KPIT

# Session-III
# Embedded System Design Cycle

# Session outline

- Embedded design process and environment
- Difference between Desktop Vs Embedded application development process
- Role of build and debugging tools
- Various Debugging tools

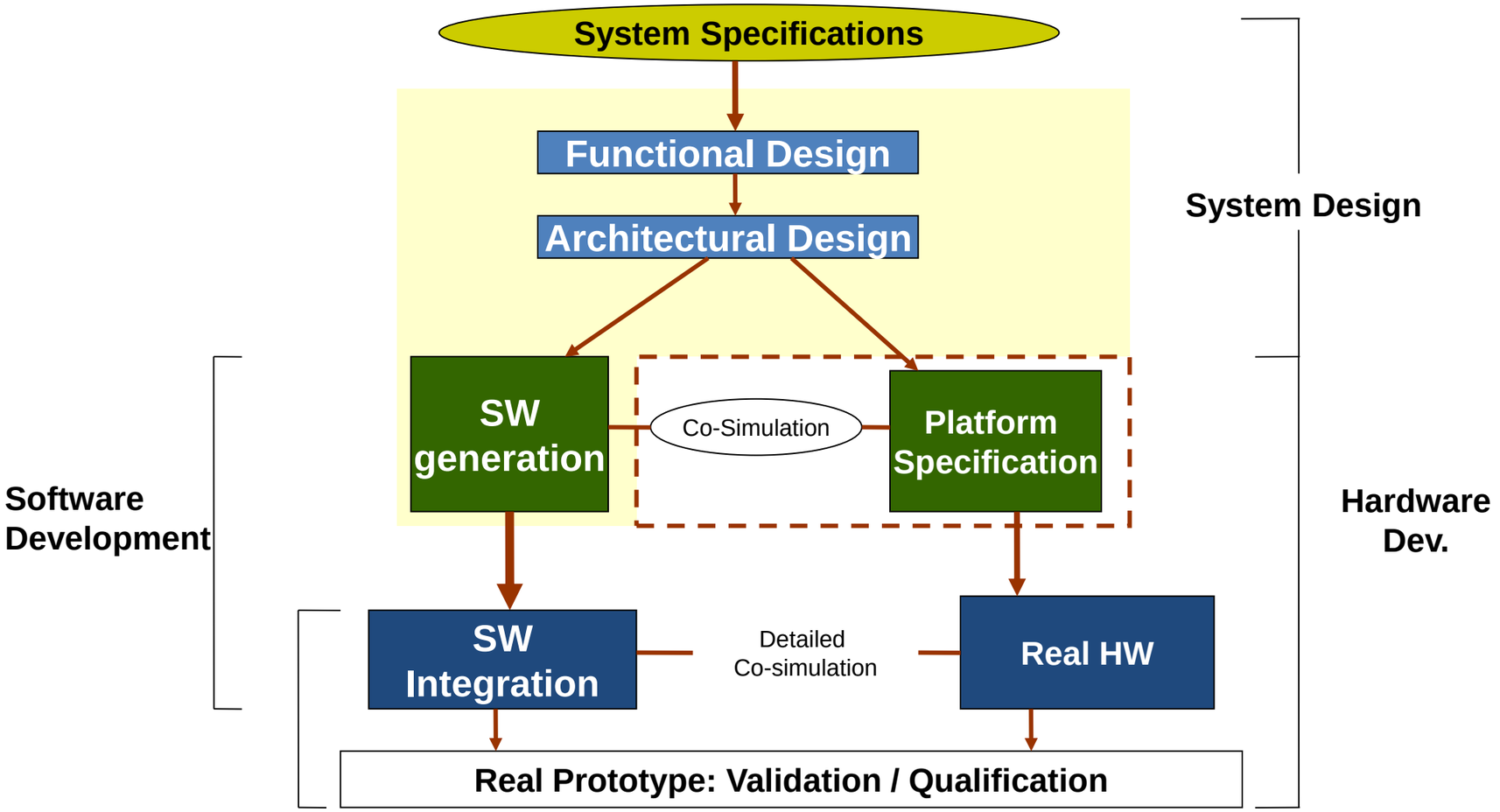
# Introduction to Embedded System Design process

- Embedded Design process can be split into various phases:

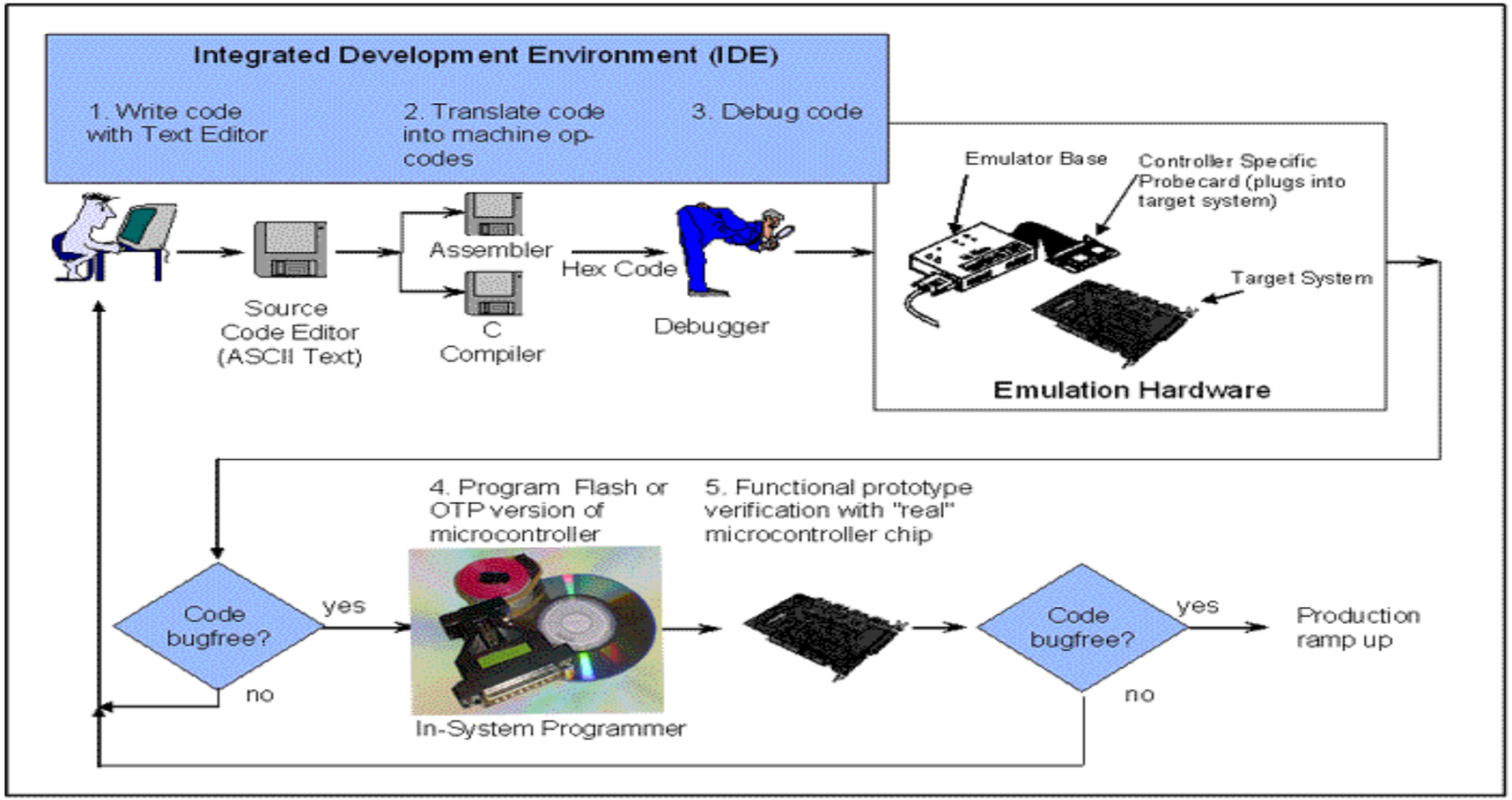
- Product specification
- Hardware/Software Partitioning
- Independent hardware- Software Design phase
- Hardware-Software Integration Phase
- Acceptance Testing (Product/system Testing) and release

Considerable amount of iteration and optimization occurs within phases and between phases.

# Embedded System Design Environment



# Typical Microcontroller Development cycle




The Microcontroller Development Cycle

*Source: **5HSBC –** Research Insight to Win Technology Markets*

# Some important aspects of Desktop based applications

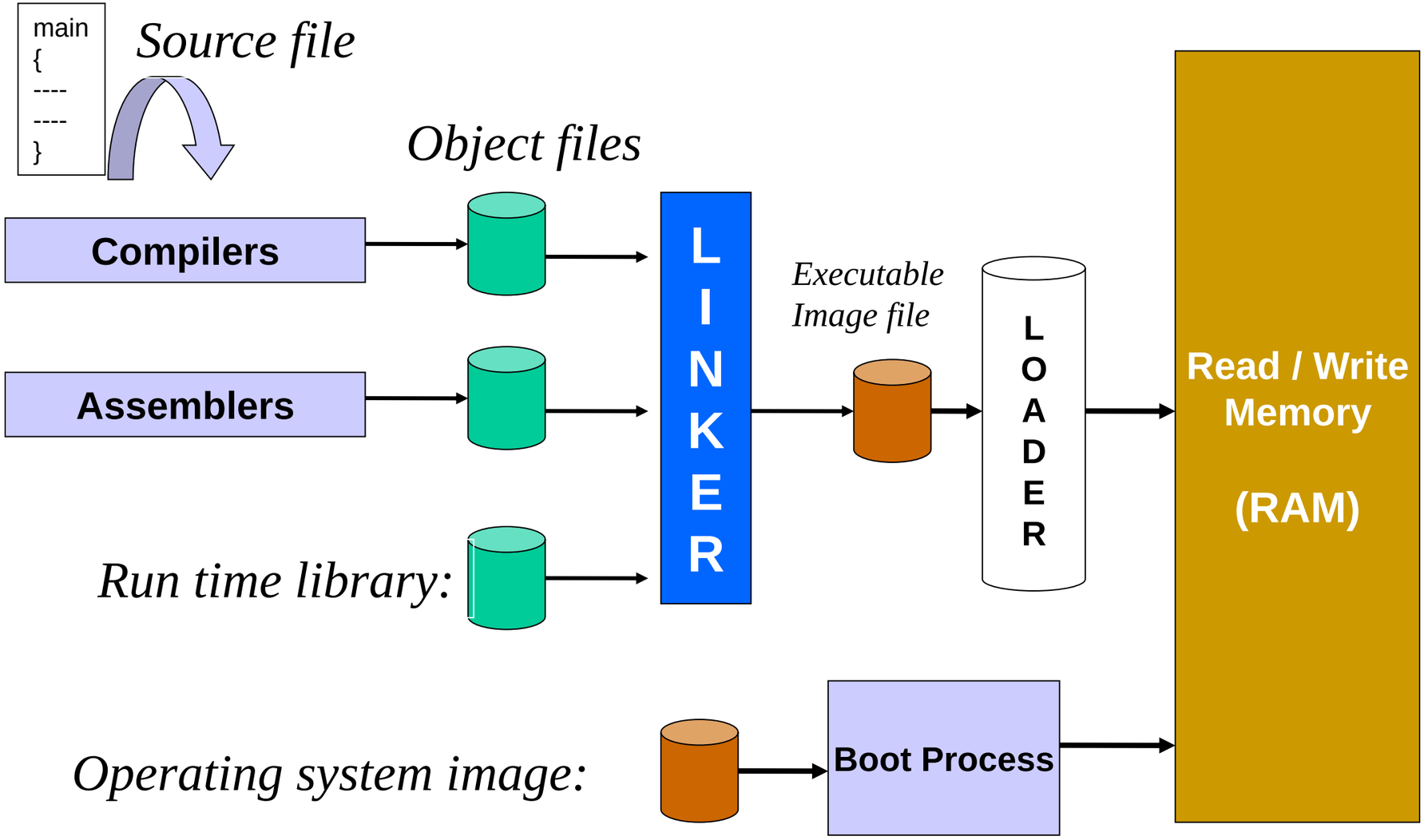
- Source code is seldom translated directly into **loadable binary images.**

- Sophisticated suites of tools translate the source into **relocatable modules**, sometimes with and sometimes without debug and symbolic information.

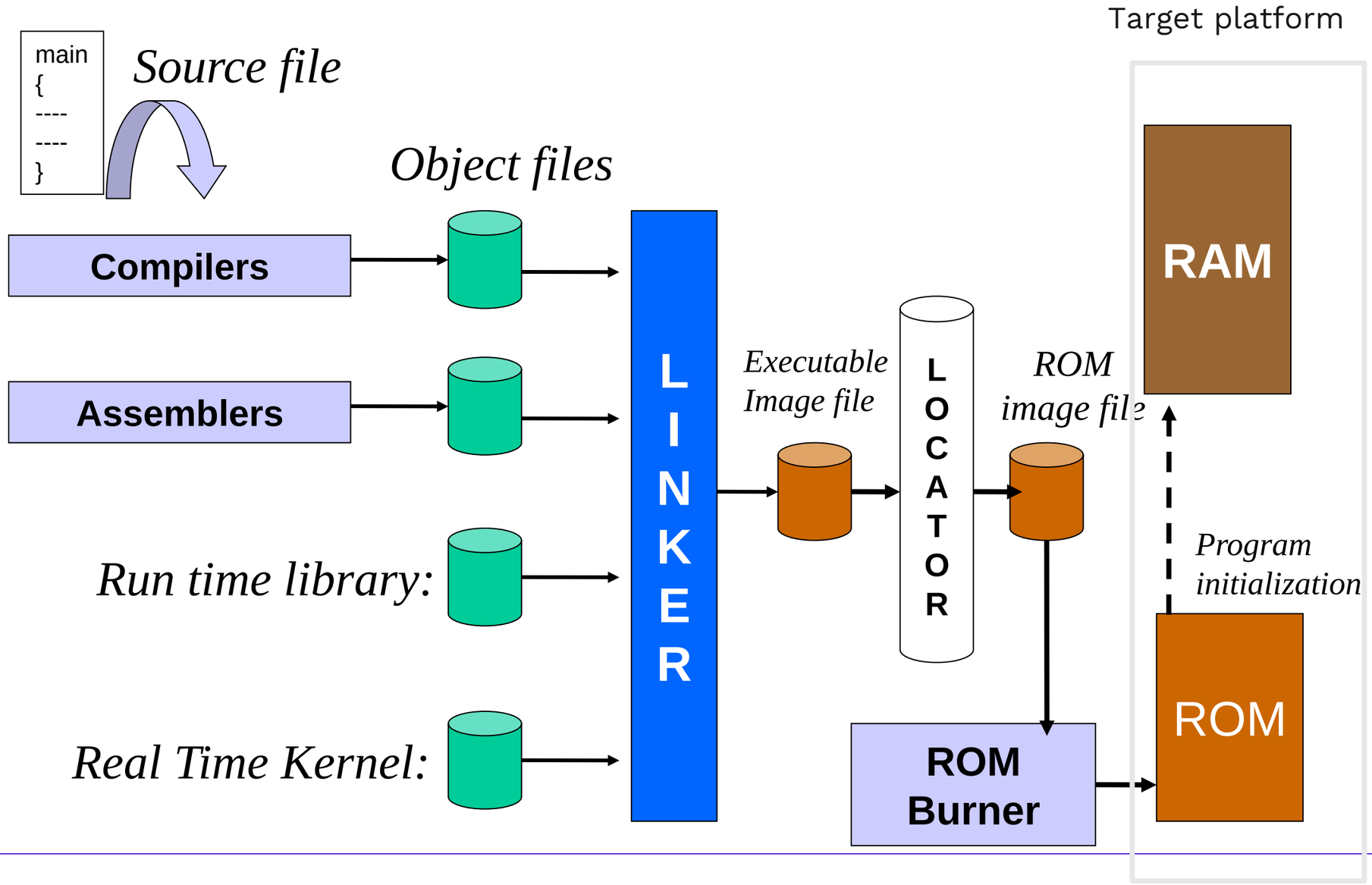
- Complex, highly **optimized linkers and loaders** dynamically combine these modules and map them to specific memory locations when the application is executed.

**Embedded systems developers don’t enjoy this luxury.**

# Build process for desktop applications



# Build process for Embedded applications



# Introduction to Build tools

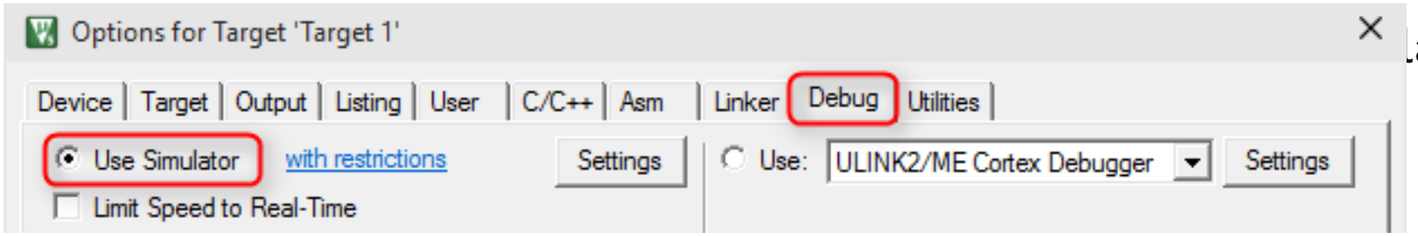
- Build tools (mostly) reside on the development platform but  allow the programmer to debug a program running on the target system.
- At a minimum these tools must:
    - ✓ Provide mechanism to run the program for the target
    - ✓ Mechanism to replace the code image on the target
    - ✓ Provide real-time monitoring of program execution on the target ( capability to observe processor registers, processor sates, etc.)

# Various debugging tool

- Instruction set simulator ISS
- In-circuit simulators
- In-circuit Emulators
- In-circuit debuggers

# Debugging with Instruction Set Simulators

- Software team  use Instruction Set Simulators (ISS) to allow them to compile their target code for their chosen microprocessor.




atform

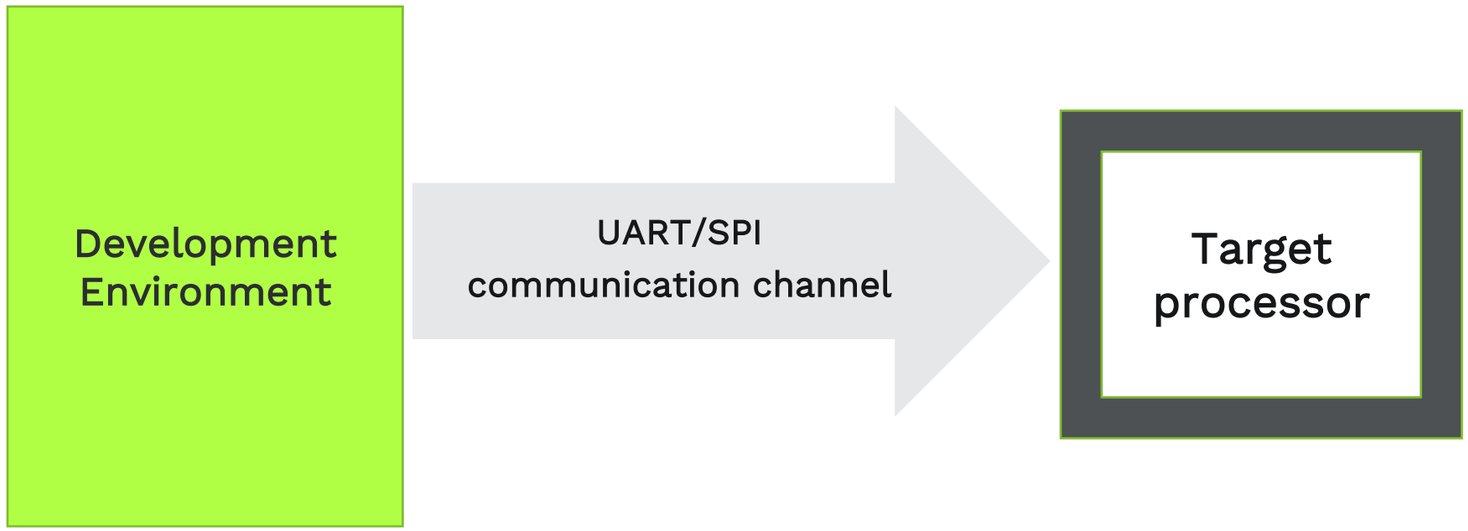
Keil uVision IDE simulator selection

Simulation helps to validate software throughout all stages of the development cycle. A high-level product design can be tested in simulation to see if it serves the intended use and meets all requirements.

# In-circuit simulators

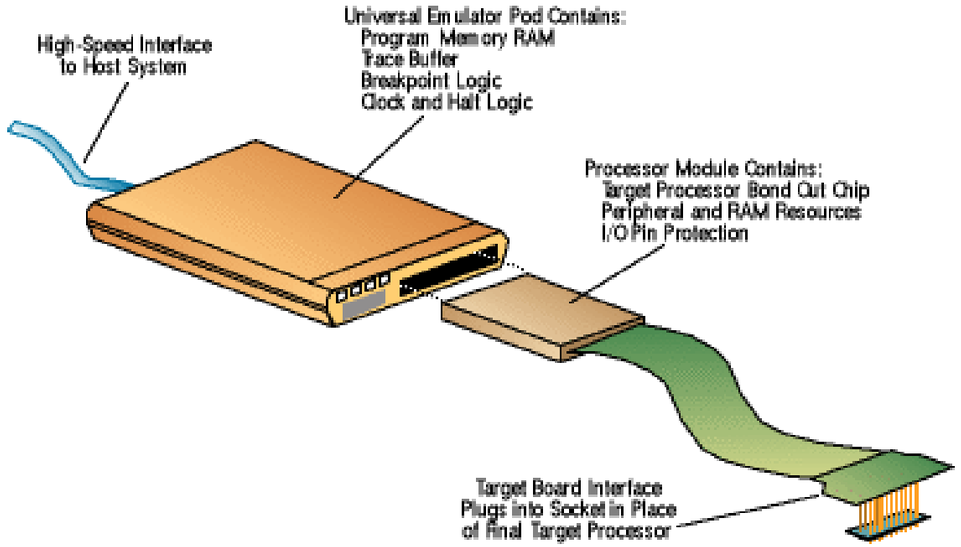
- The approach of integrating the capability of a simulator with a communication module that acts as the target processor is called In-circuit simulation.

- Stimulus is provided to the simulation directly from the processor's digital input pins, which allows the simulator to set binary values on the output pins.



# In-circuit Emulators

An in-circuit emulator is plugged into a target system in place of the embedded processor.




Some in-circuit emulators have special ASICs or FPGAs that imitate core processor code execution and peripherals, but there may be behavioral differences between the actual device and the emulator

# In-circuit debuggers

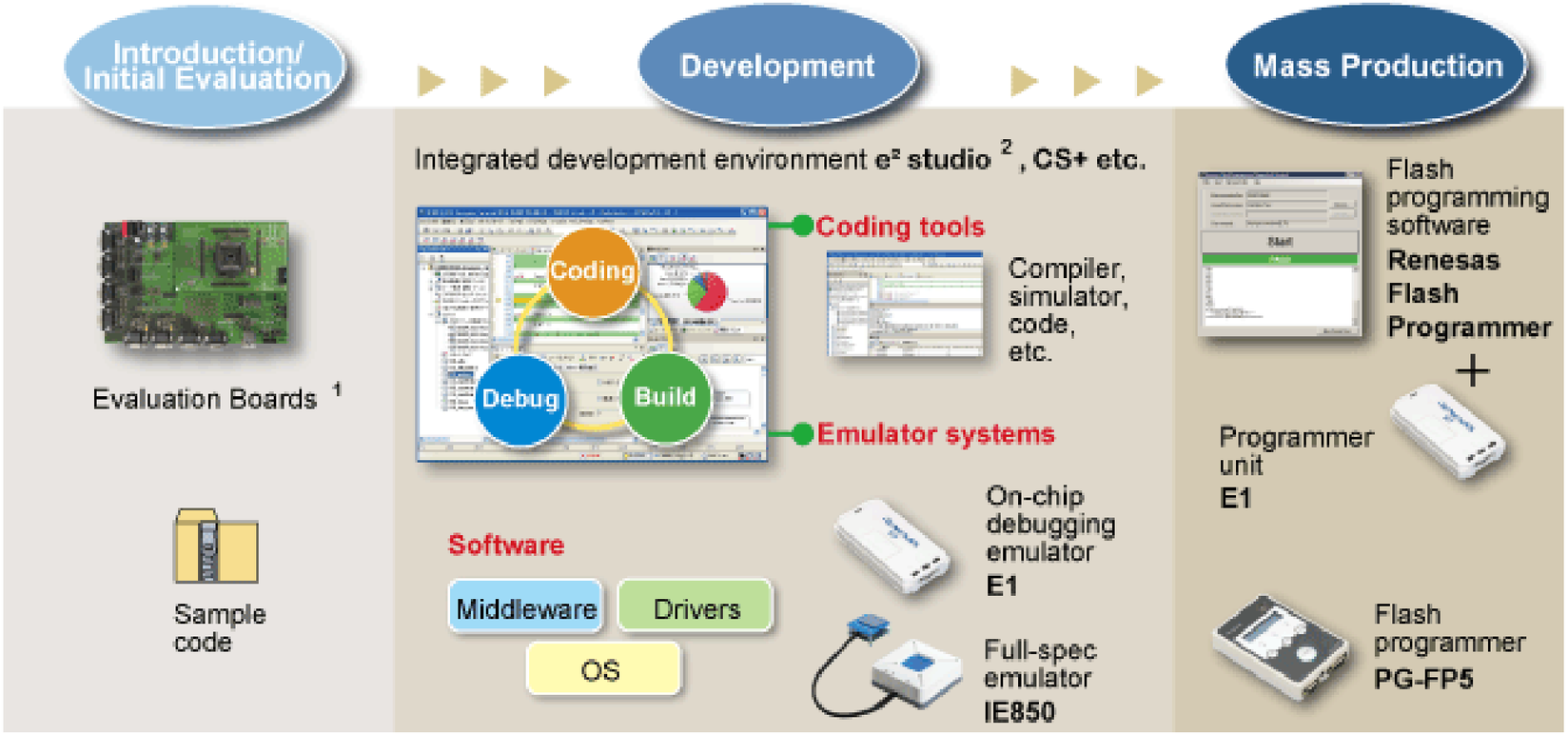
- Development tool with special silicon/hardware features to support code debug and having serial communication between host and target are typically referred to as "debuggers.“

  - ✓ a portion of the debugger resides on the host computer, and a portion resides in the target system.
  - ✓ Target system must support In-Circuit Serial Programming (ICSP) interface

- **Debugger capabilities:**
  - ✓ STOP/START the program
  - ✓ Setting breakpoints
  - ✓ Loading programs from the host
  - ✓ Viewing or modifying memory and registers
  - ✓ Running from an address
  - ✓ Single-stepping the processor


TRACE32-ICD are microprocessor development tools that provides debug and trace logic (BDM, JTAG, ETM, OCDS, NEXUS) integrated on the chip.

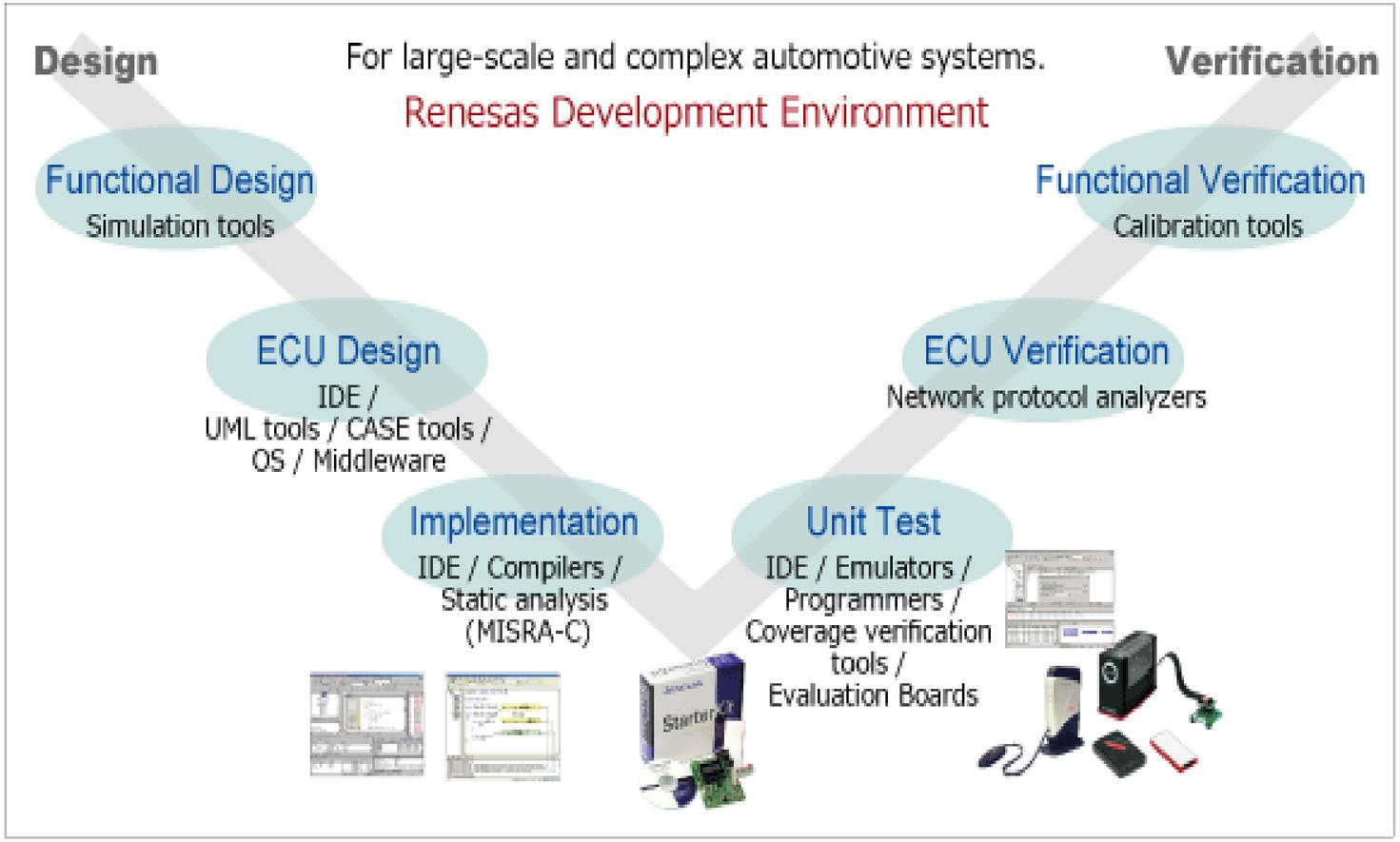
# Tools from Renesas



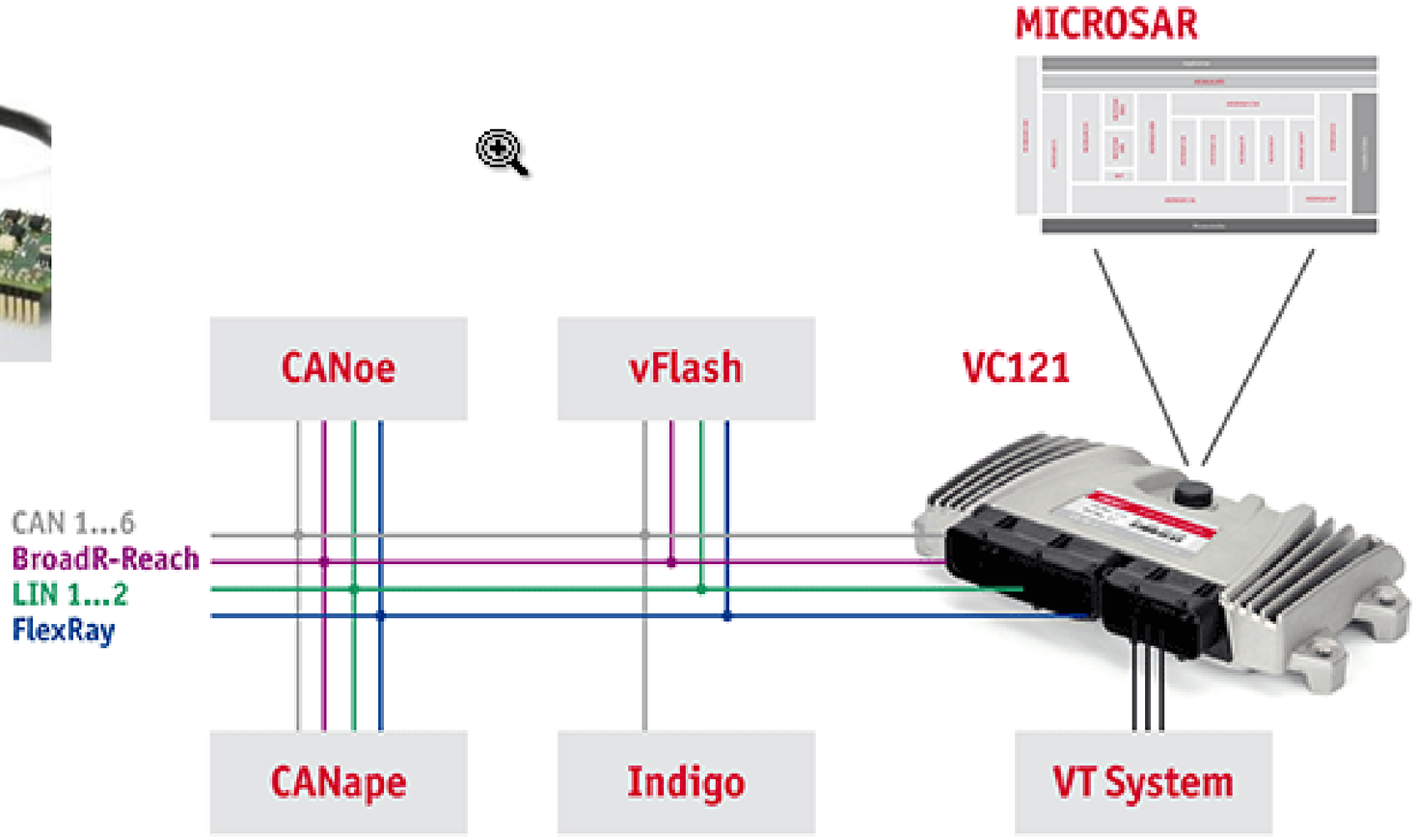

* A free evaluation version is also available for the coding tools and Renesas Flash Programmer.

1. These products are available only in Europe and America.
2. Under development. Debug support only. Renesas compiler is NOT supported.

# Tools from Renesas as per V model



# Tools from Vector

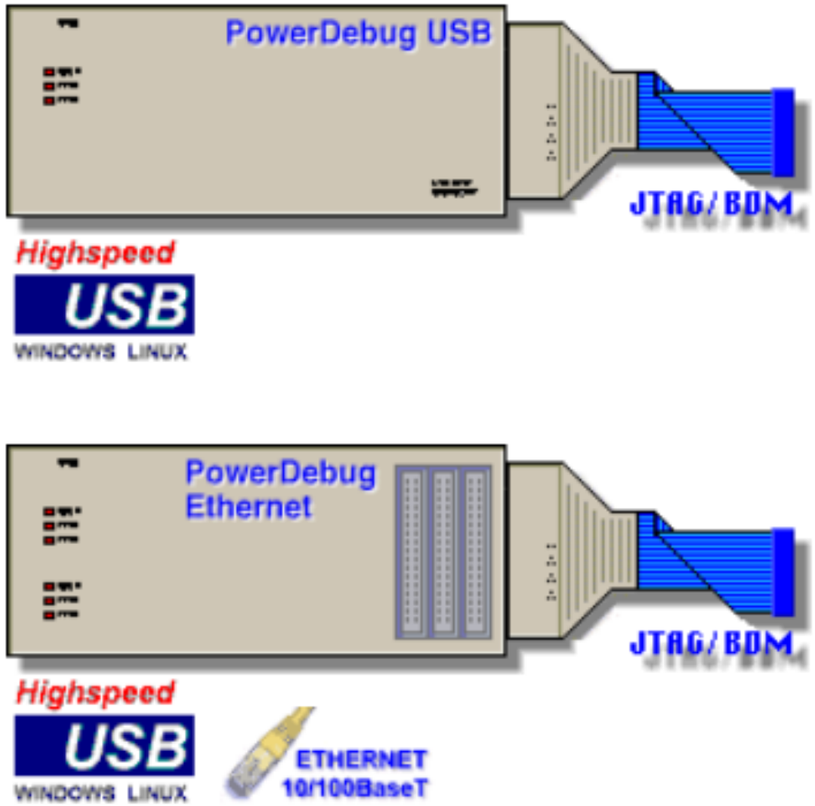



# Lauterbach Products

The Lauterbach product TRACE32-ICD supports a wide range of on-chip debug interfaces. The hardware for the debugger is universal and allows to interface different target processors by simply changing the debug cable and the software. PowerDebug Ethernet can be upgraded to PowerTrace.

The Timing/State Analyzer Module is special designed for microprocessor applications. It can work separately or in conjunction with all ICD modules. The high-speed transient recording allows very long record time when tracing peripheral lines in a microcontroller application

# Lauterbach Products




## High-Speed JTAG/BDM Debugger

- Support for a wide range of on-chip debug interfaces
- Easy high-level and assembler debugging
- Interface to all compilers
- Fast download
- RTOS awareness
- Interface to all hosts
- Display of internal and external peripherals at a logical level
- Flash programming
- Hardware breakpoints and trigger (if supported by chip)
- Trace extension available
- Multiprocessor/multicore debugging
- Software trace
- Virtual analyzer
- USB, Ethernet or Parallel Interface