2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 15-16, 2024

Developing Complex Systems using Model-Based Cybertronic Systems Engineering Methodology

Petri Solanti

Siemens EDA, DTEG

accellera SYSTEMS INITIATIVE

Agenda

- Cybertronics Systems
- Cybertronics Challenge
- Taming the Beast - Tackling the Cybertronics Challenge with MBCSE
- Design Example
  - Top-level System Modeling
  - Cybertronics Subsystem
  - System-on-Chip Subsystem
  - Architecture Performance Analysis

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Cybertronics Systems

An Overview

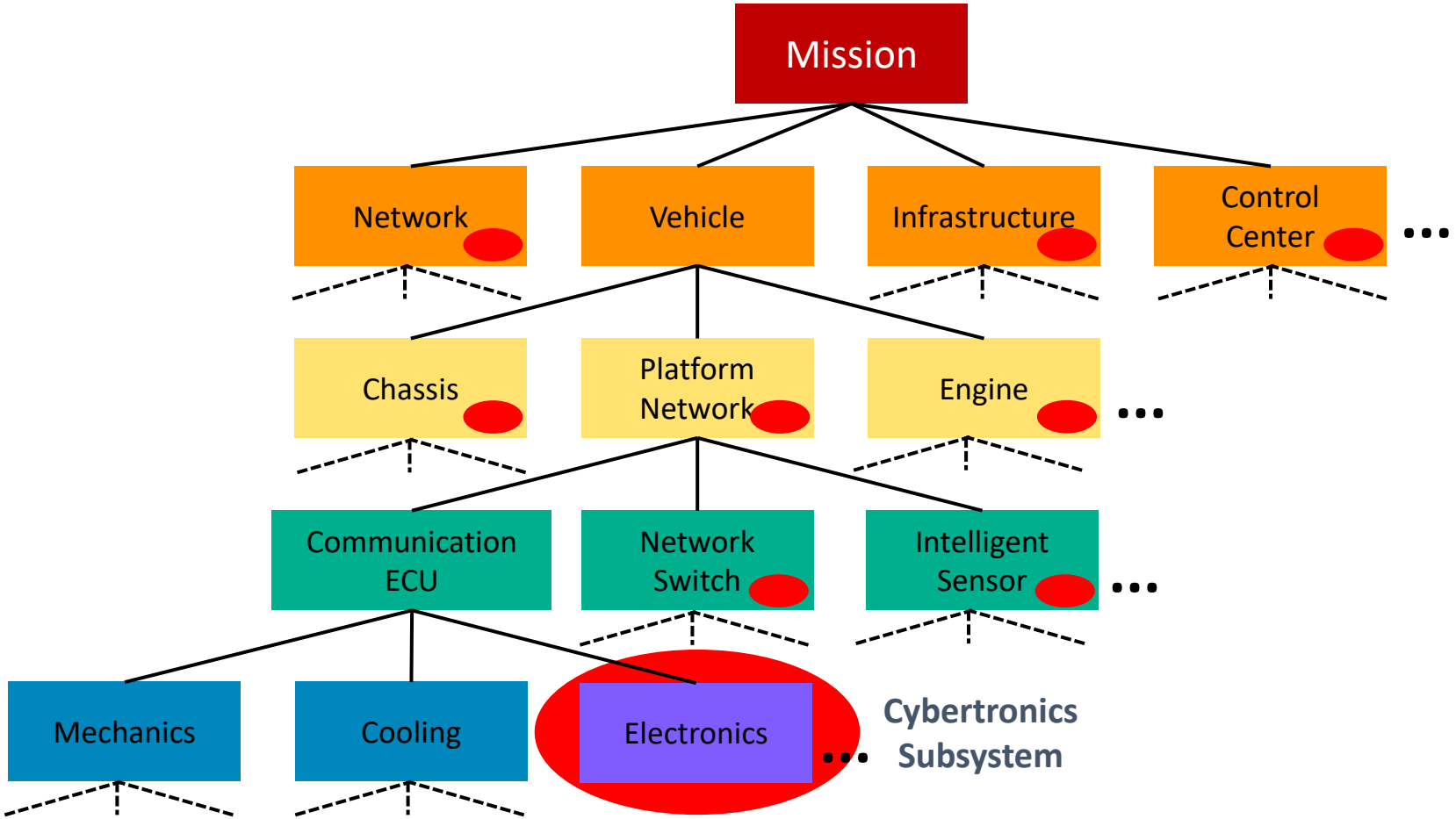
accellera SYSTEMS INITIATIVE

Definition of Cybertronics

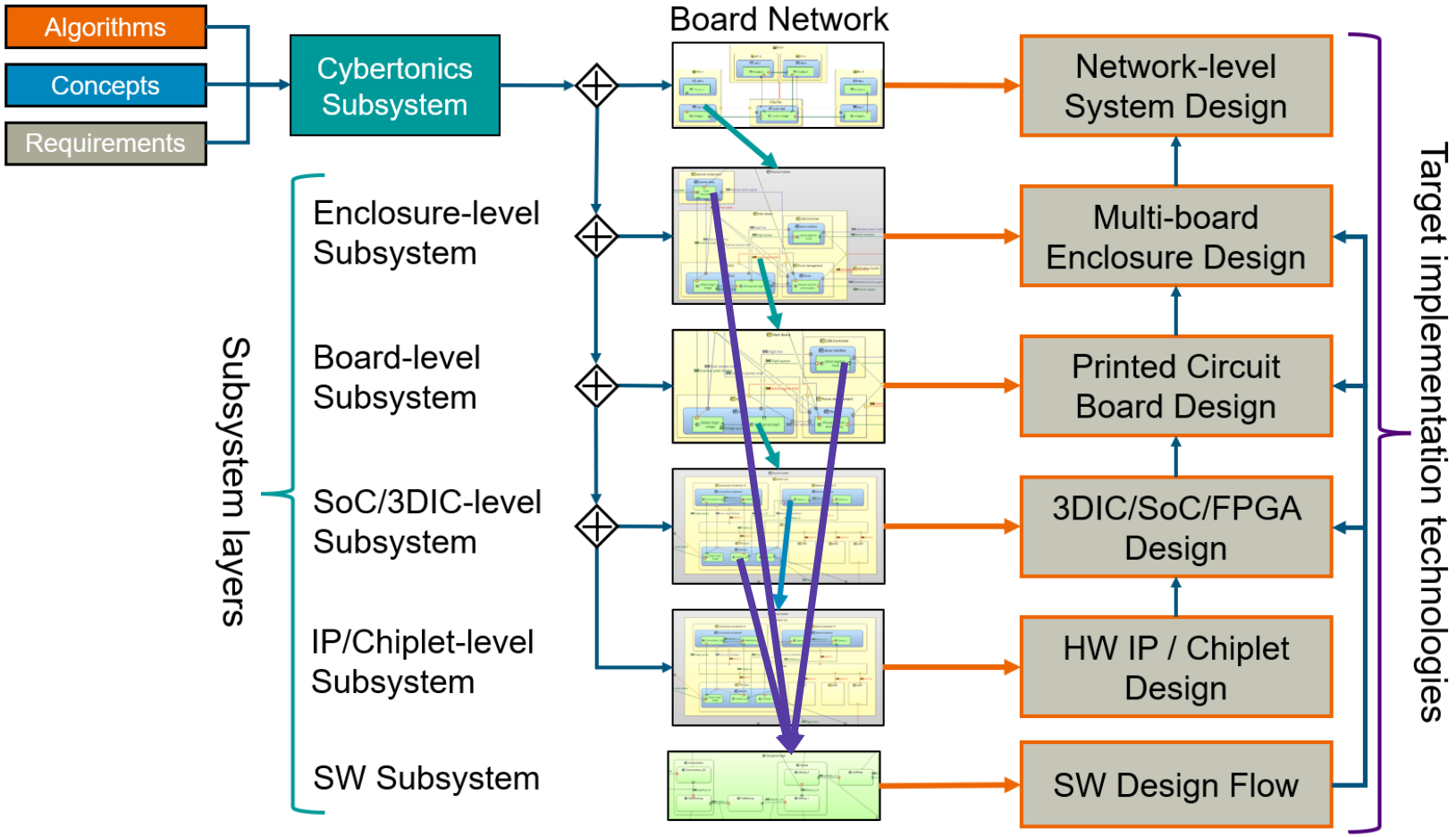
Cybertronics is the engineering discipline of functional co-architecting of electronics and software to realize new software defined, semiconductor enabled systems.

- The definition of cybertronics covers
  - Network level
  - Electronic unit (Multi-PCB enclosure)
  - PCB
  - SoC
  - FPGA and Chiplet level
  - Embedded software

System-of-Systems subsystem hierarchy



Structure of Cybertronics System-of-Subsystems



2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Cybertronics Challenge

New dimensions of system design complexity

accellera SYSTEMS INITIATIVE

Cybertronics challenge

- Cybertronics systems design challenges are not faced at any other type of system design
- The main challenges in cybertronics design
  - Multi-disciplinary system with dynamic configuration changes
  - Requirements allocation, propagation and tracing
  - Complexity
  - Function allocation
  - Verification

Multi-disciplinary systems

- Cybertronics system contains different implementation domains
  - Network
  - PCB
  - Integrated circuits (SoC, FPGA, ASIP, 3DIC, HW IP, Chiplet)
  - Embedded software
- Multiple architecture options available → Optimization based on e.g.,
  - Area and cost
  - Performance and power
- Optimization requires understanding of implementation technologies

Requirements allocation, propagation and tracing

- System-level requirements decomposed to subsystem requirements
- Parameterized requirements for automated requirements verification
- Additional requirements specified during the decomposition process
- Cybertronics requirements challenge:
  - Different requirement management tools and strategies in different domains
  - Interpretation of requirements in different design domains
  - Requirements tracing in multi-disciplinary design environment

Complexity

- Multiple dimensions to be considered
  - Computational load
  - Data movement and caching
  - Real-time requirements
  - Power consumption
- Complexity is orders of magnitude higher than in mechanical systems
  - Millions of lines of SW code in multiple SW stacks
  - Billions of gates in an average size SoC
- Managing the cybertronics complexity with traditional design methodologies is not possible

Function allocation

- Cybertronics functionality can have different implementations
  - Software executed on different processor architectures
  - Hardware accelerators on SoC or FPGA
  - Application specific standard components
  - Specific devices (e.g., network routers)
- Diversity of the implementation options leads to a huge solution space
- Different function allocations lead to different metrics (e.g., performance, cost, power consumption)
- Quick exploration of different architecture options is necessary

Verification

- Verification complexity:
  - Thousands of requirements
  - Millions of tests
  - Multi-level, multi-domain verification
- Due to multi-disciplinary system nature, a multi-level, multi-domain verification is required, but
  - Domain specific verification flows
  - Multiple repository architectures and formats
  - Flat verification approach → Huge number of tests to be traced

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Taming the Beast

Tackling the Cybertronics Challenge with MBCSE

accellera SYSTEMS INITIATIVE

What should a cybertronics system design methodology provide?

- Multi-disciplinary system modeling
  - Information hiding and abstraction
  - Model-based engineering approach for seamless communication between design teams
  - Adding domain specific information only where needed
  - Links to domain specific implementation flows
- Complexity management
  - Clear separation between function and structure
  - Gradual increment of fidelity during the modeling process
  - Dividing design into subsystems for more detailed decomposition
  - Design integrity management

What should a cybertronics system design methodology provide?

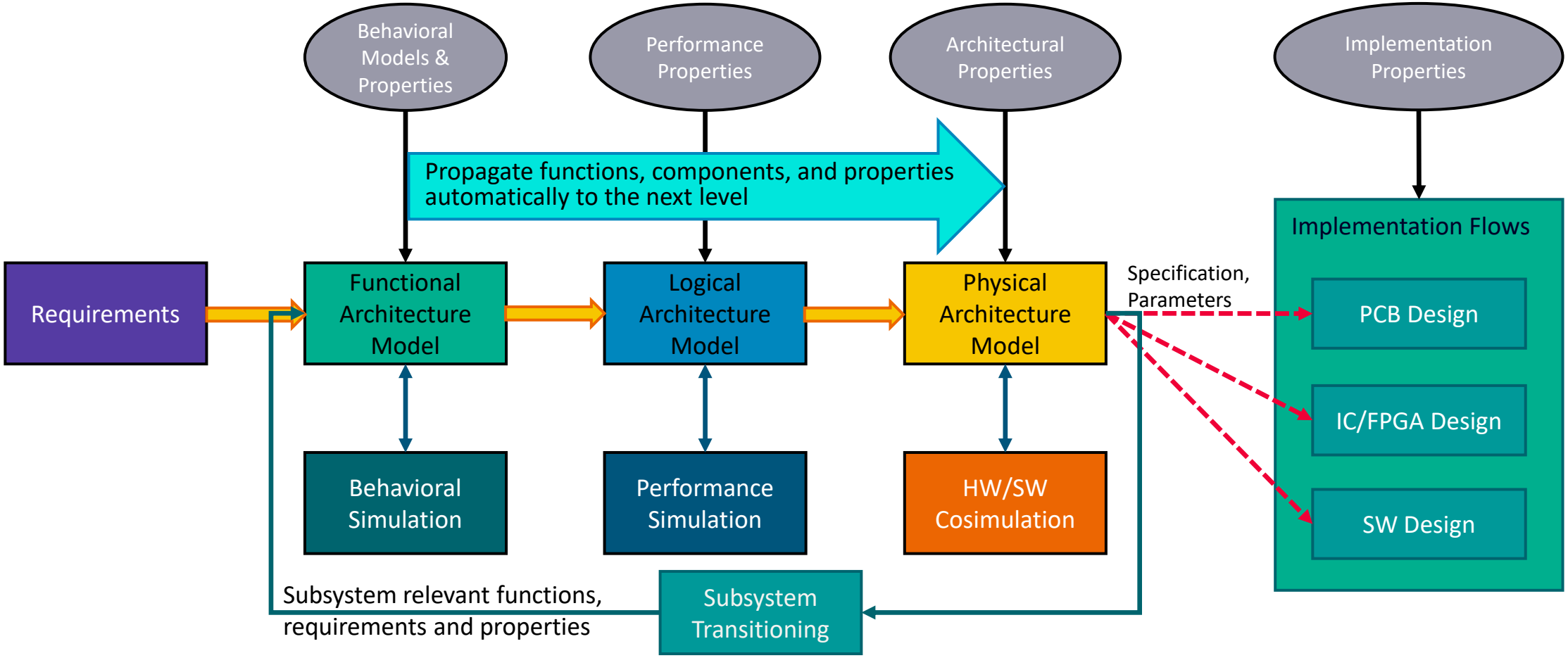
- Requirements handling
  - Assisted allocation of requirements and parameters to the model
  - Automatic propagation of requirements and parameters throughout the process
- Function allocation
  - Free allocation of functions to different structural elements
  - Exploration of different function allocations
- Verification
  - Requirements driven digitally threaded verification throughout the design process
  - Integration of domain specific verification processes

Model-Based Cybertronics Systems Engineering methodology

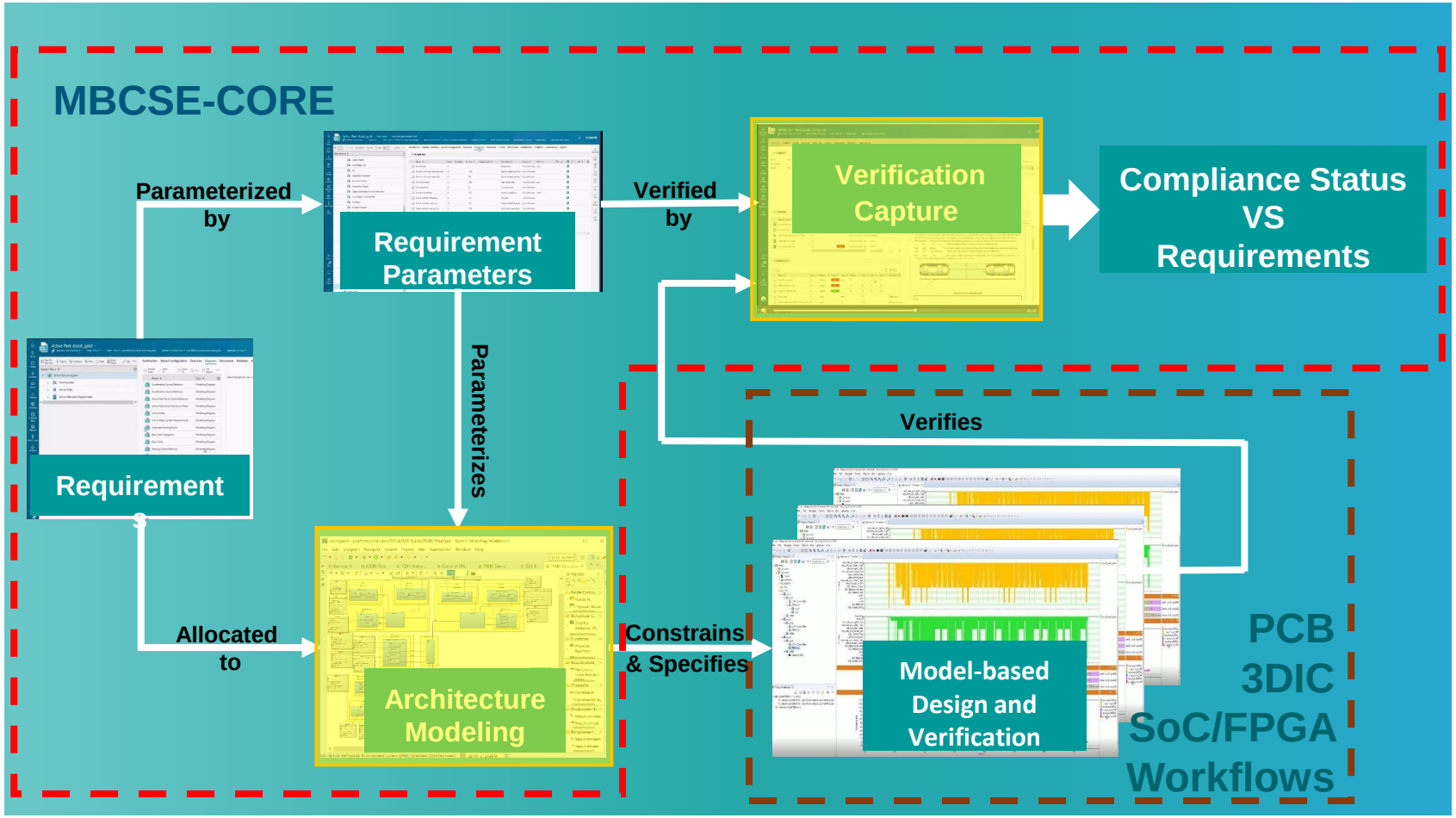
MBCSE methodology borrows concepts from several methodologies:

- Arcadia provides a structured approach to system decomposition
  - Subsystem hierarchy concept
  - Clear separation between function and structure
  - Integrated requirement and property handling
  - Automated transitions ensure integrity between design layers and subsystems
- Property Model Methodology for validation and verification
  - Formalized, property-based requirements
  - Continuous refinement of requirements and constraints
  - Simulation-based analysis
- Verification Capture Point -based digital verification threading
  - Universal data object for digitally threaded verification

Cybertronics architecture modeling process

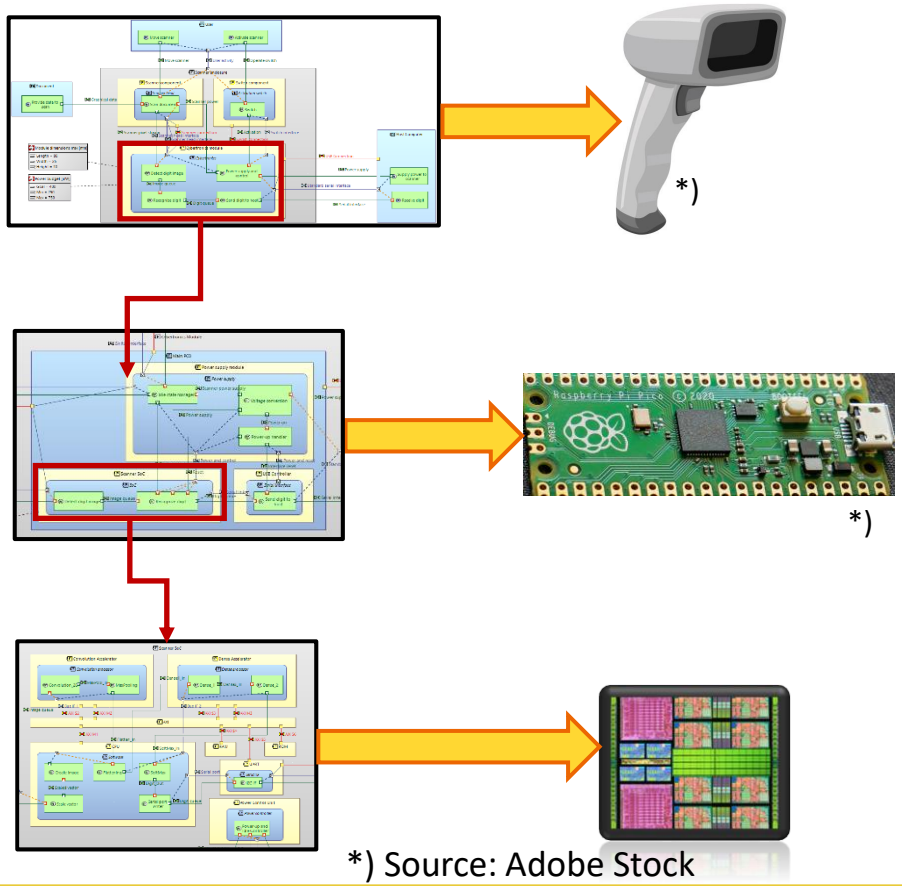


Cybertronics Systems Integration Lab Framework



Modeling multi-disciplinary systems

- MBCSE methodology enables specifying individual target implementation domain for each subsystem level
- Transitioned subsystem is treated as a component in the parent system
- Subsystem is handled as a separate project
- Domain specific implementation details are inserted using properties and they can be extracted from the model



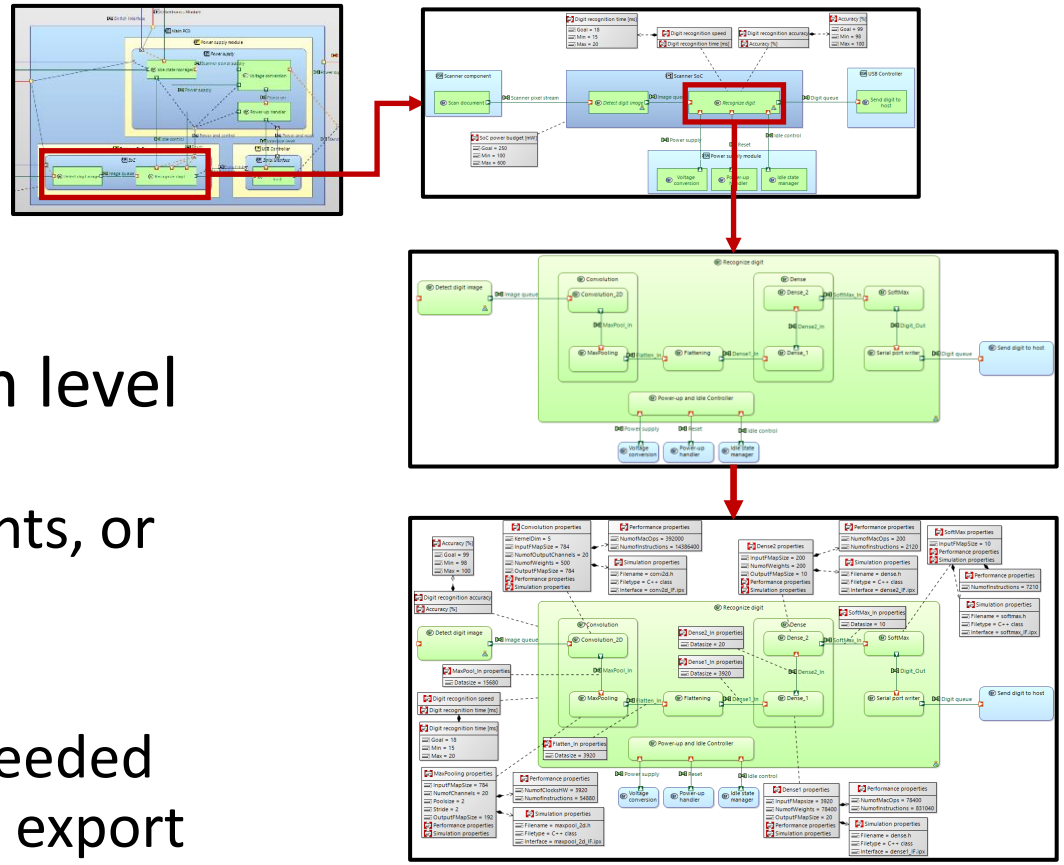
*)

*)

*) Source: Adobe Stock

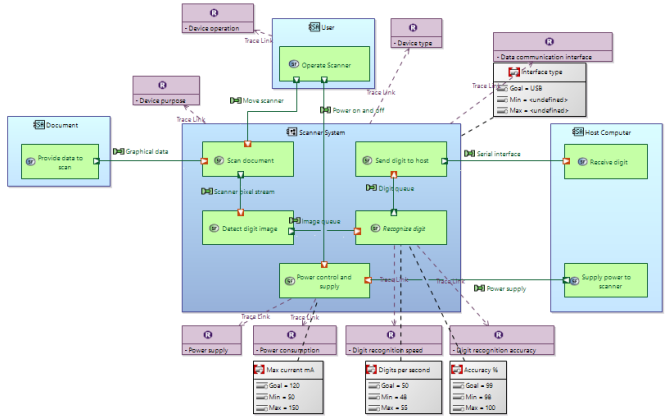
Managing the complexity

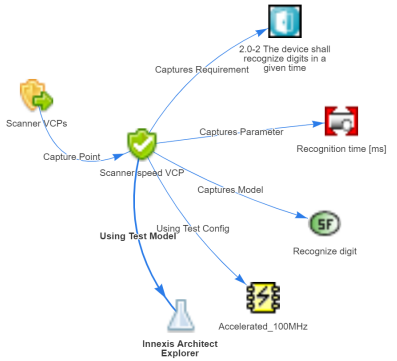
- Many techniques available
  - Subsystem hierarchy
  - Function breakdown
  - Property modeling
- Different components on a subsystem level
  - Subsystems
  - Library systems, Off-the-shelf components, or IP
- Increasing the fidelity gradually
  - Add information to model, where it is needed
  - Property groups with specific names for export



Addressing requirements handling

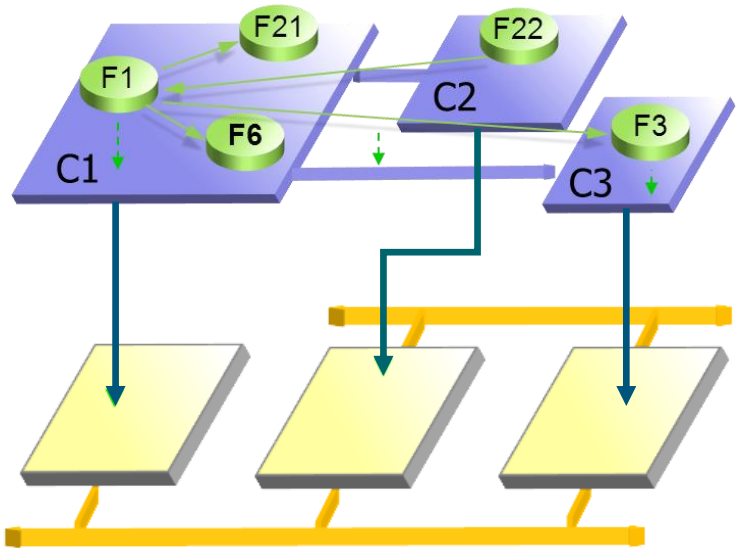
- Allocating requirements and parameters to the functions or structures
- Automatic propagation of requirements to
  - Other system layers
  - Subsystems
  - Implementation flows
- Automatic tracing of requirements through
  - Allocation
  - Implementation
  - Verification
- Verification Capture Points link requirement to verification





Function allocation

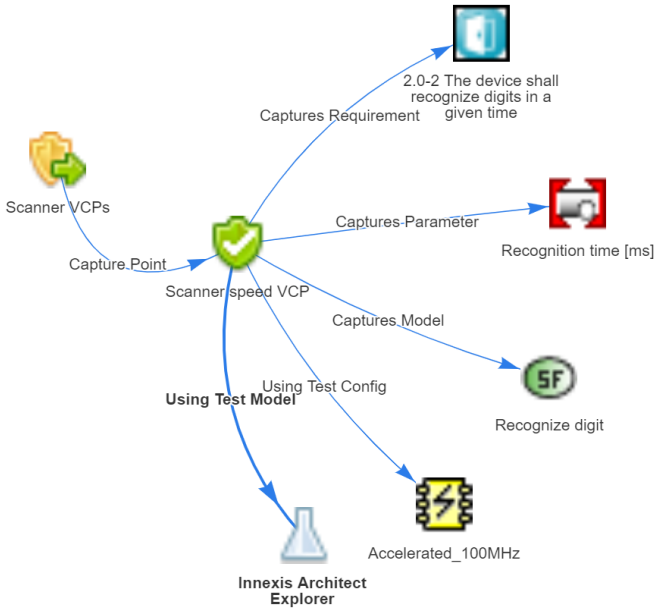
- Functional, logical and physical architecture layers
- Functional architecture:
  - Description of system functions and data exchanges
  - Related information can be added as properties
- Logical architecture:
  - Allocation of system functions to logical components
  - Allocation of function data exchanges to logical channels
  - Performance analysis based on performance properties
- Physical architecture:
  - Allocation of logical components to physical components
  - Allocation of logical channels to physical links
  - Implementation specification for domain specific flow



Source: https://mbse-capella.org/arcadia.html

Digital verification threading using Verification Capture Points

- VCP enables automated verification threading
- Verification Capture Point links together
  - Requirement and requirement parameters
  - Model or implementation to be verified
  - Test model
  - Test configuration
  - Verification result



Scanner Recognition Time

Model	Requirement	Parameter	Value	Min	Max	Certified By	Comment
Cybertronics Module	2.0-2 The device shall recognize digits in a given time	Recognition time [ms]	15.20	15	20	Lisa Verifier	SoC with CPU plus 2 accelerators, increased clock frequency to 200 MHz gets us within the budget, close to the minimum.

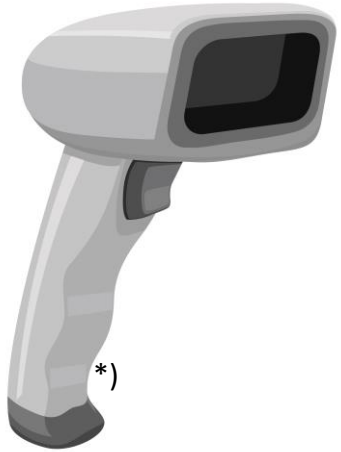
2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Design Example

accellera SYSTEMS INITIATIVE

Example system

- Hand-held scanner to read hand-written postal codes
- System design has multiple implementation domains
  - System enclosure (mechanical)
  - Electronics subsystem (PCB)
  - System-on-Chip (embedded HW/SW system)
- The example covers following aspects
  - Functional definition of the system and requirements allocation
  - Allocation of functions to mechanical and cybertronics parts
  - Decomposition of cybertronics subsystem to components on PCB
  - Optimization of HW/SW allocation on SoC
  - SoC architecture specification for digital IC implementation flow


*)


*)

*) Source: Adobe Stock

System decomposition and implementation

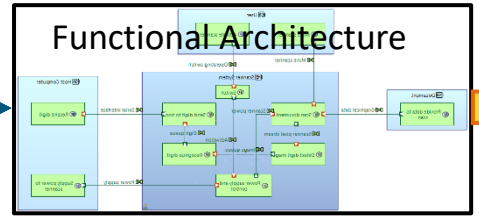
Algorithms

Concepts

Requirements

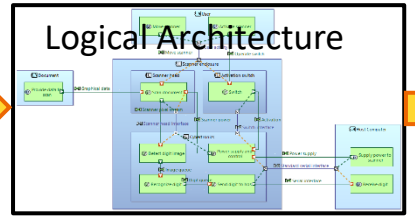
What the system shall accomplish

Functional Architecture



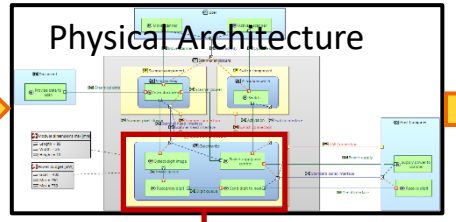
How the system will work

Logical Architecture



How the system will be implemented

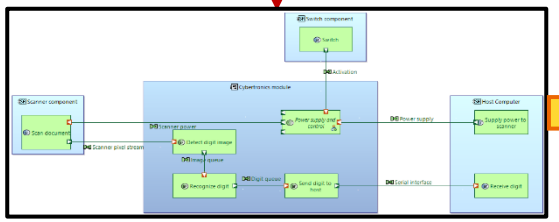
Physical Architecture

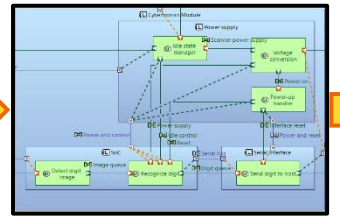


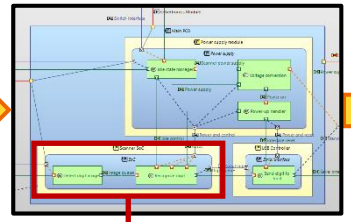
*)

Transition electronics module to a subsystem

All electronics and software functions



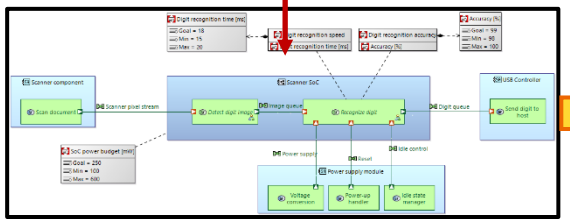


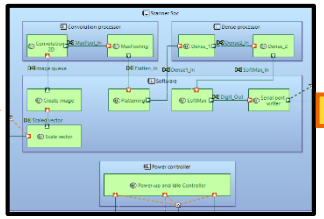


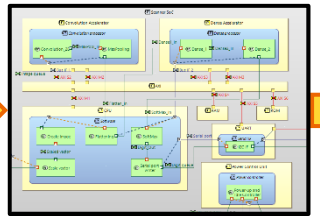
*)

Transition SoC component to a subsystem

All digital and software functions







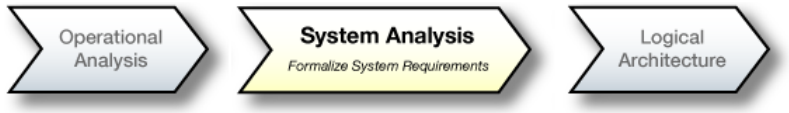
*) Source: Adobe Stock

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

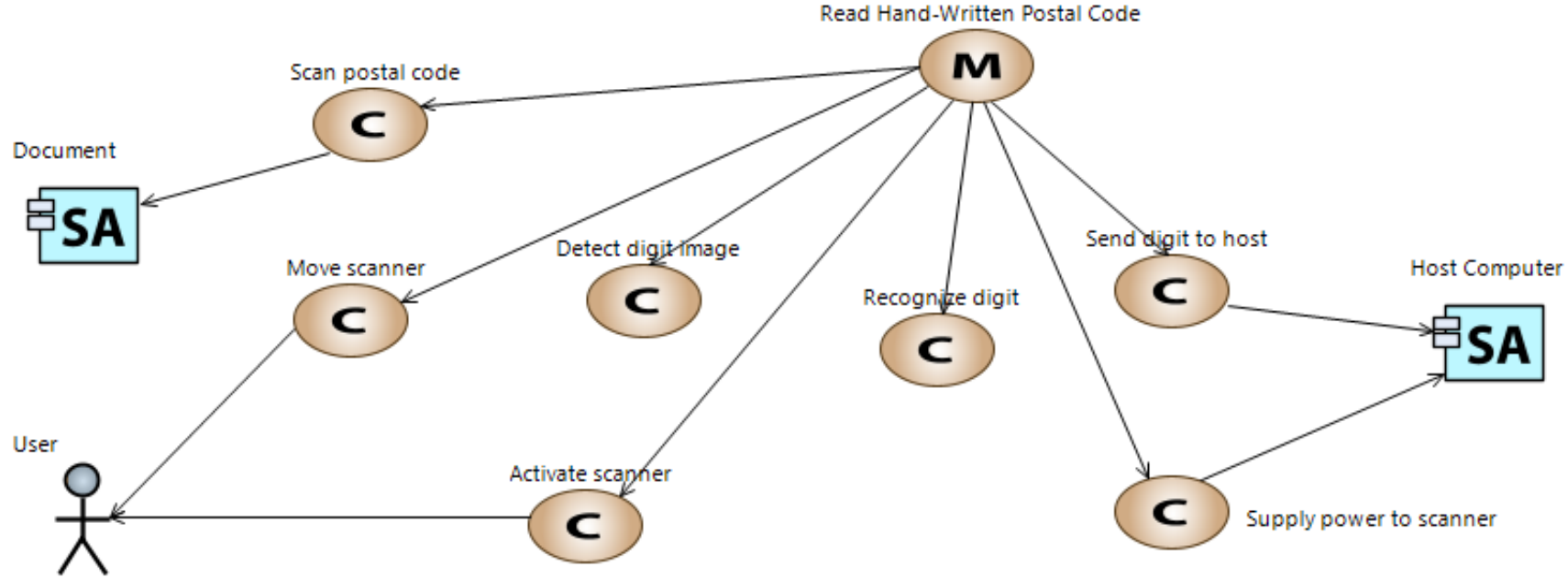
Top-Level System Modeling

accellera SYSTEMS INITIATIVE

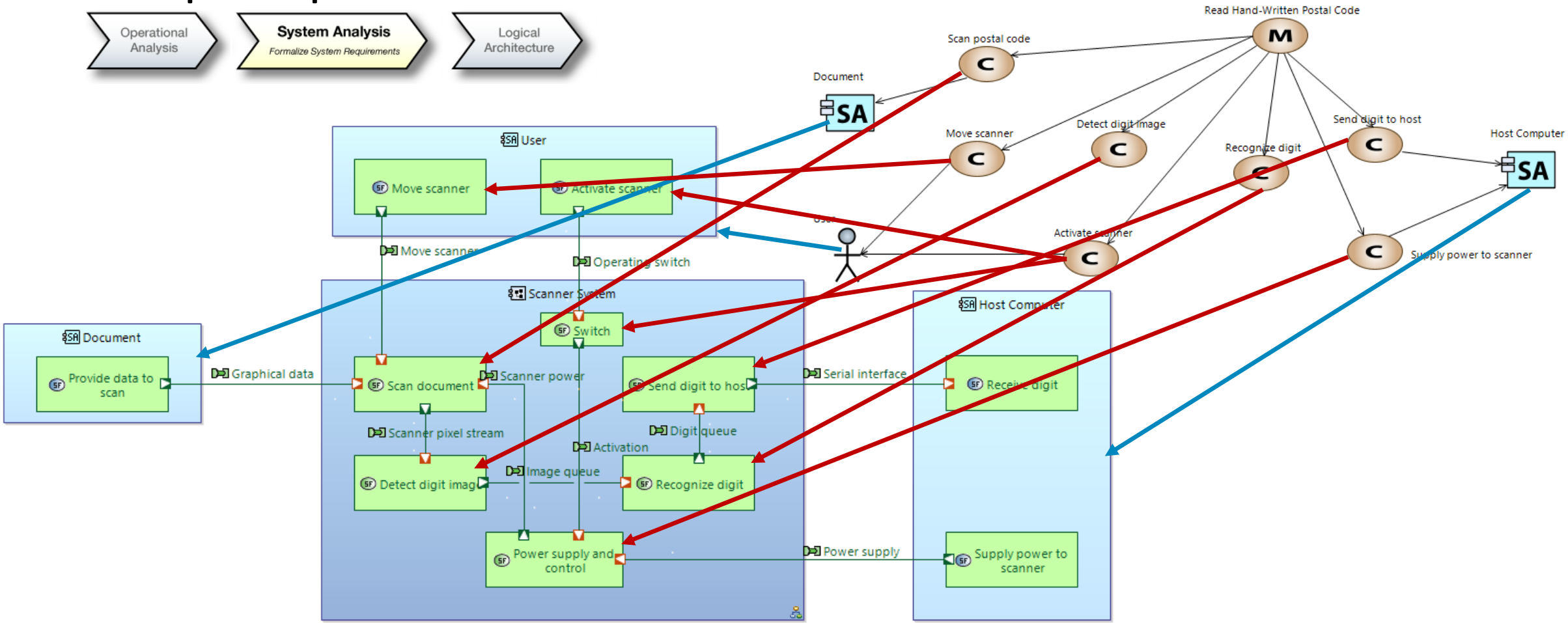
Functional capability model of scanner device



- Operational environment and capabilities of the device
- Defines device functions and actors the device is interacting with

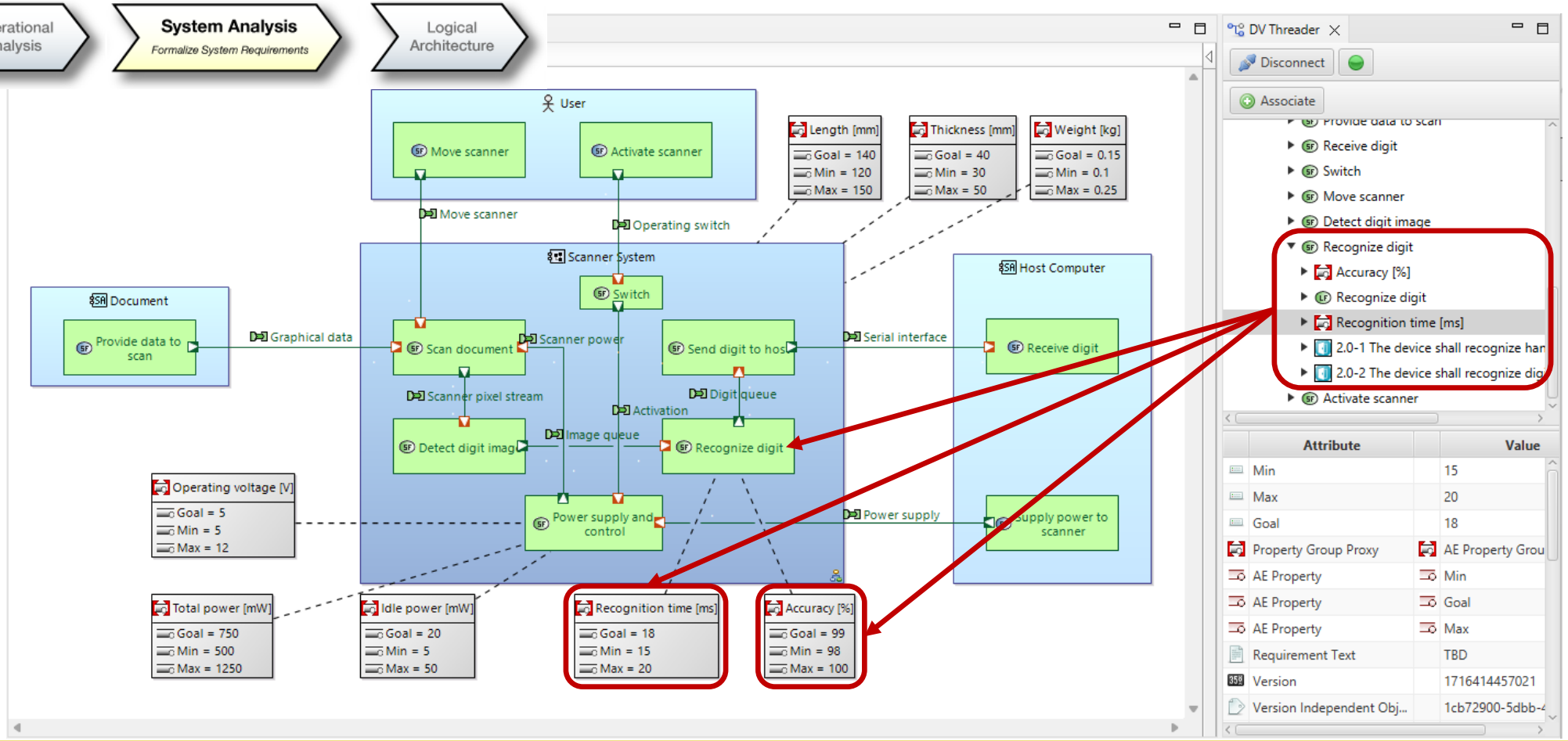


Map capabilities into a functional architecture

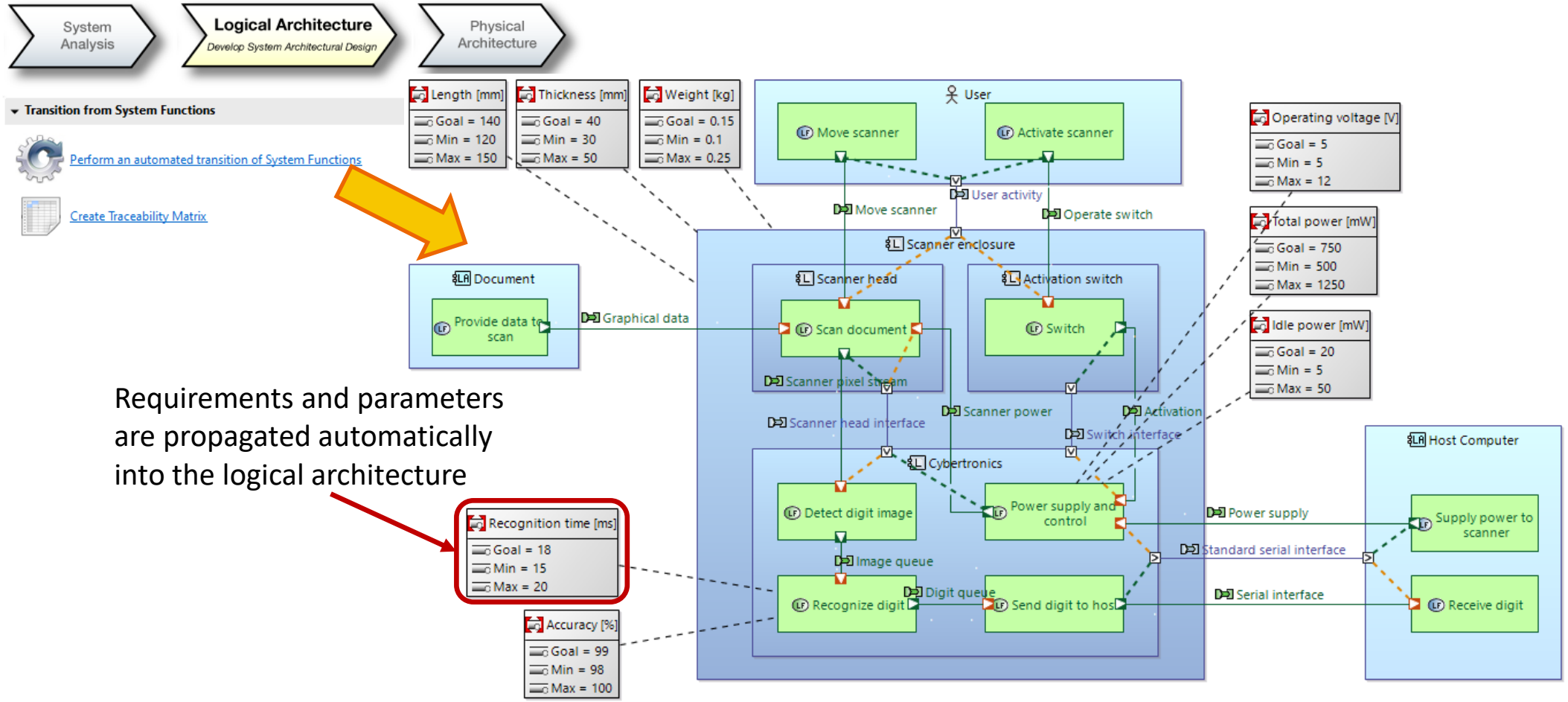


Allocate requirements and parameters

Operational Analysis → **System Analysis** *Formalize System Requirements* → Logical Architecture



Map functions to logical components

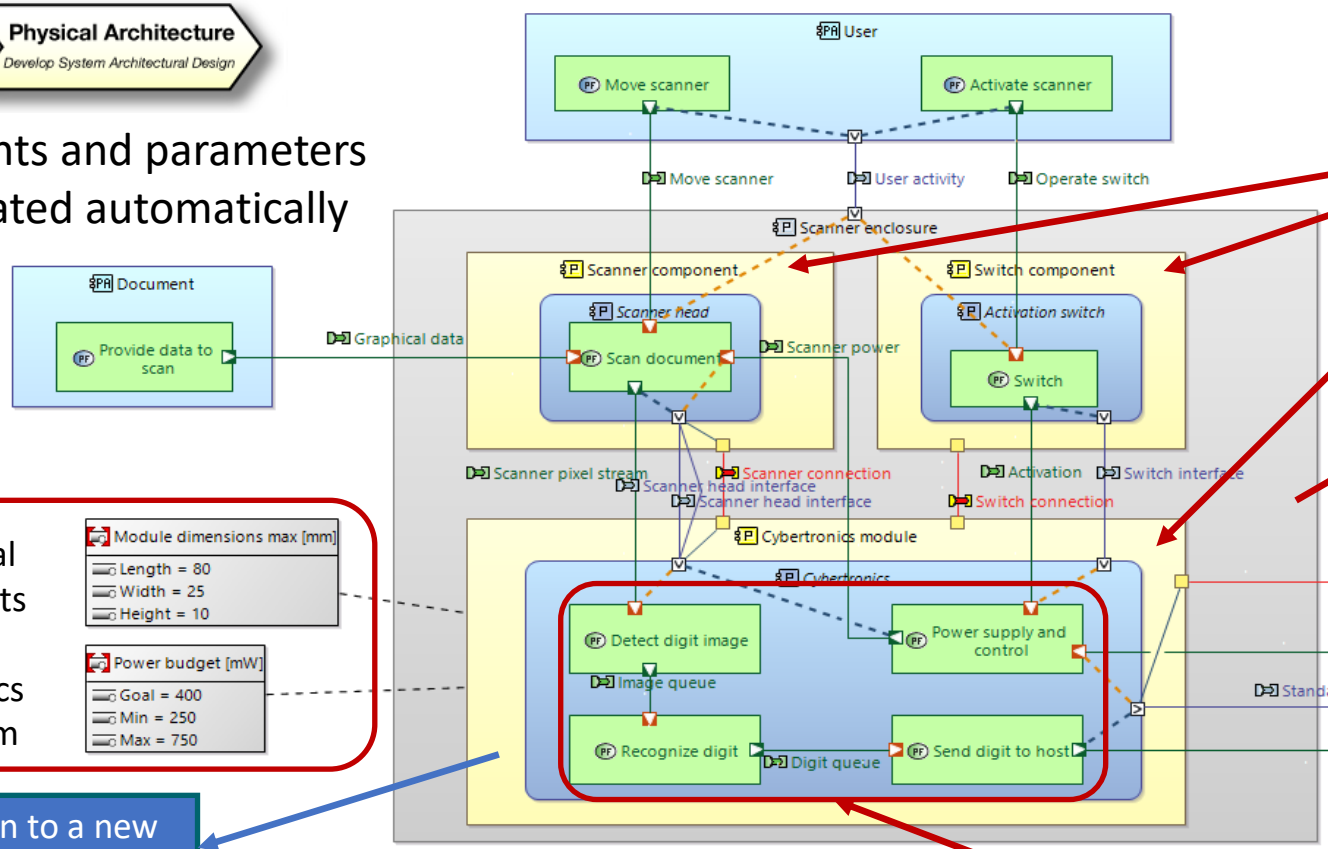


Requirements and parameters are propagated automatically into the logical architecture

Model physical architecture of the device

Logical Architecture → **Physical Architecture** *Develop System Architectural Design*

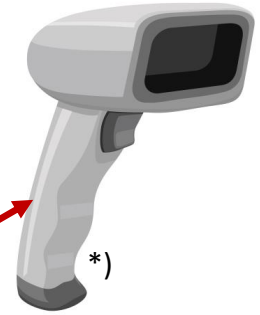
Requirements and parameters are propagated automatically



Additional constraints for electronics subsystem

Transition to a new subsystem

Physical components inside scanner enclosure



*)

Scanner enclosure

All SW and electronics related functionality allocated to Cybertronics module

*) Source: Adobe Stock

Why system to subsystem transition?

- Subsystem transition is part of the system decomposition process
  - Enables development of large systems of subsystems
  - Propagates all relevant information to the new subsystem project
  - Unlimited subsystem hierarchy
  - Integral part of Arcadia methodology
- Benefits:
  - Maintains design abstraction
  - Minimizes amount of detailed information in the upper system level
  - Maintains integrity of the design through automatic transition
  - Subsystem has visibility to the upper level only through the transition interface

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

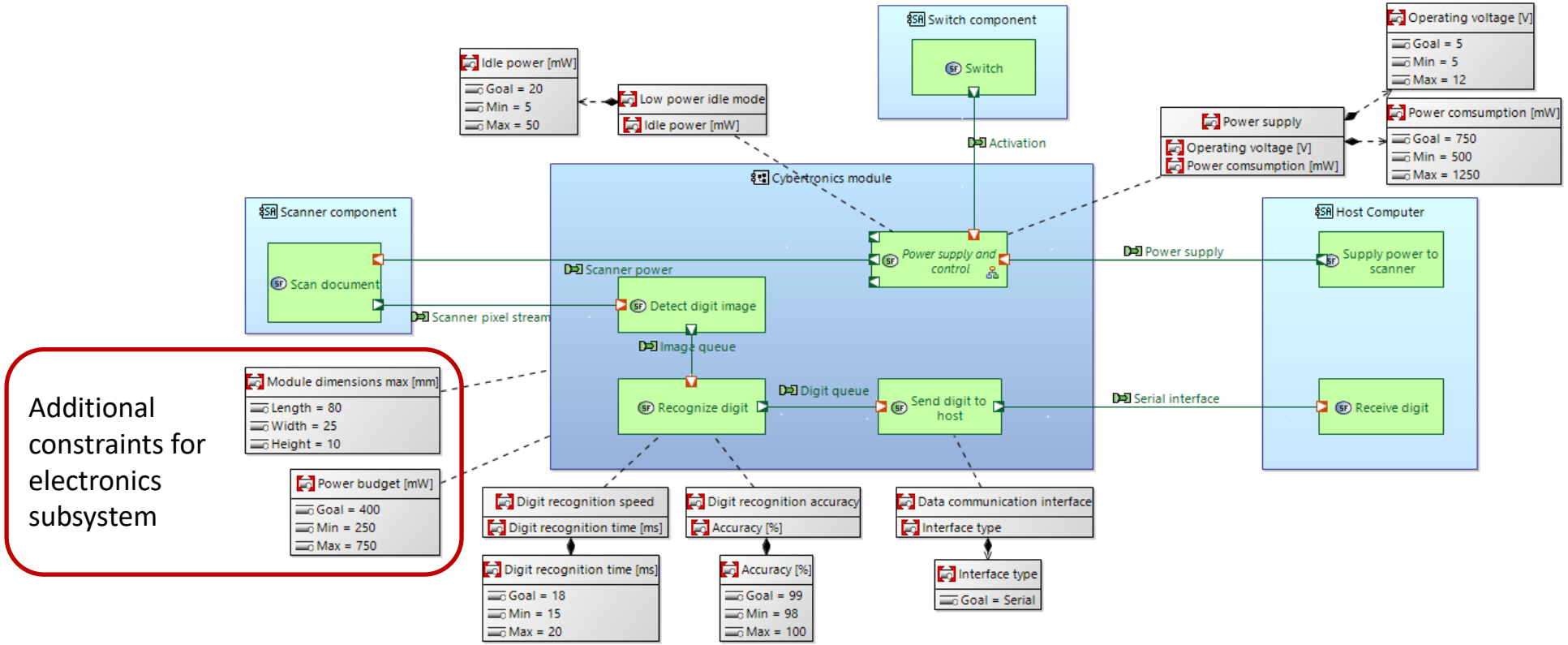
Cybertronics Subsystem

Modeling and Analysis of PCB-Subsystem

accellera SYSTEMS INITIATIVE

Transition Cybertronics Module to subsystem

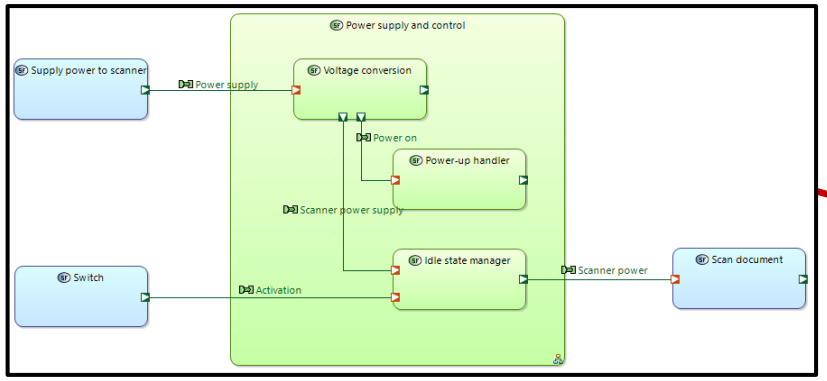
Operational Analysis → **System Analysis** *Formalize System Requirements* → Logical Architecture



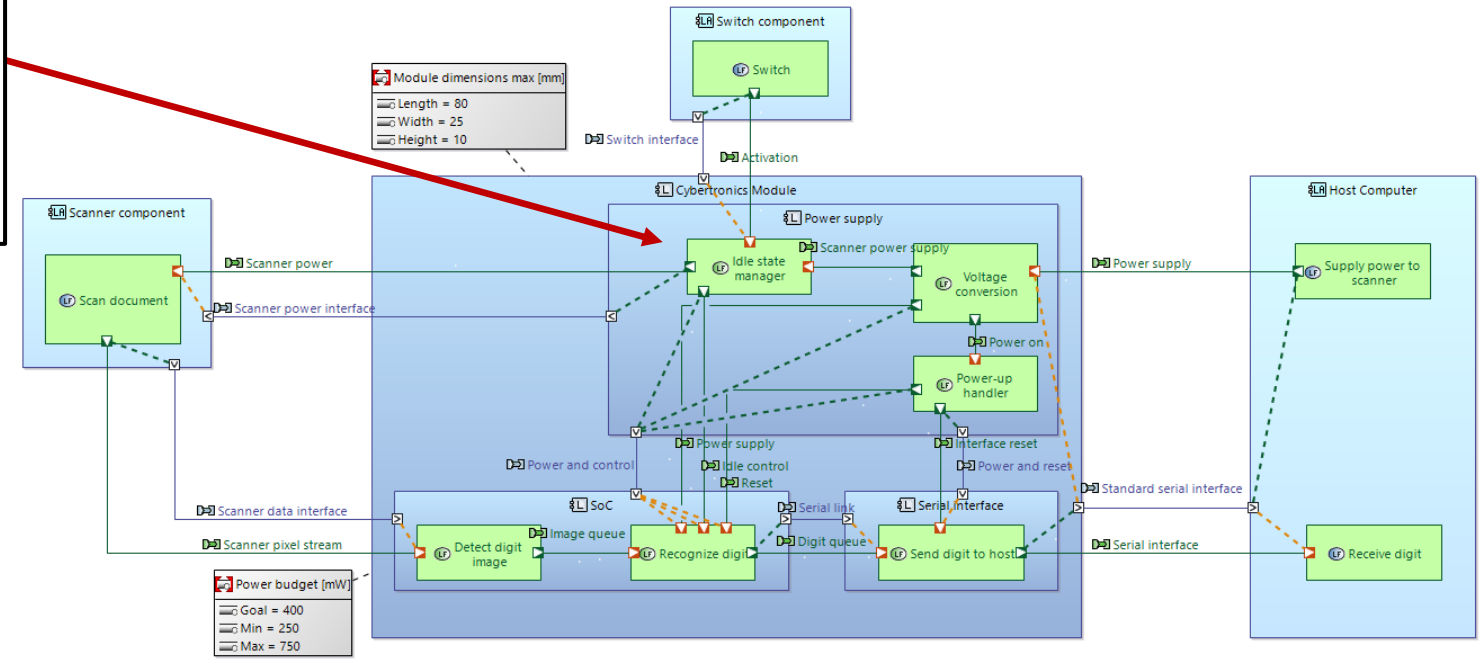
Additional constraints for electronics subsystem

Creating logical architecture

System Analysis → **Logical Architecture** *Develop System Architectural Design* → Physical Architecture

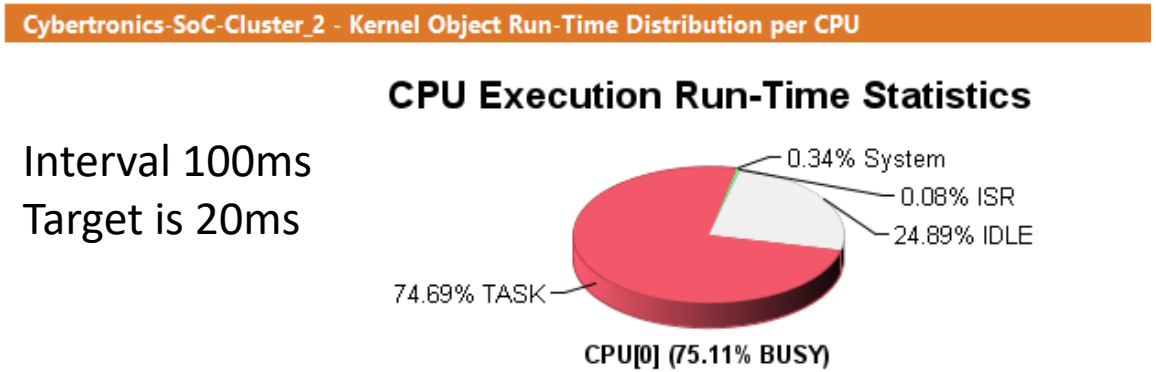


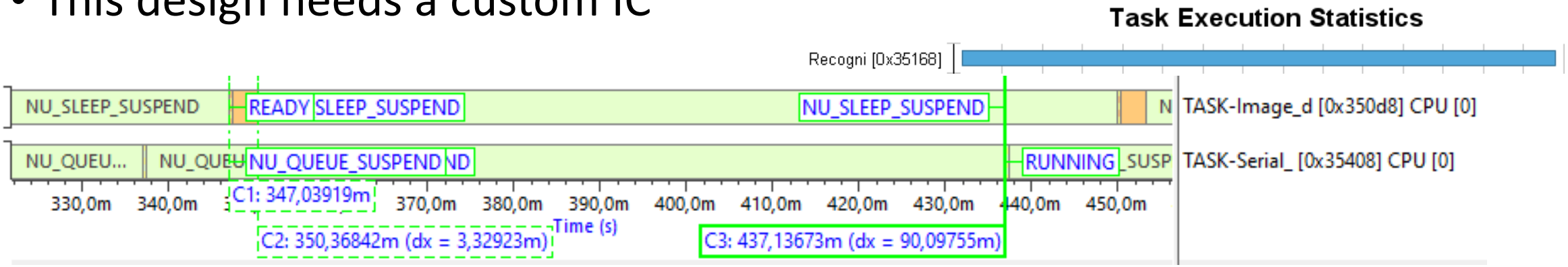
Break down power supply function
Into more detailed description



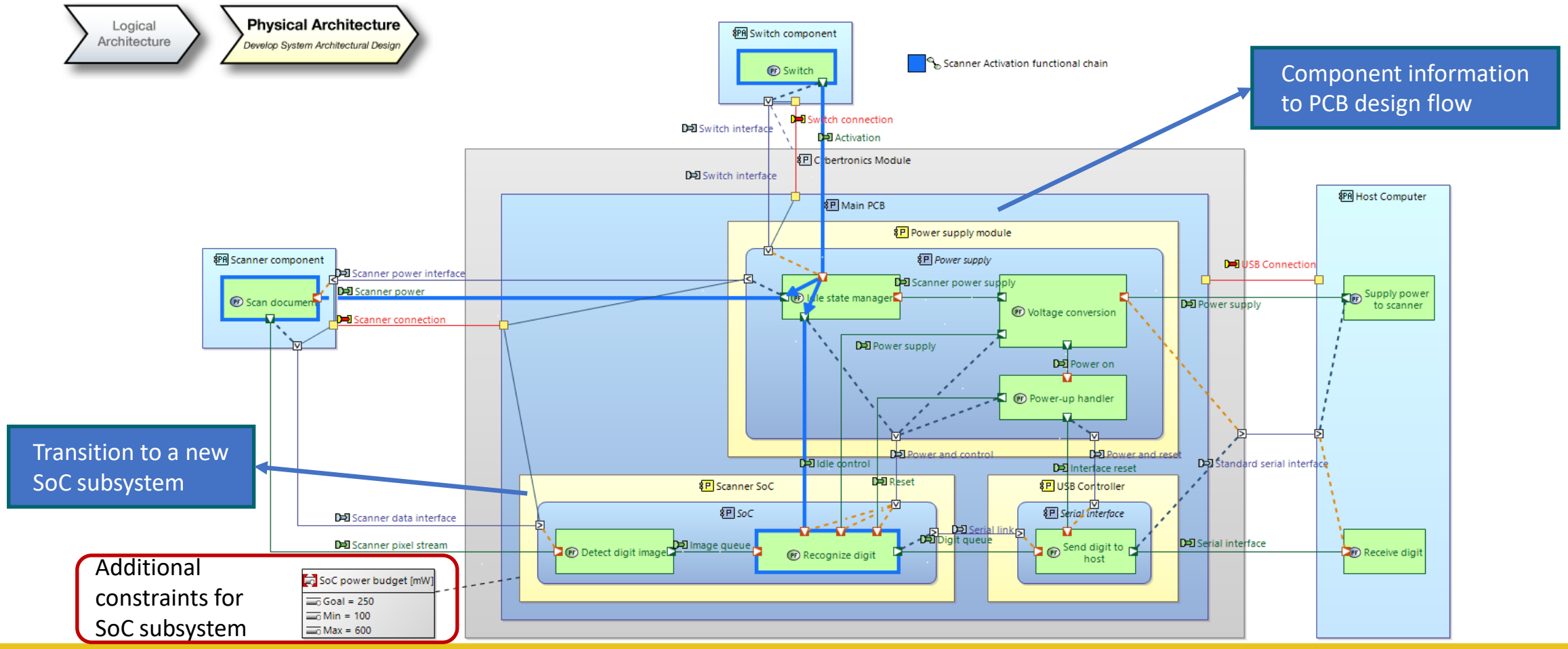
Analyzing feasibility

- Does the function allocation meet requirements, e.g.,
  - Performance
  - Power consumption
- This design needs a custom IC





Physical architecture of Cybertronics module



2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

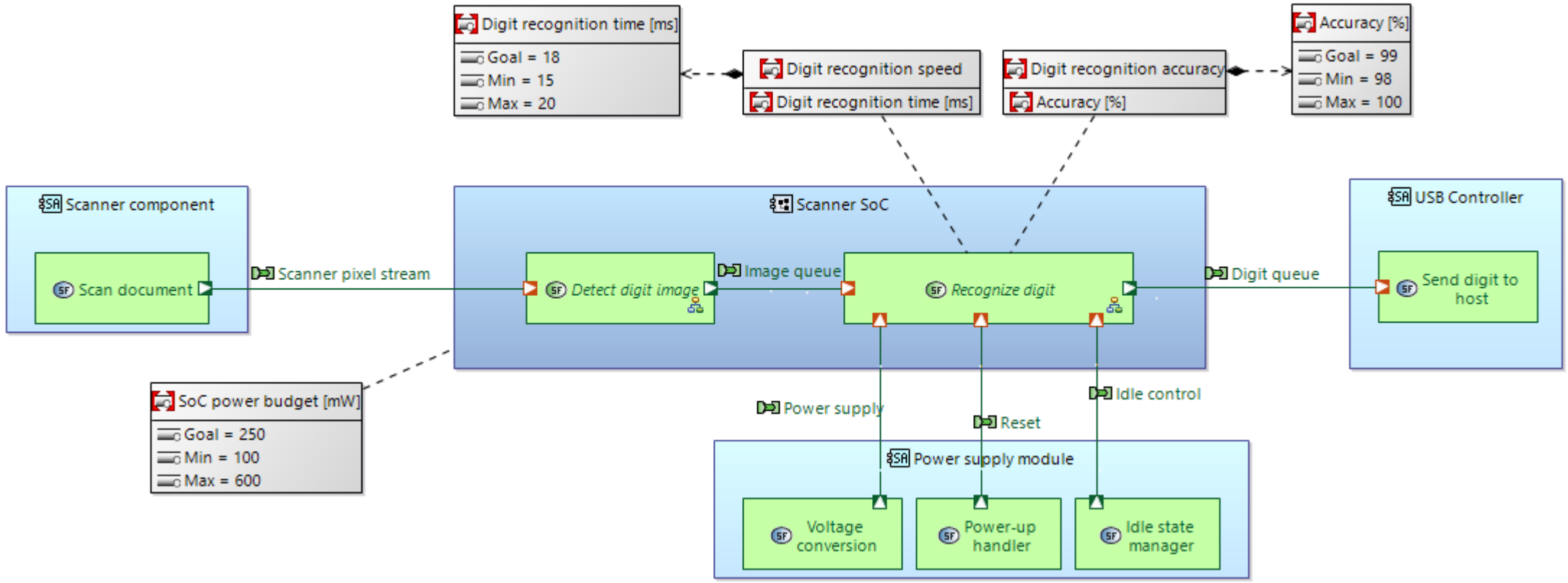
System-on-Chip Subsystem

Digging into Details

accellera SYSTEMS INITIATIVE

Transitioned SoC subsystem

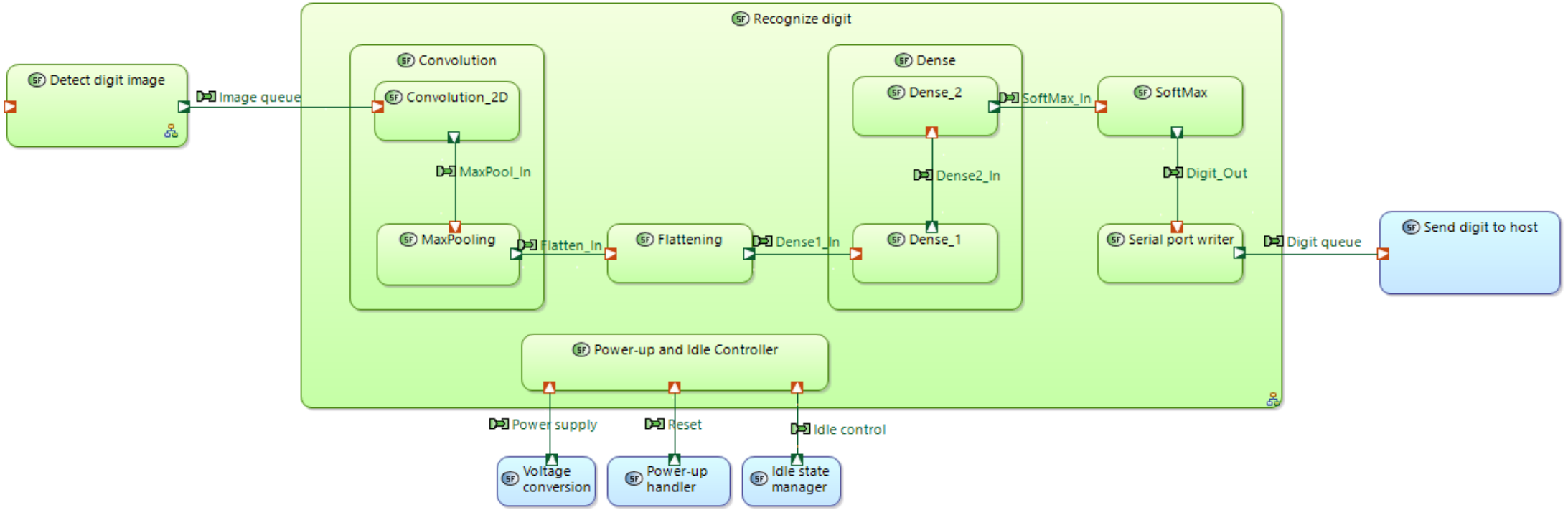
Operational Analysis → **System Analysis** *Formalize System Requirements* → Logical Architecture



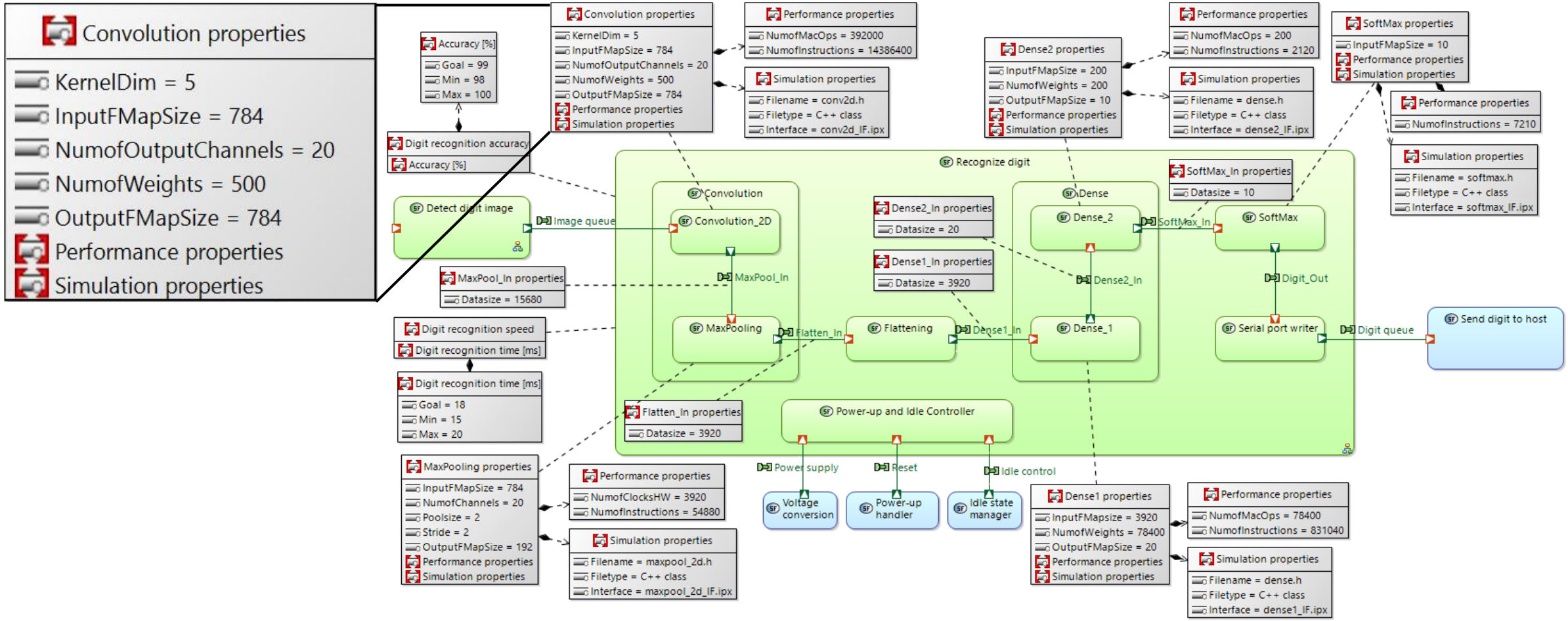
Functional decomposition of the algorithm

Operational Analysis → **System Analysis** *Formalize System Requirements* → Logical Architecture

- Each function implements a partial behavior of the architecture

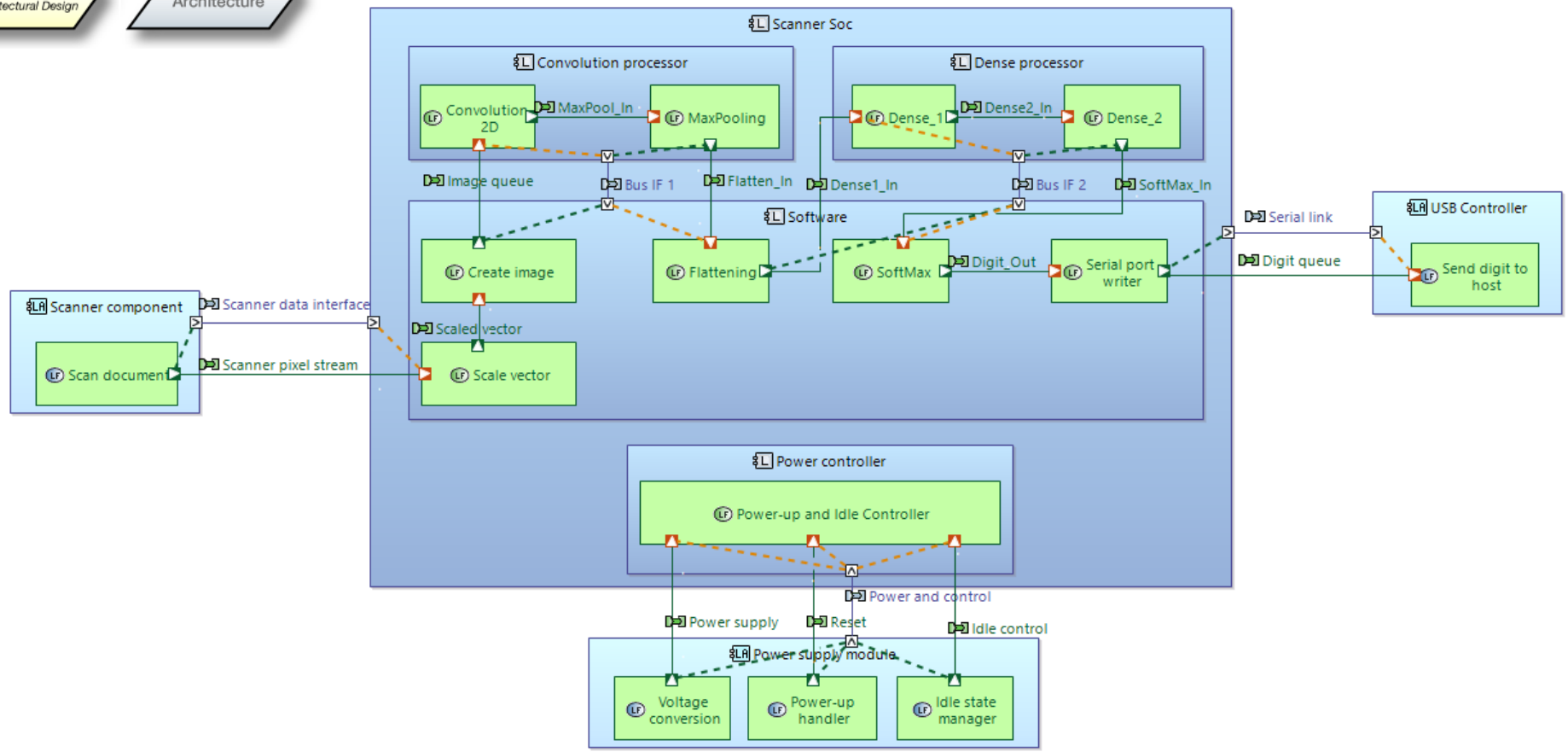


Add functional implementation properties



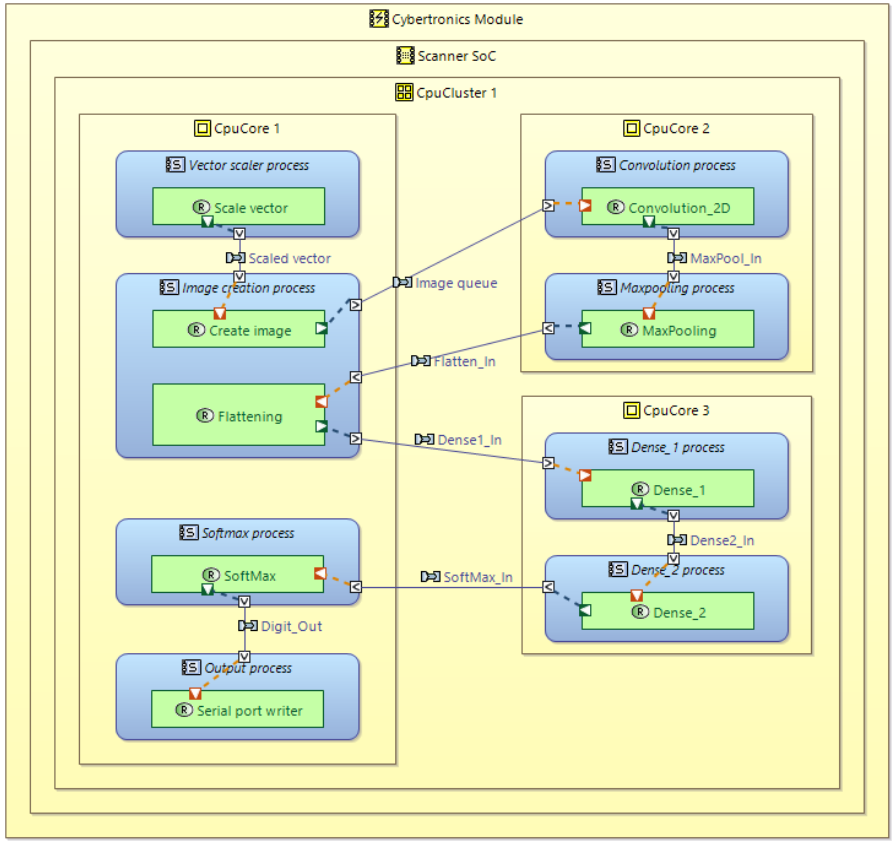
Allocate functions to logical components

System Analysis → **Logical Architecture** *Develop System Architectural Design* → Physical Architecture



Create a performance model

- Performance analysis model uses partitioning in the Logical Architecture
  - Logical System as an ECU
  - Executable functions on SoC with one or more CPU clusters
  - Logical components as CPU cores with the allocated functions on SW components.
- SW functions implemented as synthetic workloads or real SW code
- Data traffic uses a shared memory



2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Architecture Performance Analysis

Exploring Implementation Options

accellera SYSTEMS INITIATIVE

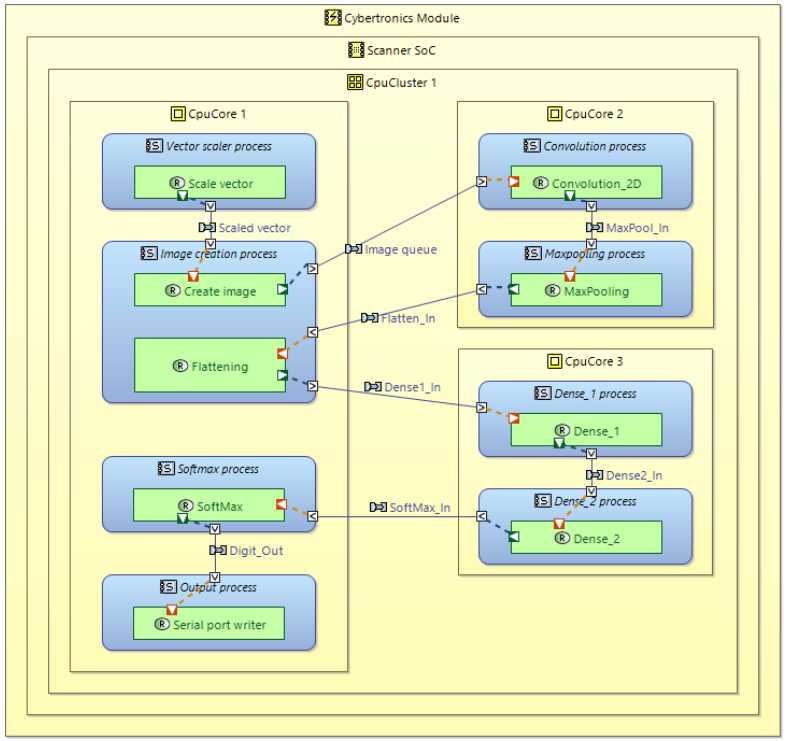
Performance analysis procedure

Performance analysis is an iterative process

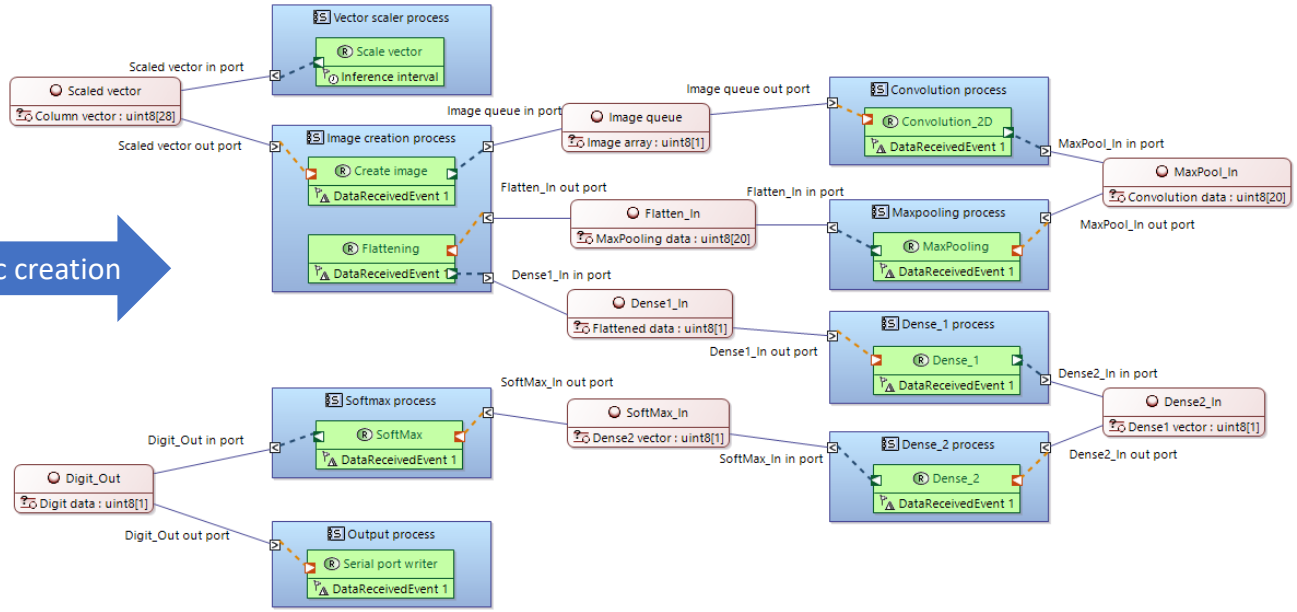
1. Coarse analysis using software implementation of all functions
   - Create an initial HW platform model
   - Allocate functions to SW tasks using synthetic load values
   - Analyze execution results and move functions between the SW tasks
   - Pseudo accelerate functions by using HW accelerator latency as load value
2. Fine-grain analysis with Generic Hardware blocks on the HW platform
   - Replace the accelerator candidates with Generic HW blocks
   - Set data I/O and processing latencies to realistic HW implementation latencies
   - Simulate with different latency combinations to specify latency and clock frequency range for accelerator IP development

Performance simulation model

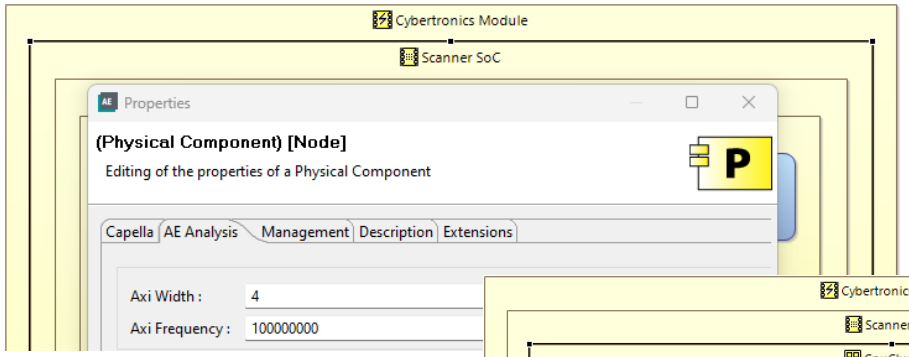
- First trying full software implementation with 3 CPU cores



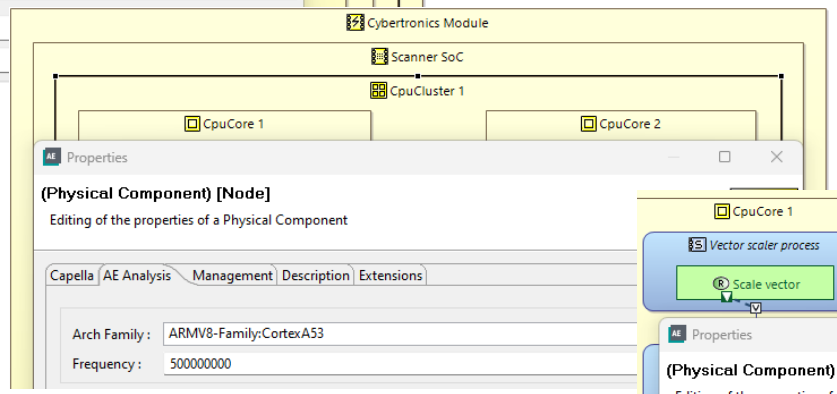
Automatic creation



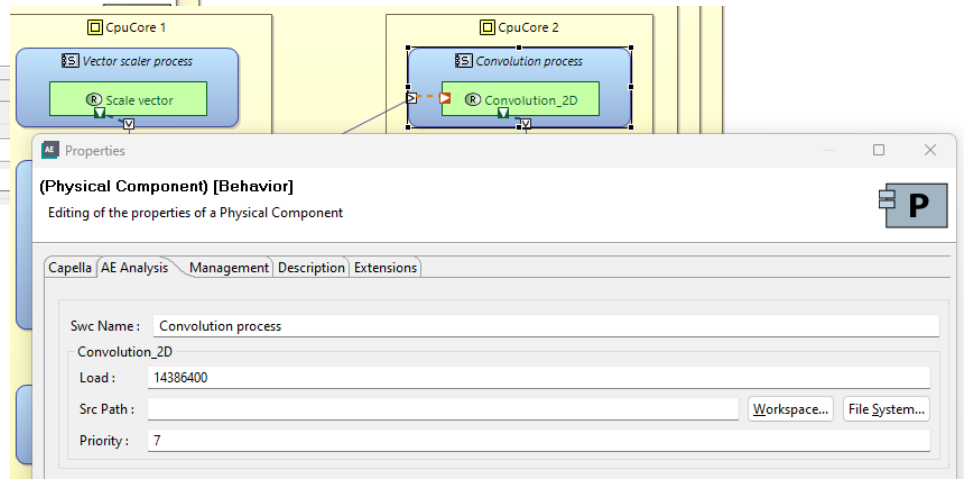
Setting up the simulation



SoC interconnect bitwidth and clock frequency

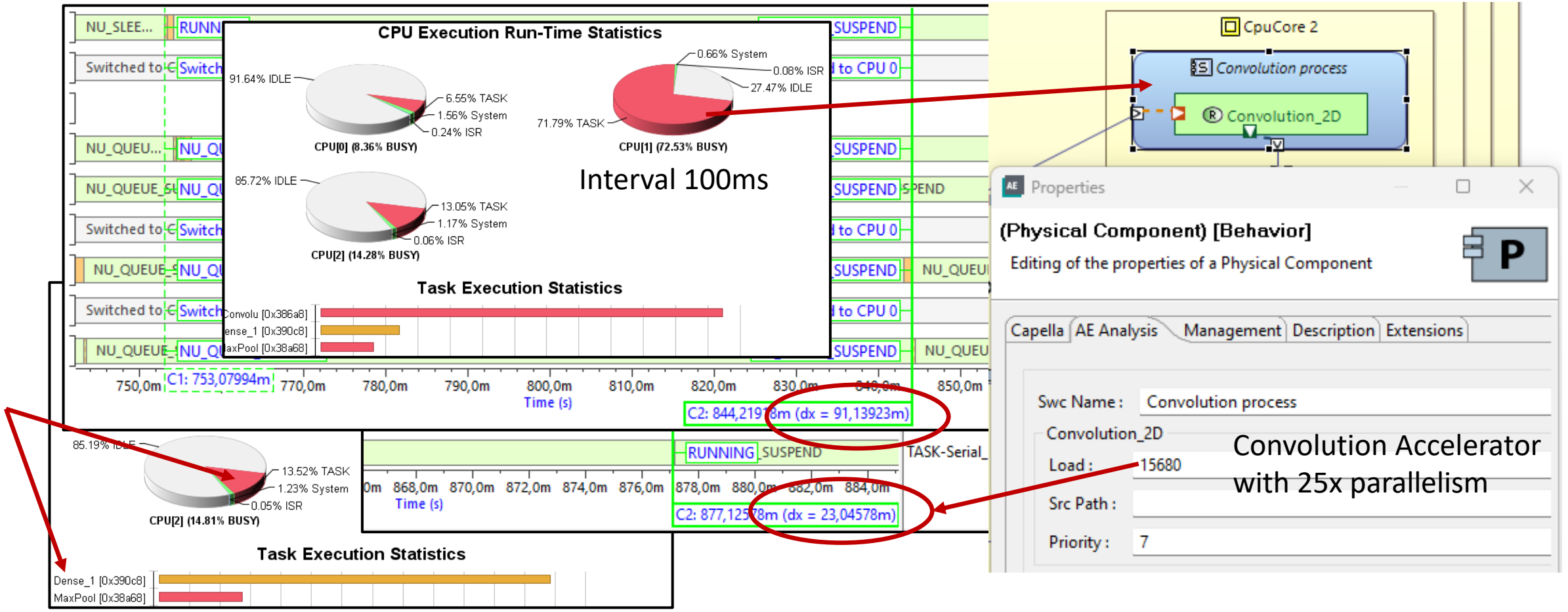


Processor family and clock frequency

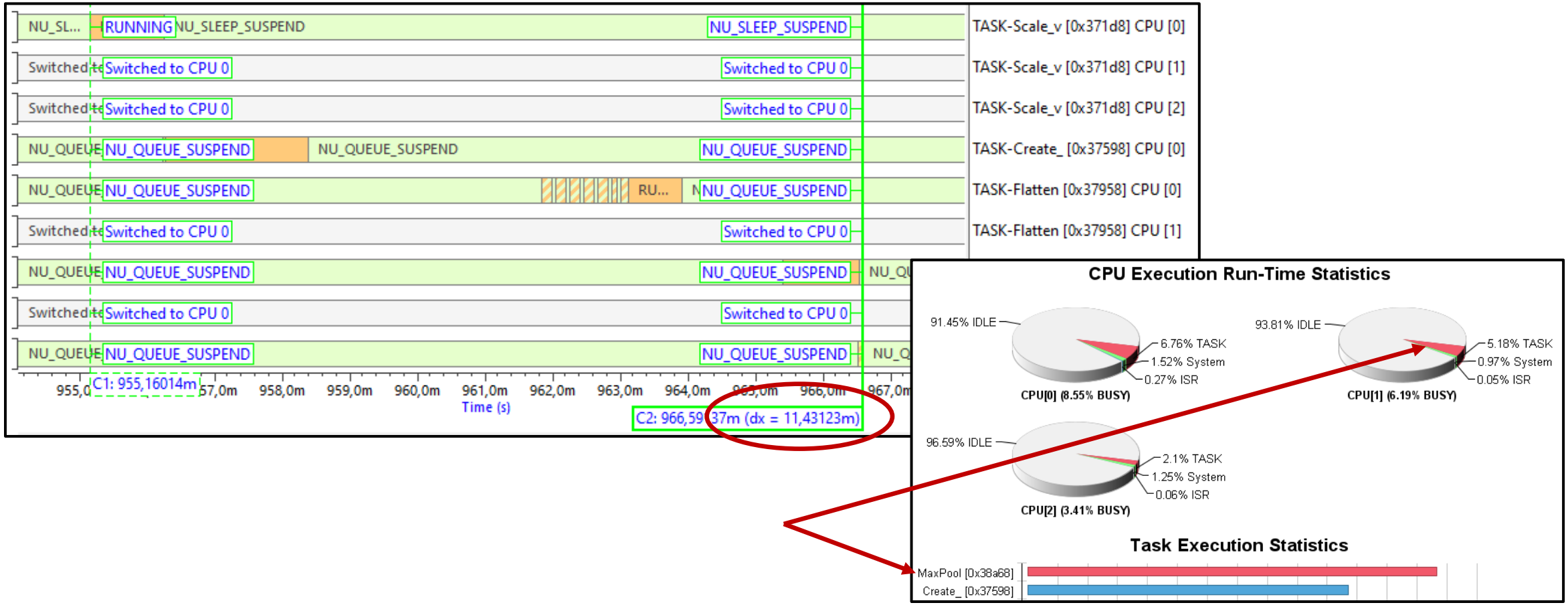


Software workload setting

Exploring different load configurations

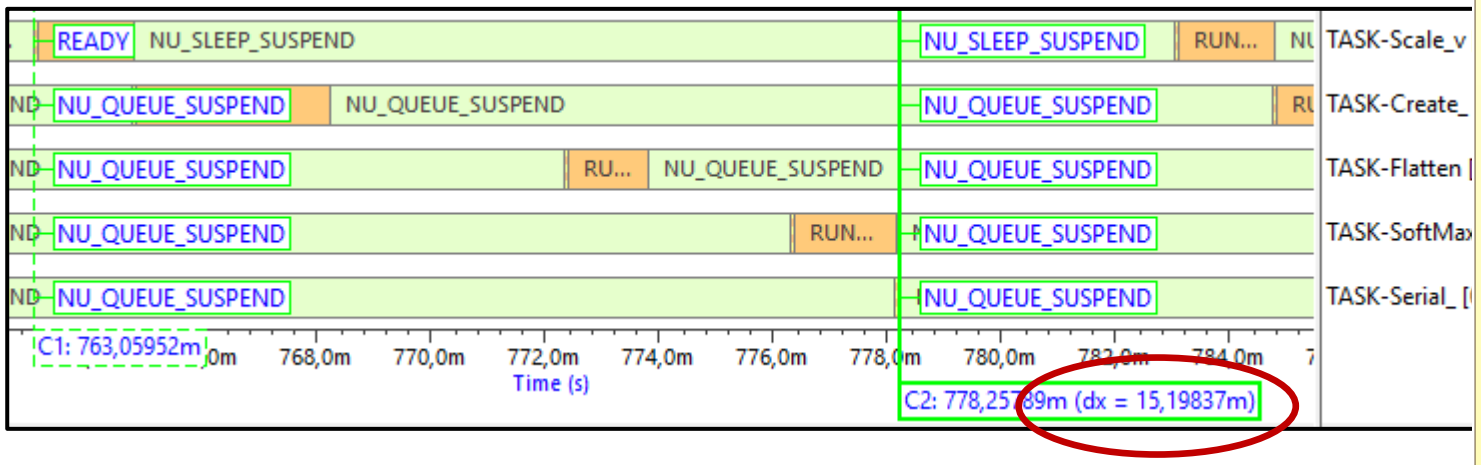


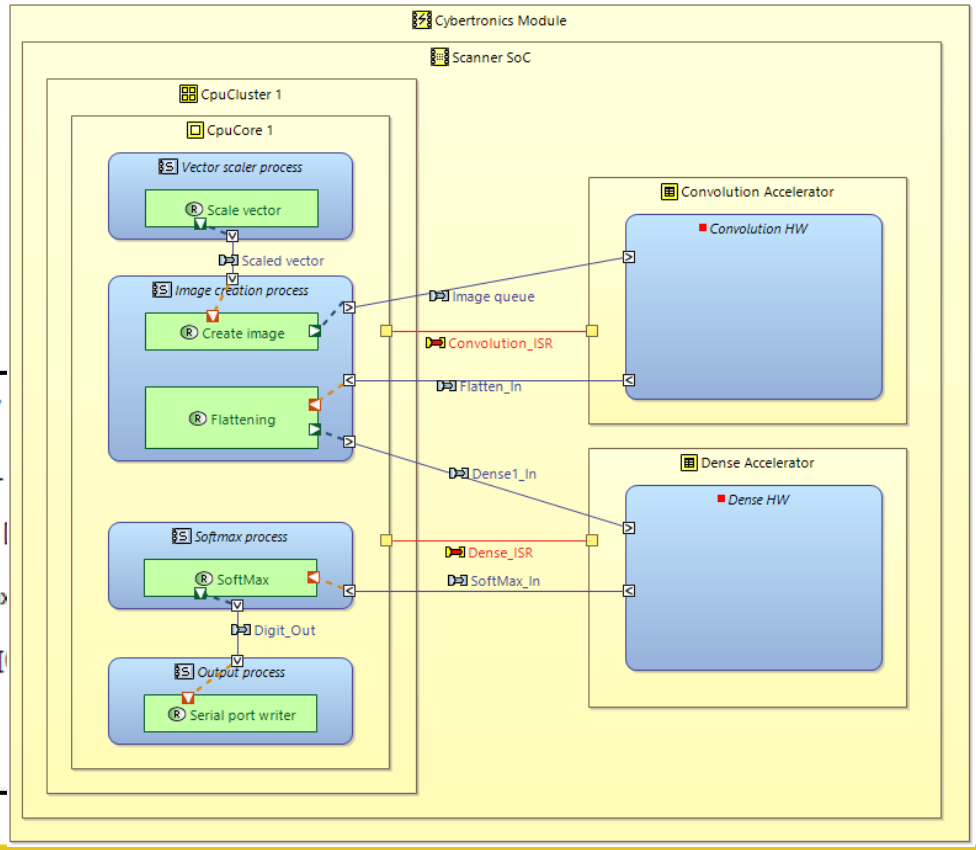
Results with 2 accelerator candidates



Performance with 2 accelerators and 1 CPU

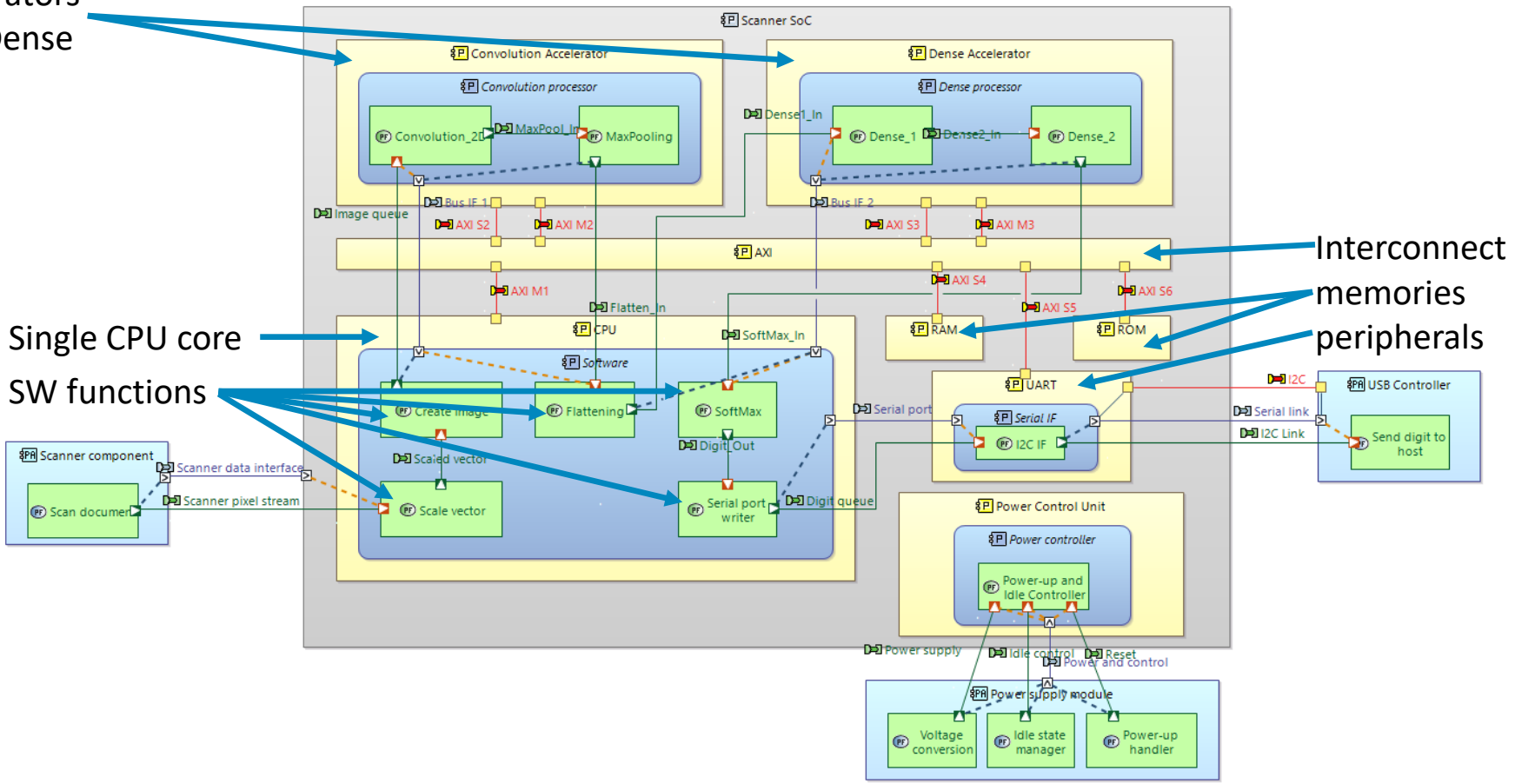
- Modify performance model
- Analyze required clock frequency
- Requirement met with 200MHz clock





Final SoC implementation architecture

Dedicated HW accelerators
for Convolution and Dense



Interconnect
memories
peripherals

Single CPU core
SW functions

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Conclusions

accellera SYSTEMS INITIATIVE

Conclusions

Multi-context modeling

Model-Based Cybertronics Systems Engineering is a new methodology that extends model-based systems engineering into the electronics domain and drives cybertronics design automation

- Formalized methodology and tooling to support
  - System modeling with multi-project and subcontractor support
  - Architecture exploration with early performance analysis
  - Verification management with digital threading
  - Paths to multiple electronics implementation domains
- MBCSE methodology enables a holistic analysis, exploration and implementation of cybertronics systems

Conclusions

Verification capture points

- Verification threading via the capture points (VCPs) enables tracking of all design decisions and verification status

VCP	Parameter	Min	Max	Value	Comment	Version	Model	AE_Project	Config
Scanner Recognition Time	Recognition time [ms]	15	20	90.13	Ran performance analysis based on using a standard CPU on the PCB. Measured value fails to meet budget by a large margin. Suggest creating a custom ASIC for this function.	1.0.0	Cybertronics	Scanner Cybertronics Module_Perf	Cybertronics Module SW Only
Scanner Recognition Time	Recognition time [ms]	15	20	91.14	Simulated an SoC architecture with 3 CPUs, and a full s/w solution, but timing exceeds budget by ~5x. Analysis suggests accelerating convolution function.	1.1.0	Cybertronics Module	Scanner SoC	SoC_Full_SW_3CPU_500HMz
Scanner Recognition Time	Recognition time [ms]	15	20	23.05	Using an accelerator for convolution, but timing is still over budget by ~20%. Analysis suggests that we also need to accelerate the dense layer.	1.1.1	Cybertronics Module	Scanner SoC	SoC_1Accel_2CPU_500HMz
Scanner Recognition Time	Recognition time [ms]	15	20	11.43	With acceleration of the dense layer also, the timing budget can be met. To be within min-max range we will constrain convolution accelerator to 5ms, and the dense layer to 2ms to achieve optimal timing.	1.1.2	Cybertronics Module	Scanner SoC	SoC_2Accel_1CPU_500HMz
Scanner Recognition Time	Recognition time [ms]	15	20	21.07	SoC with CPU plus 2 accelerators, simulated with 100 MHz clock frequency does not meet the requirement. Suggest we increase the frequency.	1.2.0	Cybertronics Module	Scanner_SoC_GenHW	Scanner SoC GenHW 100MHz
Scanner Recognition Time	Recognition time [ms]	15	20	15.20	SoC with CPU plus 2 accelerators, increased clock frequency to 200 MHz gets us within the budget, close to the minimum.	1.2.1	Cybertronics Module	Scanner_SoC_GenHW	Scanner SoC GenHW 200MHz

Questions