UNIVERSITY OF CALIFORNIA, SANTA BARBARA
Department of Electrical and Computer Engineering
ECE 122A—Fall 2023

Final Project

This project is aimed at the implementation of a high-performance adder using low voltage swing (LVS) technology from Intel. LVS technology is used in Pentium 4 processors for the 90-nm node. In this project you will get to know some of the difficulties that engineers face in the design of high-performance processors.

LVS technology has been introduced to take advantage of the 90-nm technology and overcome some of the limitations imposed by aggressive scaling. To limit power consumption, voltage-supply level is decreasing rapidly, which limits the performance of various circuit styles. In the past, Dynamic logic-based circuits were used to gain performance with low voltage supplies, but as technology scales, leakage current of the transistors strongly degrades the robustness of these gates and does not allow further performance improvement. Therefore, LVS technology is employed to conquer this problem.

The reference paper can be found on the course website. The project has four parts, each with a checkpoint (report or interview).

Part 1/4 (Report 1 Complete by 12/01 Friday)

Task 1
Study papers from Intel on LVS technology (uploaded on the course web site). Prepare a two-page report comparing the strengths and limitations of this technology with respect to other digital circuit styles that you have studied in the course.

Task 2
Prepare a brief description of how LVS-based circuits work. Explain the main components (for example, DCNs and CDLs) of an LVS-based circuit and their interactions. You should be able to draw a timing diagram for an LVS-based circuit and explain how the clock signal triggers different events (such as, pre-charge and evaluation) on different parts of circuit.
DCN: PKG, XOR and CARRY SKIP
CDL: PROPAGATE, GENERATE and KILL

Checklist:
□ Two-page review;
□ Descriptions of DCN and CDL cells with timing diagrams.

Part 2/4 (Report 2 Due on 12/05 Tuesday)

Task 3
Build the DCN and CDL cells in SUE. Extract HSPICE netlists, and then use transient simulation to evaluate the circuits. Your stimuli signals should include all the combinations of inputs. (use the 90nm technology and choose V_{DD}=1.2V)

Checklist:
□ SUE schematics for all the DCN and CDL cells;
□ Waveforms that show your cells working correctly.

Part 3/4 (Report 3 Due on 12/09 Saturday)

Task 4
Build the sense amplifier (SA) cell in SUE. Extract HSPICE netlists, and then use transient simulation to evaluate the circuits. Your stimuli signals should include an enable signal, a sinusoidal input signal and its reversed signal.

Task 5
Build the bitslice cell in SUE. Extract HSPICE netlists, and then use transient simulation to evaluate the circuits. Your stimuli signals should include all the combinations of A, B and Cin, while outputs are Sum and Cout.

Checklist:
□ SUE schematics for the sense amplifier you designed;
□ Waveforms that showing your SA working correctly;
□ SUE schematics for the bitslice you build;
□ Waveforms that show your bitslice working correctly.

Part 4/4 (Final Report – Due by 12/12 Tuesday by 11:59 PM)

Task 6
Using SUE, generate the HSPICE netlist for the 16-bit LVS-based adder, which is discussed in the referred papers. Use the 90-nm technology models from the BSIM website. Refer to the Appendix if you do not know the structure clearly.

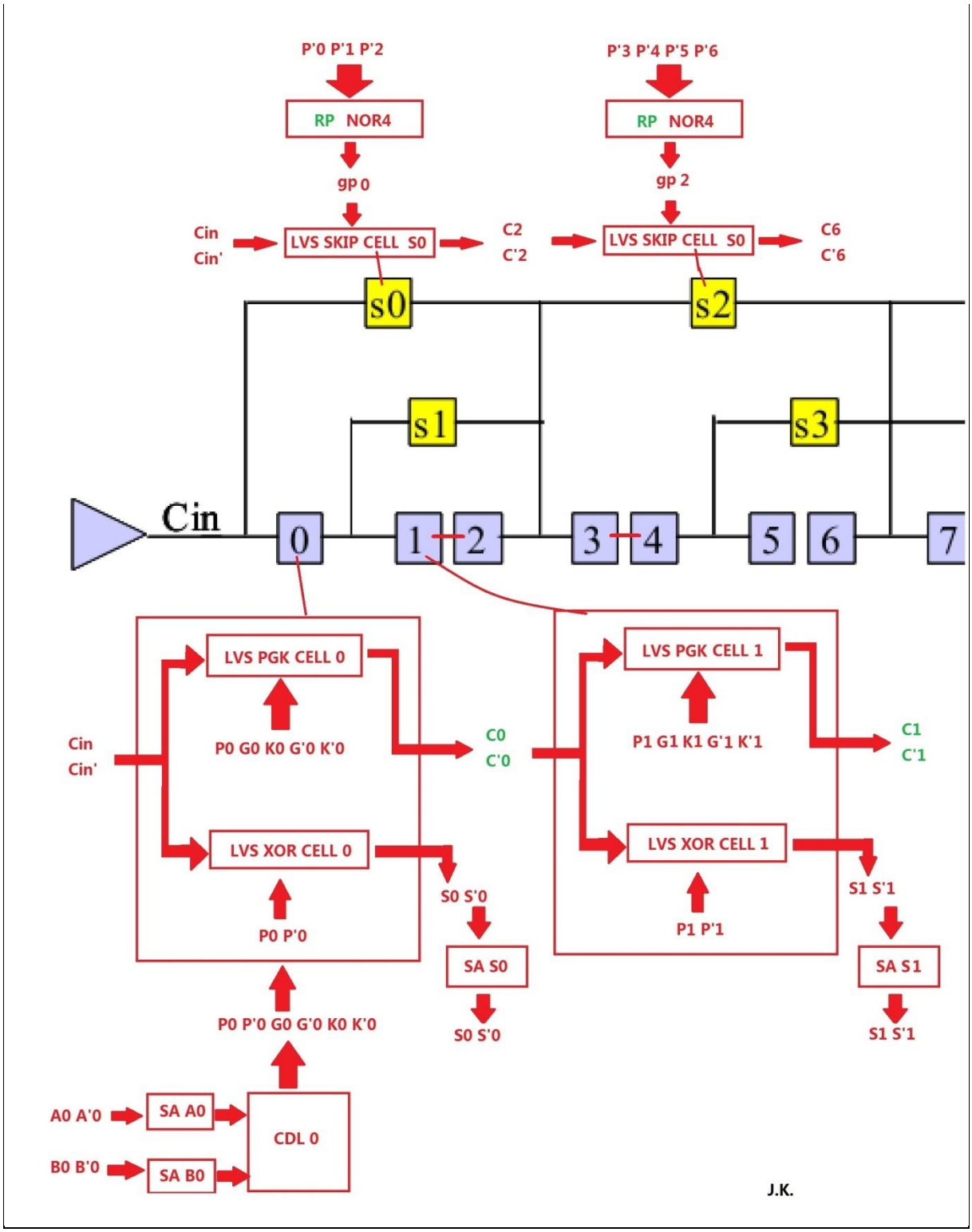
Task 7
Simulate your circuit for various (at least five) input patterns to make sure that its functionality is correct. Run simulations to capture the best and worst-case delays of your adder circuit. Optimize your circuit to decrease the worst-case delay for higher performance.

□ Each student in a group will be orally examined by TA individually. Questions could be anything about comprehension of this project.

(Final Report checklist)

Checklist:
□ Everything in Report 1, 2 and 3; all the problems should be fixed at this time;
□ SUE schematics for the 16-bit adder you build;
□ Waveforms showing your 16-bit adder working correctly;
□ Pointing out the best and the worst cases of inputs; explain why.

Appendix


P'0 P'1 P'2
RP NOR4
gp 0
P'3 P'4 P'5 P'6
RP NOR4
gp 2
Cin
Cin'
LVS SKIP CELL S0
C2
C'2
LVS SKIP CELL S0
C6
C'6
s0
s2
s1
s3
Cin
0
1
2
3
4
5
6
7
LVS PGK CELL 0
P0 G0 K0 G'0 K'0
C0
C'0
LVS XOR CELL 0
P0 P'0
S0 S'0
SA S0
S0 S'0
P0 P'0 G0 G'0 K0 K'0
A0 A'0
SA A0
B0 B'0
SA B0
CDL 0
LVS PGK CELL 1
P1 G1 K1 G'1 K'1
C1
C'1
LVS XOR CELL 1
P1 P'1
S1 S'1
SA S1
S1 S'1
J.K.