1.

An embedded product undergoes a wide range of processes — designing the architecture, developing the platform with programming language and tools, integrating processors, peripherals, and software and lastly, testing compliance and functioning.

As the demand for compact devices increases, the sizes of processors and microchips keep shrinking, which requires the development of complex control systems. It is necessary to monitor the entire embedded control system and application design processes to optimize the overall system design. Here, the model-based design approach proves to be an effective and efficient means of understanding the product parts such as commercial microcontrollers and processors as well as algorithms and code for the working of both microelectronic and embedded devices. It helps address various difficulties and complexities, which arises during the lifecycle of embedded application software through visual prototyping and simulation of models.

Model-based design (MBD) performs verification and validation through testing in the simulation environment. It covers various disciplines, functional behavior, and cost/performance optimization to deploy a product from early concept of design to final validation and verification testing.

The methodology discussed in this paper concerns the process of Model Driven Engineering that allows the representation of system and validation of its properties after needs and requirements capture. This automatic process is based on three phases: semi-formal representation, passage to the formal representation, and system validation.

2.

Evaluation of Model Provided for a braking system:

The screenshots of evaluations are provided below:

A screenshot of a cell phone

Description automatically generated

In this step, the model is read for evaluations.

A screenshot of a cell phone

Description automatically generated

This step provides the details of process followed to execute the NuSMV commands.

A screenshot of a cell phone

Description automatically generated

This screenshot shows the variables in the model

A screenshot of a computer

Description automatically generated

The reachable states are shown in the screenshot above

A screenshot of a cell phone

Description automatically generated

The analysis of the CTL specs is done is this step.

The provided specs is SPEC AG!(( encrust\_frame & !generate\_frame) →AX (state=brake\_on )) which turns out to be false as seen in the above screenshot.

The CTL spec states that the brake is on when there is an encrusted frame, although no effective frame has been generated.