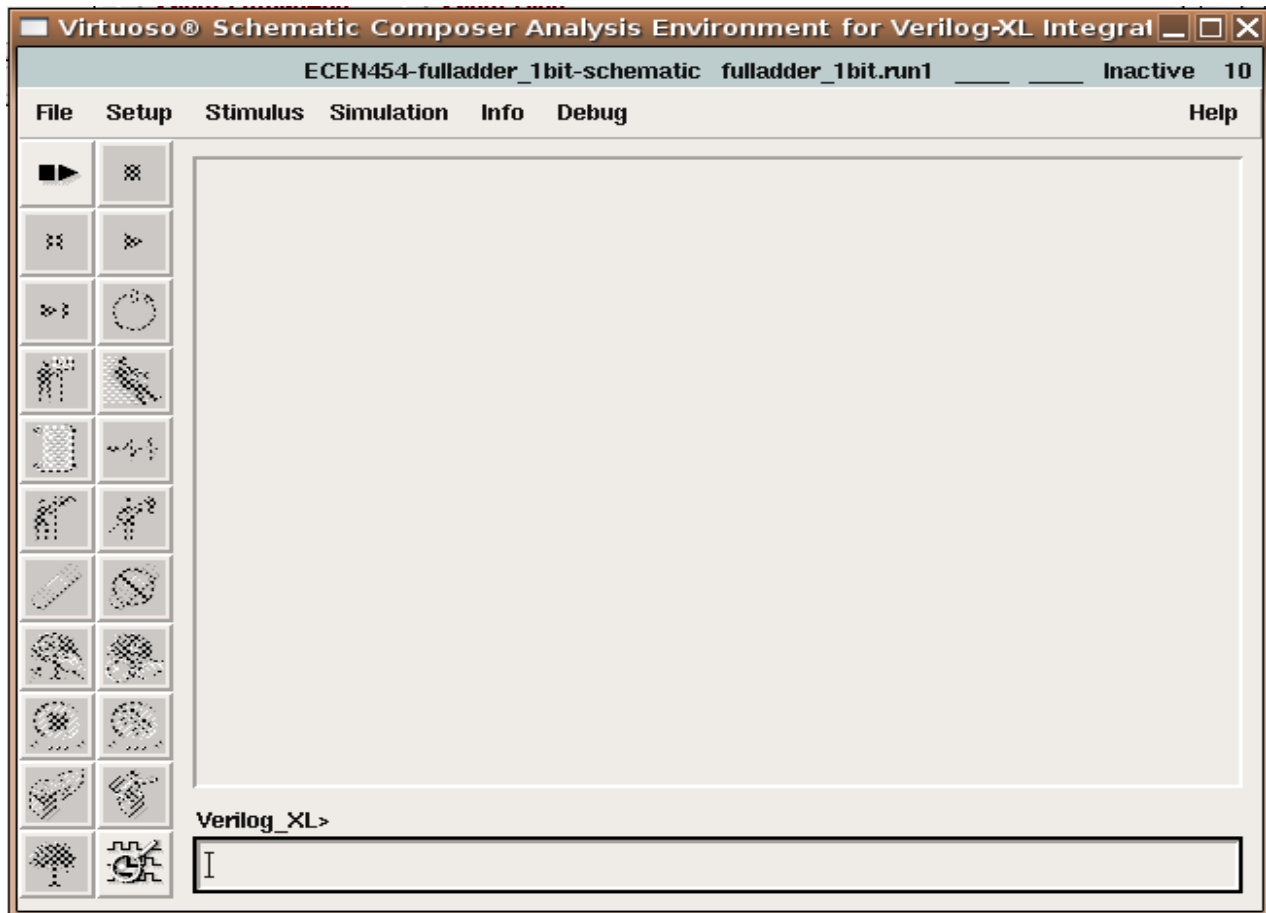


- This creates a symbol view of the fulladder in the Library. This can be observed in the Library manager. Observe Design > fulladder > symbol. This symbol consists of the fulladder block along with ports standing for the pins that were used in the design (in this case them being A, B, C as inputs and SUM, Carry as outputs).

### 3. Simulation

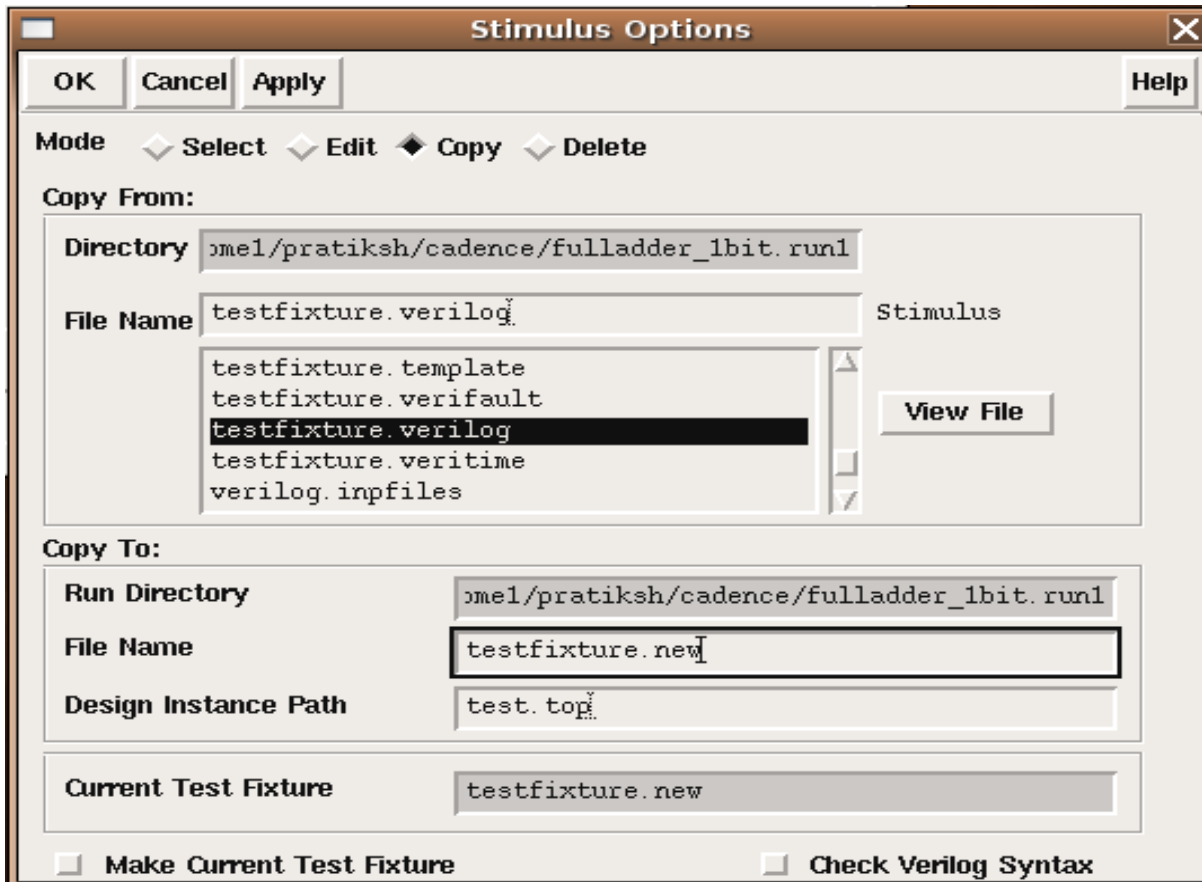
- To make sure the schematic we just completed works properly, we will simulate it using Verilog-XL logic simulator. Note that this is just the logic simulation of schematic and it does not give out accurate timing information (in fact, assuming zero gate delay model here).
- To invoke the Verilog-XL simulator select Tools → Simulation → Verilog-XL. The Setup environment form appears. Make sure the Run Directory (it will be fulladder.run1 or xxxx.run1 if you give different name of the cell), Library, View and Cell fields are filled correctly. Click Ok and Verilog-XL Integration Control window appears as follows.

# ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation



- We first need to create a test fixture where we specify the test vectors, which Verilog-XL uses to simulate. We will use all the eight possible input combinations as test vectors to validate your design. Note that in real design; generally we cannot afford the time and memory to do test on all input combinations. To open a new test fixture, select Stimulus→Verilog in the Integration Control window. It will give you warning message “There is no test fixture in ...” Click Yes. And in Stimulus Options window, choose Copy in the mode field. Choose testfixture.verilog in the Copy from File Name field and testfixture.new in the Copy to File Name field. **(Even if you do not find testfixture.verilog among the files to choose from, type out “testfixture.verilog” in the copy to field)**. Thus testfixture.verilog is copied to testfixture.new and click OK. We will modify this copy of the test fixture file. Again choose Stimulus→Verilog in the VIC window and in the Stimulus options form choose Edit as the mode. Choose the file name to be edited testfixture.new. Also choose Check Verilog Syntax at the bottom of the form. Click Ok and an editor window opens up.

## ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation



- The test fixture, which appears is the default test fixture in Verilog Hardware Description Language. A Hardware Description language (HDL) is used to describe a hardware circuit. At present we will just see very basics of Verilog HDL and study it in more detail in other labs.
- The default test fixture consists of some Verilog code and comments, which start with “//” (as in C++, in fact Verilog is similar to C). Anything following “//” is commented out. Next is an initial statement. This tells Verilog to perform the following task at start up. A begin and end tell Verilog to perform the following lines in the initial block.
- The three statements tell Verilog to set the three inputs to zero. The 1'b0 means “one bit, binary zero”. A 3-bit representation of decimal four would be 3'b100 or 3'd4 (d denotes decimal). Similarly hexadecimal can be specified as “h”. Note that since no timing information is associated with the assignment statements, all the three statements are executed in parallel. That is, all three bits are set to zero at time t = 0.
- We will add timing information right now. Here it works like a clock. Since we do not extract timing information from the circuit and this is still just logic simulation, we would feed different input signals at different time that can be seen as delay of input. The Verilog uses #time directive for adding delay, which means “halt time nsec and

# ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation

---

then proceed”. The second test vector could then be #50 C = 1'b1; //ABC = 001. Note that we are just changing the value of C, as A and B remain the same. Similarly add all the eight possible test vectors for the three inputs. You can add comments for each statement also to make it clear. When two statements are to be executed in parallel then use the same #time directive for two assignments. For example to produce vector 011 at 50 ns (if A = 0, B = 0 and C = 0 at 0ns) use #50 B = 1'b1; C = 1'b1. Save the test fixture and exit the text editor.

- Again choose Stimulus→Verilog in the VIC window and in the Stimulus options form choose Select as the mode. Choose the file name to be testfixture.new. After specifying the test fixture, run the simulation. For this click on the square and triangle bar you see on the Vertigo-XL Integration Control window just below the File menu. It will read "Start Interactive". This will create a netlist of your schematic, which tells Cadence the connectivity of the design. After some time the Verilog-XL window reads:

```
Highest level modules:  
Test Type? For help C1 >
```

If there are any errors inform your TA. To start the actual simulation, click on the box with triangle, on which a tag “continue” appears when you place your mouse on it. Later on you will see the similar message on the Verilog-XL window:

```
0 simulation events + 20 accelerated events CPU time: 0.2 secs to compile + 0.1 secs to  
link + 0.0 secs in simulation
```

```
End of VERILOG-XL 3.11.s007, Jan, 23, 2004, 12:18:38. If there is any error you will  
have to change the test fixture or the design and run the simulation again.
```

Ask your TA if you are stuck.

- If you see some problem and want to clear the simulation runs, you can go to File→Clean Current Run. Before clicking OK to clean current run, you may want to copy the testfixture file you created in some other directory so that you do not have to write it again. Once you clear the current run, you might want to try simulation again.
- There are two ways to observe the simulation results. One is to observe the simulation results in text format. In order to monitor the input and output signals, you have to add several lines at the end of your test fixture file as the following:

**initial**

```
$monitor ($time," A=%b, B=%b, C=%b, SUM=%b, CARRY=%b", A, B, C,  
SUM, CARRY);
```

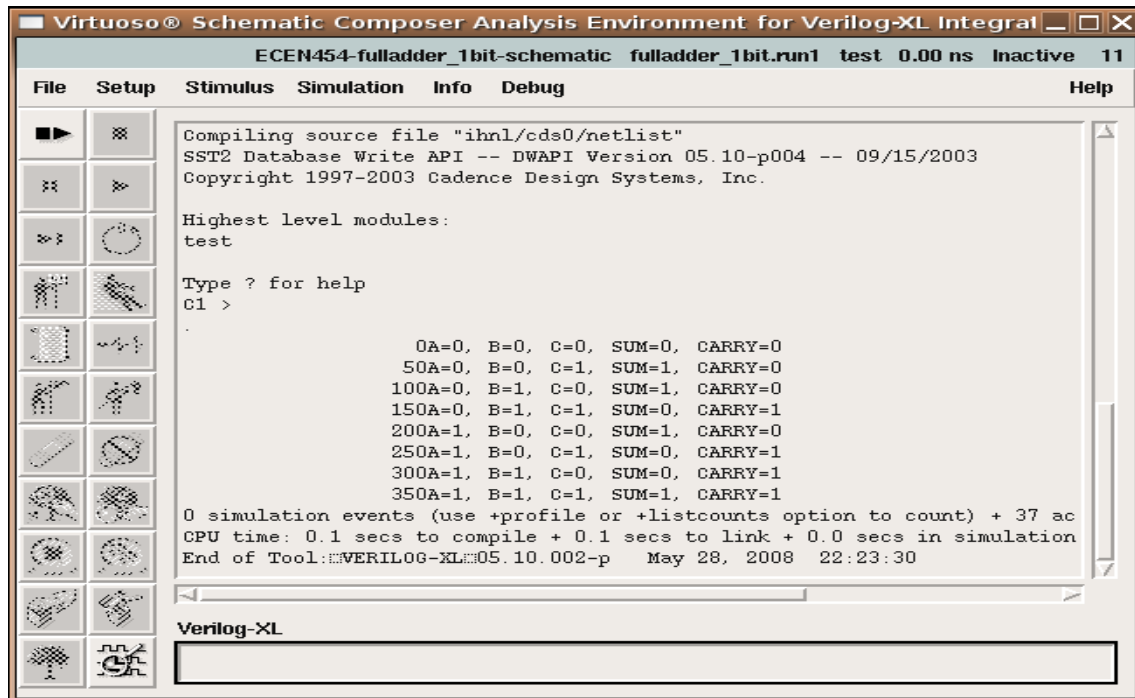
Make sure the signal name is exactly you used in the schematic. Run simulation again and you will see the simulation results.

If you want graphical results, run “simvision &” in the command line in the terminal. In

Open file menu, choose \*.trn

file in cadence/fulladder.run1/RunObject.0/shmDir/shm.db.

# ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation



## 4. Complete logic Simulation

The above sections contained instructions on how to capture a schematic and perform logic simulation. Now we have to follow these instructions to complete the entire design (keep figure 1 of “Lab 0” in mind...that is what we have to build and simulate to verify) of the pipelined adder. Make sure you name the schematics and other files you use appropriately so that you may look back anytime.

- The next step would be to design the 4-bit adder. For this, we use a new schematic window. Open a new schematic (preferably in the same library to maintain hierarchy) according to the procedures described in the above sections. We can now add instances of the “fulladder” by choosing the “Design” library in the component browser where we can select the “fulladder” as the component to be inserted. We can see that the symbol of the “fulladder” that was created previously appears as the component with 3 input pins (A,B,Cin) and 2 output pins(Sum, Carry).
- Connect 4 such full adders in series to form the 4-bit adders. This forms the carry-ripple adder. After completion of wiring, generate a symbol for the schematic since it will further be used to obtain the 8-bit pipelined adder.  
Perform stimulation for this circuit to verify its operation. It helps to check each block individually since we will be using it in a larger design.

Note: Although the adder can be implemented using various other styles such as the carry-look ahead

# ECEN 454 – Lab1: Introduction to Cadence Schematic Capture & Simulation

---

adder, carry-select adder which can compute the results in a quicker fashion, we will still use the carry-ripple adder to maintain simplicity in the basic design and consistency among students.

- After completing the 4-bit adder, we design another building block, the “4-bit register”. The register is basically a 4-bit memory element and we would use 4-FF's to build it. We provide its inputs from 4 pins and tap its outputs through 4-other pins (be careful of the direction of signal when adding a pin). Apart from these we also provide a clock and `_clock` (`_clock` stands for inverted clock signal) input to the Flip-Flops.

Create symbol for the circuit after wiring and simulate the circuit to verify its operation.

- Now, after completing all the individual building blocks to design the 8-bit pipelined adder using these building blocks in the same manner. Use the 4-bit adder, 4-bit register that you have designed and FF's to complete the design as shown in Fig 1 from Lab0. Simulate the circuit by giving the appropriate inputs to verify the operation of the circuit (a subset of test cases is enough). (The output will appear at the immediate rising clock edge following the input).

## 5. Lab Report

1. Take a plot of all the schematic designs and print out the respective simulation results. **Clearly mention the schematic name on the plot. The simulation must display the inputs and the corresponding outputs along with the time at which these signals are monitored. Points will be taken off if the results are not presented clearly.** For schematic plot choose Design → Plot → Submit in the schematic Composer window. In the Submit Plot form click Ok. It should print at ipszac. You can change the plot options if you like. For simulation results choose File → View Log File → Simulation, the simulation outputs will show up. Print out this log file.
2. Report is due by the next lab (if your lab is on Monday, then your due date is next Monday). No late submission.

## 6. Cadence Tips

1. DO NOT KILL cadence process if the machine hangs. This generates view locks, which won't allow you to open the views in edit mode. Inform your TA if cadence has crashed and your standard locks would be cleared.
2. Always SAVE your work. Whenever working continuously, do not wait till end to save. If cadence crashes your work will be lost.
3. Please make sure you log off using “exit” when you leave since other students cannot access the machine if you do not log off and leave.