# Systematic Digital Design – Boosting Design Productivity and Component Reuse through Handshakes and Interfaces

*Natalie Simson, Johannes Ecker, Endri Kaya, Wolfgang Ecker – Infineon*

This 3-hour (plus break(s)) tutorial covers the evolution of digital design, emphasizing interface- and handshake-based methodologies to improve modularity and flexibility. It explores communication protocols for synchronous and asynchronous hardware and highlights practical benefits of these concepts through two RISC-V case studies: a scalar synchronous processor and an asynchronous 2-stage processor. The tutorial is structured in four parts, each taking about 45 minutes.

## 1. A systematic of digital design

Digital design has advanced through new, higher levels of abstraction and the clustering of components, thereby reducing modeling and design details but requiring a strict discipline to be followed. Today, most digital circuits are implemented at the Register-Transfer-Level, which abstracts physical details into discrete values and discrete time, and provides predesigned elements as adders or registers. While this still grants designers control over implementation details, it also requires specifying every design aspect in every clock cycle, ultimately leading to limited modularity and flexibility.

To address these issues, methods such as IP-Reuse, High-Level Synthesis, Processor Synthesis, Memory Compilers, and tools such as Chisel and SpinalHDL have emerged. The benefit of these approaches is limited as they can be applied only to a set of design classes or still require clock-related thinking.

## 2. Interface-Based Design Flow

To overcome the existing challenges of digital design and increase productivity, a shift to modular thinking focused on component reuse is required. To achieve this, a flexible yet standardized interaction between modules is required. Therefore, in this section, we introduce Interface-Based Design in which interfaces are implemented according to a clear definition that also allows their configuration to propagate along connected interfaces.

## 3. Handshake-Based Design in Synchronous and Asynchronous Design

Interfaces play a pivotal role in Handshake-Based Design, supporting consistent management of communication infrastructure: Handshakes provide guaranteed synchronization and data exchange.

Two key methods are asynchronous and synchronous (timed) handshakes. We use these terms for handshake protocols used in synchronous and asynchronous hardware. We outline the protocols and signals required for both approaches, along with the necessary background on their design principles.

## 4. Examples

The handshake-based approaches are exemplified by two different RISC-V designs: first, a synchronous scalar 3-stage processor, and second, an asynchronous 2-stage handshake-based RISC-V processor. The examples deepen the know-how in systematic handshake-based digital hardware design, highlight the benefits of using these methods, and demonstrate the differences in employing handshakes within IPs versus the state-of-the-art RTL design.