VISUALSIM TRAINING

Training: Planning, Modeling, Simulation, Advanced Features

MIRABILIS design

Planning

- Goals & Requirements
- Tasks or Operations
- Workloads or Traffic
- Hardware or Platform
- Collaboration & Sharing
- Data Management
- Support
- Look & feel

Modeling

- Parameters/ Variables/ DS/ Fields
- Flow- System, Beh-Arch, Data flow
- Performance Power Functionality
- Hierarchical/ Class/ DI Blocks
- Traffic/ Results
- Hardware/ Software/ Network
- Abstraction
- Algorithms
- Task Graph
- Checking

Simulation and Analysis

- Types of Studies
- Parameter Ranges
- Graphical/ Batch/ Multi-Core
- Debugging/ Verification
- Requirements
- Post-Simulation
- Types of Plots
- Advanced Studies
- Accuracy
- Compare

Advanced Features

- Failure
- Integration
- Generation
- Encryption

Agenda- Part 1: Planning

- Goals and Requirements
- Tasks and Operations
- Workload or Traffic
- Hardware or Platform
- Collaboration and Sharing
- Data Management
- Support
- Look and Feel

Planning Stage of the System Modeling

It involves the solutions/patform architect along with the product team deciding on the high level aspects of the design.

It will involve the following questions and subsequently narrowing down on the answers:

1. Performance and Power Objectives: Define specific performance and power targets, balancing them for overall objectives, such as X GHz **processing speed** within Y watts **power budget**.
2. Application Mapping and Task Graph: Plan how applications or tasks will map onto the hardware platform, outlining the system's **task graph.**
3. Scalability and Reliability: Ensure the hardware system accommodates future scalability needs while maintaining high availability and reliability. This may involve redundant components or failover mechanisms to prevent downtime.
4. Security and Interoperability: Integrate security measures into the hardware design and ensure interoperability with other systems. For example, include hardware-based encryption modules and support industry-standard **communication protocols** like PCIe/UCIe.
5. Cost-Effectiveness and User Experience: Consider cost implications in hardware design decisions, such as selecting **IPs/hardware platform**. Employ strategies to enhance user experience within budget constraints, like choosing cost-effective components without sacrificing performance or usability.

Goals in system design

- High-level objectives guiding system design, quantifiable targets for functionality, performance, and other key aspects, used to drive design decisions.
- They can be later used to narrow down on the Requirements/Constraints of the system
- The box below gives an example of **system design goals for a radar system**

Performance:

High-Level Goal: Achieve a minimum target detection rate of 95% within a range of 200 kilometers.

Quantifiable Metric: Target Detection Rate

Power:

High-Level Goal: Maintain power consumption below 500 watts during normal operation.

Quantifiable Metric: Power Consumption

Functionality/Behavioral:

High-Level Goal: Ensure the radar system can differentiate between aircraft, ships, and weather phenomena with at least 99% accuracy.

Quantifiable Metric: Classification Accuracy

Reliability/Security:

High-Level Goal: Maintain system availability of at least 99.9% over a 30-day period, allowing for scheduled maintenance downtime.

Quantifiable Metric: System Availability

How to map goals to requirements?

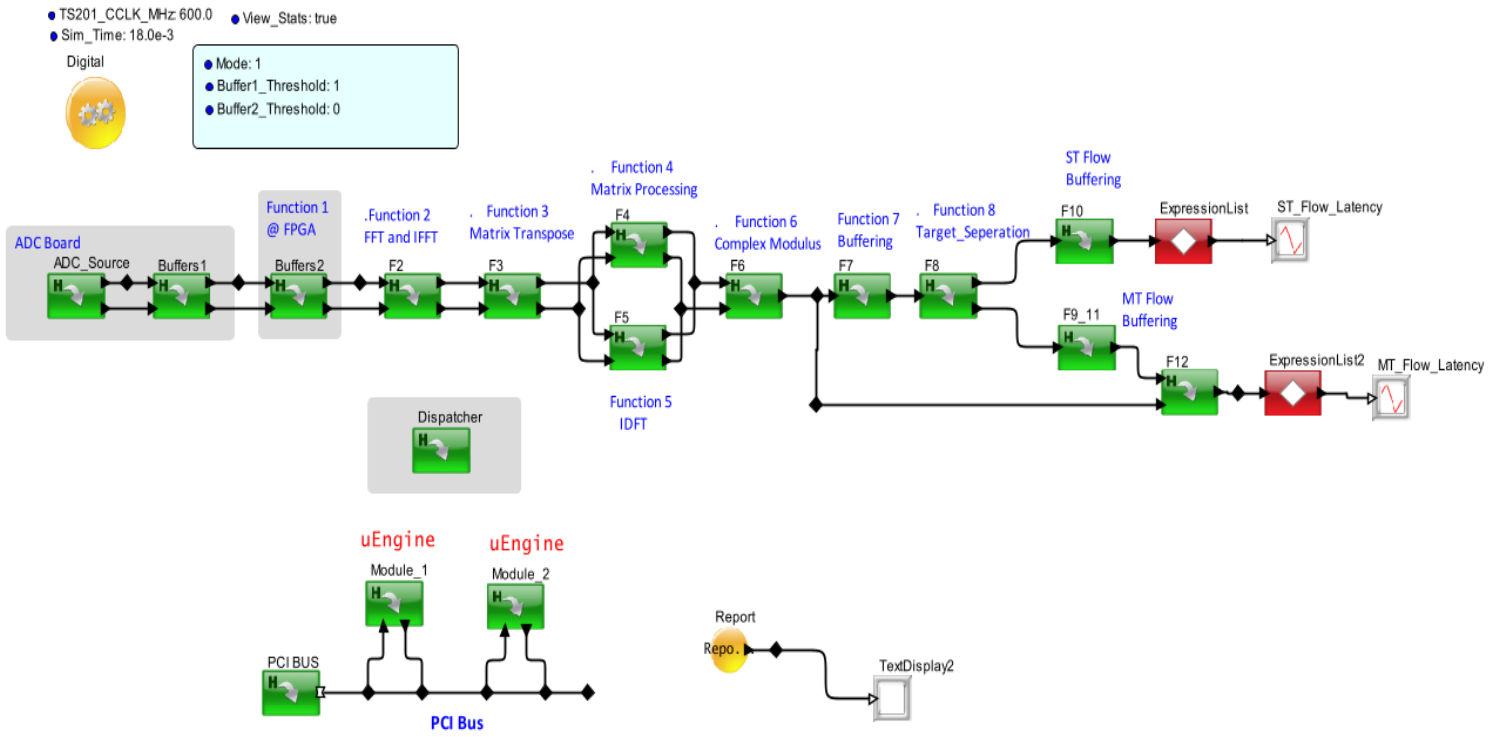


Figure : System Model with hardware + algorithms. Flow Latencies are being observed

Step 1 : Define Goals
Eg: Achieving a higher target detection rate

Step 2 : Model the system to observe certain performance metrics like latency, buffer usage, throughput.

Step 3 : Define Requirements/Design Constraints
Eg : A higher Latency will limit the target detection rate

Step 4 : optimizing the system
Eg : Optimize the radar system's signal processing algorithms, hardware capabilities, and environmental conditions **to minimize latency and improve overall performance.**

VisualSim Modeling Methodology – From Goals to verifying Requirements

MIRABILIS design

- Different HW Arch
- Components from different vendor

Mapping application to different hardware resources

Debugging

Fail

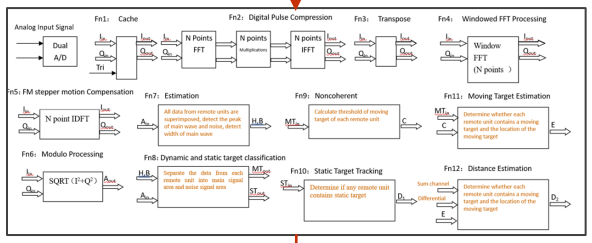
Project Goals → Capture the Workflow (Task Graph) → Task Graph Trigger Rate → List of Stats → HW Arch → Mapping → Stats Gen. → Requirements → Pass

Build a radar system that can handle 30 targets per minute

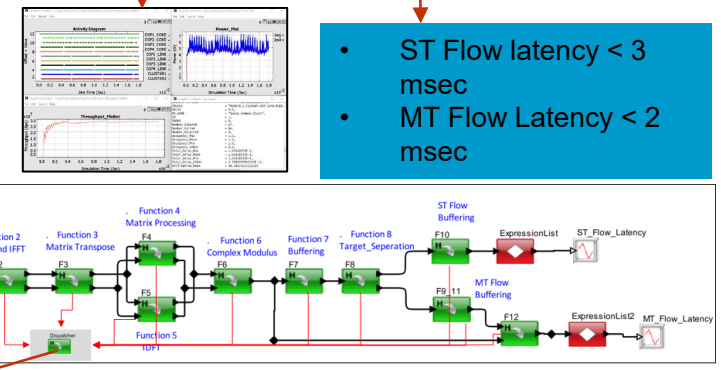
- Periodic
- Event Based

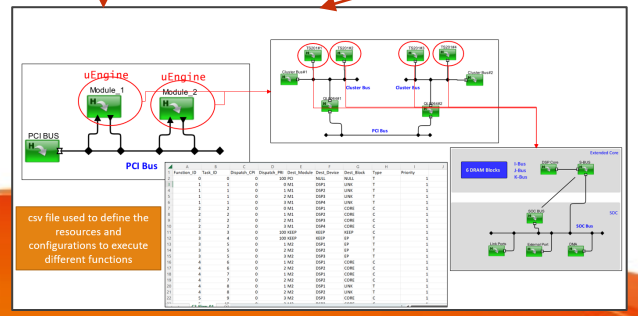
- App latency
- CPU Utilization
- Bus/Memory access latency
- Bus/Memory throughput
- Power consumption
- Buffer occupancy

- ST Flow latency < 3 msec
- MT Flow Latency < 2 msec



Built using delays





Tasks Or Operations

- **Task:** A specific unit of work that needs to be accomplished within a system. Tasks are typically smaller and more granular than workloads. They represent individual operations or actions that contribute to the completion of a larger goal.

How to Map Tasks to the System Hardware?

1. **Task Analysis:**
   - Understand the requirements and characteristics of each task in terms of CPU, memory, I/O, and other resources.
   - Analyze dependencies and relationships between tasks to identify any constraints or priorities.
2. **Resource Allocation:**
   - Assign system resources such as CPU cores, memory, and I/O devices to tasks based on their requirements and priorities.
3. **Choose the right scheduling algorithm:**
   - Optimize resource allocation to maximize system efficiency and performance while meeting the demands of individual tasks.
   - **Context Switching delays** : Involves saving the current execution context of the running process, loading the context of the next process to be executed, and transferring control to it.

Workload

- Workload is a combination of tasks
- There are three different ways to define a workload in Visualsim

Behavioral Flow :

- Define a sequence in which the hardware resources of the system will be triggered.
- Breakdown the flow into individual stages with predefined input and output

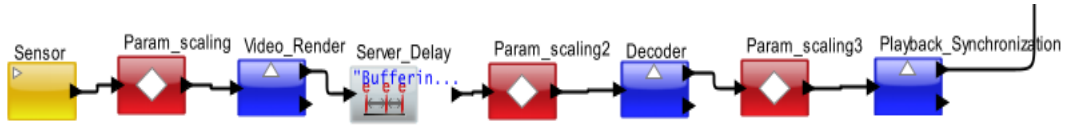


Figure : Behavioral flow example

Generate an instruction mix:

- Used to simulate the workload when the actual application software is still under development
- It can also be used to evaluate the impact of certain type of instructions on the overall performance
- Examples :
  - What will be the impact on performance when there are higher branch mispredictions?
  - higher number of loops vs a sequential code

Use real application traces:

- Used when the application is available
- Standard benchmarks like dhrystone can also be executed by fetching their traces using open source tools like GEM5

Hardware or Platform Modeling

MIRABILIS design

- Extend an SOC based system model to a complete platform model
- Components: Example of an automotive platform
  - Software applications: Extend the ECU/Microcontroller Model to include ADAS, infotainment systems
  - Sensors and Actuators : Accelerometer, Gyroscope, proximity sensor etc
  - Control systems : Speed control, Power systems, Mechanical systems etc
  - Interfaces : A high speed Ethernet interface
  - BUS/Communication Protocols : CAN BUS and nodes for communication between different control systems
  - Power Generation : Include Battery, Temperature/Thermal management

A Complete automotive platform model in VisualSim

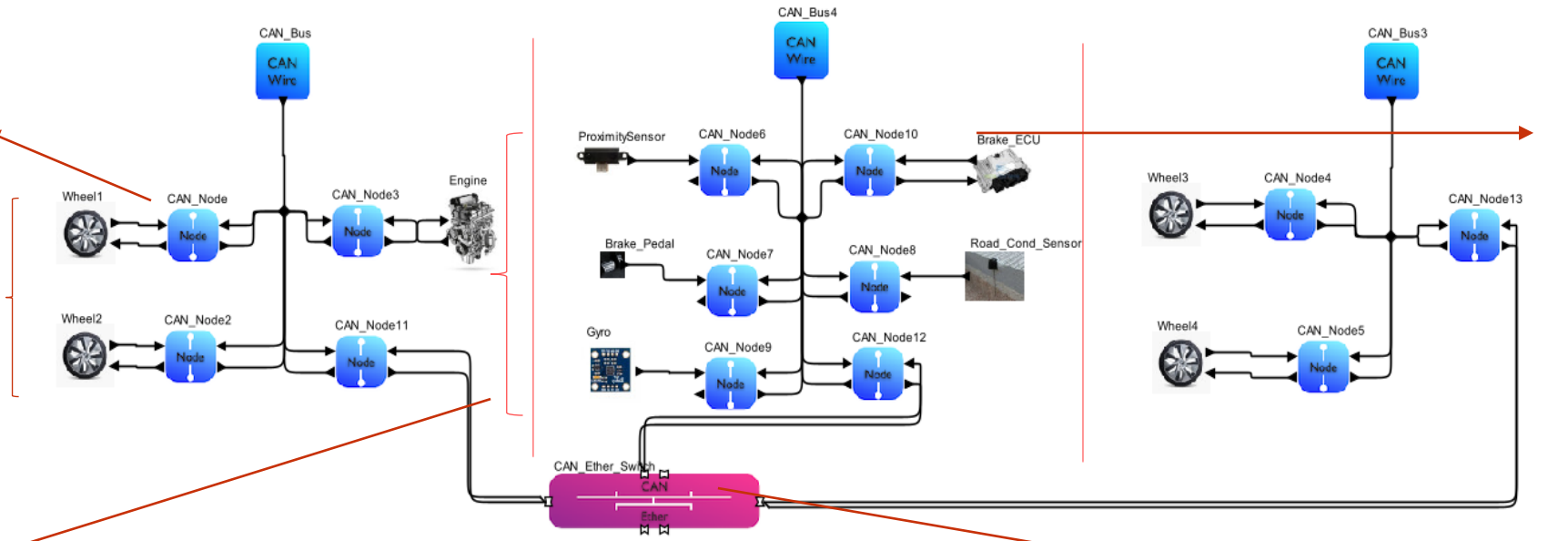
5. Communication between different control systems

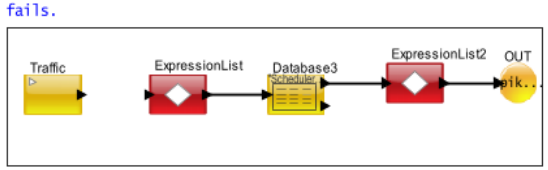
4. Mechanical parts

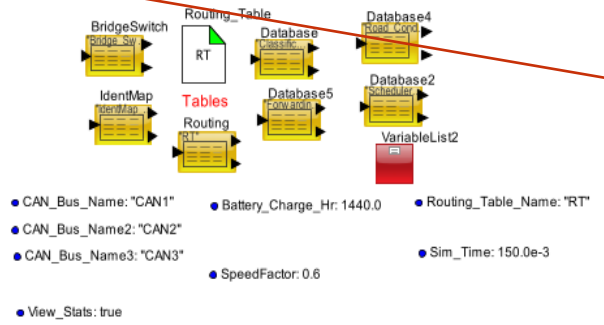
2. Sensors and Actuators

1. ECU or Microcontroller

3. Ethernet Interface







If there are no data on the Processor core 2 latency plot, then that means that processor core 2 was not being used. We have to increase the traffic rate of proximity sensor or Brake Pedal to simulate a high load case

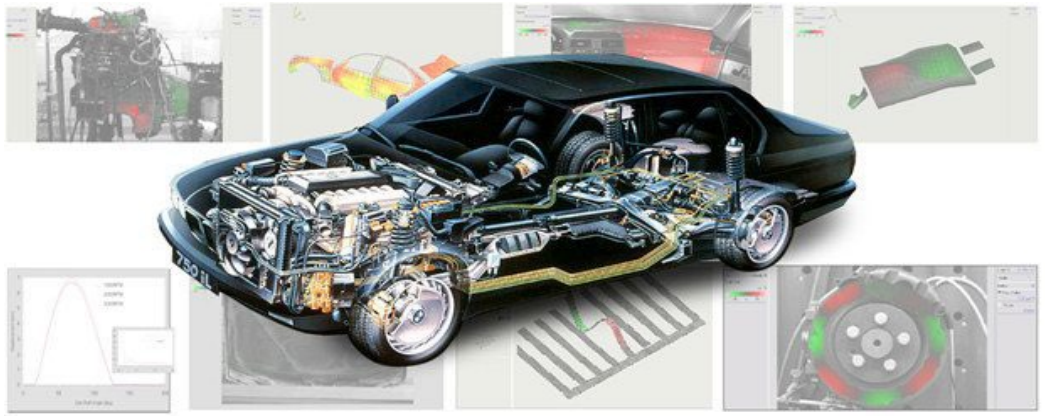
Collaboration and Sharing

Subsystems and their integration - Example

Develop or Acquire IP

SOC Platform Integration

Board level Integration



Full system Integration

Software Development – Firmware, OS, drivers

Workload analysis and Optimization

Functional testing and Validation

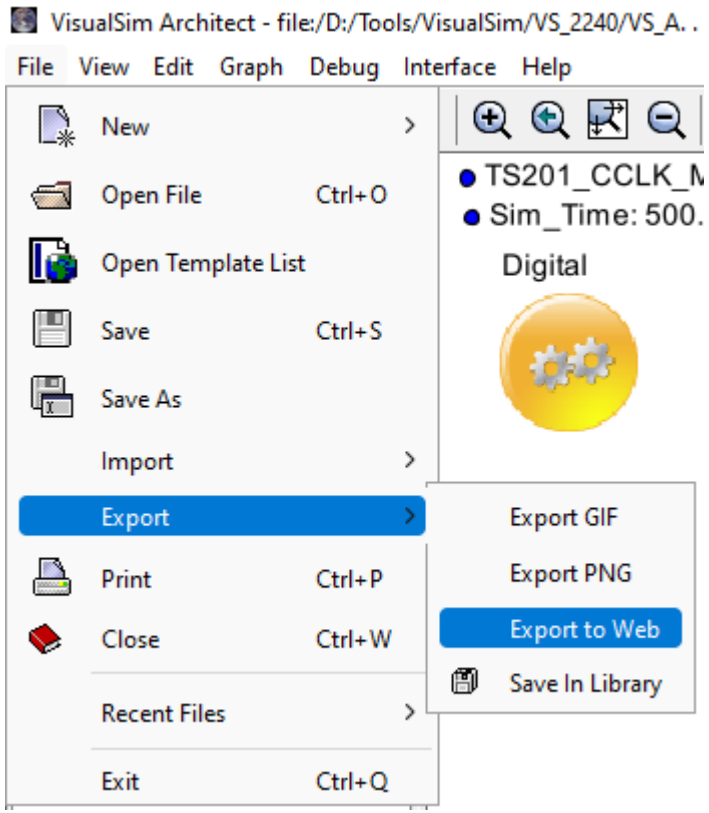
Collaboration with manufacturers and designers

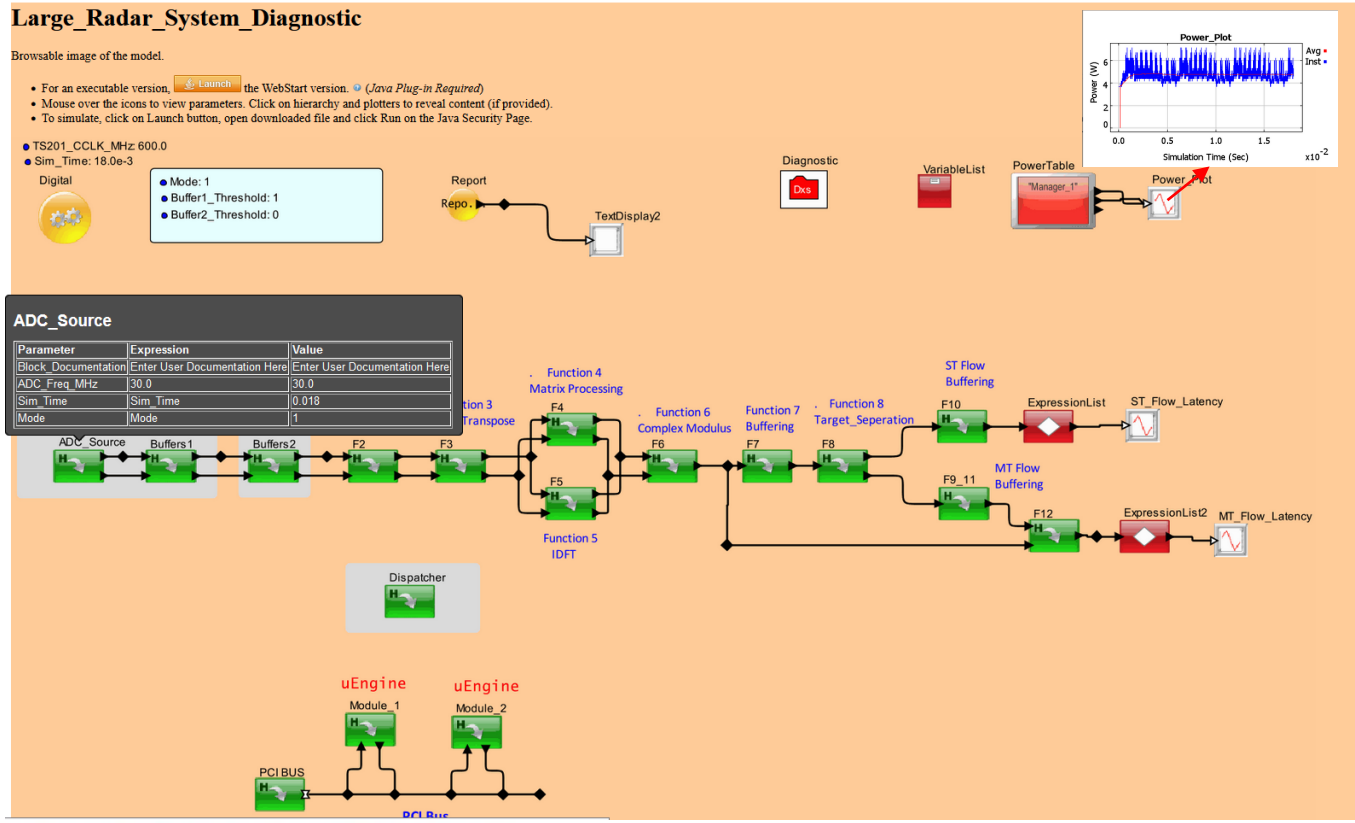
Certification and benchmarking wrt industry standards

Maintenance and Updates

Each of these tasks can be performed by individual teams using VisualSim and the Integration team can also use the outputs from these teams

Generating Documentation - Interactive, and all plots loaded





Data Management

GitHub or any version control tools

User 1

User 2

User N

Download Model and Class files
Ex: D:/Project/VS/Project1

Set CLASSPATH (in Windows – system environment and Linux - .cshrc) to include the basepath
Basepath = D:/Project/VS/

Download Model and Class files
Ex: D:/VisualSim/Project1

Set CLASSPATH (in Windows – system environment and Linux - .cshrc) to include the basepath
Basepath = D:/VisualSim/

Download Model and Class files
Ex: C:/Plan/VS/Project1

Set CLASSPATH (in Windows – system environment and Linux - .cshrc) to include the basepath
Basepath = C:/Plan/VS/

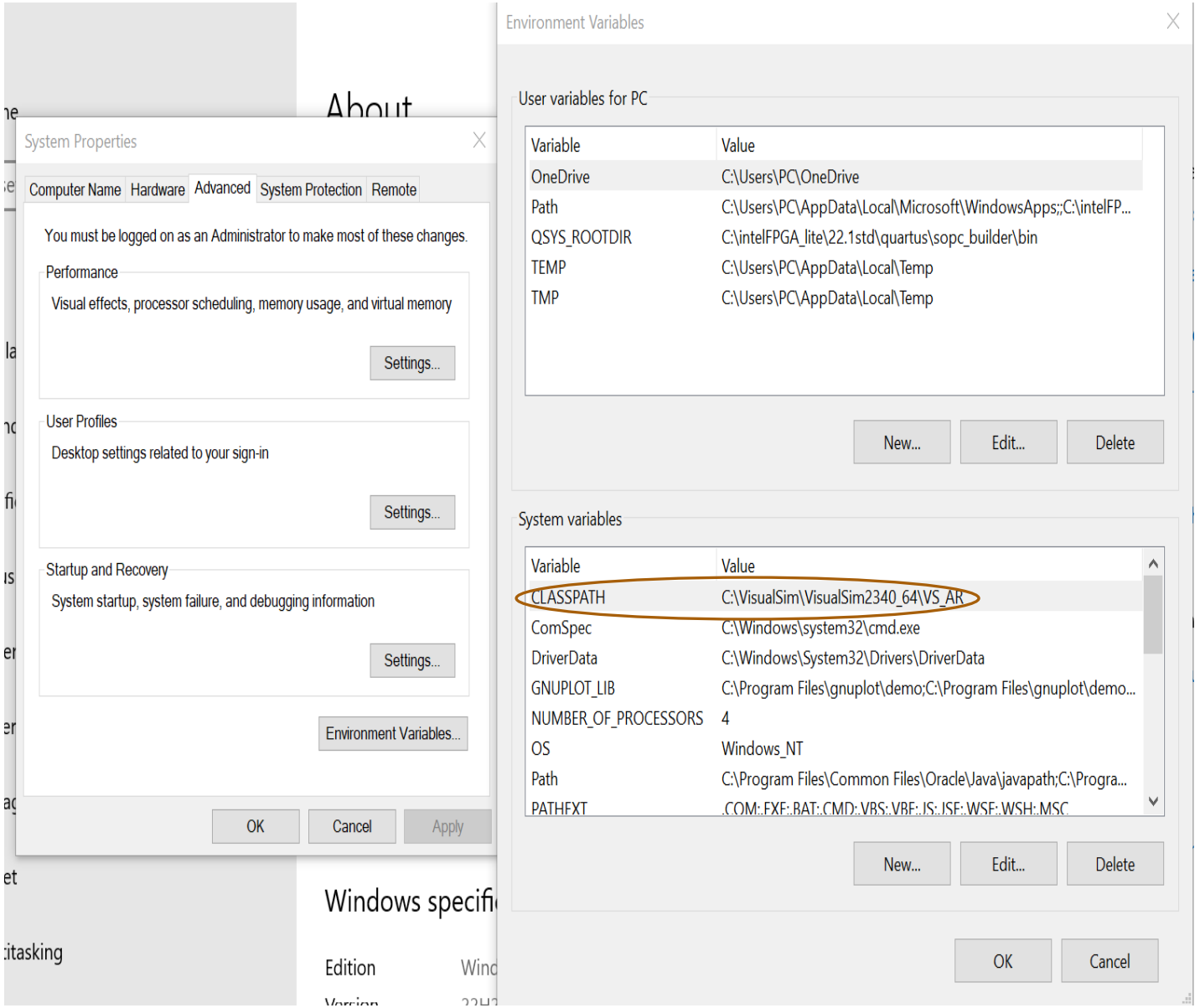
From the Basepath, it will be a standard path for all users

Adding Environmental variable in Windows:

1. Right-click on the Start Button
2. Select “System” from the context menu.
3. Click “Advanced system settings”
4. Go to the “Advanced” tab
5. Click “Environment Variables…”
6. Click “New”
7. Enter the path for the variable that is to be used.
8. For example,
9. Variable : CLASSPATH
10. Value: C:\VisualSim\VisualSim2410_64\VS_AR

In Linux :

1. Open the .bashrc file and add the export command
2. For example, Export CLASSPATH= C:/VisualSim/VisualSim2410_64/VS_AR



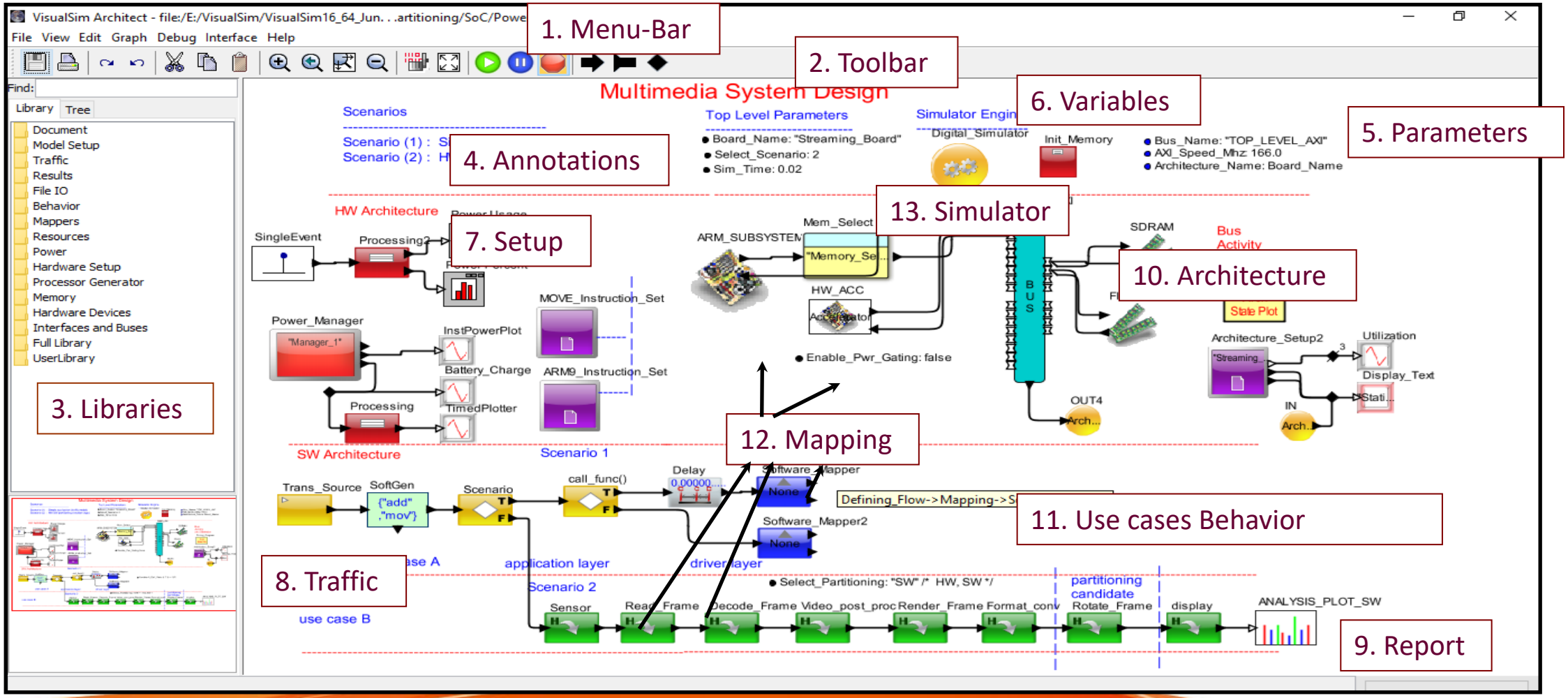
Support

online_support@mirabilisdesign.com

Skype, Whatsapp, Wechat (China)

Teams, Zoom

Look and Feel of the Hardware and Behaviour in VisualSim



VISUALSIM TRAINING