

32-bit ALU

PARAS SANJAY GIDD, Manipal Institute of Technology

July 13, 2021

Abstract

This paper presents a small block (full adder & full subtractor) of 32-bit ALU implemented on sky130 pdk using eSim. ALU is logic block which is used to do logical, arithmetic, bit-wise, bit-shift & complex operations. It can be considered as core of processor which do all the data processing as required. Depending on the need & design, capabilities of ALU can vary. Modern day Processor consists of ALU which is far more capable than the ALU proposed here. The ALU is designed based on application & need of operations to be performed. The proposed ALU is simple in design but gives a glance on how ALU works & how its built using nothing but simple basic building blocks such as AND, OR, MUX etc.

2 Implemented Circuit

1 Circuit Details

The proposed ALU is capable of 4 operations- addition, subtraction, logical AND, logical OR on 32 bit numbers. here, I have implemented addition & subtraction & with addition of logical AND which is already part of Full adder & logical OR, this block can be called as 1 bit ALU & such 32 blocks are cascaded to form 32 bit ALU of which operation can be controlled by 4:1 MUX. Talking about 32-bit ALU which i studied for this project (reference website in readme file) 32-bit ALU which takes two 32-bit values & is capable of 4 operations – addition, subtraction, bitwise AND, bitwise OR which is selected by 2 control lines & generate one 32-bit output. Along with that, it also has 3 status outputs – zero, carry, overflow. A typical 32-bit ALU is rather much more complex than this but its simplified version of ALU which can be easily implemented using basic logic gates. The 4 logic gates which are used in this are AND, OR, NOT & XOR. Along with that to choose between various operations to perform, multiplier is used which is controlled by control signal connected to its select lines. VLSI runs on 3 main factors – Area, Power & Performance, & therefore, to use area in efficient manner, we are going to use inversion for one of the inputs while performing subtraction which in result allow us to use same adder circuitry for subtraction purpose too.

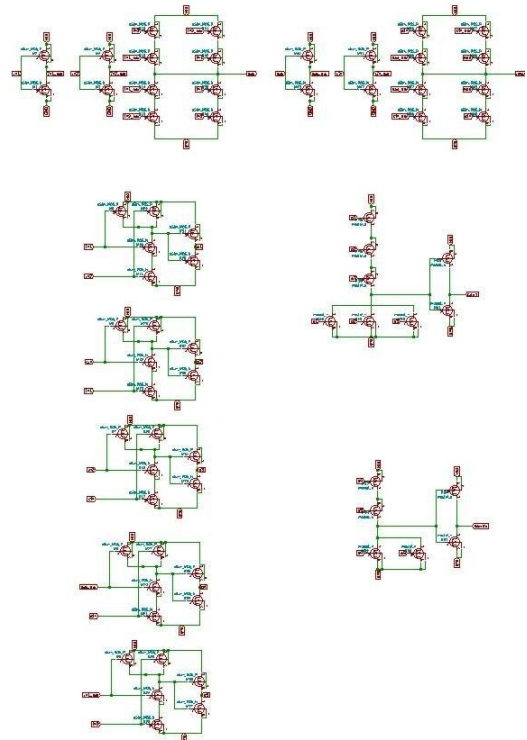


Figure 1: Implemented circuit diagram.

3 Implemented Waveforms

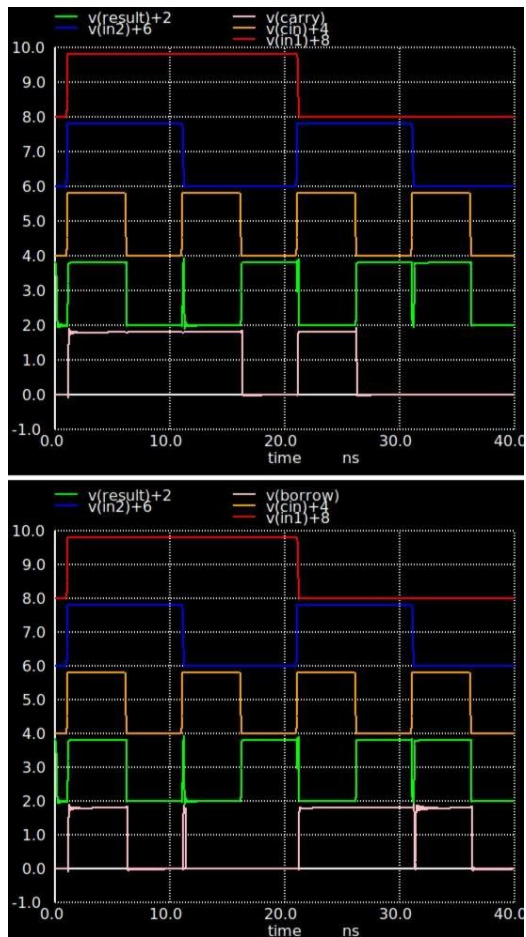


Figure 2: Implemented waveform.

References

- [1] D. M. D. Dr. Warren Toomey. Na. <http://www.csc.villanova.edu/mdamian/Past/csc2400fa13/assign/ALU.html>.
- [2] NA. Na. <http://www.visionics.a.se/Full-Adder1.aspx>.
- [3] C. Organization and D. M. Edition. Na. D. P. J. Hennessy.