IP Data Sheet

TES µEngine

The TES µEngine concept enables complex digital circuit tasks to be created as software-defined functions executed by mini-computers. Each task is assigned to a dedicated µEngine optimized for efficiency, deterministic behavior, and real-time performance.

CONCEPT

A µEngine is based on a configurable CPU called µCore. The µCore remains in source code until a generator creates its synthesizable HDL code according to a specific configuration. Only the required hardware components for the instruction set are generated, keeping the size minimal. Combined with memory and peripherals, the µCore forms the µEngine. Tasks can be defined in C language and are stored in ROM, making the µEngine a true “Software Defined Device”.

Advantages of this approach include decoupling hardware and software development, reducing time-to-market, and enabling late-stage software changes. Programmable ROM options (OTP, EEPROM, Flash) allow updates even after tape-out or in-field use.

RAISE SYSTEM

The RAISE system (Related Array of Independent Single-tasking µEngines) supports the parallel execution of different tasks, enabling the efficient implementation of complex operations. Each µEngine can be optimized for its specific function and disabled when not needed to save power.

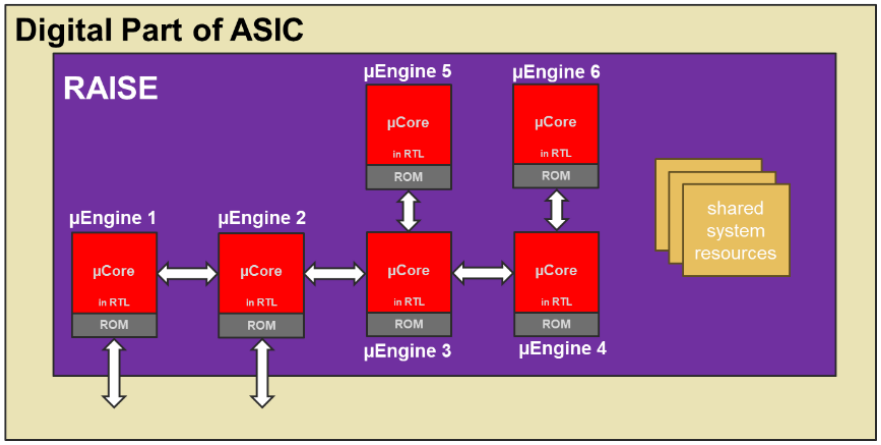

Figure 1: Block Diagram of RAISE System

An entire RAISE system with multiple µEngines operating in parallel, can be cycle-accurately simulated on a PC, ensuring predictable runtime and avoiding hardware dependency during early development.

USE CASES

- Complex state-machines
- Digital filters
- Interface controllers
- Protocol converters
- Hardware accelerators
- Specialized embedded microcontrollers
- Specialized parallel processor structures
- Feedback loop controllers (PD, PI, PID)
- Brushless DC motor controllers

BENEFITS FOR ASIC DESIGN

- Increased efficiency by converting digital design into software development
- Hardware independent and parallel Software development
- Rapid system development and evaluation:
  - Software debug and tests with RAISE Simulator
  - Peripherals modeled as executable Python scripts
  - Evaluation of system functionality during design and conception
  - Cycle-accurate instruction simulation
- Ready-to-use: Verified IP block reuse reduces verification effort
- Reduced simulation effort – software changes do not require gate-level simulation
- Flexibility and maintainability through software updates without RTL changes
- Faster time-to-market via accelerated development

Sales & Marketing Contact

TES Electronic Solutions

sales@tes-dst.com