بسم الله الرحمن الرحيم

Project Title: Transistor-Level 4-BIT ALU Design

Module: 4-BIT INCREMENT/DECREMENT UNIT Design

- Author: Ahmed Assem Mohamed

May 18, 2025

21 ذو القعدة، 1446

1. Abstraction

This report presents the design and implementation of a **4-bit Increment/Decrement Unit** — a key sub-module in the ALU architecture, implemented at the transistor level using **130nm CMOS technology.** The unit supports incrementing or decrementing a 4-bit signed input using a control signal and produces **4-bit signed output**. The circuit was meticulously developed and simulated using Cadence Virtuoso.

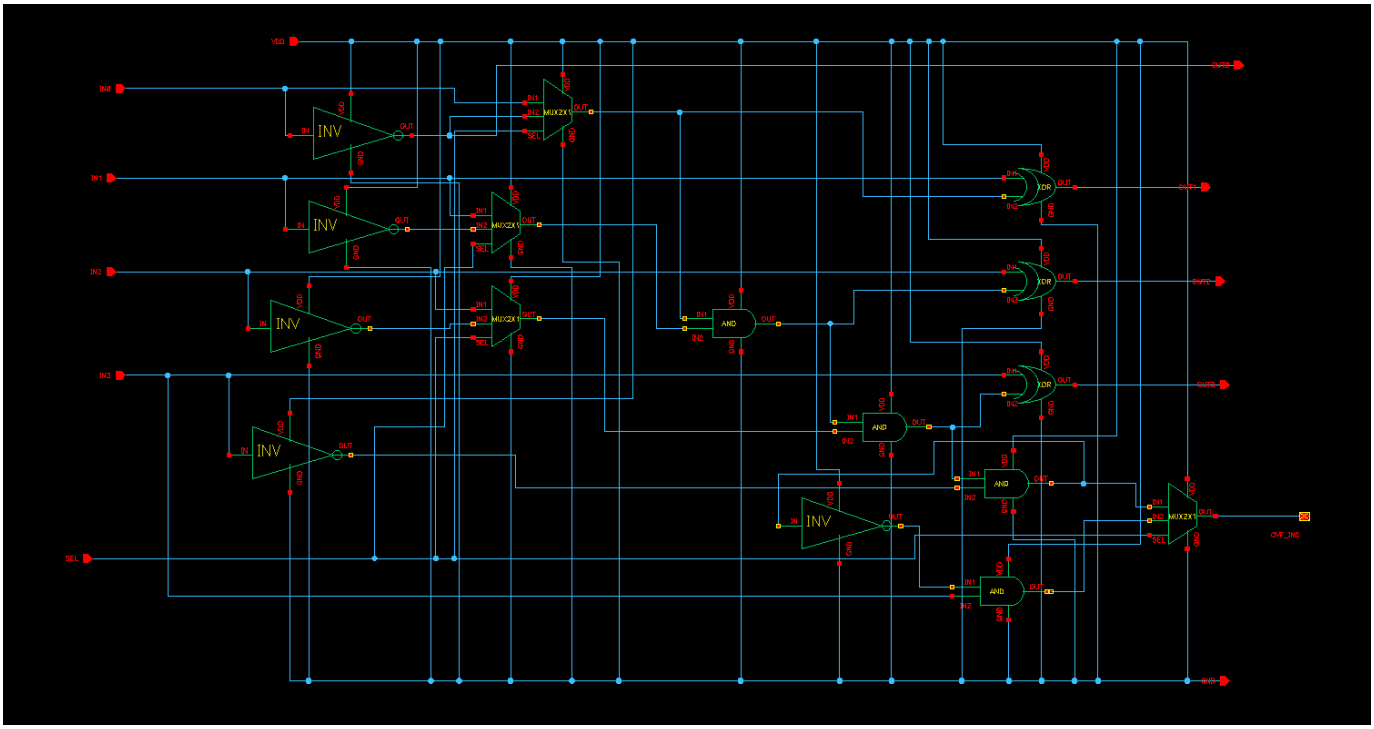
2. Logic Function and Truth Table

OUT[3:0] = A[3:0] + (1 or -1)

A (dec)	A (bin)	SEL	Operation	OUT(dec)	OUT(bin)
3	0011	0	Increment	4	0100
3	0011	1	Decrement	2	0010
-4	1100	0	Increment	-3	1101
-4	1100	1	Decrement	-5	1110
7	**0111**	0	Overflow → -8	Look at the flag	Look at the flag
-8	1000	1	Overflow → 7	Look at the flag	Look at the flag

3. Circuit Design

3.1. Schematic

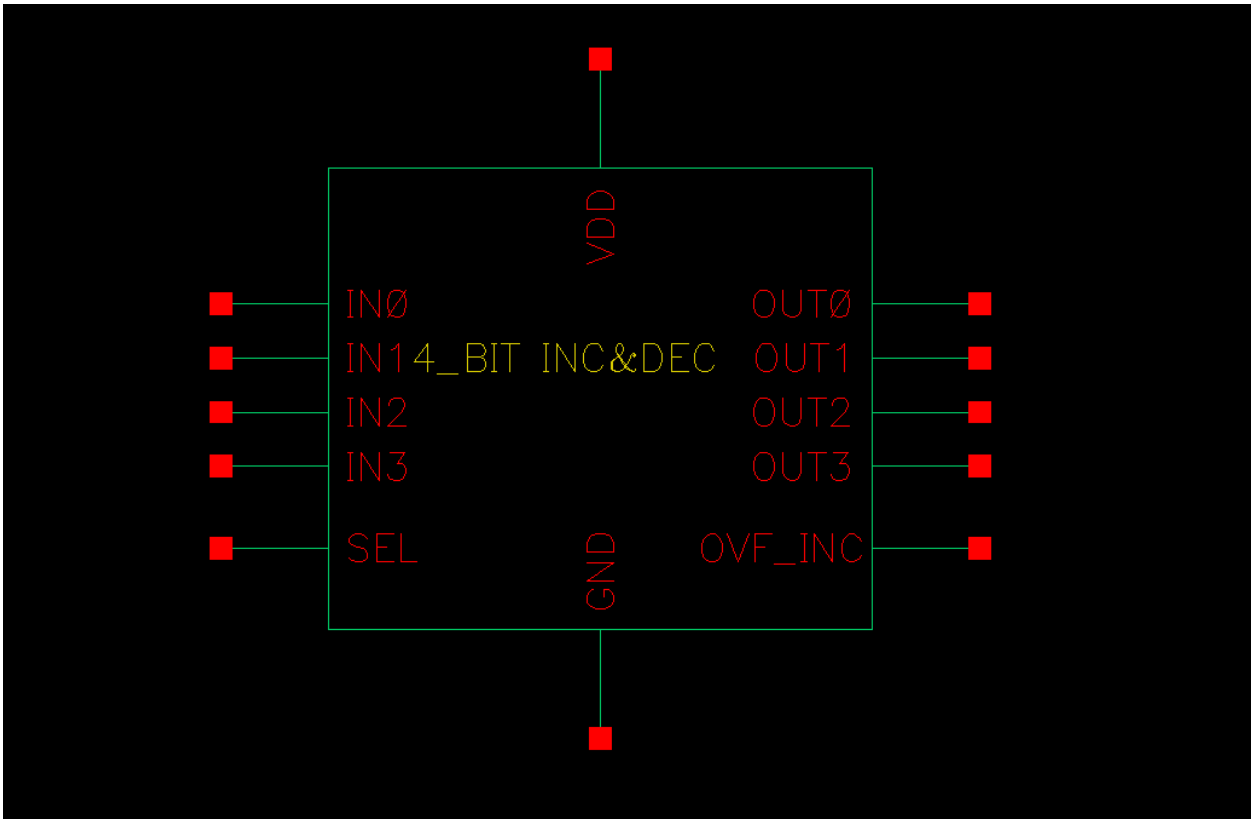


3.2. Design Approach

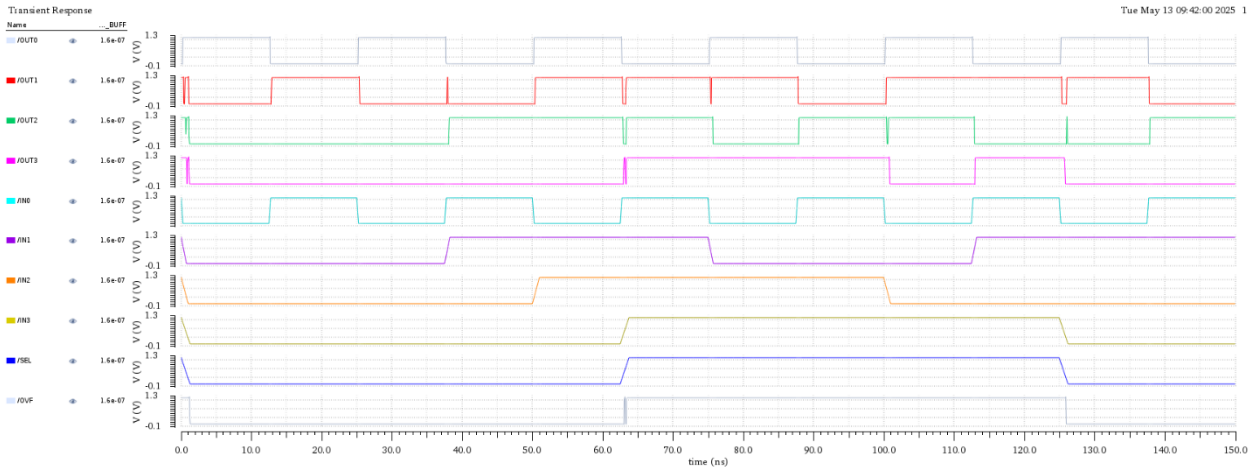
- 130nm CMOS process
- 1.2V VDD
- INPUTS : 4-BIT IN0, 1 SEL
- OUTPUTS: 4-BIT OUT
- Input Parameter: WN_INCREMENT
- Gates used: XOR, AND, INV, MUX-2x1.
- notes: they are based on signed inputs.
- Sizes: XOR: 2* WN_INCREMENT, AND: 4* WN_INCREMENT for the first one in the chain and 2* WN_INCREMENT for the rest, INV: 2* WN_INCREMENT, MUX-2X1: 2* WN_INCREMENT

- why? for XOR to be strong as the inv, but this result is only by intuition and approximated calculation while assuming CL = 15*CREF(INV).
  While for AND as the first and is needed to give the out to the following AND, and the following AND is 2*in
- OVF: is designed on the assumption that the input is signed. the OVF case occurs only when the input is 0111 for INC-CASE, while 1000 for DEC-CASE.
  therefore the OVF flag = IN0 . IN1 . IN2 . IN3' + IN0' . IN1' . IN2' . IN3
- The OVF flag is designed using 1-MUX_2X1, 2-INV, 2-AND
- The design approach is as follows: for SEL = 0 work as INC, while for SEL = 1 work as DEC.

3.3. Symbol



4. Simulation and Results



Note: the simulation result is performed with an input f = 40.0MHZ for A0, 1/2 *f A1, 1/3 * f for A2, 1/4 * f for A3. IN(A3 A2 A1 A0)