Chapter 3: Methodology

3.1. Introduction

System-level modeling is used to derive the cost, performance, and power early in the design phase. This is done by conducting trade studies on the architectures of the embedded system to select the best bus, processor, and memory. Using this methodology success and failure is determined by understanding the traffic and resource requirements of the proposed solution. System-level modeling involves the abstract representation and mapping of the hardware and software components and statistical models to generate the performance and cost numbers [44, 55]. Based on analysis requirements and availability of system information the system models can be further refined to the detailed representation.

Abstract modeling components used in system-level modeling are built by referring analysis specific details with less functional information. These representations are simpler and mimic the actual system components in a specific area of interest. The proposed research method shows optimizing the modeling effort of simulation-based system-level modeling of complex embedded applications by exploring the systematic modeling approach, technique of extracting the modeling blocks for given specification and goals, and techniques used for faster analysis and generating results for different scenarios. This approach includes the pre/post-system modeling, modeling component selection, system modeling, and analysis steps as shown in Figure 3.1.

Problem description, defining modeling objectives, pre-work, and data collection done is in pre/post-system modeling step. Based on availability of system specification and modeling components, component selection is to be done. This step involves the extraction of modeling blocks for given specification and goals and techniques used for faster analysis and generating results for different scenarios. In extraction, reusing of the modeling component of the previous work is to be considered. If the required component is not available, then the custom modeling component has to be built for system modeling. Later, the system is built using selected modeling components. After creating simulation model, the system simulation is done to get the simulation results. Simulation can be done either by manual experimental setup or by the batch simulation method. The analysis is to be done by comparing the expected results. If the expected result does not meet the requirement, then go back to the modeling steps as shown in Figure 3.1.
This approach gives the methodical way of building the analysis specific system model. This approach includes the concept of reusability and modeling component selection that reduces the modeling effort and saves modeling time.

3.2 Pre/Post System Modeling

Pre/post system modeling is the step before and after the system modeling, includes Problem Identification and Formulation, System Data Collection, and Model Results for Communication and Future Use as shown in Figure 3.2.

3.2.1 Problem Identification and Formulation

Describing challenge, solution to address challenge, problem definition, essentials, modeling goals, results to be achieved, assumptions made, and required modeling cost are mentioned in this step.
3.2.2 System Data Collection

Essential system details are acquired to build the analysis specific system model is called system data collection. In this step, the information require to construct the system model such as system description, parameter values, the unpredictable arrivals to the system, and traffic distributions details are collected.

3.2.3 Collect the Results for Different Simulation Runs and Discuss

In this step, conducted experiments with various values of different inputs and then collects the results for multiple simulations. Then construct display charts for further study and analysis.

3.2.4 Document Model for Communication and Future Use

In this stage model description, goals, considerations, parameter, simulation and analysis report, and scope for further study are captured in the form of document. This document plays as communication channel between different team members during the system development.

3.3 Modeling Component Selection

Modeling component selection is the process in which basic modeling components are selected based on followings.
• Specifying the modeling objectives based on system description and requirements.
• Defining the traffic inputs which mimics the external interaction to the system.
• Specifying the system behavior with respect to timing information with less or ignoring other information’s.
• Deciding the modeling constructs for system components.
• Build the system model, conduct simulation experiments
• Estimate, optimize, and validate.

Figure 3.3 shows the list of different modeling components categorized for system modeling. Modeling components can also be built from scratch. Discrete event simulation is considered as Model of Computation. Here the length of the simulation can be given a value or considered as default. Type of modeling objectives is to find the system latency, resource utilization, instantaneous power dissipation, buffering size, and waiting time. Modeling types can be queuing, mapping, standard and mixed. And also the kind of application can be specified. Application type can be automotive, industrial, SOC, processor, imaging, standards, mobile, computer software, network, networking, aerospace, and others [42]. Result blocks such as plotter, statistics generator and text display are used to calculate and display the performance metrics. Performance metrics is utilization, throughput, latency and buffering requirement. Workload or traffic definition is to be based upon the type of input, generation type, and attribute details. Input types can be
• A signal from sensors
• User action
• Network packet
• Input for each channel
• Interface or port
• Read or write request
• Instruction sequence to execute on a processor
• Input from interface
• Request from scheduler
• Trigger a periodic operation, and
• Execution of a scheduler
Traffic generation types are

- Standard statistical distributions
- Special or custom distributions
- Trace file input
- Scenario driven

Traffic can also be defined based on

- Its attributes as time distribution between generated events
- Data structure (packet, frame, IC pins etc.)
- Fields (define standard data headers/ control such as data, address, source, destination, priority etc.), initial values for the fields of the data structure

Characterizing the workload handled by the system in terms of the demands placed on the various resources using traffic blocks. System behavior can be represented by activity diagram, Unified Modeling Language diagram or task flow diagram. Components used for Behavior modeling are defining flow blocks, processing blocks, decision block and mappers. Hardware modeling component selected based on the modeling type, the specification of the hardware to be considered. Information required to represent the hardware in queuing modeling is attributes...
of queue and delay of execution length. In case of mapping type, details required for hardware modeling are attributes of the scheduler, processor speed, context switching time, and worst-case execution time or task cycles. In standard modeling type, data sheet details required for hardware modeling. Data-sheet information of a processor includes instruction sets, clock frequency, processor speed, cache memory and hierarchy, bit size and number of pipeline stages. Automation for modeling component selection is explored here. The user has to select the options and guidelines report generated from automation as discussed in section 3.2 and section 5.3.2.

3.4 System Modeling

System modeling is the process of creating the system model by using available and reusable modeling components. Input required for this process is system description, objectives. Steps used in system modeling not necessarily fixed order can vary depends on the levels of the details. Proposed approach is particularly for modeling complex embedded systems. The flow of this approach is includes specifying the system description and developing models for system behavior, architecture of the system, and allocating the architectures resources to the tasks of system behavior [32]. Create schematics and stream outline of the framework and converting to executable model. Examine executable model by tracing through the animation and manually identifying numbers, then comparing performance numbers with the expected results or with the real system performance [33]. This is done by system engineers, modelers verifies the considerations made in the system modeling. Figure 3.4 shows the flow of modeling, simulating and analyzing system models.

3.4.1 Simulator Selection

Simulation engine is selected on the basis of need of the simulation and modeling objectives. Discrete event simulator is used for performance modeling of embedded systems.

3.4.2 Declaration and Initialization

Declarations and their initializations are as same as in higher languages such as C and C++. It includes the parameters, data fields, global and local data declarations.
3.4.3 Workload Modeling

Impact of different traffics arrived to the system is characterized by the workload model, which mimic the video frames, signal flowing through the ports, and control signals. Arrival of traffic can be distribution type, sequential stores or retrievals and sequence of actions. Workload includes set of data, signal or information given to the port. Workload describes the test vectors given to the system model to verify analysis specific actions of the system. Traffic rate can be a defined intervals of time or it can be arrived any movements. Workload Model \( w_m \) can be depicted as in equation 3.1. Work load contains set of data fields. These fields show the timing details about inputs data such as video frames, signal flowing through the ports, and control signals, internet packets, and sequential stores or retrievals and sequence of actions.

\[
wm = f(D_s, A_t, T_r) \quad \text{Equation (3.1)}
\]

Where,

\( D_s \rightarrow \text{Data\_Structure} \)
Traffic rate $T_r$ can be specified as constant period of time, randomly arrived, deviations between two different values, or arrival rate recorded from actual system. The input traffic build such that it should be simpler that contains the less number of operations. Larger complexity of workload which includes excessive number of operations not just make the workload hard to build and oversee yet additionally make it less valuable for understanding performance. It isn't important to mimic each and every use details, instead center around necessary information well on the way to happen. The lower recurrence ones can be disregarded or joined with others into a solitary task [34].

### 3.4.4 Behavior Modeling

System behavior includes the information about sequence of tasks with their data flow execution. Behavior information of the system can be extracted from the set of tasks in the application, software program or algorithm. Extracted information is related to only timing that is essential information required for performance modeling and no other implementation details are included. Set of tasks in the behavior can be concurrent and sequential based on data dependency. Actual processing of the task on the resource is modeled such that task of the behavior model is mapped with resource of architecture model. Based on task timing, task mapping and resource speed execution timing can be estimated. Behavior model $bm$ can be expressed as in equation 3.2.

$$bm_{queue} = f(tf, q, dt)$$  \hspace{1cm} \textit{Equation (3.2)}

\textit{Where}

- $tf$ \textit{Task flow (task diagram)}
- $q$ \textit{Queue}
- $dt$ \textit{processing delay}

Each process or task in the behavior modeled is described by a queue with delay as given in the equation 3.3.
\[ q = f(ql, nq, qs) \]  \hspace{1cm} \textit{Equation (3.3)}

Where

\( ql \rightarrow \text{queue length} \)
\( nq \rightarrow \text{number queues} \)
\( qs \rightarrow \text{Queue scheduling (FIFO, LIFO, Preemptive etc)} \)

Processing delay can be either approximated calculated as task WCET. In modeling Queue with the delay is used for representing a task when there is no architectures detail available that is needed usually in the early phase of the system design [58]. This kind of modeling is essentially used to capture the scheduling schemes in network system and to determine the resources contention. Here process is defined as function of finite or infinite number of data fields.

\subsection*{3.4.5 Architecture Modeling}

Hardware part of the system is categorized in to processing, storing and data transferring units. These units are called resources. These resources modeled specific to performance analysis that is only timing details with minimal functional details. Hardware and software components of embedded systems are modeled using Ptolemy simulator based modeling environment. Entire system is modeled such that the program to be executed on processor is allocated by scheduling. Architecture model \( am \) represented as in equation 3.4.

\[ am = f(tm, rs) \]  \hspace{1cm} \textit{Equation (3.4)}

Where

\( rs \rightarrow \text{resources} \)
\( tm \rightarrow \text{task mapping} \)

Resources are characterized by resources speed, context switching, resource queue length, and resource scheduling as given in equation 3.5.

\[ rs = f(rp, cs, qlr, qsr) \]  \hspace{1cm} \textit{Equation (3.5)}

Where

\( rp \rightarrow \text{resources speed} \)
\( cs \rightarrow \text{context switching} \)
\( qlr \rightarrow \text{resource queue length} \)
\( qsr \rightarrow \text{resource scheduling (FIFO, LIFO, RR, Preemptive, etc)} \)
Information required modeling the task mapping as given equation 3.6.

\[ tm = f(tn, tt, rn, tp) \]

*Equation (3.6)*

Where

\( tn \rightarrow \text{task number} \)

\( tt \rightarrow \text{task time} \)

\( rn \rightarrow \text{Resource name} \)

\( tp \rightarrow \text{Task priority} \)

Task time can be either approximated or can be calculated by cycle counting of the given program and using equation 3.7.

\[ tt = \frac{\sum_i i * C}{rp} \]

*Equation (3.7)*

Where

\( i \rightarrow \text{instruction} \)

\( C \rightarrow \text{cycle count} \)

Hardware modeling is done to identify the differences in performance numbers for various hardware platforms. Based on this performance difference right platform can be selected. Details at lower level of abstraction include both timing and functionality. Modeling at this level is more accurate but less flexibility in alternative selection. To modify the one system configuration to another system configuration will take more modeling effort. This kind of modeling is used specifically for verification purpose where hardware part of the system design decided.

**3.4.6 Mapping**

Mapping in modeling mimics the process of allocating the resource to the task. Each task in modeling is characterized by number of execution cycles estimated by set of instructions of particular task or by approximating the task timing.

**3.4.7 Statistics Modeling**

In statistics modeling expressions are used to compute the system performance. These are the expressions for particular analysis.
3.4.8 Post Processing

To conduct the simulation experiments parameters and input conditions of the model is required to set different values for each execution of the module. Huge number of simulation runs required for different parameters and their various values. In such case batch simulation considered as better approach than setting values of parameters for each simulation run. Multiple simulation runs are needed for analysis and optimization of non-functional metrics by varying system parameters and input variables. System architecture is explored by conducting the comparative study on simulation results of different hardware configurations mapped. In each simulation experiment behavior model is mapped to the model of specific hardware configuration. In each simulation performance numbers are computed by recording flow of actions with respect to timing. That is arrival of task request, waiting in queue to access resource and run the task [29].

3.5 Simulation

Simulation of the system is the execution of the system model. Simulation is used to the modeling where the modeling of the complex system is required. Simulation is an inexpensive approach to conduct the experiments and evaluate the system specification before implementation. Particularly execution is required to predict the timing bottlenecks of the system to be developed by considering various input conditions and system configurations. Discrete event simulation is used in this approach in which system is considered to be responding simultaneously for arrival of events.

Discrete event simulation with Model of Computation (DMOC) used as simulation engine in this system level performance modeling and analysis. In simulation model is executed with the fixed period of time. In each execution traffic occurrence to the system, system responses to occurrence, and execution of concurrent events, is captured and maintain the history. Discrete event simulation is less elaborate than the continuous simulation, but it is a lot less complicated after implementation, therefore, it is used most among all the applications.
3.5.1 Simulation Based Experimental Setup for Different Hardware Configuration

Determining measures used for performance analysis, input factors and their distinctive levels, scenarios of different experimental setup during simulation period, anticipated that yield information would be corresponded are examined and characterized in this stage.

3.5.2 Perform Simulation

Simulation runs generates the design metrics numbers such as cost, latency, utilization and can be plotted with least/most results or standard deviations. There are two options made for simulation based experiment setup. In manual simulation requires to changing the parameter values and run the simulation for different conditions. Multiple simulation runs for different values of parameters and collecting and plotting the results is lengthy process. For example to run the simulation for a parameter of its 100 different values, it take 100 simulations runs, also collect the results for 100 different values, if there are lot of parameters then again it becomes a lengthy process. In such cases, batch simulation and post processing is required for analysis. Following steps are required to perform batch simulation and to collect results. Algorithm is built to produce the batch document. Batch simulation algorithm takes the information about the name of different parameters and different values of each parameter and it generates the batch file as shown in Figure 3.8 for multiple simulation runs.

3.6 Analysis

Analysis is the process of understanding the variation in design metrics with respect to the changes input parameters at different simulation. Power, performance, and cost metrics can be predicted, optimized and system failures can be realized early in the design stage. Only performance analysis is considered in proposed work.

3.6.1 Performance Analysis

Performance modeling is a structured yet repeatable method in accordance with the model the overall performance concerning the System [59]. Theoretical representation of system performance is as given in equation 3.8.
ts1 = (Tj + TMj), ∀j ∈ P

Where

ts1 → system latency
Tj → task processing time mapped with resource
TMj → memory access time
P → set of components

If the two elements j and k want to access the resource simultaneously, one has to wait due to resource contention depicted as equation.

\[ ts2 = (Tj + TMj + TWj), ∀j ∈ P \]

Where

Ts2 → system latency with considering TWj
TWj → total waiting time to access resource

Waiting time depends on the number of request in the queue, processing time, scheduling algorithm, and priority. Calculating TWj is complicated because of random arrival of the requests and system complexity. Hence simulation based performance modeling and analysis is the best solution to calculate the waiting time. Performance modeling and analysis is a type of simulation based system level modeling. In this case each unit is built with specific to timing analysis. It starts amid the early periods of the application outline and proceeds all through the application lifecycle. This kind of modeling is essentially used where mathematical modeling is not possible particularly for complex systems such as complex network systems with scheduling mechanism. And this approach is essentially used to determine the waiting time based on resource contention [43]. In performance modeling, the system can be viewed as a collection of interconnected hardware and software resources that provides service to a community of users. Performance modeling involves characterizing hardware and software resources that include application software flow, the scheduling algorithm used, the speed of the hardware devices, also characterizing the workload handled by the system in terms of the demands placed on the various resources.

Advantages of performance analysis optimizes the timing related bottlenecks before releasing the products to market and can wind up with a report of separated situations that aides rapidly
observe what is critical. Validated specification from simulation and analysis gives the confidence to meet the objectives. Performance assessment assumes a vital part in architectural exploration. It helps to decide hardware and software part of the system and make to select right platform for implementation. These choices are made based on different nonfunctional requirements. Now, the values of various performance metrics can be determined by the number of time units the system was in a particular set of states. Without losing the generality, to evaluate performance, assumption made that the state transition in a system takes place with the clock cycle. Performance evaluation approaches at various abstraction levels try to determine the number of times the system was in different states. The concept of state is described differently at various approaches and depends on the abstraction levels. This type of modeling is used where the complete system details are not available and this modeling is done before design and implementation. This type of modeling is suits where modeling using series of problematic equations is not possible such as complex network systems with scheduling mechanism [45]. And this approach is essentially used to determine the waiting time based on resource contention. In this approach, discrete event simulator used and detail description of the components are abstracted by cycle counting and profiling. Hence the faster simulation reduces the evaluation effort.

3.7 Enhancement of System Level Modeling Approach

To enhance the system level modeling technique of generating modeling guidelines included along with proposed methodology as shown in Figure 3.5. In this process idea or specification and system requirements are of transformed to executable simulation model and establishes the batch simulation setup to speed-up the system modeling and simulation. This involves collecting the system goals, analysis details, and its component specification by the questionnaire list, identifying and mapping the library blocks, generating the modeling data sheets. Effort of learning and selecting the simulation models can be avoided by generating the guideline report. Questions are framed based on available modeling frame works, experience. Data required for modeling is categorized into behavior of the system, simulation type, Modeling objectives, application type as mentioned in Figure 3.6(a). The first step in the data collection is that the system designer has to describe the system. Modular has to select the type of application of the proposed system as shown in Figure 3.6(b). Modeling objectives can be estimation and
evaluation of performance, power, cost or other metrics. Performance design metrics can be the latency, utilization, and throughput. Power design metrics can be considered for system-level modeling are instantaneous, static, and dynamic voltage scaling. System cost is estimated based on the resources used.

Figure 3.5: Enhancement of Proposed Approach

Figure 3.6(a): System Data Collection

Another option is for customizing the analysis as shown in Figure 3.6 (c). Information required to develop the behavior model is related to behavior of system flow and input traffic. Input information is required to develop the stimulus to the system model under simulation is called
workload model. Workload model includes the stream of required system information and rate of requests to the system [46].

**Figure 3.6(b): Application Type**

**Figure 3.6(c): Modeling Objectives**

Types of inputs are classified as shown in Figure 3.6 (d) based on application as user action, network packet, input from interface, signal from sensors, software, and user defined. In Special or Custom Distribution, trace file input's scenario driven and Standard Statistical Distribution options are available for the User Defined type. Information required to develop Standard Statistical Distributions are field type, traffic distribution, and logic if required. The three types of traffic distributions are exponential, fixed, and mean. And three types of field type are custom, trigger, and header. Required system flow description as shown in Figure 3.6(e) is either in the form of task flow diagram or UML diagram. Each task in the system can be represented based on available details. If architecture details are not available then the task is modeled by queue with delay. Here randomize the value of queue length and execution length during simulation. If the resource details are available, then task and resources are modeled separately.

Timing details required to model the task that is by counting instructions and their machine cycles or relative intervals. Cycles counts per task calculated by profiling or cycle counting if the
set of instructions is available for a particular task otherwise relative-time for the task is considered. In this case hardware and software, components can be modeled separately. Information required to model software component is processing length, mapping hardware name, priority, and id. Information required to model hardware architecture is resource name, resource speed, scheduling type, and context switching time. Micro architecture and instruction details can be used for further refined modeling.

Modeling guidelines generation is included to speed-up the simulation. Figure 3.7 shows simulation environment setup using automation. Batch algorithm included for automatic generation of batch file form the different inputs and their different values.

![Figure 3.6 (d): Data Collection for Workload Modeling](image)

Figure 3.6 (d): Data Collection for Workload Modeling
The information required to generate the file for batch simulation are number of parameters, name of each parameter, and their different values as shown in Figure 3.8.

Figure 3.9 shows the view of generated batch file which includes the information of path for the simulation model, parameters assigned with values. Each line represents individual simulation runs in which same set of parameters with different values are mentioned.
Figure 3.7: Simulation Setup

Figure 3.8: Batch Simulation Setup
3.8 Summary

The proposed approach includes five steps. Acquisition of system information and defining the modeling objectives are discussed in the first step. Step two explores the mechanism of translating the acquired system details to the modeling guidelines. Step three shows structure of modeling in Ptolemy's simulator based modeling environment. Steps four discussed the simulation; here batch simulation fastens the experimentation. Step five shows the analysis part of the approach. Analysis part of this approach is discussed in Chapter 5 and Chapter 6 by considering case studies.