SAN JOSÉ STATE UNIVERSITY
Charles W. Davidson College of Engineering
DEPARTMENT OF ELECTRICAL ENGINEERING

EE 271 **Final Design Project** **Fall 2015**

Implement and Analyze Different Types of 64-bit Adders

The purpose of this project is to implement, optimize and analyze performance of 4 different implementations of 64-bit adders: (1) Ripple-Carry-Adder (RCA), (2) Carry-Look-Ahead (CLA-2L) with 4-bit groups and 2-level P-G generator, (3) Carry Selected Adder (CSA-EQG) with 4-bit groups and (4) Carry Selected Adder (CSA-UEQG) with unequal-bit groups for highest speedup. Each adder will take 1 clock to complete the calculation. **Analysis will be performed in details for timing and area**. All your work, including the details of your implementation, tests, measurements, analysis and EDA tool setting and configurations must be clearly described as stated in the project report template.

I. Circuit Interface

The **exact** interface ports of all adder circuits must be as below:

- `op1[63:0]`
- `op2[63:0]`
- `sum [63:0]`
- `crout`
- `clock`
- `reset`

Note that for this project, **the analysis is very important and should be documented clearly and neatly.** The report must include exact information as requested. Do NOT skip the information and do NOT report extra information. Grading will be based on the quality and completeness of your project and report.

II. Libraries and EDA Tools

- Use Toshiba library at **`/export/apps/toshiba/sjsu/synopsys/tc240c`**/ for your synthesis and analysis. This is a technology library with 250 nm technology operating at 2.5V. For best and worst delays (hold and setup time violation checks), use **`tc240c.db_BCCOM25`** and **`tc240c.db_WCCOM25`**, respectively. Note that in using Toshiba library, the `tc240c.workview.sdb` is the symbol library that you should use.
- RTL-level simulations can be performed on any simulator but gate-level (netlist) simulations must be performed with either **Synopsys VCS** or **Cadence NCVERILOG**. Synthesis must be performed with **Synopsys Design Vision** (or **Design Compiler**)
- All simulations must be performed at both RTL and gate (netlist) levels

III. Implementation and Testing

The implementation must be done at RTL level in order for you to control the architecture of the adders. You are **NOT** allowed to use Verilog arithmetic operators +, -, *, and / for hardware implementation. You must make sure that the final optimized adders that you implemented are the types of adders that you intended to implement.

Besides testing each module during the project development, you are required to develop final test cases to comprehensively test the operations of the designs and display (by using both `$display` system function and waveforms) the input operands and output results together with timing data to represent the circuit delay performance. The final test cases must be able to perform pre-synthesis (RTL) functional verification and **post-synthesis (netlist) functional and timing** verifications. Select a set of test data such that you can re-verify the static timing (during the synthesis) with dynamic timing (during post-synthesis simulation). Select the data point that you want to output in order for you to exam the dynamic timing delays as function of the input operands.

In order to run VCS with `netlist`*, you need to point to the Verilog source of the target library. As an example of synthesizing with Toshiba library, the VCS command for simulating your netlist named* `adder_tc240c_netlist` *should be as below:*

```
vcs +v2k –debug_all –gui –y
/export/apps/toshiba/sjsu/verilog/tc240c +libext+.tsbvlibp
adder_TB.v adder_tc240c_netlist.v
```

Moreover, you can use Cadence simulator (`ncverilog`) as below:

```
   ncverilog -y /export/apps/toshiba/sjsu/verilog/tc240c
   +libext+.tsbvlibp +access+r adder_TB.v adder_tc240c_netlist.v
```

where `.tsbvlibp` is the suffix of Verilog source files of `tc240c` library and `/export/apps/toshiba/sjsu/verilog/tc240c` is the library directory

Since tc240c library has nanosecond time scale and 10 picosecond time-precision, the timescale directive below should be included in the Verilog testbenches

```
   `timescale 1 ns / 10 ps
```

Note that some Verilog files in this technology library are encrypted (protected) by Cadence EDA tool and VCS may not be able to decrypt them. If that is the case, you will then get error messages. Using `ncverilog` for netlist simulation should be fine since the library was provided to us by Cadence. If anyway you get error message about file protection, try to find out in your netlist which encrypted Verilog files that are used and then try to replace them with non-encrypted files. Below is the list of encrypted files.

- `tsbLD2SFprim.tsbvlibp`
- `tsbSCK2prim.tsbvlibp`
- `tsbMUXXprim.tsbvlibp`
- `tsbSCK1prim.tsbvlibp`
- `tsbCTLprim.tsbvlibp`
- `tsbFD2BASICprim.tsbvlibp`
- `tsbFD4BASICprim.tsbvlibp`
- `tsbCHKprim.tsbvlibp`
- `tsbRCK1prim.tsbvlibp`
- `tsbCLD3SFprim.tsbvlibp`
- `tsbCLD4SFprim.tsbvlibp`

- `tsbMAJprim.tsbvlibp`
- `tsbLDP1prim.tsbvlibp`
- `tsbFD1BASICprim.tsbvlibp`
- `tsbCFD3BASICprim.tsbvlibp`

IV. Design Project Report and Report Submission

EE271 final project is an individual project. Each student will work on the project independently and no information should be shared among students. Each student is required to turn-in a CDROM or memory stick that includes a formal “Final Project Report" (SOFT-COPY) with a required format (a template report is provided) and the whole project materials such as Verilog codes, netlist, simulation waveforms, displays from system functions, circuits and reports from the simulation and synthesis tools. Students are responsible for providing completed information such that grader can re-simulate and re-synthesize the design for grading purposes.

The CDROM or memory stick must have one file named "`Report.xxx`", one file named "`README.xxx`", and one directory named "`Project`". The "`Report.xxx`" file is the final project report in MSWORD or PDF, the "`README.xxx`" file explains steps to simulate and synthesize your project, and the "`Project`" directory is the parent directory that contains files, scripts, outputs, etc. of the project. Files and sub-directories under the parent directory “`Project`” must be organized as they are on your computer account such that grader can just copy the whole `Project` structure to his/her computer account and will be able to simulate and synthesize your project correctly. Please label directly on the CDROM or memory stick your exam seat # and your full name.

The project report must be in the right format and must include completed information about the project as shown in the report template. Please download the report template on the class canvas and edit your report directly from the sample. Do not change the format, fonts, and page setup of the sample report but just fill-in and writing the requested information. Do NOT skip the information and do NOT report extra information. Grading will be based on the quality and completeness of your project and report.

Write the information below directly on your CDROM or memory stick, NOT on the cover or the envelope, so that grader can contact you if there is any problem with your turn-in.

- **EE271 Fall 2015**
- **Your exam seat number and your full name**