

Siener

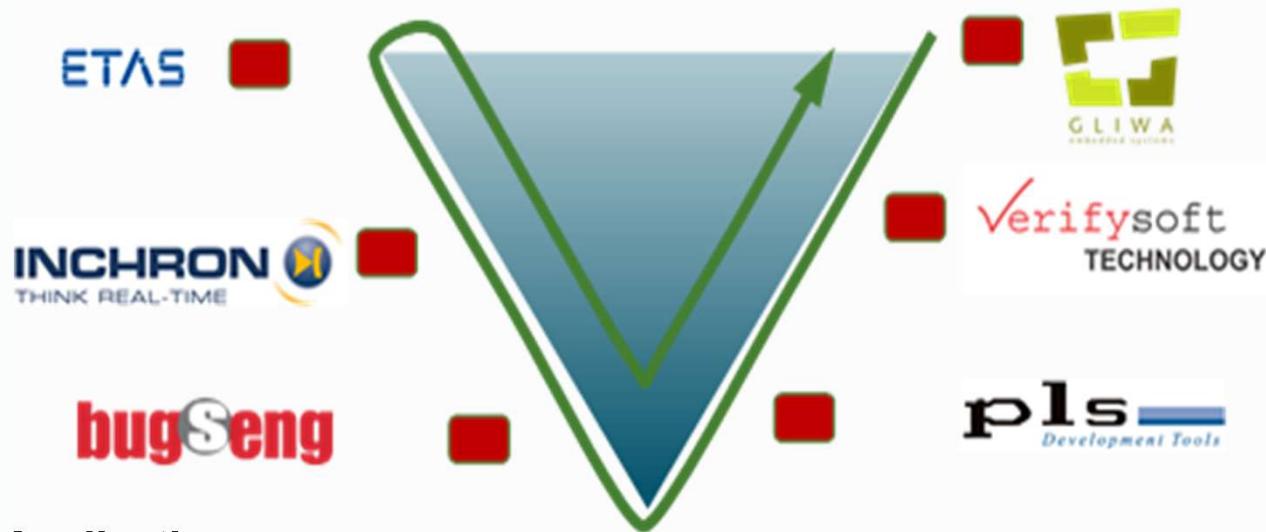


Feb. 2020

王治国

北京西能电子科技发展有限公司

Siener Embedded Tools World

**Focus Applications:**

- ✓ BMS, Inverter and VCU in HEV/EV
- ✓ ECU for Powertrain in Diesel Engine and Gasoline Engine
- ✓ ECU for Function Safety (ISO 26262 and IEC 61508)
- ✓ ECU Compliance AUTOSAR
- ✓ ADAS and Fuel Cell Vehicle



INCHRON公司介绍

INCHRON Tool-Suite产品介绍

ChronSIM with INCHRON Tool-Suite

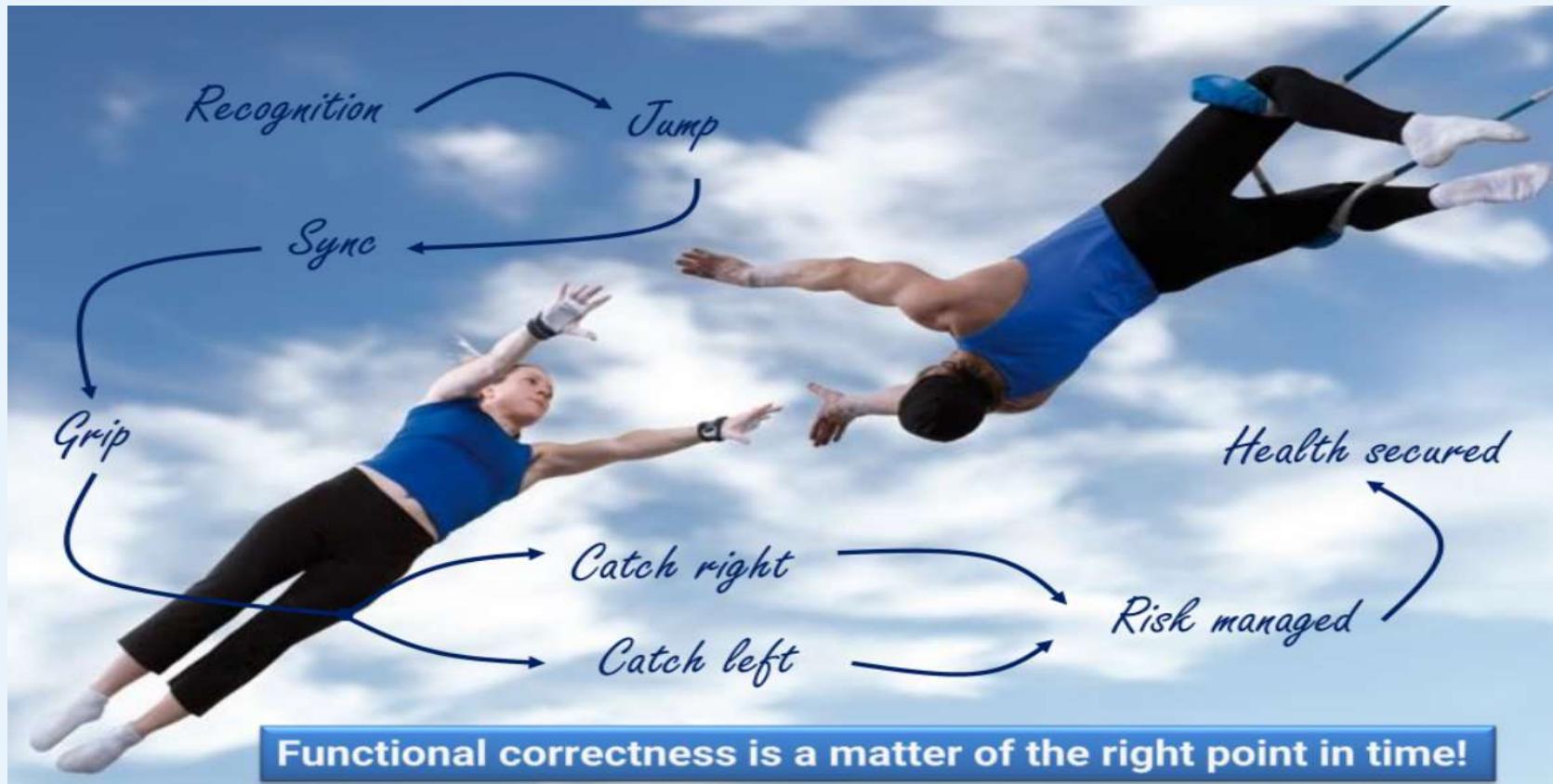
Multi-core in AUTOSAR OS

ADAS Radar Example



The Real-Time Aspect

Siener





INCHRON是全球领先实时系统架构设计和自动优化解决方案提供商, INCHRON Tool-Suite提供了强大的功能:

- 涵盖了从单核到多核再到多CPU到分布式系统的汽车电气架构
- 建立嵌入式系统架构的设计模型, 包括C/C++语言输入
- 基于时间特性仿真, 对嵌入式系统的功能和动态性能进行分析, 并自动优化
- 支持AUTOSAR Task, runnable, ISR, IOC, Semaphor, Spinlock, 核间任务激活
- 支持Hypervisor, CAN/Ethernet/Flexray
- 支持状态图, 甘特图, 历史统计, 事件链, 负荷率、需求图等图形显示
- 支持3rd party trace信息输入, eg. GLIWA T1, PLS trace, Lauterbach Trace
- 支持Python script脚本命令, 支持集成IBM Rational DOORS需求管理





Successfully Proven by Our Customers

Siener

© INCHRON GmbH 2019



Audi



DAIMLER



HYUNDAI



RENAULT
Passion for life



BOSCH



DELPHI



veoneer



+ undisclosed customers

INCHRON Confidential



Real-Time Solutions

Siener



**Powered by
diagnostic methods
and tools from**

INCHRON 





Seamless Integration of Tools and Services

Siener



pls
Development Tools



ESG
expresslogic

SOLIN AG
Technologien verbinden



**Strong
Partners**



HANCOM
HANCOM MDS

OPENSYNERGY

IBM

LAUTERBACH
DEVELOPMENT TOOLS

i SYSTEM

inaya



INCHRON公司介绍

INCHRON Tool-Suite产品介绍

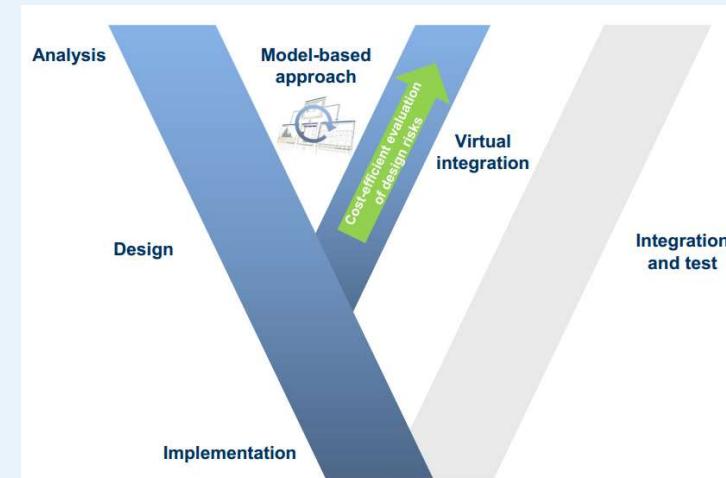
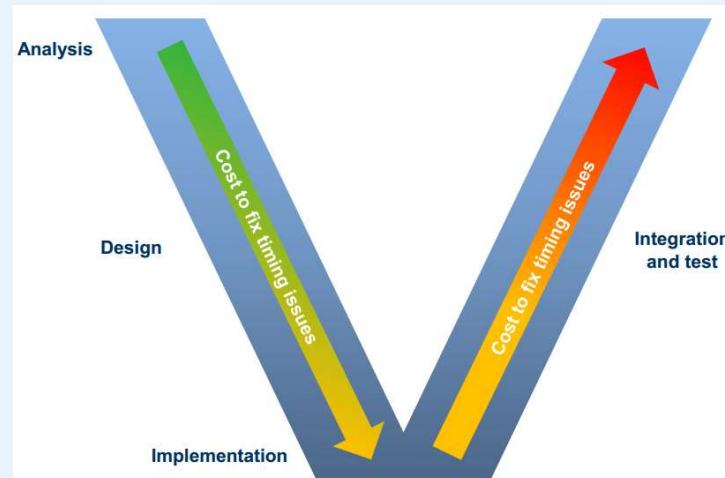
ChronSIM with INCHRON Tool-Suite

Multi-core in AUTOSAR OS

ADAS Radar Example



- INCHRON Tool-Suite是一个嵌入式系统的设计、诊断和测试工具
- 基于MS Windows的应用程序
- 在虚拟原型上仿真、分析和深入预测嵌入式系统（含C源代码）的功能和动态行为





INCHRON Tool-Suite的典型用户

Siener

INCHRON Tool-Suite适合以下用户：

- 系统工程师：在嵌入式系统的设计阶段使用INCHRON Tool-Suite来寻找最佳系统架构。
- 开发工程师：在开发阶段使用INCHRON Tool-Suite来分析和优化嵌入式软件或所选软件模块的动态行为。
- 测试工程师：在测试阶段使用INCHRON Tool-Suite验证动态行为，以验证嵌入式系统的实时功能。

INCHRON Tool-Suite的基础专业性：

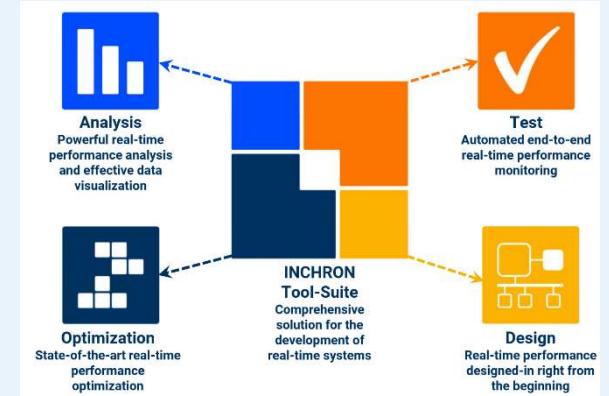
- 电气工程或计算机科学。
- 熟悉嵌入式系统和C编程语言，以及相应的实践经验



INCHRON Tool-Suite典型应用领域

Siener

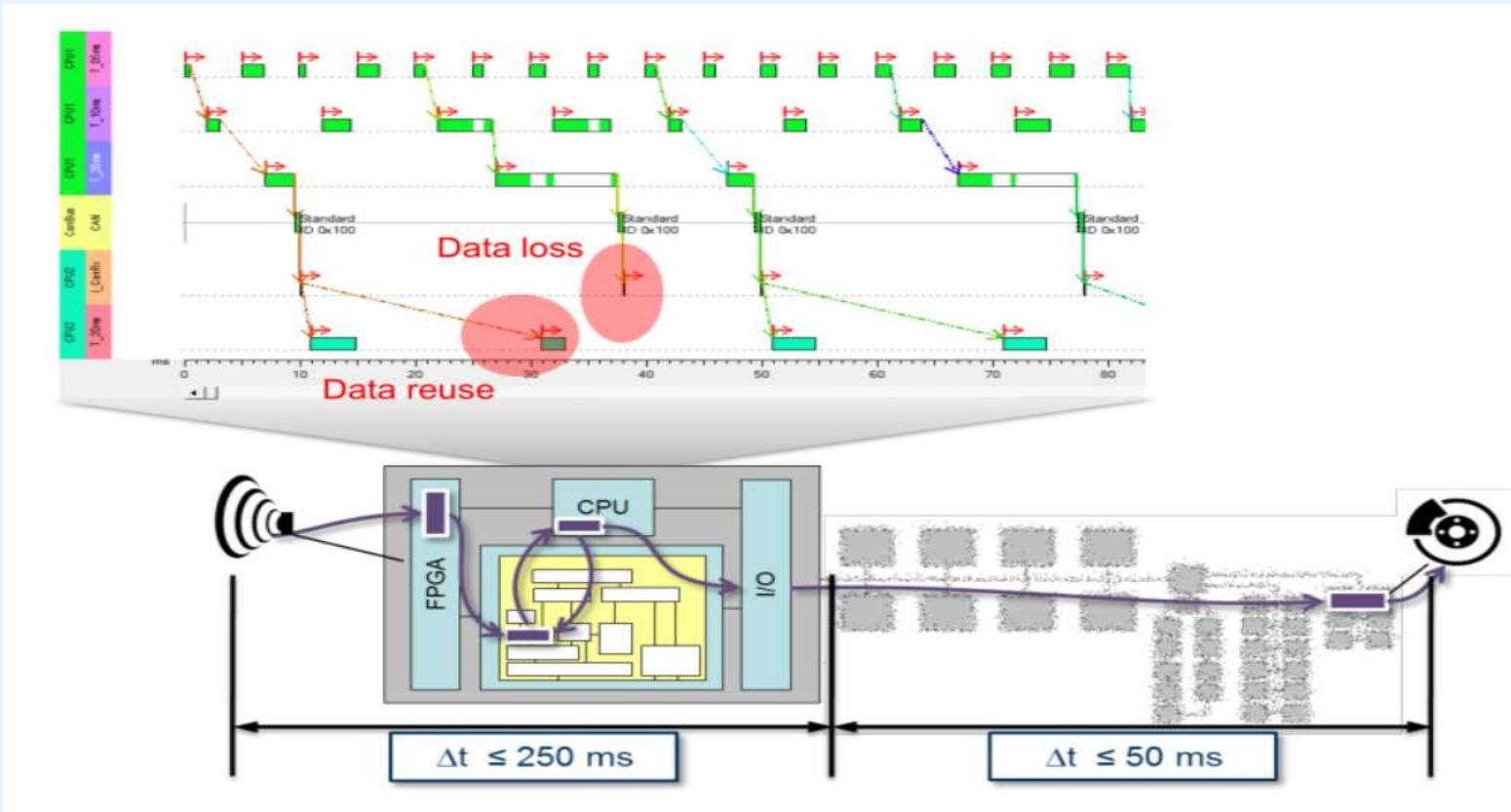
- 设计嵌入式系统
 - ◆ 设计时间预算、系统需求和初始软件架构
 - ◆ 仿真和分析，对系统功能和动态性能进行评估
- 嵌入式软件的分析与测试
 - ◆ 优化架构后，编写代码替代软件架构的时间预算
 - ◆ 分析系统动态行为，预防时间问题，优化并验证系统需求
- 已有代码的重用
 - ◆ 将已有代码和新的架构集成，评估已有代码在新的系统架构中的效果
- 嵌入式系统的扩展功能的评估
 - ◆ 容易改变系统设置和变量值
 - ◆ 避免扩展功能对系统功能和动态性能的不利影响





Safety-Critical Systems: Design Challenges

Siener





ISO 26262 Part 4 and 6 Recommend Simulation

Siener

Table 3 — System design verification

| Methods | ASIL | | | |
|--|-------------|----|----|----|
| | A | B | C | D |
| 1a System design inspection ^a | + | ++ | ++ | ++ |
| 1b System design walkthrough ^a | ++ | + | 0 | 0 |
| 2a Simulation ^b | + | + | ++ | ++ |
| 2b System prototyping and vehicle tests ^b | + | + | ++ | ++ |
| 3 System design analyses ^c | see Table 1 | | | |

^a Methods 1a and 1b serve as a check of complete and correct implementation of the technical safety requirements.

^b Methods 2a and 2b can be used advantageously as a fault injection technique.

^c For conducting safety analyses, see ISO 26262-9:2011, Clause 8.



ISO 26262 Part 4 and 6 Recommend Simulation

Siener

Table 6 — Methods for the verification of the software architectural design

| | Methods | ASIL | | | |
|----|--|------|----|----|----|
| | | A | B | C | D |
| 1a | Walk-through of the design ^a | ++ | + | o | o |
| 1b | Inspection of the design ^a | + | ++ | ++ | ++ |
| 1c | Simulation of dynamic parts of the design ^b | + | + | + | ++ |
| 1d | Prototype generation | o | o | + | ++ |
| 1e | Formal verification | o | o | + | + |
| 1f | Control flow analysis ^c | + | + | ++ | ++ |
| 1g | Data flow analysis ^c | + | + | ++ | ++ |

^a In the case of model-based development these methods can be applied to the model.

^b Method 1c requires the usage of executable models for the dynamic parts of the software architecture.

^c Control and data flow analysis may be limited to safety-related components and their interfaces.

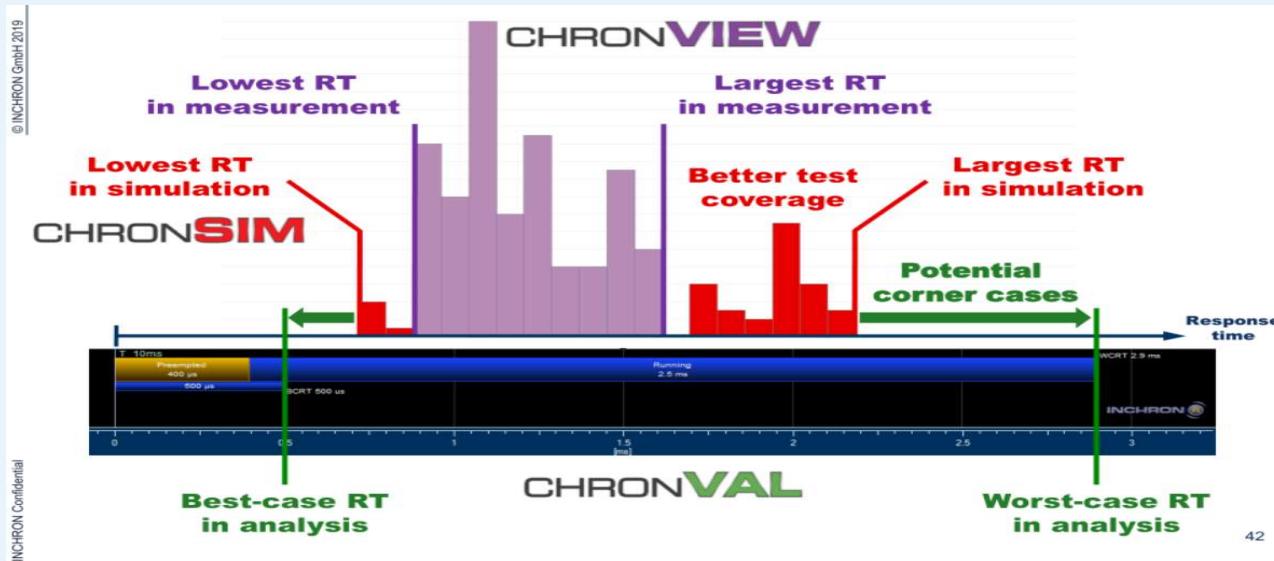


INCHRON Tool-Suite工具套件的主要license

Siener

- CHRONSIM
- CHRONVAL
- CHRONVIEW
- CHRONOPT
- CHRONBUS

- 单机版
- 浮动版
- 年软件升级和技术服务





- User Manual _ INCHRON Tool-Suite Software Version 2.9
- AN001 Reference Manual of Macros and Functions
- AN003 Importing Trace Files with chronVIEW
- AN002 User Defined Schedulers
- Online help

- Examples
 - doc\examples 安装文件夹下

- Demo
 - AUTOSAR OS
 - Multi-Cor Optimization
 - ADAS Radar SDF

The screenshot shows a Windows-style application window for the INCHRON Tool-Suite User Manual. The title bar reads "File Edit View Go Bookmarks Help". Below it is a toolbar with various icons. The main area has tabs for "Contents", "Index", and "Bookmarks", with "Contents" currently selected. A search bar includes fields for "Search" and "Chapter 3: Basics".
Table of Contents:
Chapter 3. Basics
3.1. INCHRON Tool-Suite Features
3.2. INCHRON Tool-Suite Modes of Operation
 3.2.1. The Project Mode
 3.2.2. The Simulation Mode
 3.2.2.1. Functions for Simulation Control
 3.2.2.2. Functions on Simulation Trace Files
 3.2.2.3. Functions for Simulation Visualization
 3.2.2.4. Usage of the Simulation Control Window
 3.2.3. The Validation Mode
 3.2.4. The Viewer Mode

The following chapter gives you a brief introduction in the structure and handling of INCHRON Tool-Suite.

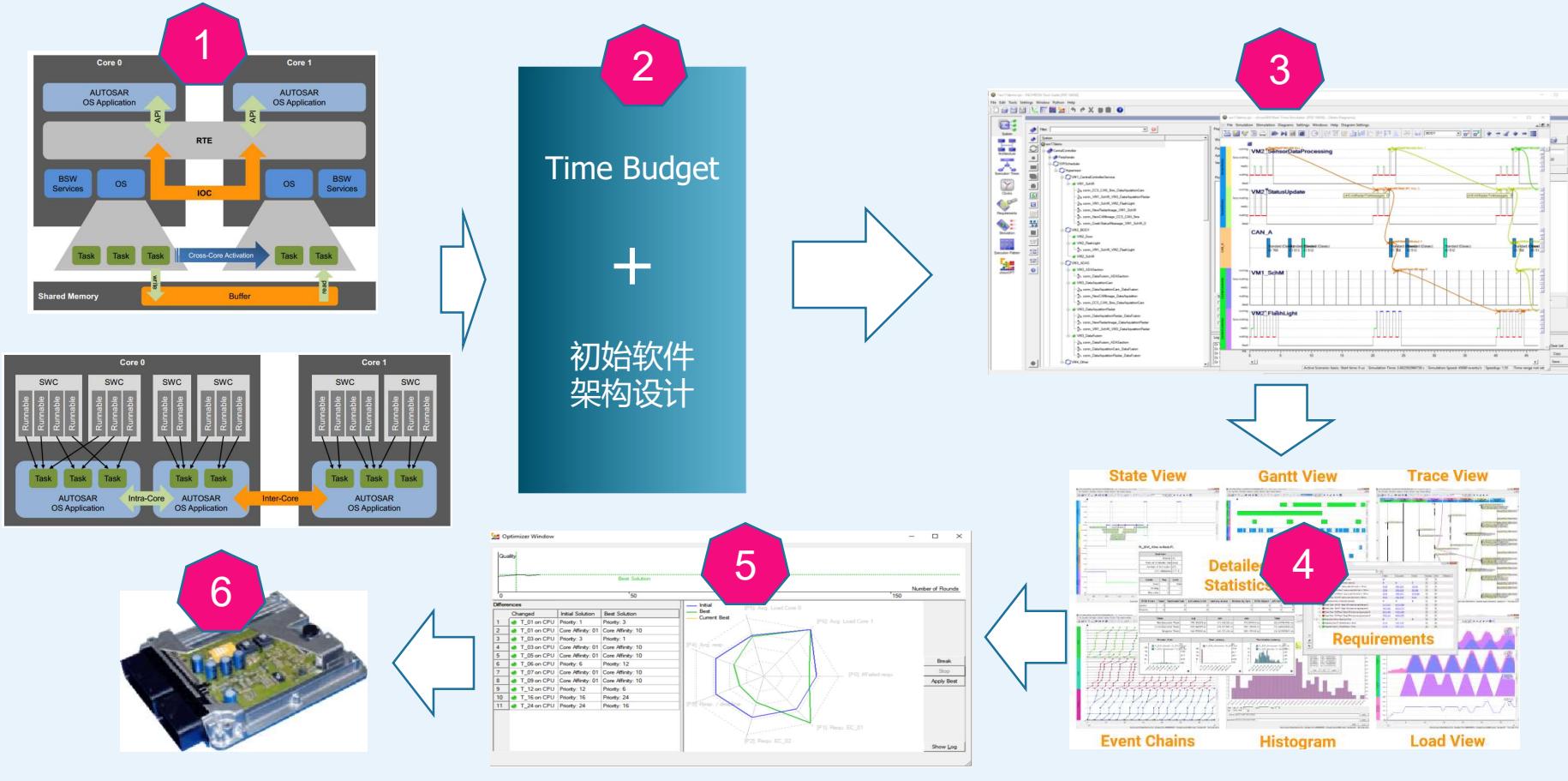
3.1. INCHRON Tool-Suite Features

The term system architecture is always regarded as the view on the target system. An INCHRON model of the target system consists of:



使用INCHRON Tool-Suite工具流程

Siener





INCHRON Tool-Suite各文件格式

Siener

- *.IPR --INCHRON Project file
- *.ISM--INCHRON Target Models file
- *.ISL--Stimulation Scenarios file
- *.IRQ--Requirements file
- *.IVP--View Profiles file
- *.CSV or *.TXT--Stimulation Data file

- *.ISF-- INCHRON simulation trace data
- *.ITA--INCHRON Trace Archive
- *.IDX--INCHRON simulation trace index data

- *.HTML, *.CSV, *.CVR, *.IPR--File Format for Reports

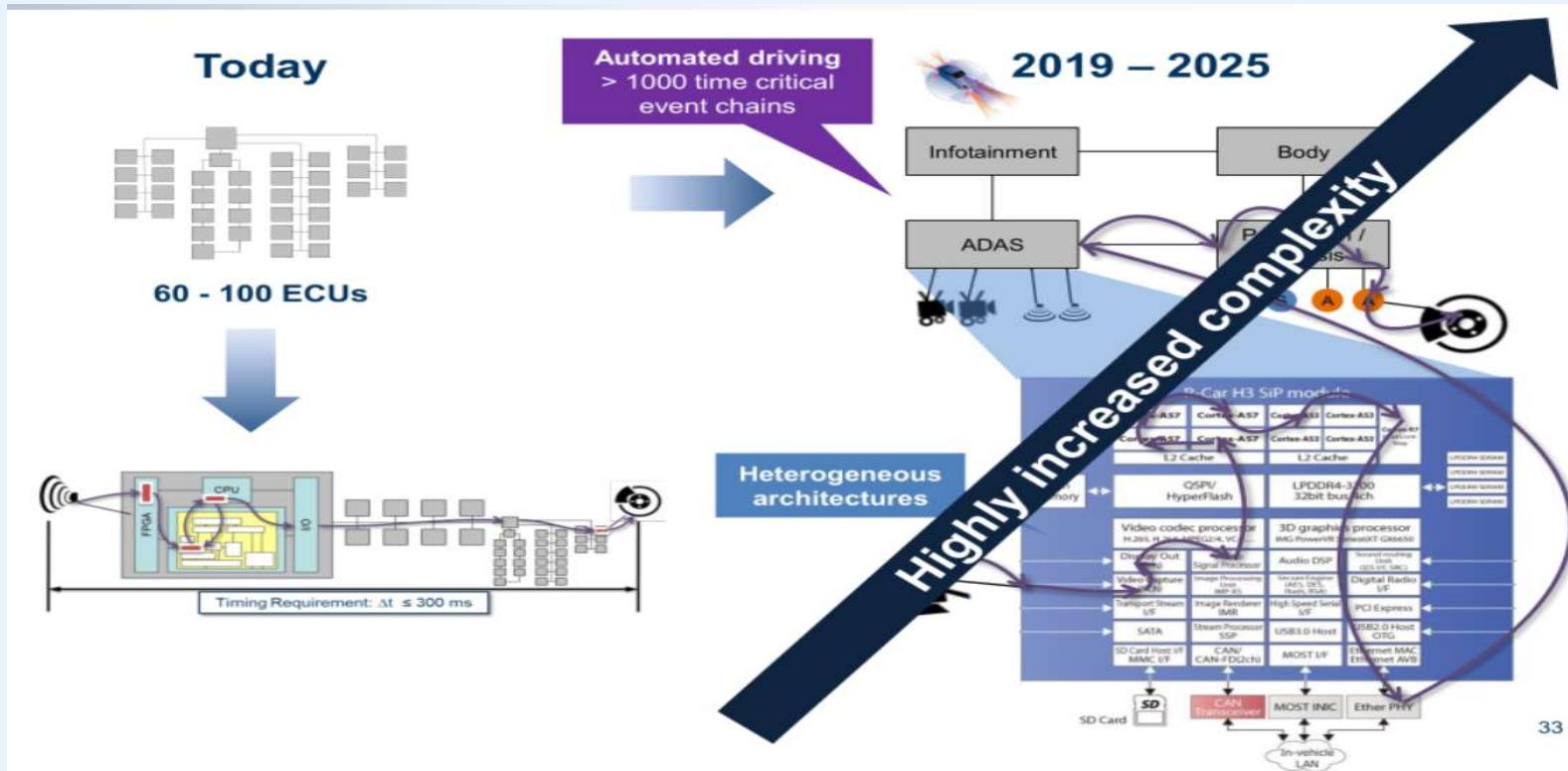
- *.EPS, *.JPG, *.PNG, *.PS, *.TIF--Graphic Formats



使用INCHRON Tool-Suite工具套件的好处

Siener

未来汽车电子构架会走向更集成化：

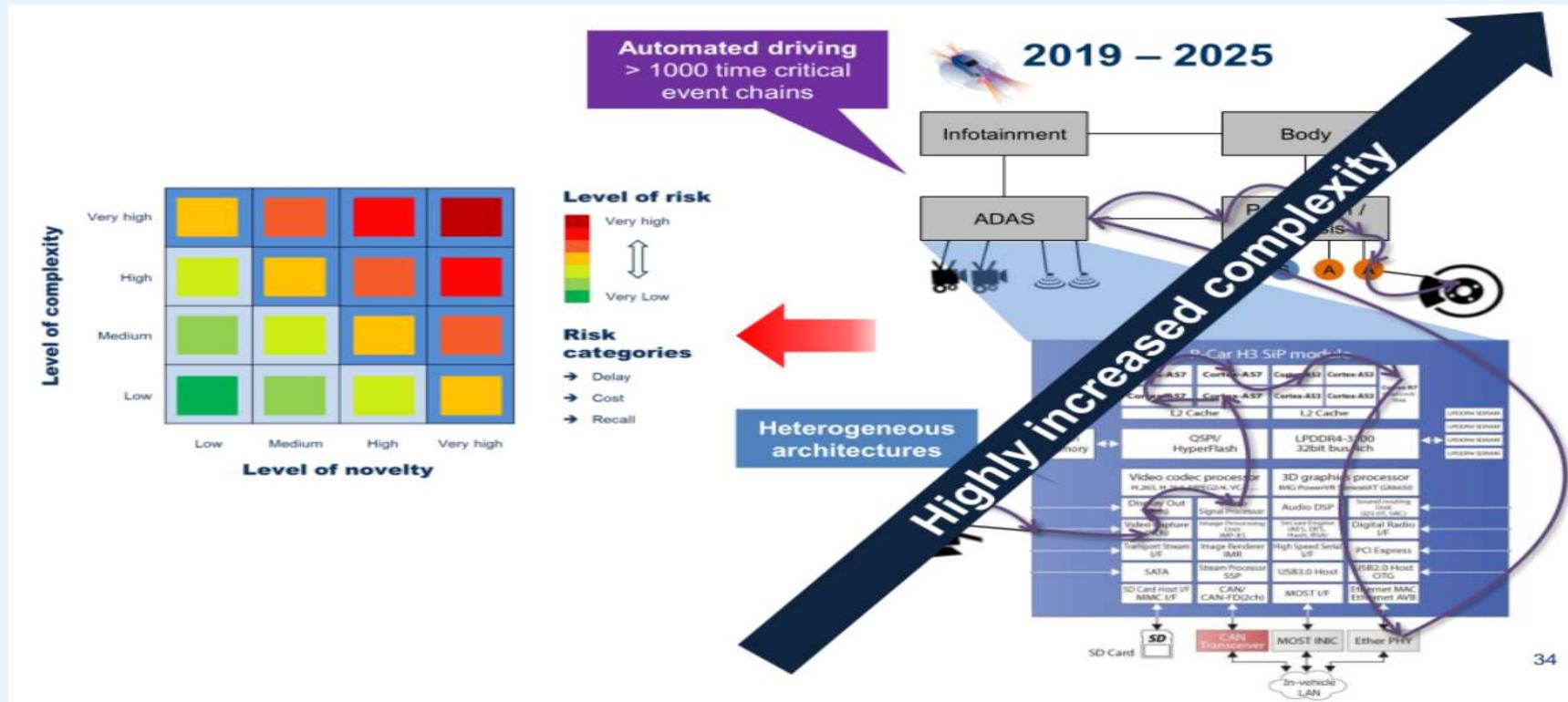




使用INCHRON Tool-Suite工具套件的好处

Siener

项目风险评估越来越高：

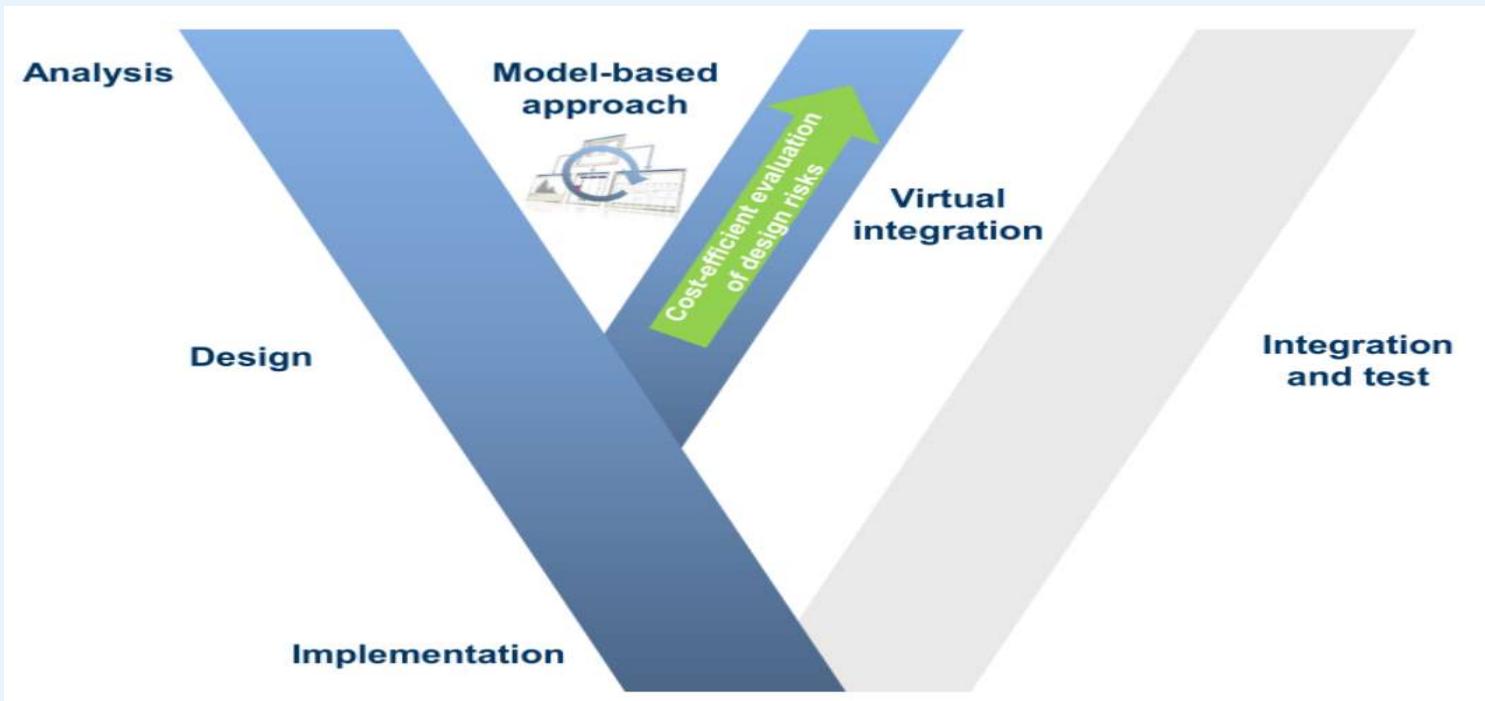




使用INCHRON Tool-Suite工具套件的好处

Siener

通过INCHRON Tool-Suite工具套件可以在早期就能检查出时间错误：





INCHRON Tool-Suite工具套件的好处

Siener

INCHRON Tool-Suite使您及早能够识别时间问题，并消除设计缺陷，从而带来以下受益：

➤ 缩短产品上市时间

在构建真正的原型之前，INCHRON Tool-Suite使系统工程师在开发过程中尽早分析、优化和验证设计，减少重新设计，直接缩短产品上市时间。

➤ 减少研发支出

工程师可以分析和优化设计，而无需构建原型，减少重新设计，避免项目延迟和缩短产品上市时间，会直接减少研发成本。

➤ 降低产品成本

INCHRON Tool-Suite可在不同的硬件架构上模拟嵌入式软件，使工程师能够选择最具成本效益的设计。

➤ 降低产品召回风险

通过避免实时问题提高了嵌入式系统的质量。工程师可以在开发阶段检测并消除实时问题，从而大大降低产品召回的风险。

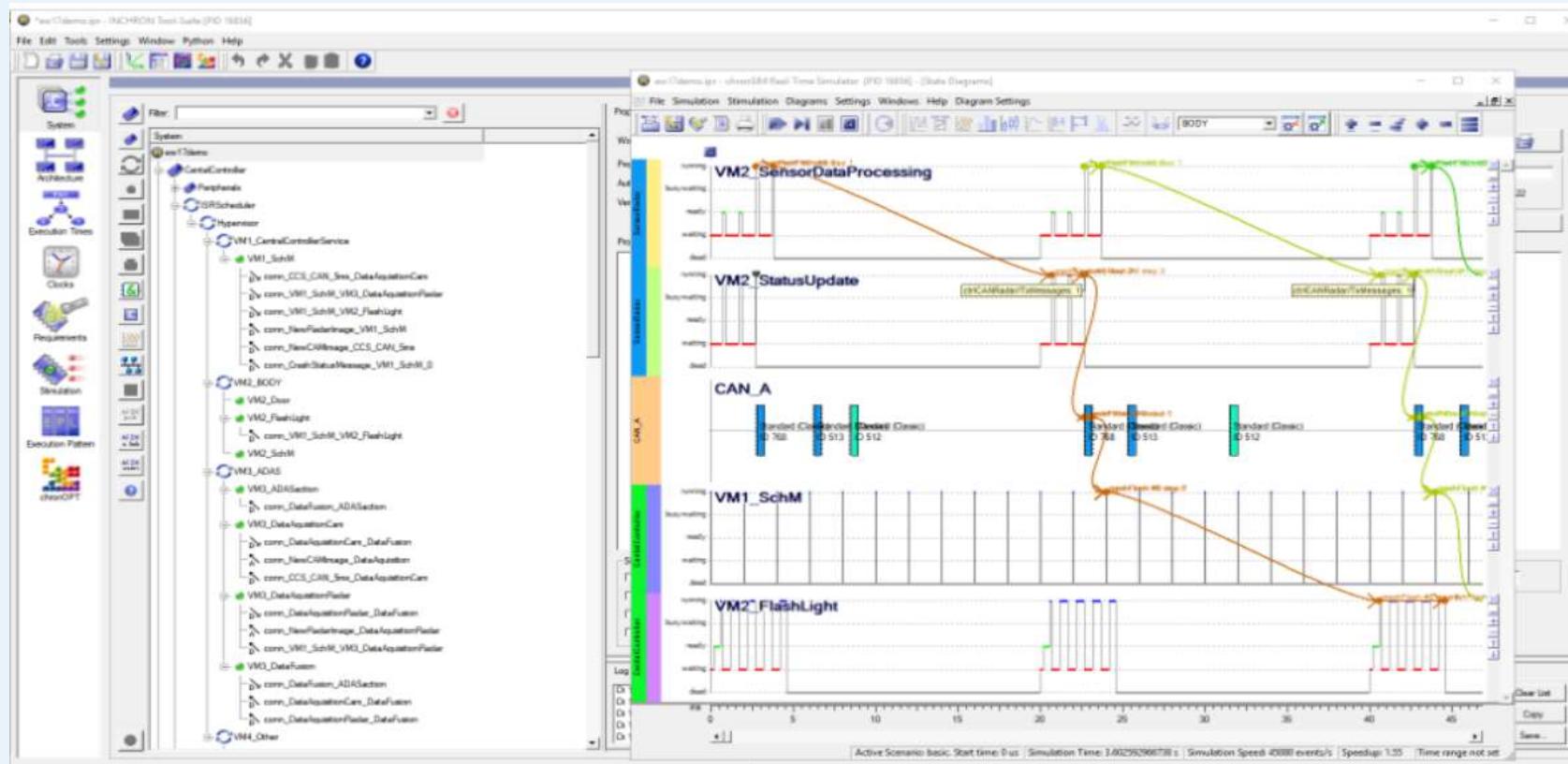
➤ 新产品功能的影响

INCHRON Tool-Suite可以快速回答是否可以将新功能集成到现有系统架构中，还是需要进行重大调整。



Design: Example

Siener

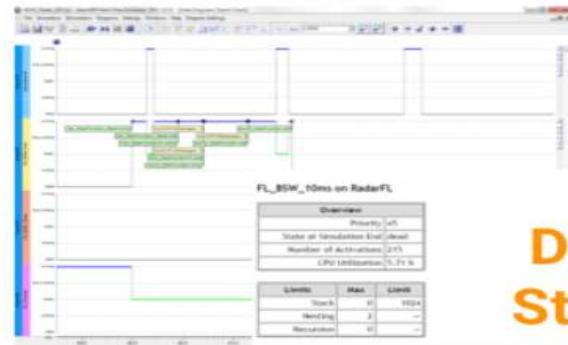




Visualization & Analysis Deliver Deep Insights

Siener

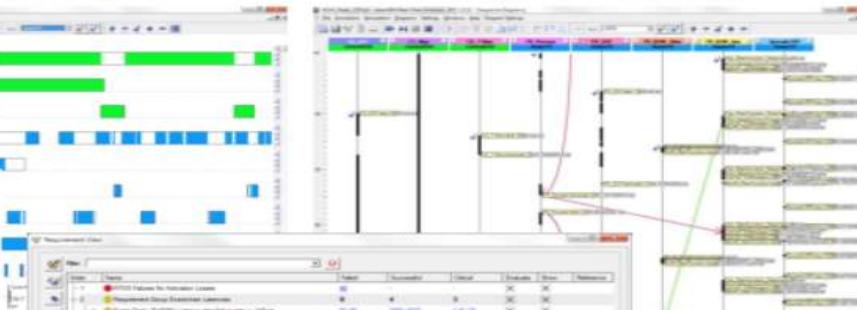
State View



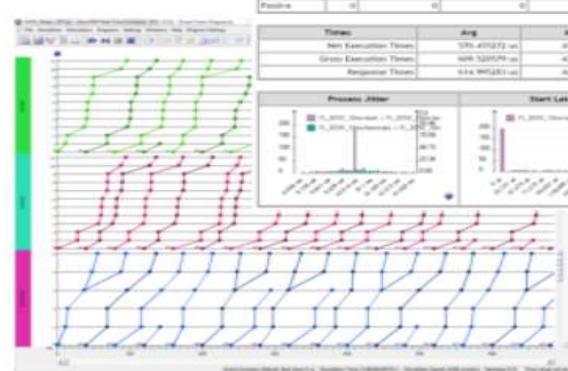
Gantt View



Trace View

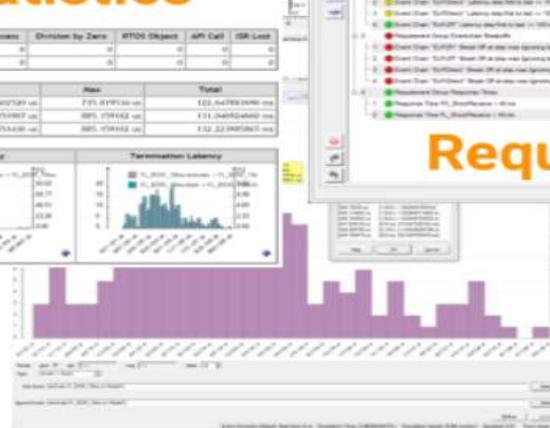


Detailed Statistics



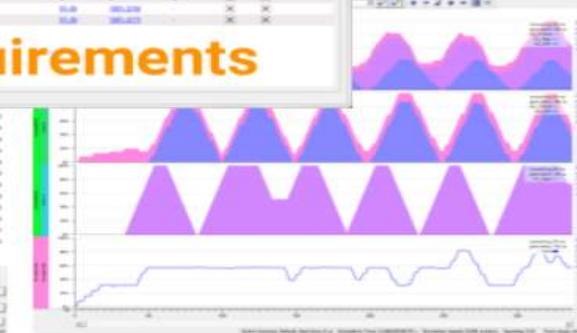
Event Chains

Requirements



Histogram

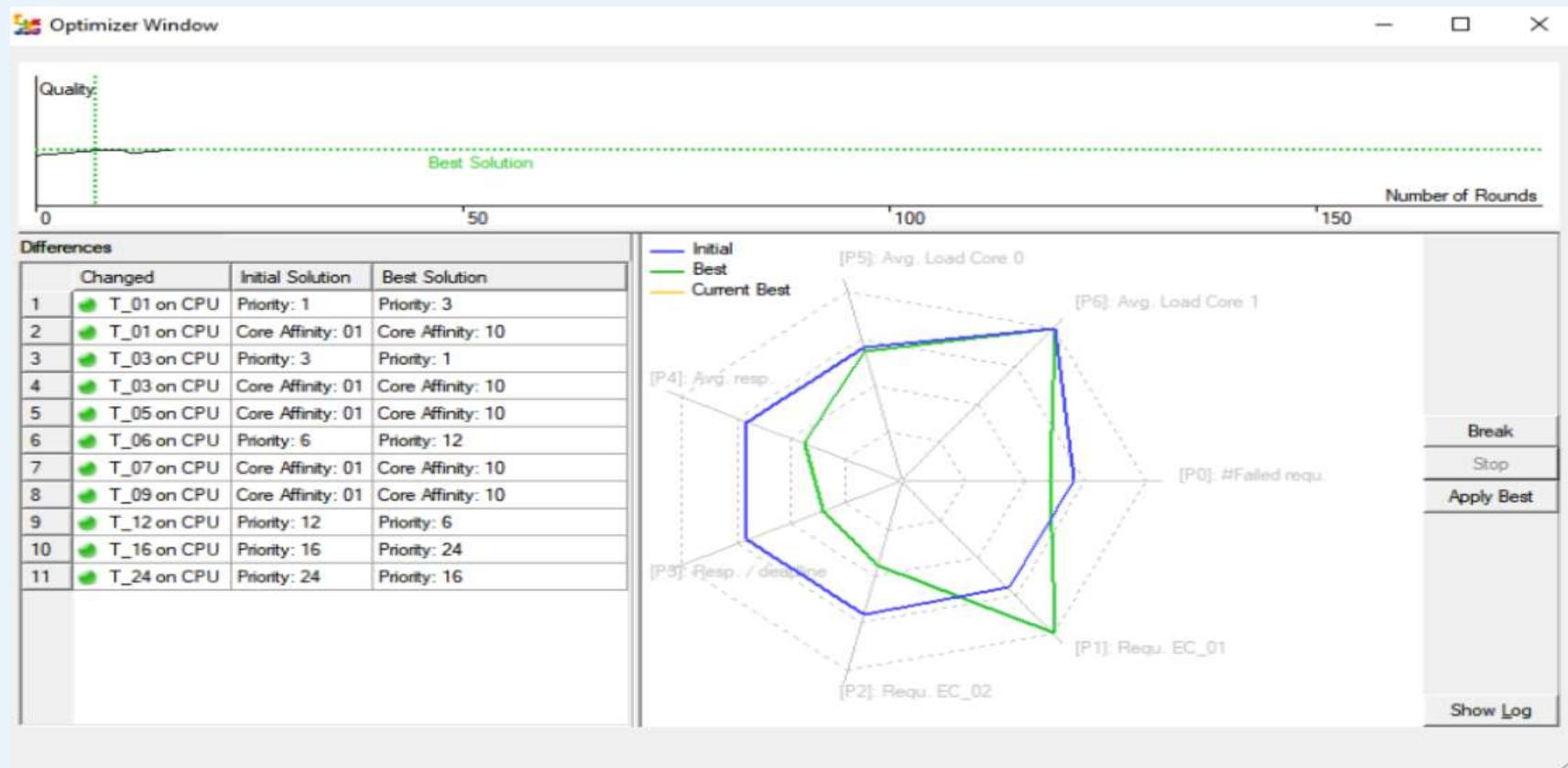
Load View





Optimization (Example)

Siener





ADAS--Heterogenous Distributed Systems

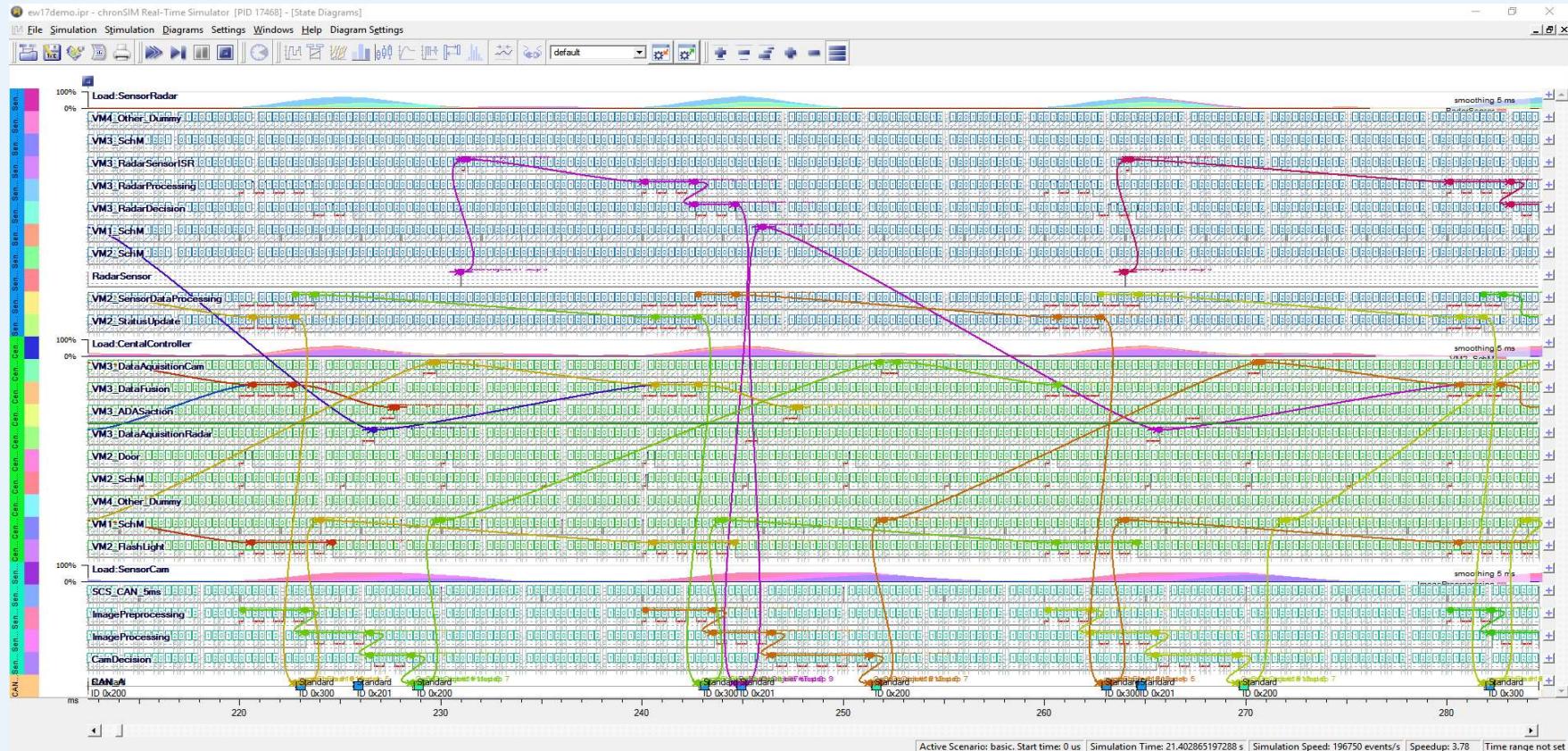
Siener





Analyzing Distributed Systems Comprising Hypervisors

Siener





INCHRON公司介绍

INCHRON Tool-Suite产品介绍

ChronSIM with INCHRON Tool-Suite

Multi-core in AUTOSAR OS

ADAS Radar Example



Scheduling

- OSEK, AUTOSAR OS, cooperative / preemptive, fixed-priority, round robin with time slices, EDF, TDMA, Thread X, ARINC 653
- Dedicated scheduling API for the definition of any custom scheduling algorithm

Simulation / Worst-case analysis

- Full support of multi-core and multi-MCU systems
- Multiple non-synchronized time bases
- CAN, Flexray, LIN, Ethernet (with TCP/UDP over IP and QoS), AS5643, AFDX, SPI, UART
- chronSIM fully supports C/C++ as a modeling language and provides various APIs for the definition of customer specific models
- Stochastical distribution models for CET and stimulation
- Fault injection

Requirements

- Formalized timing requirement definitions
- Event chains



INCHRON Tool-Suite:chronSIM

Siener

Scheduling parameters depending on the CPU model, the scheduling type, and scheduling strategy

Net execution time behavior

Data and activation flows (inter process communication)

CPUs, peripherals, processes, functions, messages etc.

The screenshot shows the INCHRON chronSIM interface. On the left, there is a project tree under 'System' labeled 'MyProject'. It includes a 'MyCpu' node with 'ISRScheduler' and 'Default Scheduler' children, and an 'Interrupts' node with 'NewISR' child. A 'NewProcess' node is selected and highlighted in blue. On the right, a configuration dialog for 'NewProcess' is open, divided into 'General', 'Timing', and 'Connections' tabs. The 'General' tab is active, showing fields for 'Name' (set to 'NewProcess'), 'Entry Function' (set to '(undefined)'), and various scheduling checkboxes like 'Active at Boot' (checked) and 'Preemptable' (checked). It also includes 'Activation Limit' (set to 1), 'Priority' (set to 0), 'Stack Limit' (set to 1024 bytes), and 'Stack Group' (set to 'none'). Below this is a 'Runnable Entities' section with a table header 'Name Order'. At the bottom of the dialog are buttons for 'Edit Core Affinity...' and 'Edit Scheduling Points...'. A large orange text overlay at the bottom center reads: 'Goal: Predict the system behavior based on a model!'



ChronSIM: Abstraction level

Siener

A chronSIM model can comprise different levels of abstraction:

- Tasks and ISRs
- Tasks, ISRs, and Runnables (basically first-level functions under a task or ISR)
- Tasks, ISRs, and Runnables with C/C++ implementation

These abstraction levels can be combined as required by the progress of a project and the availability of sophisticated model information.

The screenshot shows the ChronSIM software interface. On the left, there is a tree view of the system structure:

- System
- MyProject
 - MyCpu
 - ISRScheduler
 - DefaultScheduler
 - MyProcess
 - TranslationUnits
 - Sources.c
 - MyFunction
 - MyProcess

A blue callout box on the left states: "C/C++ can be used to implement the simulation behavior of processes and functions". An arrow points from this box to the "Sources.c" node in the tree.

On the right, a detailed configuration dialog for the "MyProcess" task is open:

| General | Timing | Connections | | | | |
|---|--|-------------|------|-------|--|--|
| Name: MyProcess | Entry Function: MyProcess() in Sources.c | | | | | |
| <input checked="" type="checkbox"/> Active at Boot | <input type="checkbox"/> Idle Task | | | | | |
| <input checked="" type="checkbox"/> Preemptable | | | | | | |
| Activation Limit: 1 | Stack Limit: 1024 bytes | | | | | |
| Priority: 0 | Stack Group: none | | | | | |
| Runnable Entities | | | | | | |
| <table border="1"><thead><tr><th>Name</th><th>Order</th></tr></thead><tbody><tr><td></td><td></td></tr></tbody></table> | | | Name | Order | | |
| Name | Order | | | | | |
| | | | | | | |

A blue callout box on the right states: "Here, the simulation behavior of the task MyProcess is defined by the implementation given in Source.c". An arrow points from this box to the "Entry Function" field in the dialog.



ChronSIM: Abstraction level

Siener

The screenshot shows the INCHRON Tool-Suite interface. On the left, the project structure is displayed under the 'System' tab, showing a hierarchy from 'MyProject' down to 'Sources.c' which contains 'MyFunction' and 'MyProcess'. A callout box highlights this structure with the text: "Model elements in the INCHRON Tool-Suite must be valid C/C++ identifiers even if no code is used!"

In the center, a callout box points to the 'Timing' tab of the 'MyProcess' task configuration dialog. It contains the following text: "In addition to predefined macros from the API, the full C/C++ syntax can be used to specify / implement the behavior of the model!"

The 'Timing' tab displays the following C/C++ code:

```
C Sources.c x
1 #include <autosar.h>
2 #include <chronosim.h>
3
4 void MyFunction()
5 {
6     //function code executed during simulation
7 }
8
9 TASK(MyProcess)
10 {
11     DELAY(300, unit_us);
12     Schedule();
13     while(!finished())
14     {
15         DELAY(gaussian(500, 10),unit_us);
16     }
17     DELAY(datasize*10, unit_us);
18 }
19 }
```

The 'Timing' tab also includes configuration options for the task:

- Name: MyProcess
- Entry Function: MyProcess() in Sources.c
- Active at Boot:
- Idle Task:
- Preemptable:
- Activation Limit: 1
- Stack Limit: 1024 bytes
- Priority: 0
- Stack Group: none

Below the configuration tabs, there is a 'Runnable Entities' section with a table:

| Name | Order |
|------------|-------|
| MyFunction | |
| MyProcess | |



ChronSIM: Abstraction level

Siener

If no implementation is provided, the real-time simulation behavior is determined by the settings in the Timing tab:

The screenshot shows the ChronSIM interface with the 'Timing' tab selected. On the left, a tree view shows a project structure: 'System' > 'MyProject' > 'MyCpu' > 'ISRScheduler' > 'DefaultScheduler' > 'MyProcess'. A blue callout box points from the text below to the 'Simulation Replacement' section of the 'Timing' tab. The 'Timing' tab contains the following settings:

- Execution Time**:
 - Best case: 500 us
 - Worst case: 2 ms
- Simulation Replacement**:
 - WCET (radio button)
 - BCET (radio button)
 - Uniform (radio button)
 - Normal (radio button, selected):
 - Mean: 1 ms
 - Sigma: 300 us
- Blocking Time**:
 - Blocking Time: 0 us

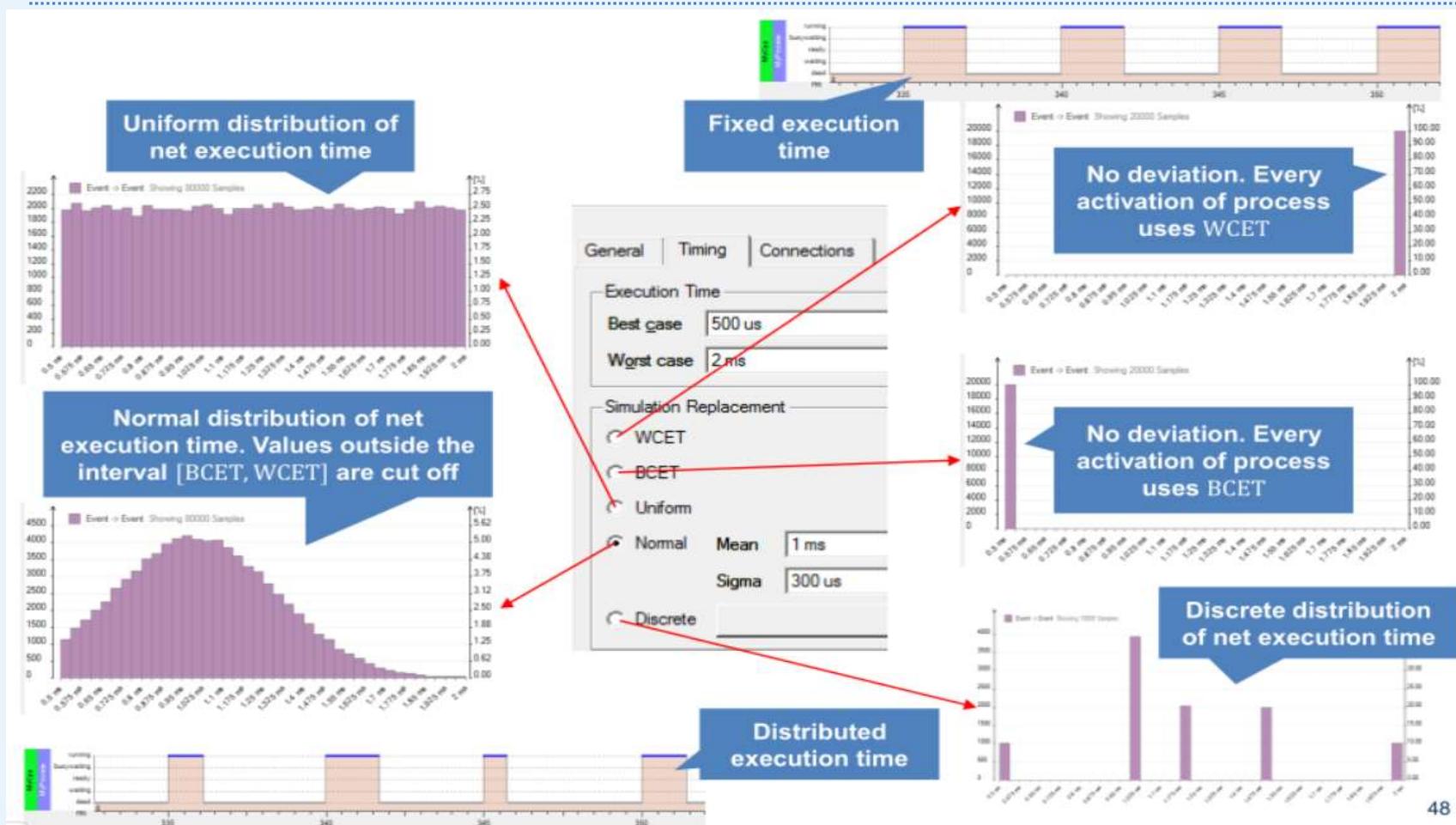
A histogram chart titled "Event -> Event: Using CPU Task" is displayed on the right, showing a bell-shaped distribution of execution times.

Simulation Replacement setting is used to define a specific distribution or fixed value for the net execution time of a process during the simulation



ChronSIM: Execution time distribution

Siener





ChronSIM: Stimulation scenarios

Siener

A Tool-Suite project can contain multiple stimulation scenarios.

In order to use several scenario definitions at the same time, a composite scenario must be defined.

The screenshot shows the 'Stimulation' tab of the ChronSIM software interface. On the left, there's a small icon of a stimulation device with three red dots. Below it, a blue callout box contains the text: 'Stimulation generators are defined in the context of a basic scenario'. In the center, a table titled 'Current Stimulation Scenarios' lists four entries:

| Default | Name | Type | Event Preview | Description |
|-------------------------------------|-----------|----------------------|---|-------------|
| <input type="checkbox"/> | A | Stimulation Scenario | 10 blue downward arrows | |
| <input type="checkbox"/> | B | Stimulation Scenario | 10 blue downward arrows, with the last one having a double-headed vertical arrow below it | |
| <input type="checkbox"/> | C | Stimulation Scenario | 10 blue downward arrows, with the first two having double-headed vertical arrows above them | |
| <input checked="" type="checkbox"/> | Composite | Composite Scenario | 10 mixed red and blue downward arrows | |

Below the table are buttons for 'Add Scenario...', 'Add Composite Scenario...', 'Edit...', and 'Delete'.

Three blue callout boxes provide additional information:

- 'Set the active scenario' points to the 'Default' column of the 'Composite' row.
- 'Composite scenarios are collections of references to a subset of the existing scenarios in a project' points to the 'Event Preview' column of the 'Composite' row.



ChronSIM: Stimulation dialog

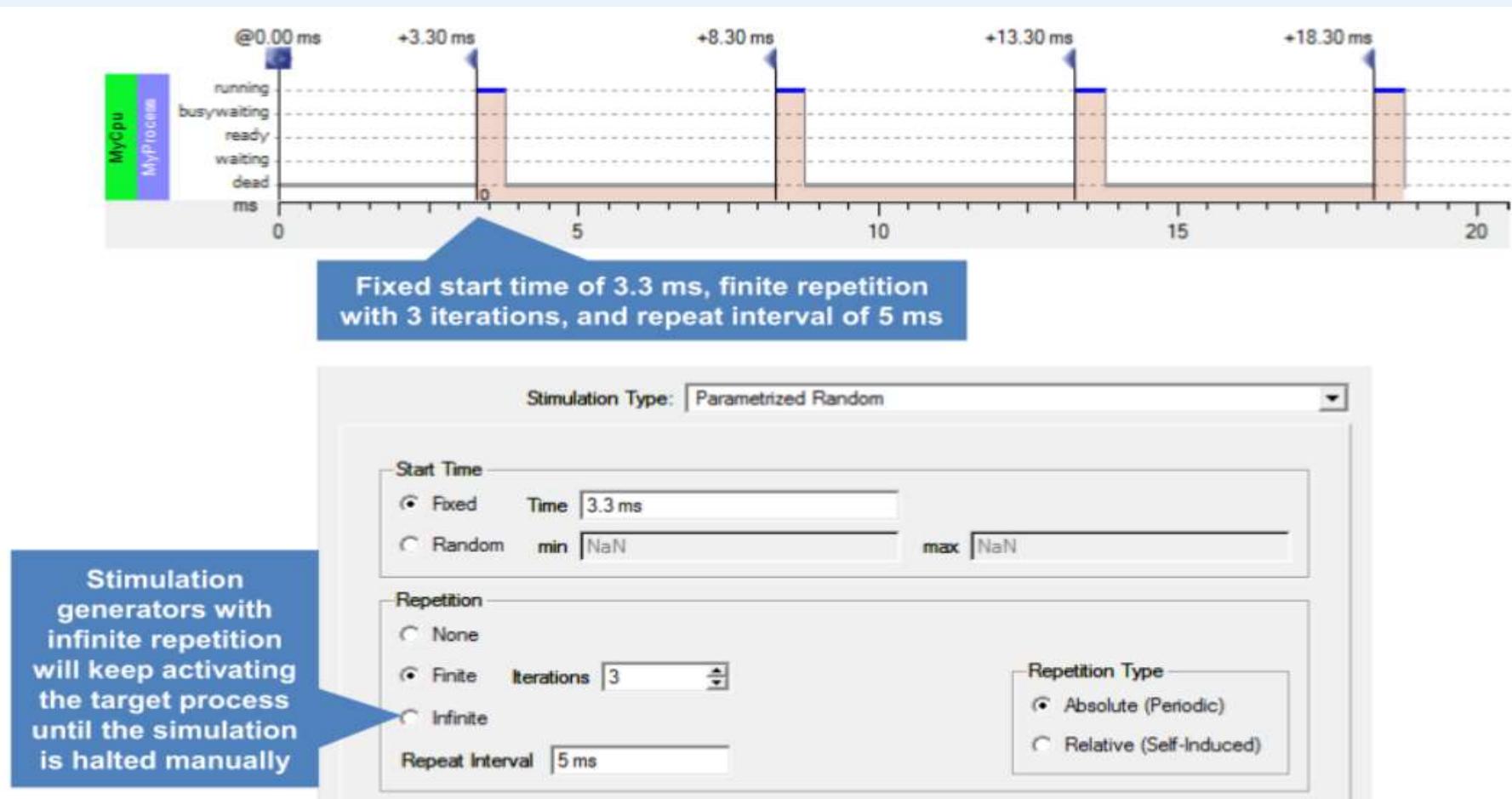
Siener

The screenshot shows the ChronSIM Stimulation dialog with two tabs: "Target" and "Generator". The "Target" tab is selected, showing a stimulation target of "MyProcess on MyCpu" and a clock selection of "MyCpu". A red circle highlights the "Target" tab. A blue callout points to the "Stimulation Target" field with the text: "Depending on the model type, not only processes but also other system tree objects (e.g. runnables or messages) can be valid stimulation targets". Another blue callout points to the "Clock" dropdown with the text: "The clock which defines the time base of this stimulation generator". The "Generator" tab is also shown, with a dropdown menu open for "Stimulation Type", listing options like "Parametrized Random", "Burst Pattern", "Read from CSV File", etc. A red circle highlights the "Generator" tab. A blue callout points to the "Start Time" section with the text: "Start time and repetition are basic parameters of most time-triggered stimulation generators".



ChronSIM: Parametrized random stimulation

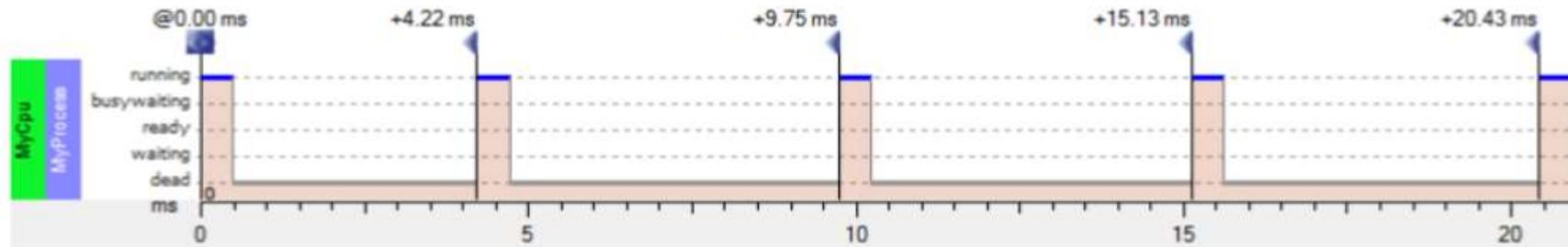
Siener





ChronSIM: Absolute (periodic) with normal distribution

Siener



Repetition

None

Finite Iterations

Infinite

Repeat Interval

Random Variation

None

Normal sigma

Uniform min max

Limitation

Minimum Inter Arrival Time

The repeat interval is also the mean value for the normal distribution

The minimum time distance between two consecutive activations

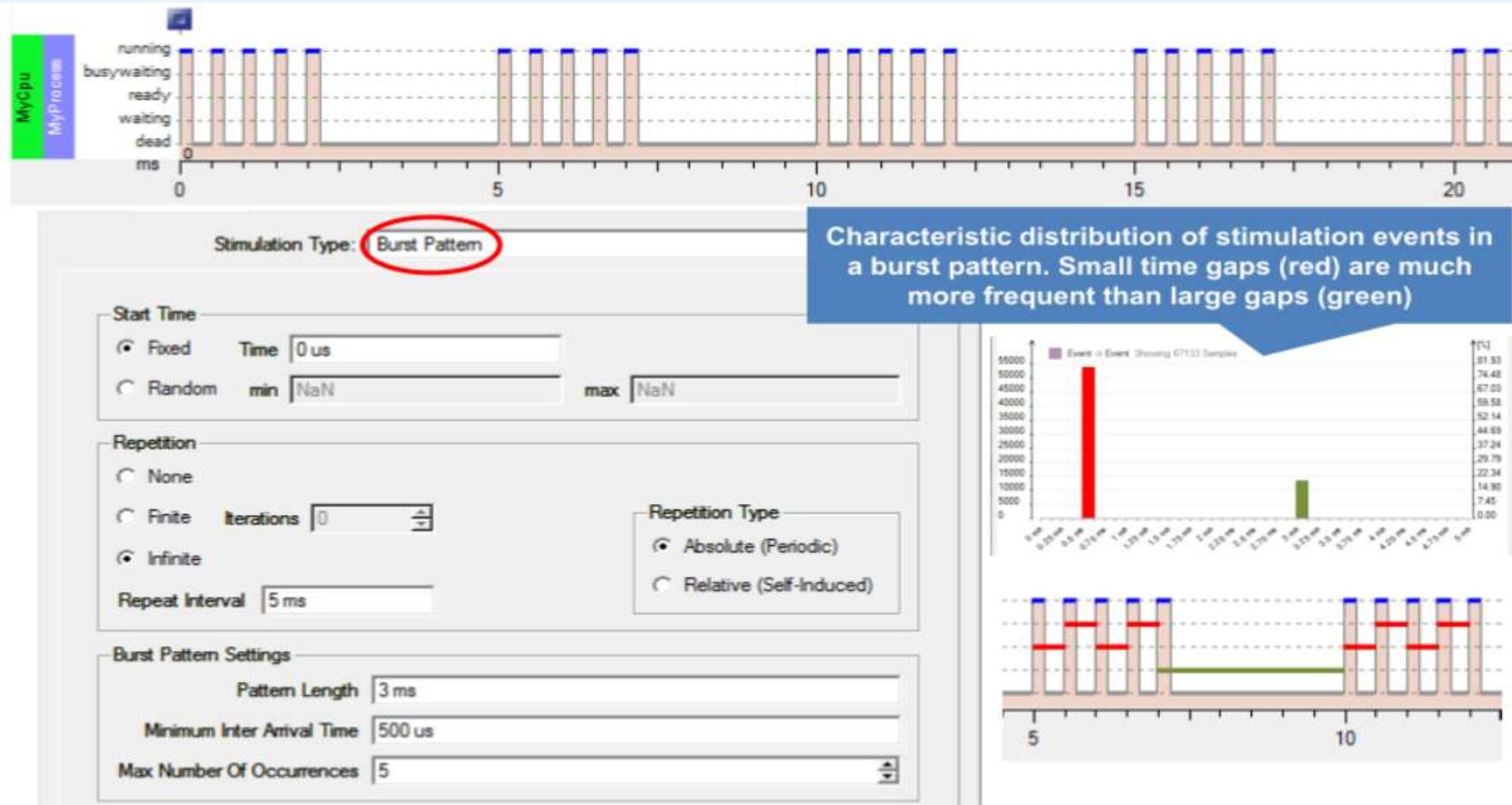
Resulting distribution of stimulation events





ChronSIM: Burst pattern

Siener





ChronSIM: Induced list based stimulation

Siener

A list based stimulation induced by a source generator

The event pattern (red) defined by the list is induced by the source event (blue)

Target ListBased on MyCpu

Clock: MyCpu

Stimulation Type: Induced List Based

Source Generator

Generator: 1 Synthetic (Start Time=0 us)

1.200000000 ms
3.400000000 ms
4.700000000 ms
5.500000000 ms
5.900000000 ms

Preview

ms 0 5 10 15 20 25 30 35 40 45 50

Current Stimulations

| Type | Stimulation Target | Start Time | Repetition | Repeat Interval |
|------|---------------------|-----------------|------------|-----------------|
| 1 | Parametrized Random | MyCpu:Base | 0 us | Infinite |
| 2 | Induced List Based | MyCpu:ListBased | - | - |



Siener

Import execution time and stimulation information from trace and debug tools





ChronSIM: Fixed-priority preemptive scheduling

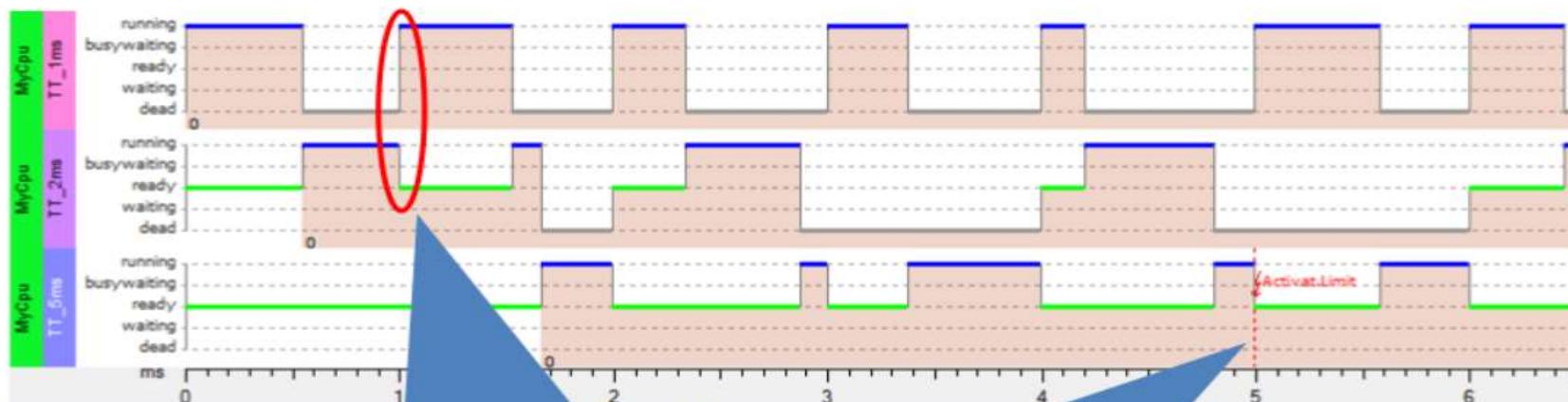
Siener

ISRs always have a higher priority level.

| Processes | | |
|-----------|------------------|----------|
| | Name | Priority |
| 1 | I_Demo | 100 |
| 2 | DefaultScheduler | |

Rate-monotonic priority assignment (RMA) is considered optimal. However, integration of sporadic processes can be difficult.

| Processes | | | | | |
|-----------|--------|--------------------------|----------|-------------------------------------|------------------|
| | Name | Active at Boot | Priority | Preemptable | Activation Limit |
| 1 | TT_1ms | <input type="checkbox"/> | 10 | <input checked="" type="checkbox"/> | 1 |
| 2 | TT_2ms | <input type="checkbox"/> | 9 | <input checked="" type="checkbox"/> | 1 |
| 3 | TT_5ms | <input type="checkbox"/> | 5 | <input checked="" type="checkbox"/> | 1 |



Lower priority tasks are preempted by those with higher priority. Thus, the system can more rapidly respond to real-time events.

Activation limit violations occur when a new activation arrives before the previous activation has terminated.



ChronSIM: Requirement editor

Siener

Add new requirement group. Groups are for mere structuring of requirements.

Add new requirement.

Architecture
Execution Times
Clocks
Requirements

Toggle displayed name (from / to user-defined label).

Available requirement types.

Requirements

| Order | Name | State | Evaluate | Show |
|-------|--|-------|----------|------|
| 1 | Requirement Group Mesut's requirements | X | X | X |
| 1 | Latency from "start T2 on CPU1" to "start T2 on CPU1" > 95 ms | X | X | X |
| 2 | Latency from "start T2 on CPU1" to "start T2 on CPU1" < 105 ms | X | | X |

| Order | Name |
|-------|-------------------------------|
| 1 | Mesut's requirements |
| 1 | T2.start jitter (lower bound) |
| 2 | T2.start jitter (upper bound) |

Type of New Requirement:

- Requirement Group
- Event Timing**
- Response Time
- Load
- Event Chain Timing
- Event Chain Break Off
- Event Chain Join
- Event Chain Multiple Processing
- Call Nesting
- Recursion
- RTOS Failures
- Illegal Event
- Event Periodicity
- Process Net Execution Time
- Event Net Execution Time
- Event Sequence
- Event() -> Event()
- Net Slack Time



ChronSIM: Event timing requirement

Siener

An event timing requirement can be used to evaluate the time distance between any two events relative to a trigger event

Custom requirement description

Start event which defines the start (time) of the measurement

End event which defines the end (time) of the measurement

Trigger event which defines when to start the evaluation of the requirement

Timing condition for time distance between start and end event.

A relative or absolute prewarn margin can be used to define a third evaluation result category (critical) besides failed and success

Requirement Evaluation

OK Cancel

Help

Edit Event Timing Requirement

Description: T2.start jitter (lower bound)

Trigger Evaluation at Event: start T2 on CPU1

Start Event: start T2 on CPU1

End Event: start T2 on CPU1

Distance: > 95 ms

Prewarn Margin: 0 %

failed success

relative

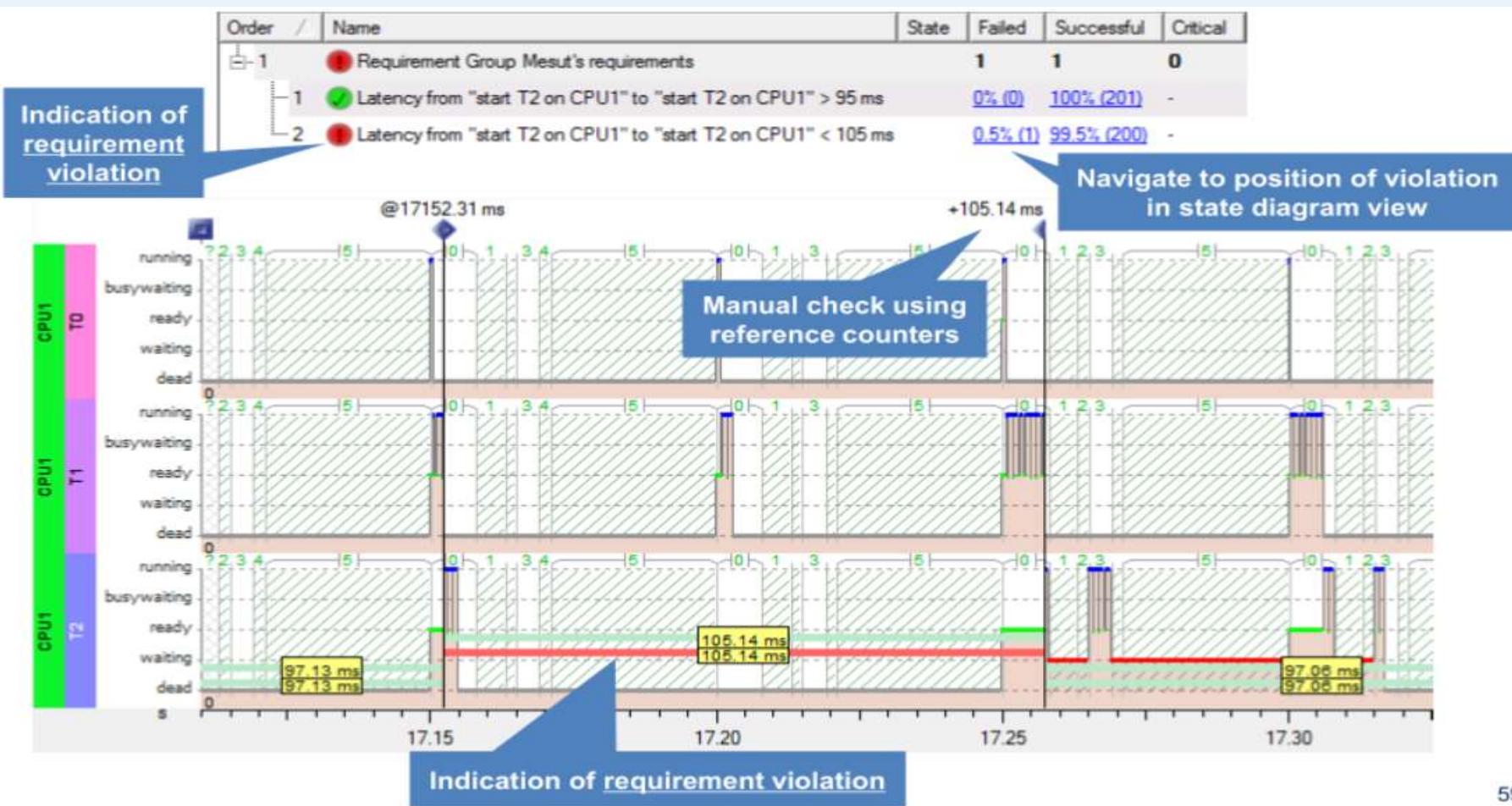
Requirement Evaluation

Evaluate Show



ChronSIM: Event timing requirement evaluation

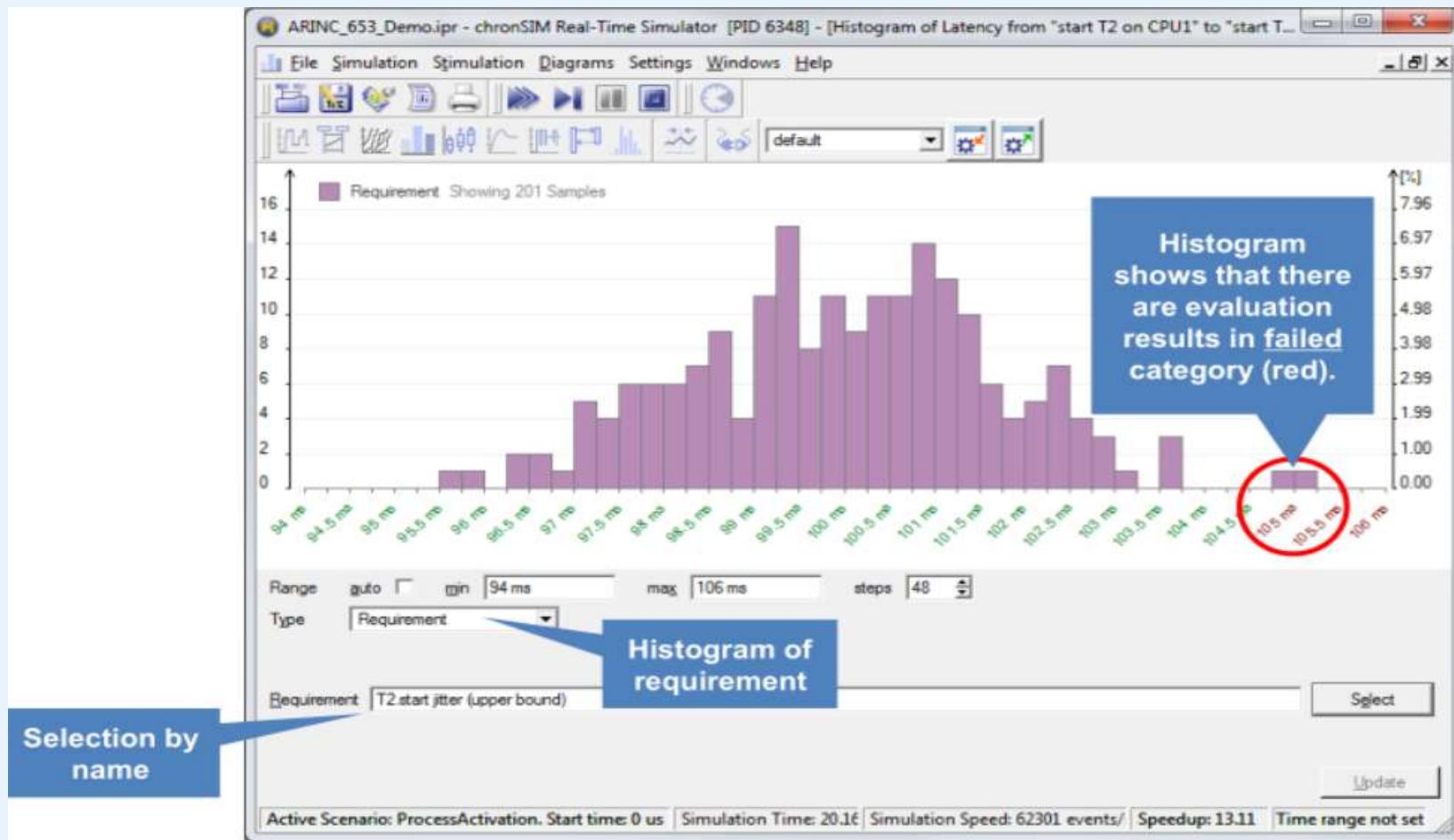
Siener





ChronSIM: Histogram of requirement evaluation

Siener



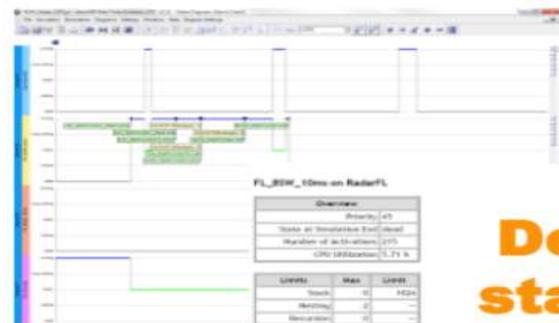


ChronSIM

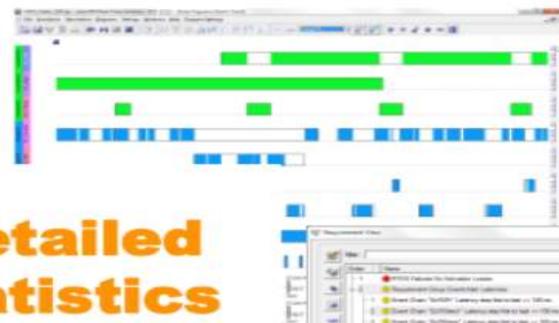
Siener

Interactive views and statistics let you explore the dynamic behavior of your system

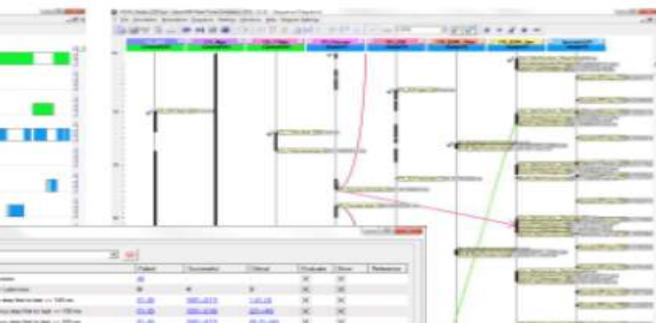
State view



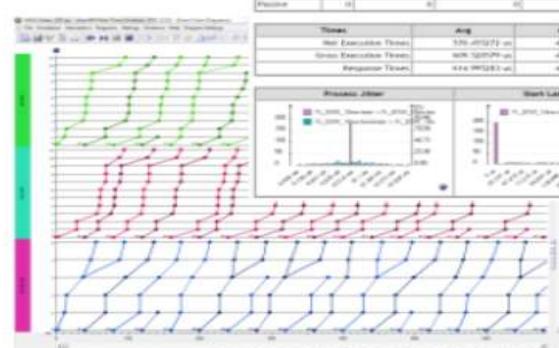
Gantt view



Trace view



Detailed statistics

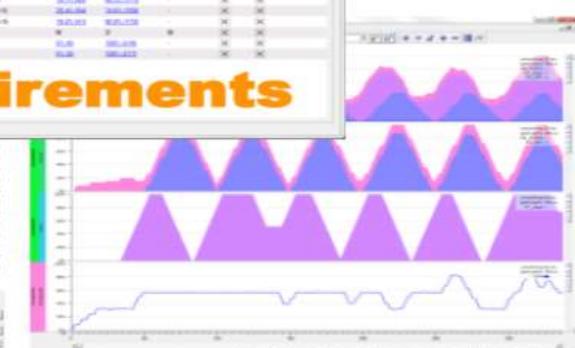


Event chains



Histogram

Requirements



Load view



面向未来，合作双赢

Siener



上海销售：刘先生 16621058313
E-mail: liuchj@siener.cn
网址: www.siener.cn

渴望人才，思索人才！
生命不息，探索不止！

我们希望客户开发能力建立，
我们希望客户关键技术攻克，
我们希望客户产品量产！！！

北京销售：张先生 15718829626
E-mail: zhanghao@siener.cn
网址: www.siener.cn

武汉销售：肖先生 13125062646
E-mail: xiaott_aurix@sina.com
网址: www.siener.cn