ECE-529 FINAL PROJECT

Carry Ripple Adder Implementation using Data Driven Dynamic Logic

Version 1.0
Dr. Erdal Oruklu

Due date: Friday, May 10th, 2013
This is an individual project. You cannot work with a partner.

1. INTRODUCTION

In this project, your objective is to design and implement adder circuits using Data Driven Dynamic Logic (D^3L) and 45 nm technology node. You will use *Cadence IC 6.1* tools for schematic entry, Synopsys *Hspice* for simulation and Synopsys *Nanosim* for power analysis.

We will be using predictive technology model (PTM) files from NCSU. You can find a lot of information at the following link:

http://www.eda.ncsu.edu/wiki/FreePDK45:Contents

There are also tutorials which could be helpful.

2. YOUR ASSIGNMENT

You will design and implement a **4-bit Carry Ripple Adder** circuit**.**

- You will implement adder circuits in two different forms described in equations below:

  1. $S = A \oplus B \oplus C_i$ and $C_{out} = (AB + BC_i + AC_i)$

  2. $P = (A \oplus B);\ G = AB;\quad S = P \oplus C_i;\ C_{out} = G + PC_i$

- You will implement these two adder circuits using two different methods and compare their performance:

  i) **Data Driven Dynamic Logic (D^3L) :** Different from conventional dynamic domino logic, which exploits a clock signal, D3L uses a subset of the input data signals for pre-charging the dynamic node, thus avoiding the clock distribution network. Please refer to [1] for more information on D3L implementation.

ii) **Split Path Data Driven Dynamic Logic (sp-D^{3}L):** sp-D3L minimizes the delay and energy penalties caused by the series-connected PMOS transistors within the PUN of n-type D^{3}L gates. The key idea is to split the PDN of the generic gate into *m* evaluation sub-networks: one for each product subfunction. In this way, the number of series-connected devices in the pre-charging PUN and the gate input capacitances are minimized. Consequently, the delay and the energy dissipation are reduced with respect to the D3L counterpart. Please refer to [1] and [2] for more information on sp-D3L implementation.

Figure 1 illustrates implementation of the generic sum of product (SP) function using both methods:

$$F = \left(I_{1,1} \dots I_{k,1}\right) + \cdots + \left(I_{1,m} \dots I_{n,m}\right) \quad (1)$$

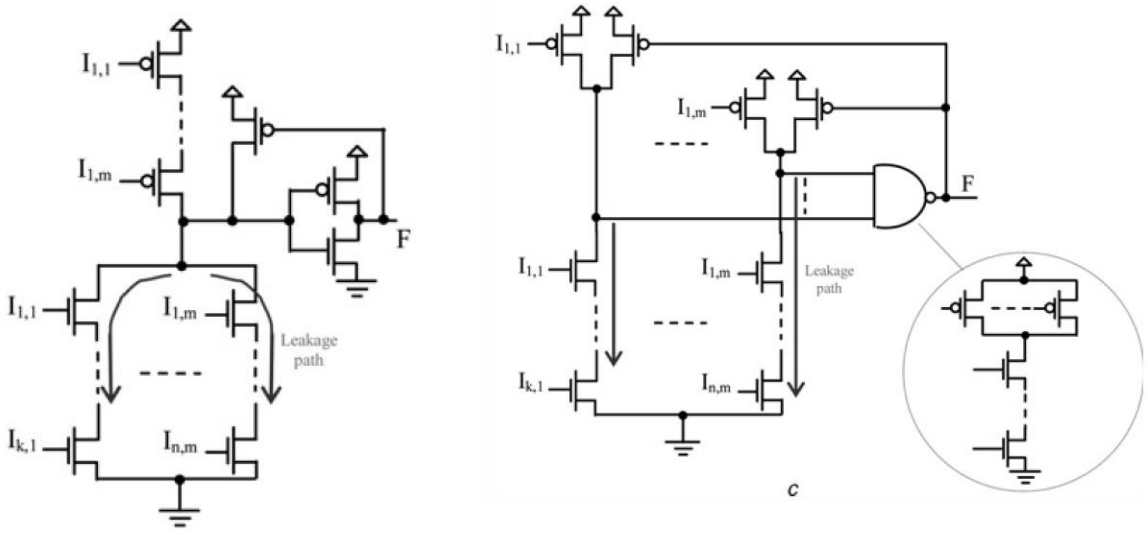

Figure 1. a) D^{3}L implementation of F b) sp-D^{3}L implementation of F

3. DELIVERABLES

Your objective is to compare performance of these implementations with respect to delay and power (dynamic + leakage)

- The following items <u>should</u> be in your report *for **four separate implementations**: (2 adders circuits using 2 different methods)*
  - Circuit schematics
  - Transient analysis waveforms, showing the proper operation of the 4-bit adder circuit
  - Power analysis reports from **Hspice**
  - Power analysis reports from **Nanosim** for leakage currents

- Tables for performance comparison among these four implementations

4. TOOL ENVIRONMENT SETUP AND INITIALIZATION:

In order to initialize the environment for the FreePDK45nm library, Cadence ic6.1, Synopsys Hspice and Nanosim tools, source the following scripts in the given order:

% source /import/scripts/ece429.cshrc
% source /import/scripts/hspice.cshrc
% source /import/scripts/synopsys2012.cshrc

You will also need to source the following script when you create a new folder. (This script copies the necessary library items to your current directory. These are "*cds.lib"* and "*defs.lib").*

% ece429-init-dir

The above step will be only required if you create a new folder and it should be the last script that you source.

5. PROJECT FLOW:

The project has three phases:

- **Phase I:** Schematic entry using Cadence Virtuoso

  You may want to use hierarchical design entry by creating symbols for each gate you have designed. Please refer to ***Hierarchy based schematic design tutorial*** at the blackboard.
- **Phase 2:** Simulation using Analog Design Environment (ADE-L)

  ADE is the main tool for setting up simulation parameters and creating the waveforms for transient analysis. You will have to choose Hspice for the simulation tool and create a link for the technology model files. Please refer to **ECE 429 -** ***Tutorial 1*** at the blackboard.

- **Phase 3:** Power analysis with Hspice and Nanosim

  For power characterization, you will perform two simulations:

  i. Hspice power characterization:

  For Hspice power characterization, you will need to edit the spice netlist file "input.sp" created by the ADE simulation steps.
  You can add the following lines for obtaining power related results:

  .PRINT TRAN POWER
  *This gives total instantaneous power results for each transient point

  .MEASURE TRAN avgpwr AVG POWER FROM 0 TO 100E-12
  * This gives the average total power in the circuit for the specified duration of transient analysis

  .MEASURE TRAN maxpwr MAX POWER FROM 0 TO 100E-12

*This gives the maximum total power in the circuit from the specified duration of transient analysis

Other examples (especially if you want to see power consumed by a single component):
.MEASURE TRAN avgpwr_for_m0 AVG P(m0) FROM 50E-12 TO 100E-12
This gives the average power dissipated in the transistor (m0) for the specified duration of transient analysis (from 50ps to 100ps).

.MEASURE TRAN avgpwr_for_v0 AVG P(v0) FROM 50E-12 TO 100E-12
This gives the average power drawn from the voltage supply v0 for the specified duration of transient analysis.

IMPORTANT: You can do this for the supply voltage you have selected for Vdd and it should give you approximately the average total power consumed.
Remember, you can use *P(component name)* for specific reports.

ii. Nanosim power characterization
For nanosim, you will use the same Hspice netlist file (***Make sure to remove .MEASURE statements or comment-out them, or nanosim will complain***).

Copy the "power.cfg" file provided to you (blackboard) and place it in the directory where 'input.sp' Hspice netlist is.

% nanosim –nspice ./input.sp –C ./power.cfg –o ./power

The results are displayed and written to *power.log* file.

Nanosim provides detailed breakdown of the total power dissipated. The important items you will need in your report are:
Average supply current
Average capacitive current
Average wasted current
Average static wasted current
Average dynamic wasted current
Wasted current percentage
Average block power (This should be a similar value you get with Hspice AVG POWER statement)

Alternatively, you can also use the *Nanosim GUI* tool. Please refer to Nanosim GUI tutorial provided at the blackboard.

6. OPTIONAL:

This bonus part can add 25 more points to your project score. (Maximum would be 125 points).

Another possible way to implement addition is given below:

$$S = ABC_i + \overline{C_{out}}(A + B + C_i) \quad and \quad C_{out} = (AB + BC_i + AC_i)$$

Implement the adder circuit given above using both D^{3}L and Sp-D^{3}L techniques and compare results with the other implementations.

7. REPORT:

Project reports are due on Friday, May 10, 2013 at midnight. ***Only blackboard submissions are accepted***. You also need to upload your HSPICE netlist files as well. Your project will be graded on how well your design works, the efficiency of your design, the level of detail of your design, improvements or extra features in your design, the quality and correctness of your report, and the thoroughness of your testing methodology.

The report should be limited to 20 pages at most and be legible when printing on letter-size papers. Please use common sense to format your report, e.g. report with small fonts/figures will not be graded. Your codes/screen shots/results can be attached as the appendix which will not count toward the 20-pages limit. However, you should discuss them within the 20-pages limit.

All the writings, results, codes, and screen shots should be by yourself. COPY without proper CITATION, and extensive COPY from other materials including but not limited to project instructions and textbooks, will be treated as PLAGIARISM and called for DISCIPLINARY ACTION. You should clearly separate your contribution from existing works, e.g. to separate your implementation from the implementation that is already given. NEVER share your reports with others. The following sections are recommended as an effective way to organize your writing in your reports.

• **Abstraction**

In less than 100 words, briefly discuss your contributions in the project.

• **Introduction**

Summarize the motivation of the project. Highlight the engineering and design challenges in this project and the methods to overcome these challenges.

• **Background**

Give concise descriptions of the adder circuits and D^{3}L and sp-D^{3}L architectures. Note that you should cite various references properly, e.g. the project instructions and the textbook.

• **Architectural Exploration of Adders**

Discuss the trade-offs among the adders and then motivate your choice of adder architecture and parameters. What are the PUN network transistors in D3L and sp-D3L designs for Sum and Cout? How do you size the transistors (W/L)? Explain your decision.

• **Bonus part**

Discuss your solution to the bonus design problem if you decide to solve it.

• **Functional Validation and verification**

Discuss the approaches taken to validate and verify your designs, e.g. test benches and the waveforms. Justify your claims of correctness with simulation results.

• **Results**

Discuss and explain the differences of the designs in terms of performance and cost (speed and power). Create charts/tables to tabulate the performance results and scaling in terms delay and area when the adder is expanded from 1 to 4 bits. Make sure you reach a conclusion with respect to best adder architecture.

• **Conclusion and Future Work**

Summarize your contributions and discuss possible future works.

• **Appendix**

Verilog code/screen shots/results listing.

• **References**

Good luck and have fun!

REFERENCES

[1] Ramin Rafati Sied Mehdi Fakhraie Kenneth Carless Smith, "A 16-Bit Barrel-Shifter Implemented in Data-Driven Dynamic Logic (D3L)", IEEE Transactions On Circuits And Systems—I: Regular Papers, Vol. 53, No. 10, October 2006.

[2] F. Frustaci M. Lanuzza P. Zicari S. Perri P. Corsonello, "Low-power split-path data-driven dynamic logic", IET Circuits Devices Syst., 2009, Vol. 3, Iss. 6, pp. 303–312