IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Boolean equations for Computer Based Interlocking (CBI)

Vandana
CRL Ghaziabad
Bharat Electronics Limited
India

Akhilesh Singh Tomar
CRL Ghaziabad
Bharat Electronics Limited
India

Abstract*— *Boolean equations are used in computer-based interlocking to ensure safe and efficient operation of complex transportation and industrial systems. Boolean equations use boolean algebra to represent and control the interlocking logic within these systems. They define the relationships between various inputs and outputs to make critical decisions for safe system operation. Inputs include sensor data, switch positions, train locations while outputs control signals like track switching and signal indications. By representing the system's behavior as a set of logical equations, robust interlocking rules can be created that prevent conflicting actions and maintains safety. Boolean equations are a tool in computer based interlocking system, enable the precise specification and control safety critical operations. They provide the logic to prevent accidents and ensure reliable and efficient operation of transportation and industrial systems. This paper describes about the boolean equations, their implementation methodology, validation mechanism and advantages in safety critical systems.

Keywords*— *Electronic Interlocking, Track Circuit, Interlocking Logic, Relay Interlocking, Trackside Equipment, Control Unit, Block Signaling, Point Machine, Fail-Safe Design, Boolean Algebra, Boolean Logic, Boolean Expressions, Digital Circuits, Logic Design

I. INTRODUCTION

Boolean equations in computer-based interlocking systems use Boolean algebra, logical operators, variables, truth tables, simplification techniques, and formal verification methods. This mathematical foundation ensures that these systems operate safely and efficiently, making them a vital component in the modern world of transportation and industry.

Boolean equations in computer-based interlocking systems are used to control and ensure safe operations. These equations are typically constructed based on the logic of the system and are implemented using digital logic gates. Here are some sample Boolean equations and their implementation methods:

a. Track Switch Control:

Boolean Equation: If the track switch is in position "A," then the signal should show "Green."

Implementation Method:

Signal_Green = Track_Switch_A

In this case, the Boolean equation is straightforward: the signal shows "Green" if and only if the track switch is in position "A." This can be implemented using a simple AND gate.

b. Block Occupancy Detection:

Boolean Equation: If a block is occupied by a train (Occupancy_Sensor = 1), and the signal is not red (Signal_Red = 0), then there should be no power supplied to the block (Power_Block = 0).

Implementation Method:

Power_Block = (Occupancy_Sensor AND NOT Signal_Red)

This equation represents a safety condition. It ensures that if a block is occupied by a train and the signal is not showing "Red," power to the block is cut off. This can be implemented using an AND gate followed by a NOT gate.

c. Crossing Barrier Control:

Boolean Equation: The crossing barriers should be down (Crossing_Barrier_Down = 1) if either a train is approaching (Train_Approaching = 1) or a maintenance operation is in progress (Maintenance = 1).

Implementation Method:

= (Train_Approaching OR Maintenance)

In this case, the Boolean equation is a logical OR operation. If either a train is approaching or maintenance is in progress, the crossing barriers should be down. This can be implemented using an OR gate.

d. Emergency Stop Button:

Boolean Equation: If the emergency stop button is pressed (Emergency_Stop_Button = 1), all signals should show "Red" (Signal_Red = 1), and all switches should be set to a predefined safe position (Switch_Safe_Position = 1).

Implementation Method:

Signal_Red = Emergency_Stop_Button
Switch_Safe_Position = Emergency_Stop_Button

This equation enforces a safety measure. When the emergency stop button is pressed, it sets all signals to "Red" and switches to a predefined safe position. This can be implemented directly by connecting the emergency stop button to the signals and switches.

These are just a few examples of Boolean equations used in computer-based interlocking systems. The implementation methods typically involve using digital logic gates (AND, OR, NOT) to combine inputs and create the desired control signals. These equations and implementations are designed to ensure the safe and efficient operation of complex transportation and industrial systems.

II. METHODOLOGY

Figure 1 depicts and electronic interlocking system in two out of two architectures. The field equipments like track, signal, points etc provide inputs in AC, DC or pulse form. These are converted in digital format and fed into the interlocking units. Only when both the outputs match, the final output is same. The information is displayed after processing on the maintenance terminals. There are two network buses for redundancy.

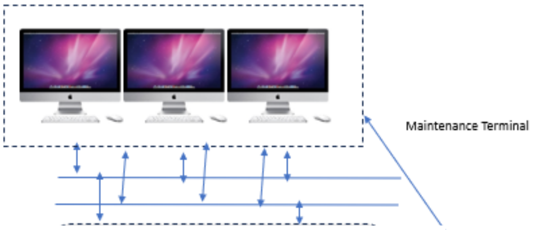


Fig 2 Boolean Equations Processing

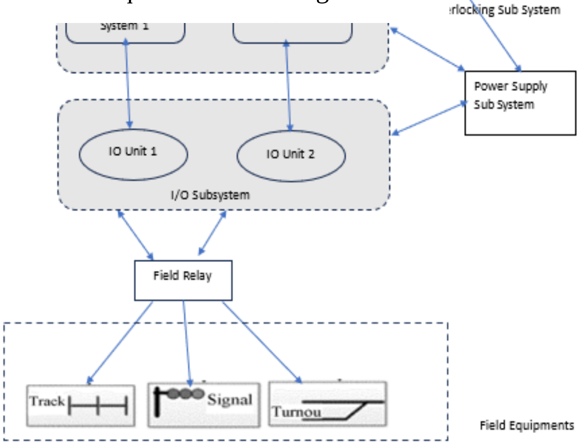
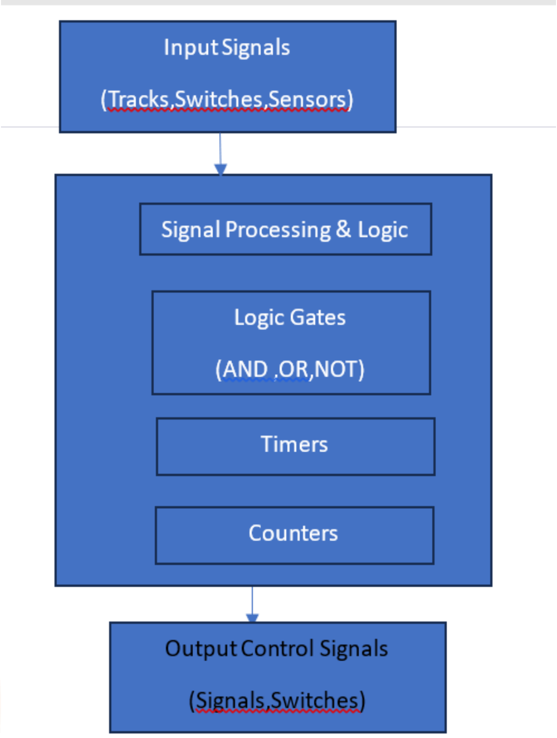


Fig 1 Interlocking System in 2002 Arch

This block diagram in Fig 2 provides representation of how Boolean equations and logic components are used to process input signals and generate output control signals in a computer-based interlocking system. In practice, these systems can be highly complex, with numerous inputs, outputs, and intricate logic.



In this block diagram,

a) Input Signals: These represent the various input sources, such as track switches, sensors, train locations, and other relevant data sources. These inputs provide information about the current state of the system.

b) Signal Processing & Logic (Boolean Equations): This block encompasses the core of the interlocking logic. Boolean equations, logic gates (AND, OR, NOT), timers, and counters are used to process the input signals and make decisions based on the defined logic rules.

c) Logic Gates: Logic gates, including AND, OR, and NOT gates, are used to combine and manipulate the input signals according to the Boolean equations. They determine the state of various control signals and outputs.

d) Timers: Timers may be incorporated to handle time-based logic, such as ensuring that signals remain red for a minimum duration or to initiate certain actions after a specific time interval.

e) Counters: Counters may be used to keep track of events or conditions, such as the number of trains passing through a particular section of track.

f) Output Control Signals: These signals represent the control outputs of the interlocking system. They include signals that control train movements, track switches, crossing barriers, and other relevant components.

III. FINDINGS

Graphs can be a useful visual representation of Boolean equations in computer-based interlocking systems. However, Boolean equations themselves are typically expressed in terms of logic gates, truth tables, or algebraic expressions rather than traditional graphs.

Let's consider a Boolean equation:

```
Signal_Green = Track_Switch_A
```

Here, we want to represent the logic that the "Signal_Green" will be active if and only if "Track_Switch_A" is in a certain state (usually ON or 1). We can create a basic flowchart-style diagram to visualize this logic

```
+-----------------+
|                 |
| Track_Switch_A  |
| (ON/1)          |
|                 |
+--------+--------+
         |
         |
         V
+-----------------+
|                 |
| Signal_Green    |
| (Active)        |
|                 |
+-----------------+
```

In this diagram:

- The "Track_Switch_A" block represents the input.
- The "Signal_Green" block represents the output.
- The arrows indicate the flow of logic.

This diagram visually conveys that when "Track_Switch_A" is in the "ON" state (1), the "Signal_Green" becomes active.

For more complex Boolean equations, similar diagrams can be created, expanding the flowchart-like representation to incorporate multiple inputs and logic gates. However, Boolean equations are often more abstract and are typically analysed using truth tables, logic gates, or simulation software rather than traditional graphical representations.

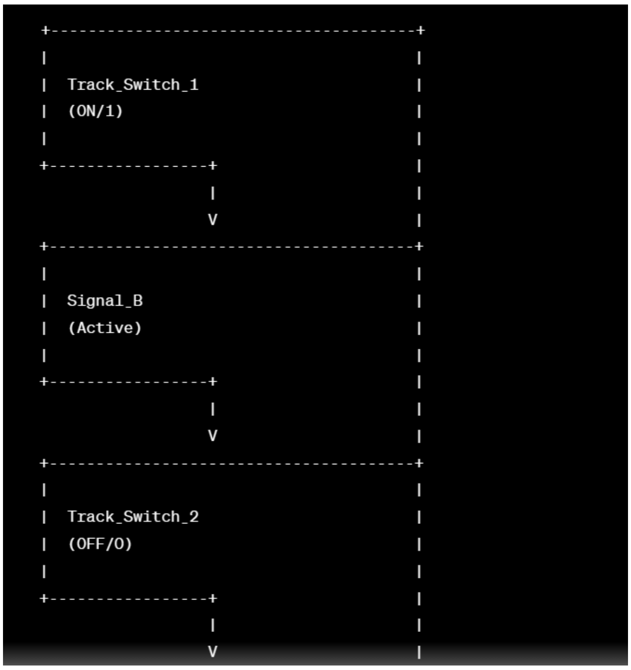
Let's consider a complex Boolean equation:

Signal_A = (Track_Switch_1 AND (Signal_B OR (NOT Track_Switch_2)) AND (Track_Switch_3 OR (Signal_C AND Signal_D)))

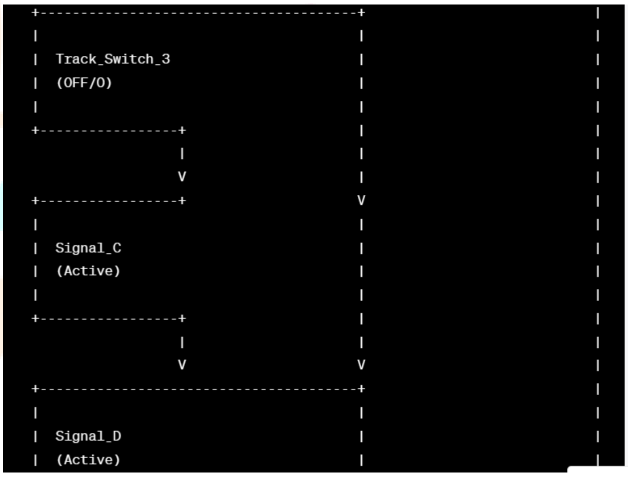
In this equation:

- "Signal_A" depends on the state of "Track_Switch_1," "Signal_B," "Track_Switch_2," "Track_Switch_3," "Signal_C," and "Signal_D" through a complex combination of logical operators.

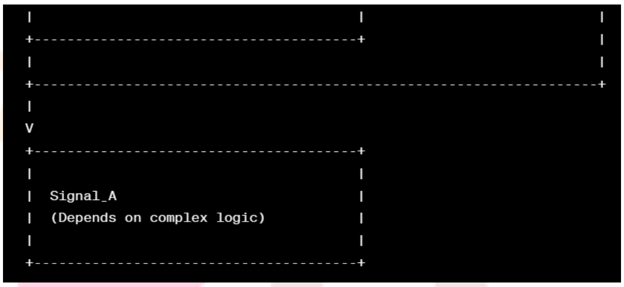
Representing this equation graphically can be challenging, but creation of a simplified diagram that illustrates the dependencies and logic assists in analyses:



```
+------------------------------------+
|                                    |
|  Track_Switch_1                    |
|  (ON/1)                            |
|                                    |
+---------------+                    |
                |                    |
                V                    |
+------------------------------------+
|                                    |
|  Signal_B                          |
|  (Active)                          |
|                                    |
+---------------+                    |
                |                    |
                V                    |
+------------------------------------+
|                                    |
|  Track_Switch_2                    |
|  (OFF/0)                           |
|                                    |
+---------------+                    |
                |                    |
                V                    |
```



```
+----------------------------------+                         |
|                                  |                         |
|  Track_Switch_3                  |                         |
|  (OFF/0)                         |                         |
|                                  |                         |
+---------------+                  |                         |
                |                  |                         |
                V                  |                         |
+---------------+                  V                         |
|                                  |                         |
|  Signal_C                        |                         |
|  (Active)                        |                         |
|                                  |                         |
+---------------+                  |                         |
                |                  |                         |
                V                  V                         |
+----------------------------------+                         |
|                                  |                         |
|  Signal_D                        |                         |
|  (Active)                        |                         |
```



```
|                                  |                         |
+----------------------------------+                         |
|                                                            |
+------------------------------------------------------------+
|
V
+----------------------------------+
|                                  |
|  Signal_A                        |
|  (Depends on complex logic)      |
|                                  |
+----------------------------------+
```

In this diagram:

- "Track_Switch_1," "Signal_B," "Track_Switch_2," "Track_Switch_3," "Signal_C," and "Signal_D" are represented with their respective states or conditions.
- Logical operators such as "AND," "OR," and "NOT" are indicated.
- "Signal_A" is the ultimate output of this complex equation, depending on the states of various inputs and logical operations.

The plots are given below

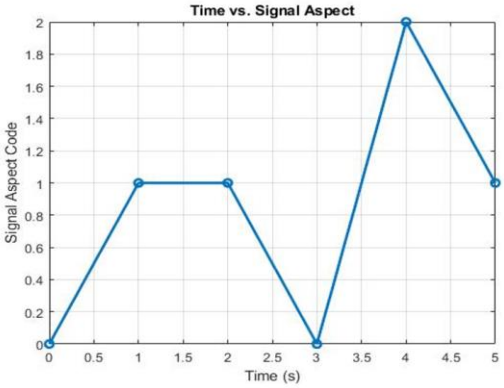


Fig 3 Time vs. Signal Aspect Plot

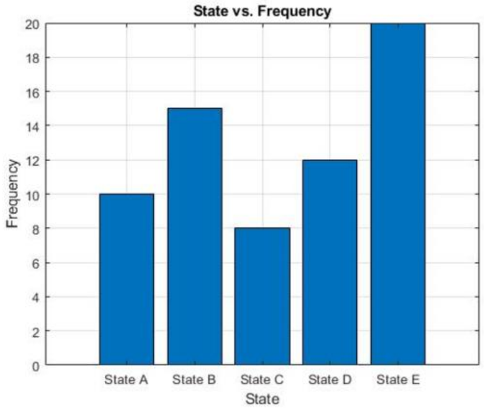


Fig 4 State vs. Frequency Plot

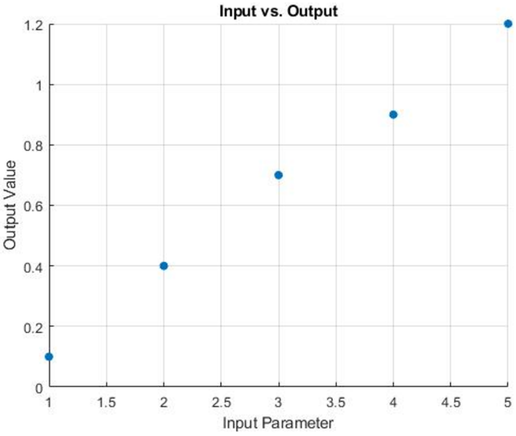


Fig 5 Input vs. Output Plot

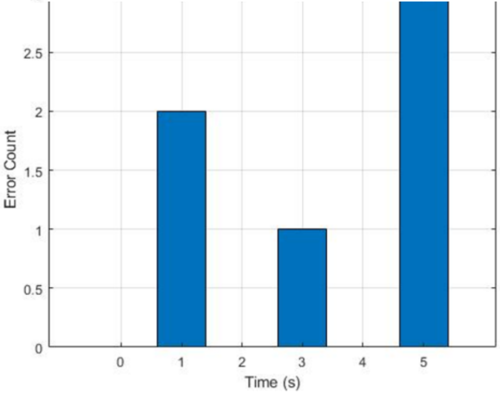


Fig 6 Time vs. Error Plot

Real-world complex interlocking systems would involve more inputs, outputs, and intricate logic, making the graphical representation considerably more complex.

IV. TESTING & VALIDATION METHODOLOGY

Testing and Validation of Boolean equations in computer based interlocking systems is a critical step to ensure the logic operates as expected and meets the safety requirements.

The steps and approaches to test Boolean equations are

a.Simulation Software: Use specialized simulation software designed for interlocking systems or digital logic systems. These tools allow to create digital model of the system and test its behaviour in various conditions.

b.Test Cases: Develop a set of test cases that cover different scenarios and edge cases. Consider situations such as train movements, signal changes, switch positions, and sensor inputs.

c.Truth Tables: Create truth tables for each Boolean equation. List all possible combinations of input values and the expected output. This provides a clear reference for validation.

d.Manual Testing: Manually evaluate the Boolean equations for correctness. Check the logic against system requirements and safety rules. Ensure the equations represent the desired behaviour.

e.Simulation: Use the simulation software to input various combinations of inputs and observer the outputs. Verify that the outputs match the expected results from the truth tables and your manual evaluation.

f.Edge Cases: Pay special attention to Test situations that are the limits of your system’s capabilities to ensure it behaves correctly.

g.Formal Methods: Use formal methods such as model checking or theorem proving to mathematically verify the correctness of your boolean equations.

h.Hardware Testing: Implement the boolean equations in actual hardware of interlocking system and conduct physical tests. Ensure that the system operates in real world environment.

i.Safety Validation: Prioritize safety validation. Ensure that the Boolean equations enforce safety rules and prevent hazardous situations, such as train collisions or incorrect switch positions.

j.Documentation: Document the testing process thoroughly, including the test cases, simulation results and any issues or anomalies encountered.

k. Iterative Process: Testing and validation should be an iterative process. If any issues or discrepancies are found revise the boolean equations and retest until the system behaves correctly and safely.

l. Peer Review: Peer or expert review shall enable to review and validate the Boolean equations.

m. Regulatory Compliance: The interlocking system should comply to the industrial standards. Validation of the boolean equations is essential to meet the standards.

n.Continuous Monitoring: Even after initial validation, it is required to continuously monitor the performance of your interlocking system in real-world operations and implement mechanisms for detecting as well as responding to faults or anomalies.

V. ADVANTAGES

Boolean equations offer several advantages in the context of computer-based interlocking systems some of the key advantages are:

a. Precision and Clarity: Boolean equations provide a precise and clear way to represent complex logic. They leave no room for ambiguity, making it easier for engineers to design and understand the interlocking logic.

b. Safety Assurance: Boolean equations are instrumental in enforcing safety rules and conditions. They allow for the explicit definition of safety-critical situations and actions, reducing the risk of accidents and ensuring the well-being of personnel and assets.

c. **Efficient Control:** These equations facilitate the efficient and automated control of various components, such as signals, switches, and sensors. By expressing logical relationships, they enable smooth and conflict-free operations.

d. **Modularity:** Boolean equations promote modularity in system design. Different components of the interlocking system can be represented by separate equations, allowing for easier testing, maintenance, and replacement of individual parts without affecting the entire system.

f. **Scalability:** Boolean equations can scale to accommodate the complexity of large and intricate interlocking systems. As systems grow, additional equations can be added to represent new logic or functionalities.

g. **Testing and Validation:** Boolean equations can be rigorously tested and validated. This includes simulating various scenarios, using truth tables, and sometimes employing formal methods to verify correctness and compliance with safety standards.

h. **Real-Time Response:** Computer-based interlocking systems rely on Boolean equations to make real-time decisions. These equations ensure that responses to changing conditions happen swiftly and accurately.

i. **Adaptability:** Engineers can adapt and refine Boolean equations as requirements evolve or new scenarios arise. This adaptability allows interlocking systems to stay up-to-date with changing operational needs.

j. **Standardization:** Boolean equations can be standardized, which simplifies design and integration efforts. Standardized logic components can be reused across different projects, reducing development time and costs.

k. **Fault Tolerance:** Boolean equations can incorporate redundancy and fault-tolerant logic. This means that even in the presence of failures, the system can continue to operate safely and efficiently.

l. **Reduced Human Error:** By automating complex decision-making processes, Boolean equations reduce the potential for human error in operating and controlling the system.

m. **Documentation:** Boolean equations provide a structured and documented way of representing the interlocking logic. This documentation is invaluable for system maintenance, troubleshooting, and regulatory compliance.

VI. ADVANTAGES

Boolean equations are a fundamental and indispensable tool in the realm of computer-based interlocking systems, which play a vital role in ensuring the safe and efficient operation of complex transportation and industrial networks. These equations provide a formalized means of expressing the intricate logic that governs how various elements within these systems interact and make critical decisions.The takeaways include

a)Safety Assurance: Boolean equations are primarily designed to enforce safety rules and prevent hazardous situations. They enable the precise representation of safety-critical conditions and actions, reducing the risk of accidents and ensuring the well-being of both personnel and valuable assets.

b)Efficient Control: These equations allow for efficient and automated control of various components, such as signals, switches, and track occupancy detection systems. By defining logical relationships, Boolean equations ensure that operations are conducted smoothly and without conflicts.

c)Complex Logic Handling: Computer-based interlocking systems often involve complex and interconnected logic. Boolean equations provide a systematic way to represent and manage this complexity, making it easier to design, test, and maintain these systems.

d)Testing and Validation: Boolean equations must undergo rigorous testing and validation to ensure they perform as expected. This process includes simulating various scenarios, using truth tables, and sometimes employing formal methods to verify correctness and compliance with safety standards.

e)Iterative Design: Designing Boolean equations is an iterative process. Engineers continually refine and update these equations as requirements evolve or new scenarios arise, ensuring that the interlocking system remains robust and adaptable.

f) Mathematical Foundation: Boolean algebra serves as the mathematical foundation for these equations, providing a rigorous framework for expressing logic and making decisions based on binary values (true/false or 1/0).

g) Real-World Applications: Boolean equations find practical applications in diverse industries, including railways, metro systems, industrial automation, and more, where the safe and reliable control of complex operations is paramount.

REFERENCES

[1] Title: "Design of Electronic Interlocking System for Indian Railways"
Authors: S. C. Rath, A. Kumar, and R. N. Sengupta
Published in: 2013 1st International Conference on Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN)
Link: IEEE Xplore
[2] Title: "An Overview of Electronic Interlocking Systems for Railway Applications"
Authors: K. M. Mohan, T. S. Selvan, and S. Anish Kumar
Published in: 2018 International Conference on Circuit, Power and Computing Technologies (ICCPCT)
Link: IEEE Xplore