International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI:10.55041/ISJEM06007
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

3-Input K-Map Function Implementation in VLSI

|Mr.Mashalkar Y.S. |Gavkare Shravan |Nagapure Shivam |Jadhav Shailesh|Jadhav Prasad |

Senior Lecturer, Department of Electronics And Telecommunication

Students, Department of Electronics and Telecommunication Vishweshwarayya Institute Of Engineering & Telecommunication, Almala, India

Abstract-This paper presents the design and implementation of a three-input K-Map function using VLSI technology. Karnaugh Map (K-Map) is a simplification technique used to minimize Boolean expressions. The simplified logic is implemented using basic logic gates and further realized using VLSI design concepts. The objective is to reduce hardware complexity, power consumption, and propagation delay. The implementation is verified using simulation tools and logic design methods.

***Keywords*—**K-Map, VLSI, Boolean Algebra, Logic Gates, Digital Design

I. INTRODUCTION

This paper presents the design and implementation of a three-input K- Map function using VLSI techniques. It provides a systematic approach for simplifying Boolean expressions and implementing them in digital circuits. The design focuses on reducing hardware complexity, power consumption, and improving performance. All standard steps are followed for three main reasons: (1) ease of Boolean simplification, (2) efficient hardware implementation, and
(3) uniformity in digital design. The process includes K-Map construction, expression minimization, and logic circuit realization. The proposed method ensures optimized circuit design and is suitable for VLSI-based digital system applications.

II. LITERATURE REVIEW

1. Early digital design techniques relied on algebraic simplification methods, but graphical approaches like the Veitch diagram and Karnaugh Map improved the efficiency and accuracy of minimizing Boolean functions
2. The Karnaugh Map (K-map) method is widely used for simplifying small-scale Boolean expressions, particularly in 3-variable systems, enabling reduced logic complexity and easier hardware implementation.
3. In CMOS VLSI design, minimized Boolean functions directly contribute to reduced transistor count, lower power consumption, and optimized chip area.
4. Advanced minimization techniques such as the Quine– McCluskey algorithm provide exact solutions but are computationally complex compared to K-map methods for small input functions.

III. NEED FOR 3-INPUT K-MAP FUNCTION IMPLEMENTATION

1. Simplification of Boolean Functions
2. The 3-input Karnaugh Map is essential for simplifying Boolean expressions with three variables, reducing complex.
3. Reduction in Hardware Complexity

Minimization using K-map decreases the number of logic gates required, which directly reduces circuit complexity.

4. Optimization in CMOS VLSI design

Simplified logic functions result in fewer transistors, leading to optimized chip area, lower fabrication cost, and better scalability in VLSI systems.

5. Power Consumption Reduction

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI:10.55041/ISJEM06007
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

By minimizing redundant logic operations, switching activity is reduced, which helps in lowering dynamic power consumption in digital circuits.

6. Improved Speed and Performance

Reduced logic levels lead to smaller propagation delay, thereby enhancing the overall speed and performance of combinational circuits.

IV. SYSTEM ARCHITECTURE OVERVIEW

1. Input Stage (Variable Definition)

The system consists of three input variables (A, B, C), representing the fundamental inputs to the Boolean function. These inputs define the possible combinations for logic evaluation.

2. K-Map Construction and Mapping

A 3-input Karnaugh Map is constructed to represent all possible input combinations ($2^3 = 8$). The given Boolean function is mapped into the K-map using minterms.

3. Logic Simplification Unit

Adjacent cells containing logic ‘1’ are grouped to minimize the Boolean expression. This step reduces redundant terms and simplifies the function for efficient implementation.

4. Gate-Level Design

The simplified Boolean expression is converted into a logic circuit using basic gates such as AND, OR, and NOT. This stage defines the functional behaviour of the circuit.

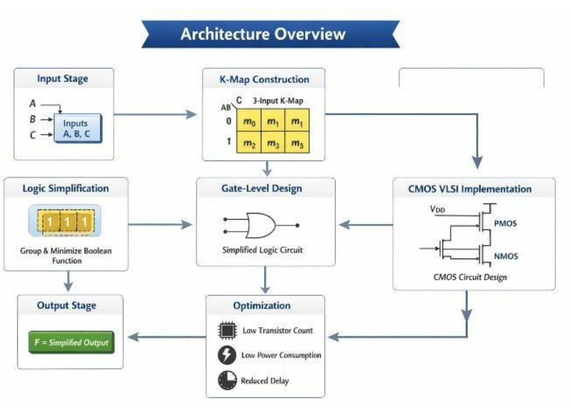


Fig (a) SYSTEM ARCHITECTURE OVERVIEW

V. THREE-INPUT K-MAP OVERVIEW

A 3-variable K-map consists of:

- 3 inputs: A, B, C
- 8 possible combinations ($2^3 = 8$)
- Organized in a 2×4 grid

$$F = C$$

ISSN: 2583-6129
DOI:10.55041/ISJEM06007

VI. IMPLEMENTATION OF 3-INPUT K-MAP FUNCTION IN VLSI

A. Problem Definition

A three-input Boolean function is considered with inputs (A), (B), and (C). The function is implemented using Karnaugh Map (K-map) minimization techniques to achieve an optimized logic expression suitable for CPLD-based realization.

B. K-Map Representation

The 3-variable K-map consists of 8 cells corresponding to minterms (m_0) through (m_7). The map is arranged as:

AB \ C	0	1
00	m0	m1
01	m2	m3
11	m6	m7
10	m4	m5

Based on the given function (example): [F(A,B,C) = \sum m(1,3,5,7)]

B. Minimization Process

Grouping adjacent 1s in the K-map:

- Group 1: (m1, m3 \right arrow A'C)
- Group 2: (m5, m7 \right arrow AC) Final simplified expression:

[F = C(A' + A) = C]

D. VLSI Implementation

The minimized logic is implemented using the programmable logic structure of the ATF1508 CPLD:

1. **Input Assignment**

- Inputs (A, B, C) are mapped to general- purpose I/O pins

2. **Logic Mapping**

- The simplified function is implemented using internal logic microcells

3. **Output Generation**

- Output (F) is assigned to an output pin
- Since the minimized function is (F = C), only a direct connection is required, demonstrating optimal resource usage.

E. Design Advantages

- Reduced gate count
- Lower propagation delay
- Efficient utilization of CPLD resources
- Simplified routing and power consumption

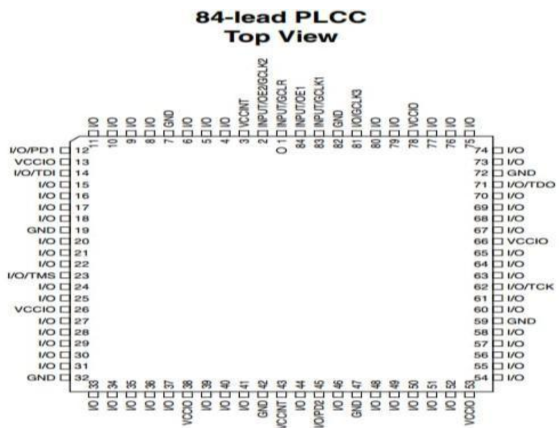
F. Practical Considerations

In real VLSI design:

- Timing constraints must be verified
- Propagation delay should meet system requirements
- Proper pin assignment (as per PLCC layout) ensures signal integrity

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI:10.55041/ISJEM06007
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

VII.ATF1508 – QUICK DATASHEET



1. **Device:** CPLD for Boolean logic, K-map, FSMs

2. **Features:** ~128 microcells, 5–10 ns speed, ISP via JTAG, non-volatile, low power

3. **Power:** 5 V core (VCCINT) & I/O (VCCIO),
multiple GND pins, use decoupling caps

4. **I/O Pins:** Programmable as input/output/bidirectional; connect A, B, C inputs and F output

5. **Clock Pins**: Global clocks (GCLK1-3), optional for simple logic

6. **JTAG:** TDI, TDO, TMS, TCK for programming

7. **Package:** 84-lead PLCC, pin numbering counterclockwise

8. **Architecture:** AND-OR arrays and flip-flops for combinational logic

9. **Software:** Programmed via Win CUPL/ATMISP using Boolean equations or truth tables

VIII. RESULT:

A. Software:

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL; use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all; entity santosh is
Port ( A : in  STD_LOGIC;
B : in  STD_LOGIC;
```

C : in STD_LOGIC; Y : out STD_LOGIC);
end santosh;

architecture Behavioral of santosh is begin

Y <= (not A and not B and C) or (not A and B and C) or
(A and not B and C) or (A and B and C);
end Behavioral;

B. VIEW RTL SCHEMATIC:

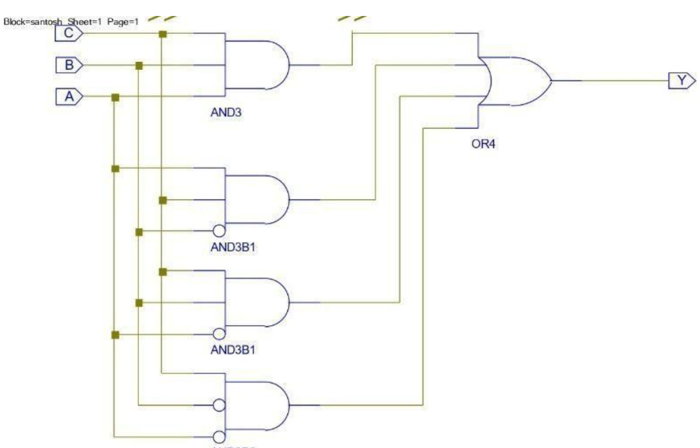
The RTL (Register Transfer Level) Schematic in Xilinx ISE shows a visual representation of the digital design before synthesis, illustrating how inputs, outputs, and internal logic are connected.

In this project, it represents the 3-input logic block with inputs A, B, C and output Y, showing how combinational logic (AND, OR, NOT) produces the correct Boolean function.

Benefits: verification against the K-map, error detection before programming, and clear design visualization.

The RTL schematic bridges the conceptual Boolean function and its implementation on the ATF1508 CPLD, ensuring correct and reliable operation.

C. LOGIC SYNTHESIS



Logic synthesis is the process of converting a Boolean function into an optimized digital circuit using basic logic gates such as AND, OR, and NOT. In this work, a 3-input function is first simplified using a Karnaugh Map (K-map) to obtain a minimal expression. This minimized expression is then implemented using logic gates, where each product term is realized by an AND gate and combined using an OR gate to produce the final output.

IX. FUTURE ENHANCEMENTS:

- Extend to 4-input or 5-input K-Maps for more complex functions
- Implement using advanced CMOS processes (28nm or below)
- Incorporate clocked sequential logic elements
- Explore FPGA prototyping for rapid iteration
- Implementation using HDL (Verilog/VHDL).
- Integration into FPGA-based systems.

X. CONCLUSION

This paper presents the implementation of a three-input K-Map function using VLSI design. The Boolean expression is simplified to achieve an optimized circuit with reduced gates, area, and power consumption. The results confirm efficient and reliable digital design.

XI REFERENCE:

[1] R. L. Ashenhurst, "The Decomposition of Switching Functions," *Proceedings of the International Symposium on the Theory of Switching*, pp. 74–116, 1959. Available: https://doi.org/10.1109/FOCS.1959.7

[2] E. J. McCluskey, "Minimization of Boolean Functions," *Bell System Technical Journal*, vol. 35, no. 5, pp. 1417– 1444, Sep. 1956.
Available: https://doi.org/10.1002/j.1538- 7305.1956.tb03835.x

[3] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, Aug. 1986. Available: https://doi.org/10.1109/TC.1986.1676819

[4] S. Hassoun and T. Sasao, "Logic Synthesis and Verification," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1820–1833, Nov. 2002.

Available: https://doi.org/10.1109/JPROC.2002.804686