1 Experiment No. 10

2 Experiment Title

Design and Analyze a 4 Bit ALU on MICROWIND 3.0

3 Objective

The main objectives of this report are:

- To design and implement a 4-bit Arithmetic Logic Unit (ALU) using nMOS pass transistor logic

4 Theory

A 4-bit Arithmetic Logic Unit (ALU) is a critical component of digital systems, designed to perform fundamental arithmetic and logical operations. In this experiment, the ALU is built using nMOS pass transistor logic to implement various functions, including addition, subtraction, and basic logical operations such as AND, OR, and XOR. The use of nMOS pass transistor logic provides a compact and efficient design approach, making it suitable for integration in digital circuits. The functionality and performance of the ALU are explored in detail during this study.
For any column k of a binary adder, there are three inputs: the corresponding bits of the input numbers A_k and B_k, and the 'previous carry' C_{k-1}. There are two outputs: the sum S_k and the new carry C_k. The truth table for the k-th column of the binary adder is shown in Table 1.

Table 1: Truth table for binary adder

Inputs			**Outputs**	
A_k	B_k	C_{k-1}	S_k	C_k
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

For any column k of a binary adder, there are three inputs: the corresponding bits of the input numbers A_k and B_k, and the 'previous carry' C_{k-1}. There are two outputs: the sum S_k and the new carry C_k. The truth table for the k-th column of the binary adder is shown in Table 1.
Standard adder equations, which fully describe the entries in Table 1, are as follows:

$$S_k = H_k\overline{C}_{k-1} + \overline{H_k}C_{k-1},$$
$$C_k = A_kB_k + H_kC_{k-1},$$

where H_k is the *half-sum*, defined as:

$$H_k = \overline{A_k}B_k + A_k\overline{B_k}.$$

Here, C_{k-1} represents the previous carry, with $0 \leq k \leq n-1$ for n-bit numbers. These equations can be directly implemented using AND-OR functions or, more economically, with EXCLUSIVE-OR gates.

For custom implementations, restating the requirements in another way may be advantageous.

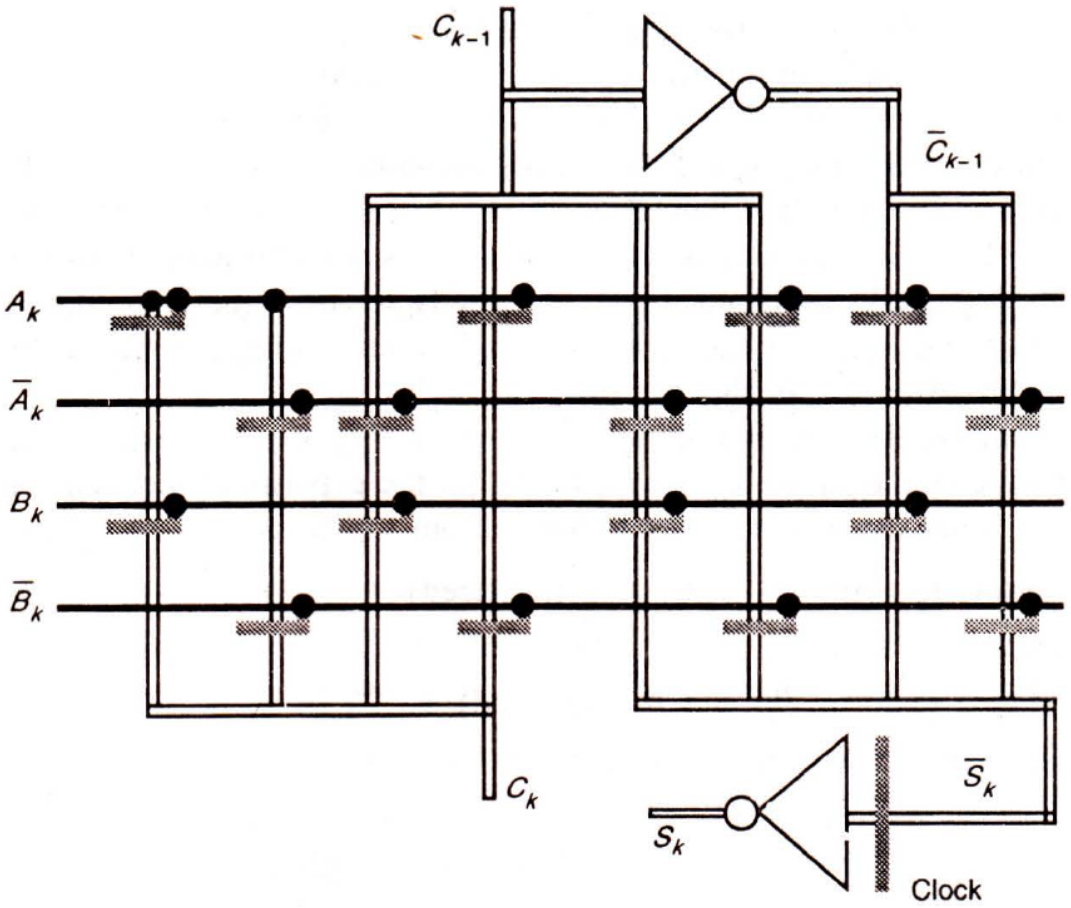

Figure 1: Multiplexer (n-switches)-based adder logic with stored and buffered sum output.

The adder requirements can be stated as:

1. If $A_k = B_k$, then $S_k = C_{k-1}$,
2. Else, $S_k = \overline{C}_{k-1}$.

Similarly, for the carry C_k:

1. If $A_k = B_k$, then $C_k = A_k = B_k$,
2. Else, $C_k = C_{k-1}$.

Alternatively, the carry condition can be summarized as:

1. $C_k = 1$ when $A_k = B_k = 1$,
2. $C_k = 0$ when $A_k = B_k = 0$.

5 Schematic Layout

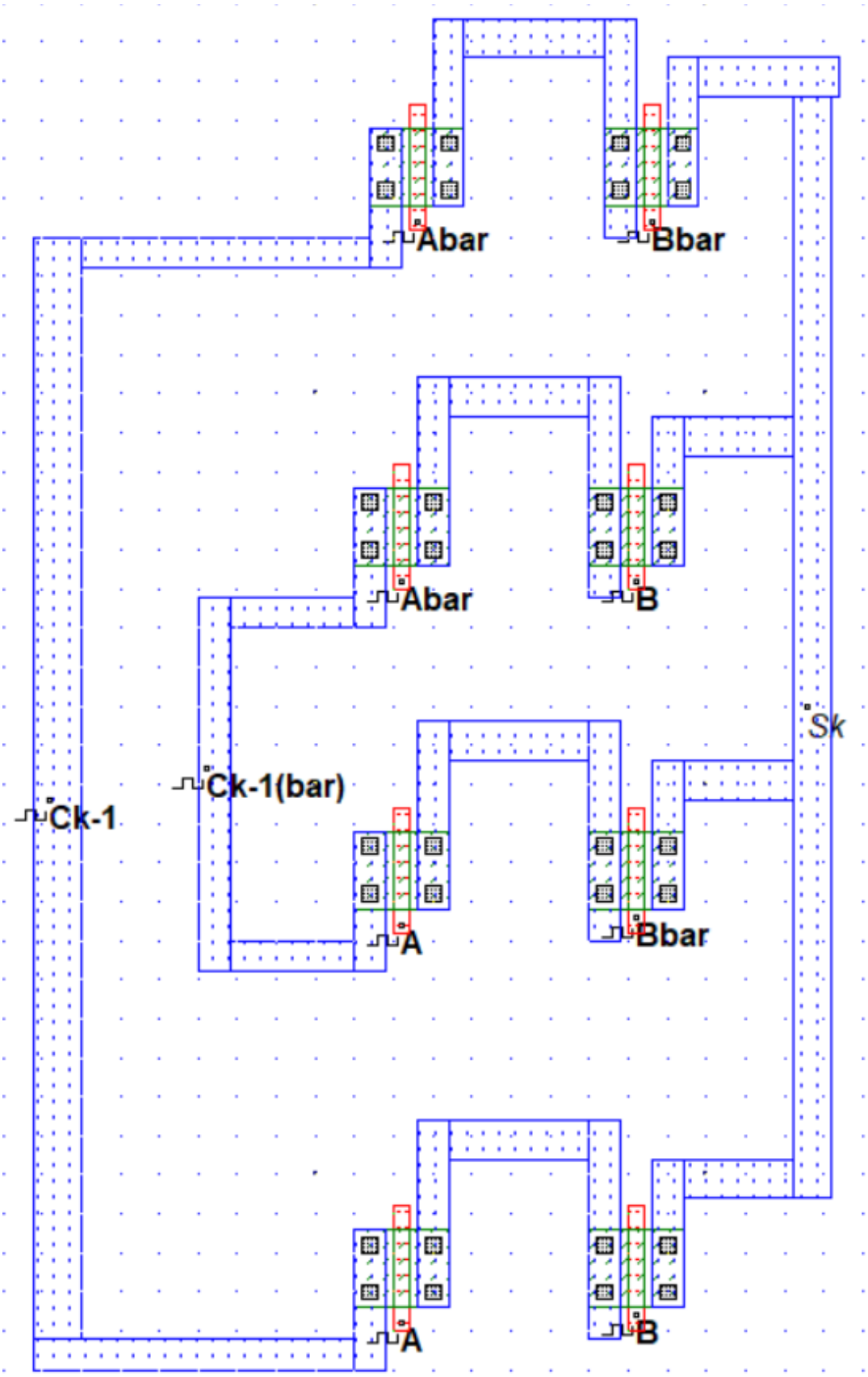

Figure 2: Design of Sum (Sk) for ALU

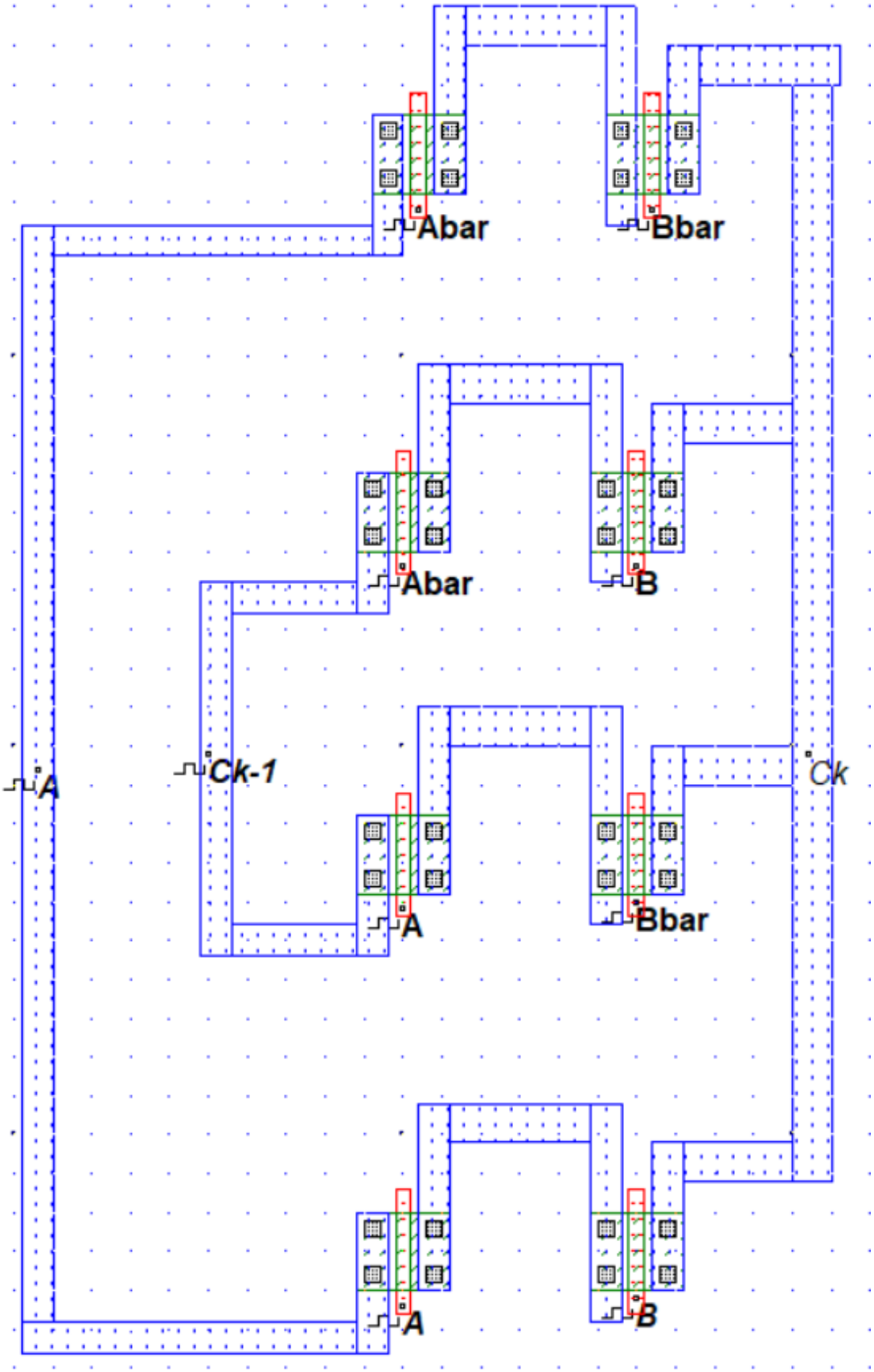

Figure 3: Design of Carry (Ck) for ALU

6 Specification

Table 2: MOSFET Dimensions for nMOS and pMOS Transistors

MOS	**Width** (μm)	**Length** (μm)	**Width** (λ)	**Length** (λ)
nMOS	0.600	0.120	10	2
pMOS	0.600	0.120	10	2

6.1 Specifications of ALU

Table 3: Parameters of Input Clock Signals for A,B Ck-1 & Ck-1bar

Parameter	**Value**	**Unit**
High Level (V)	5.00	V
Low Level (V)	0.00	V
Time Low (tl)	1	ns
Rise Time (tr)	0.001	ns
Time High (th)	1	ns
Fall Time (tf)	0.001	ns

(a) Input clock signal of A

Parameter	**Value**	**Unit**
High Level (V)	0.00	V
Low Level (V)	5.00	V
Time Low (tl)	1	ns
Rise Time (tr)	0.001	ns
Time High (th)	1	ns
Fall Time (tf)	0.001	ns

(b) Input clock signal of Abar

Parameter	**Value**	**Unit**
High Level (V)	5.00	V
Low Level (V)	0.00	V
Time Low (tl)	2	ns
Rise Time (tr)	0.001	ns
Time High (th)	2	ns
Fall Time (tf)	0.001	ns

(c) Input clock signal of B

Parameter	**Value**	**Unit**
High Level (V)	0.00	V
Low Level (V)	5.00	V
Time Low (tl)	2	ns
Rise Time (tr)	0.001	ns
Time High (th)	2	ns
Fall Time (tf)	0.001	ns

(d) Input clock signal of Bbar

Parameter	**Value**	**Unit**
High Level (V)	5.00	V
Low Level (V)	0.00	V
Time Low (tl)	4	ns
Rise Time (tr)	0.001	ns
Time High (th)	4	ns
Fall Time (tf)	0.001	ns

(e) Input clock signal of Ck-1

Parameter	**Value**	**Unit**
High Level (V)	0.00	V
Low Level (V)	5.00	V
Time Low (tl)	4	ns
Rise Time (tr)	0.001	ns
Time High (th)	4	ns
Fall Time (tf)	0.001	ns

(f) Input clock signal of Ck-1bar

Table 4: Parameters for Vdd+ and Vss-

Parameter	**Value**	**Unit**
Vdd+	5.00	V
Vss-	0.00	V

7 Output Waveshape

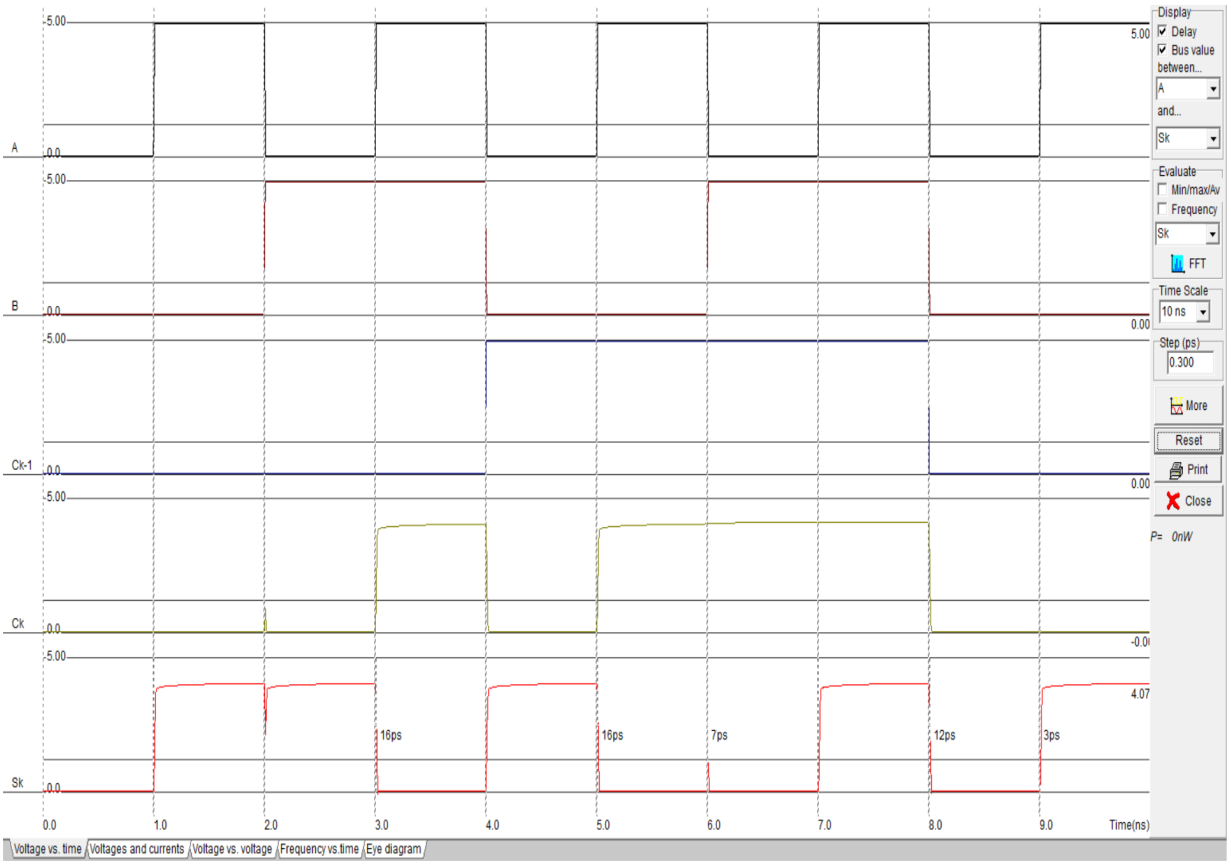

Figure 4: Output Waveshape of 4-Bit ALU

8 Discussion

In this experiment, the design and implementation of a 4-bit Arithmetic Logic Unit (ALU) were carried out using NMOS pass transistor logic. The operation of addition was successfully performed. The ALU's functionality was derived based on standard logic equations for arithmetic operations. For addition, the sum (S_k) and carry (C_k) were calculated using truth tables and equations to ensure correctness for all possible input combinations. The sum (S_k) and carry (C_k) were implemented using nMOS transistors. Specifically, for the sum (S_k), the following logic was applied.If $A_k = B_k$, then $S_k = C_{k-1}$. Otherwise, $S_k = \overline{C}_{k-1}$.Similarly, the carry (C_k) was determined.If $A_k = B_k$, then $C_k = A_k = B_k$. Otherwise, $C_k = C_{k-1}$.
The behavior of the ALU was verified by testing various input conditions.The use of nMOS pass transistor logic resulted in a compact and efficient design. However, some fluctuations were noted during the implementation due to fast and sharp input transitions, which slightly affected the ALU's stability in certain cases. Overall, the experiment demonstrated the feasibility of designing an ALU using nMOS pass transistor logic, providing insights into transistor-level implementation of arithmetic operations.