P HARI RAM NIKIL

21/9/25

TITLE: 8 BIT ALU

THEORY

This 8-bit ALU circuit uses two 74LS283 4-bit full adders and a bank of 74LS86 XOR gates controlled by an operation select signal to perform both addition and subtraction on two 8-bit binary inputs.
How Addition and Subtraction Work

•Addition: The circuit directly adds the inputs from two DIP switch arrays (representing numbers A and B) as unsigned binary values.

•Subtraction: The subtraction mode is enabled by activating the operation select line, which causes the XOR gates (74LS86) to invert the B input, and sets the carry-in to 1 for the first 74LS283. This implements two's complement subtraction:

- Each bit of B is XORed with the subtract signal, flipping the bits when subtracting.
- Carry-in is set high, adding 1 to form (-B).

TRUTH TABLE

A (bit)	B (bit)	SUB (0=ADD, 1=SUB)	Result (A+B or A-B)	Carry Out
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	0	1
0	1	1	1 (0-1=-1)	0
1	0	1	1	1
1	1	1	0	1

CIRCUIT INFO

•Component Functions:

- •74LS283: Two chips form the 8-bit binary adder/subtractor core.
- •74LS86: Eight XOR gates invert the B input for subtraction.
- •DIP Switches: Provide selectable input values for A and B.
- •LEDs: Display result outputs.
- •SUBTRACT control: Selects between direct addition and two's complement subtraction.

•Operation Summary:

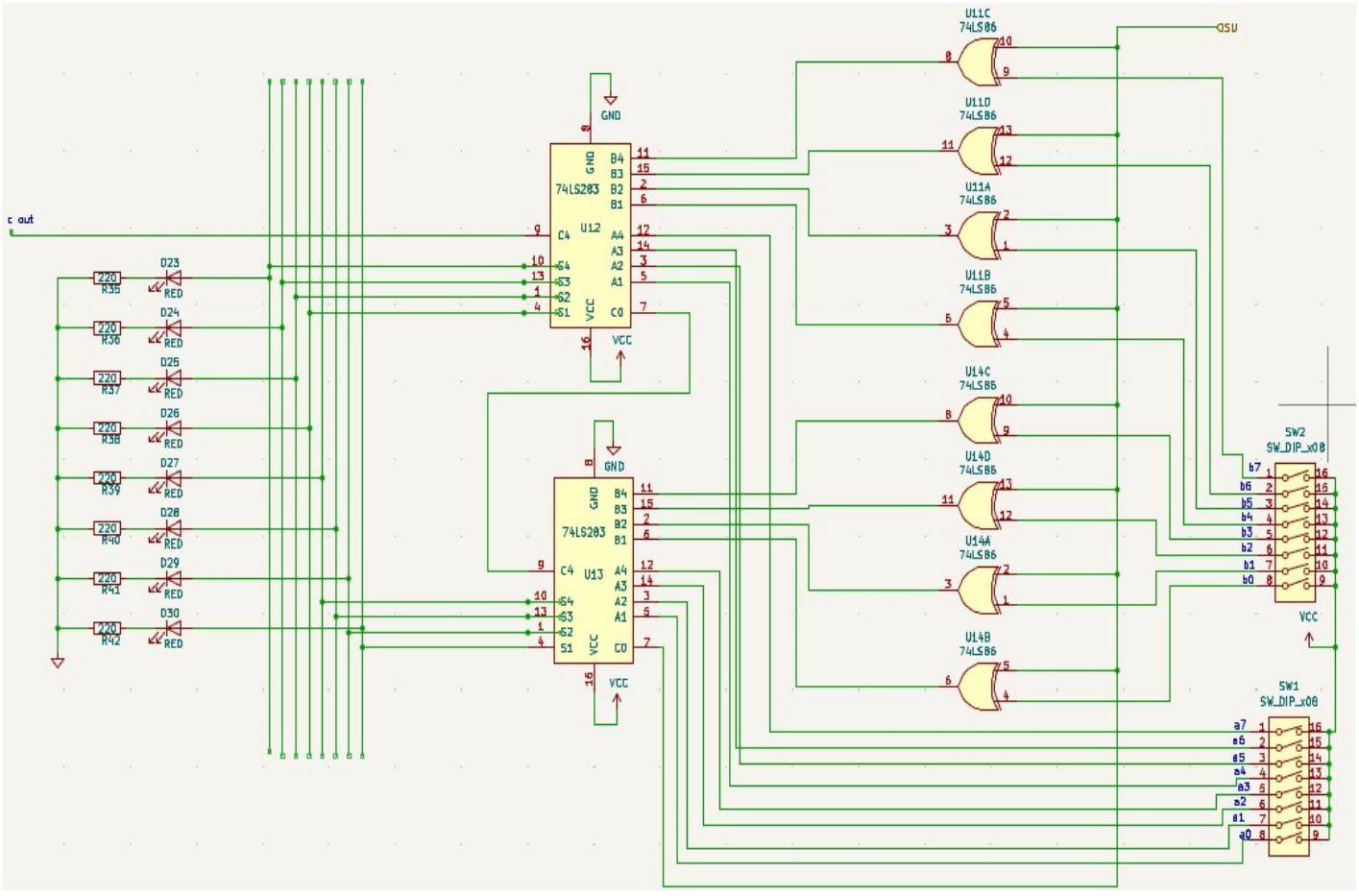
- •Addition: Result = A + B
- •Subtraction: Result = A + (~B) + 1 (where ~B is B inverted bitwise)

•The ALU circuit provides only addition and subtraction, based on the state of the operation select signal.

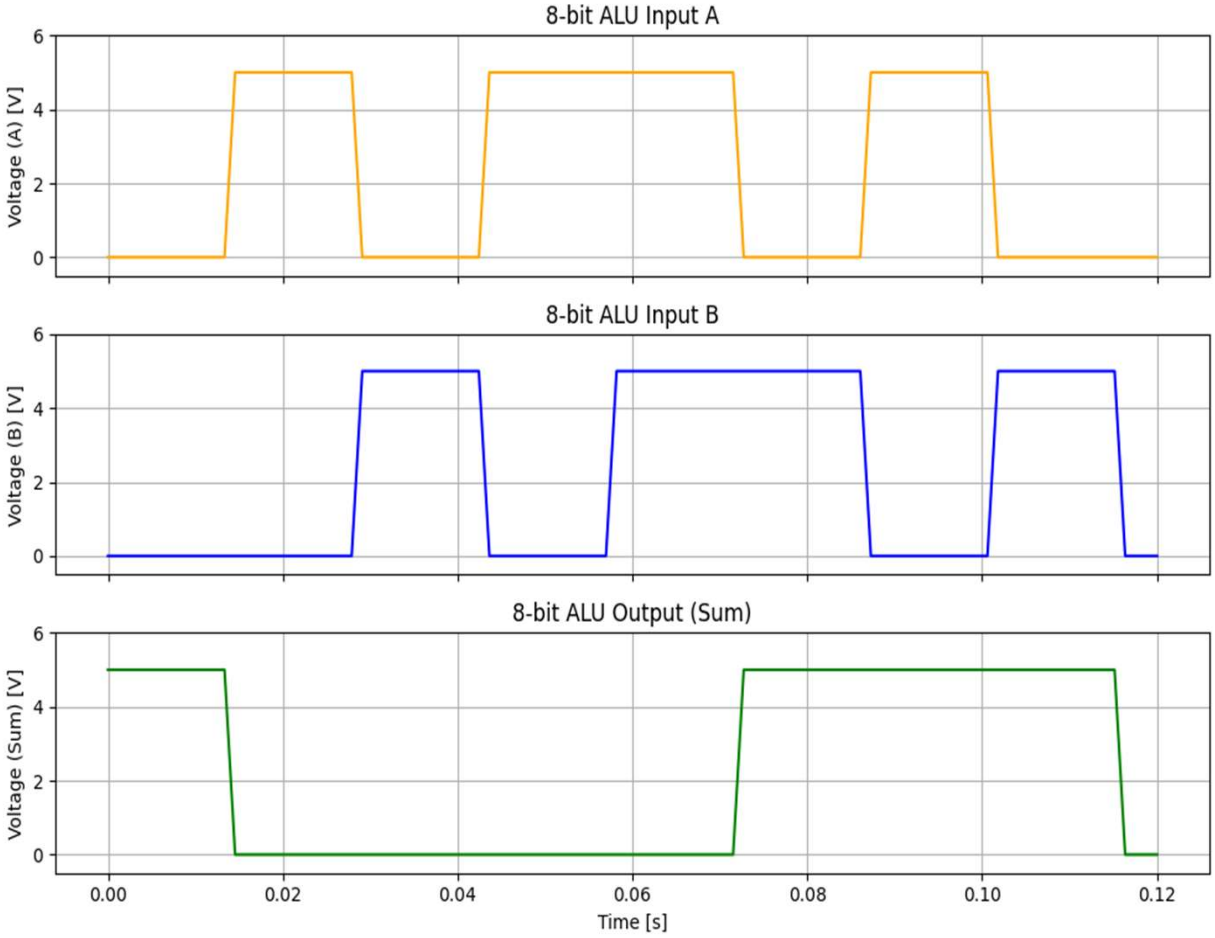
•The truth table explains how outputs depend on input bits and the control signal.

•Subtraction is implemented using two's complement arithmetic common in digital systems.

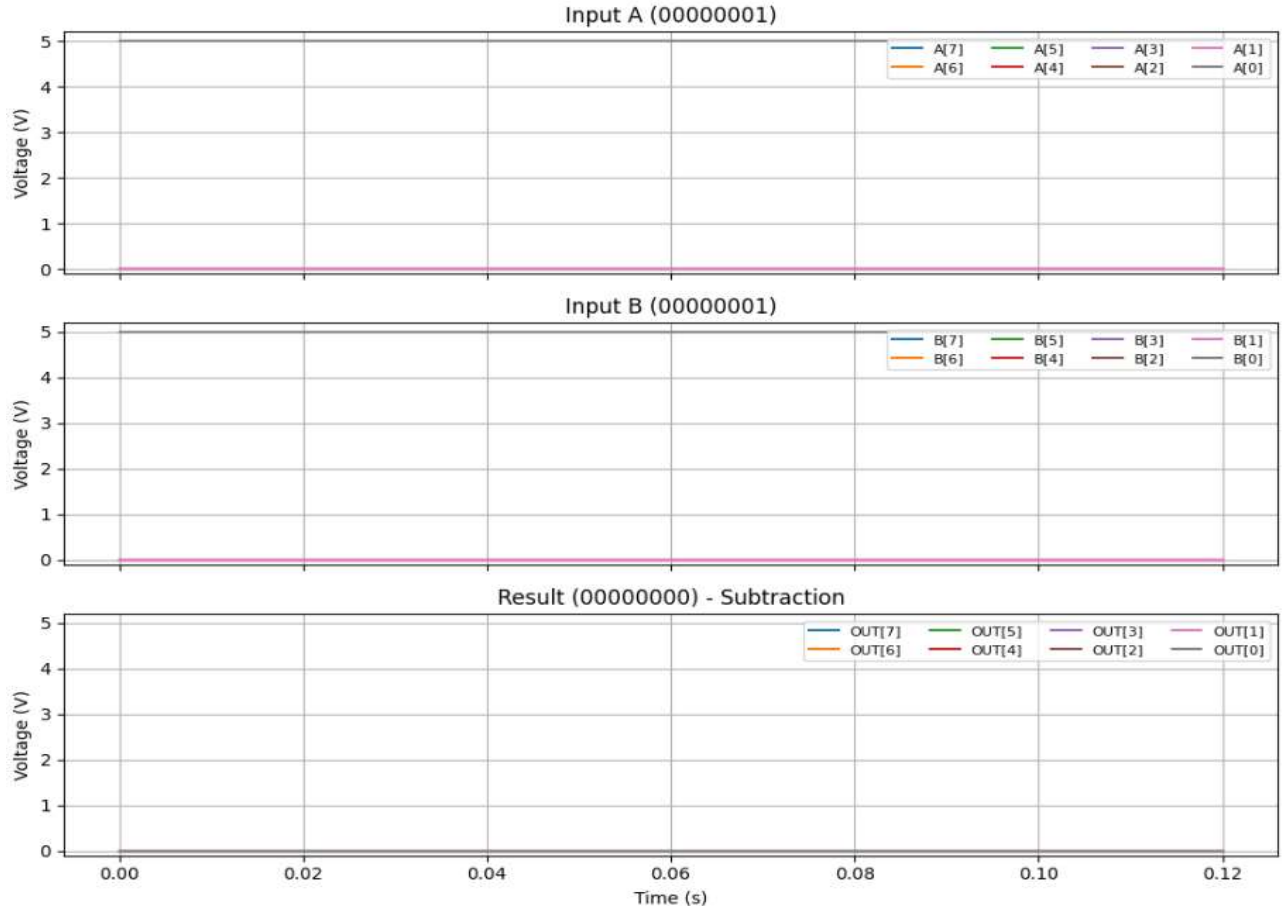
8 BIT ALU SCHEMATIC



ADDER PYTHON SIMULATION



SUBTRACTOR PYTHON SIMULATION



ADDER TRUTH TABLE

A3A2A1A0	B3B2B1B0	C_in	S3S2S1S0 (SUM)	C_out
0000	0000	0	0000	0
0000	0001	0	0001	0
0001	0001	1	0011	0
1111	0001	0	0000	1

SUBTRACTOR TRUTH TABLE

A	B	Bin	D (Difference)	Bout (Borrow Out)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

REFERENCES

- https://www.semanticscholar.org/paper/Design-of-Low-Power-4-bit-ALU-Using-Adiabatic-Logic-Turaga-Vanama/16ab11d6142791a1366e696658491888391285 98?p2df
- https://eater.net/
- https://www.instructables.com/8-bit-ALU-Arithmetic-Logic-Unit/