TAKNEEK PS - ZENITH

(100 Points)

How I met my Computer

(This PS is devised in collaboration with a Professor)

Introduction :

The development of modern embedded systems and IoT platforms requires **tight integration of hardware and software**. Traditionally, physical prototyping using FPGA or ASIC is employed to validate processor designs and operating system behavior. However, this approach is **costly, resource-intensive, and limits design exploration** during the early stages of research.

MATLAB Simulink offers a high-level environment for modeling custom processor architectures, memory systems, and interconnects. At the same time, **FreeRTOS** provides a modular, lightweight real-time operating system capable of handling scheduling, synchronization, and I/O management for embedded devices.

By designing a **custom multi-core computer architecture entirely in Simulink** and **building a complete software stack on top of it using FreeRTOS**. Instead of just communication protocols, FreeRTOS will be extended to include **task scheduling, hardware abstraction drivers, synchronization primitives, inter-core coordination, and user-level applications**.

This approach provides a **flexible and cost-effective framework** for hardware/software co-design, enabling experimentation and evaluation of new architectures without the need for costly prototypes.

TAKNEEK PS - ZENITH

(100 Points)

Task/Problem Statement :

The challenge is to **design and simulate a custom multi-core processor architecture in MATLAB Simulink** and build a **full software stack using FreeRTOS** to run on it. Unlike narrow use cases that only focus on inter-task communication, here FreeRTOS will be used as the foundation for **all system-level software**, including:

- **Core Task Scheduler** – enabling preemptive/multi-core scheduling.
- **Device Drivers** – memory-mapped drivers for timers, mailboxes, UART, and interrupts.
- **Synchronization Primitives** – semaphores, mutexes, and barriers for safe parallel execution.
- **Application Tasks** – implementing real-time workloads (e.g., producer-consumer, master-worker, real-time coordination).

At the hardware level, a **custom architecture must be modeled**, including:

- CPU pipeline stages (instruction fetch, decode, execute, memory, write-back).
- Multi-core interconnects (shared memory, bus, cache support).
- Peripherals (UART, SysTick timer, mailbox/interrupt systems).

The integrated system must then be evaluated under **different workloads and multi-core configurations**.

Deliverables:

Custom Simulink Architecture *(30 Points)*:

- Parameterizable CPU pipeline design.

- Multi-core system with interconnects, shared memory, and peripherals.

FreeRTOS Software Stack *(25 Points)*:

- BSP (Board Support Package) including startup code and context switching.
- Device drivers for memory, timer, and inter-core hardware.
- Scheduler, synchronization primitives, and interrupt handling.
- Real-time application tasks built on FreeRTOS.

Integration & Co-Simulation *(15 Points)*:

- Loading FreeRTOS binaries into Simulink's instruction memory.
- Running workloads with logging via UART or Simulink scopes.

Benchmarks & Performance Analysis *(20 Points)*:

- Contention & Synchronization Overhead Ratio
- Task scheduling latency and context-switch overhead.
- Inter-core synchronization efficiency.
- Interrupt handling response time.
- CPU utilization and throughput under workloads.

Documentation & Reports *(10 Points)*:

- Block diagrams of CPU, interconnects, and memory hierarchy.
- Description of FreeRTOS extensions and drivers.
- Results of benchmarks with graphs and performance comparisons.

Submission Format :

Submit your solution in zip format on the Google Form in the Discord Server later.

The zip file should contain -

TAKNEEK PS - ZENITH

(100 Points)

---

(1) **Technical Report (PDF):** Explanation of hardware design, software integration, methodology, and performance analysis.
**Mention benchmarking and numerical analysis in your report clearly and separately under a unique header*

(2) **Simulink Models:** CPU, multi-core, and peripherals.

(3) **FreeRTOS Codebase:** BSP, scheduler, drivers, and applications.

(4) **MATLAB Scripts:** Automation of simulations and performance logging.

Deadline: 2nd September 23:59.

Evaluation Metrics :

Custom Architecture Quality: Correctness and completeness of CPU and interconnect design.

Software Stack Implementation: Successful integration of FreeRTOS with drivers, scheduler, and applications.

Functionality: Ability to run and demonstrate real workloads.

1. Producer-Consumer: Data Transfer Integrity Rate
2. Real-Time Tasks: Deadline Miss Ratio
3. Multi-core Synchronization: Mutual Exclusion Verification Score

More demos may include: Multi-producer/consumer stress test, buffer overflow handling, deadlock avoidance test, semaphore signaling, priority-based preemption, Interrupt Latency

Contention & Synchronization Overhead Ratio: This calculates the percentage of CPU time wasted waiting on locks, barriers, or memory access, rather than performing useful computation.

Instrument your FreeRTOS code. Use the system timer to measure two things during the parallel sort:

1. Total Execution Time (t_{total}): The full wall-clock time from start to finish.

TAKNEEK PS - ZENITH

(100 Points)

---

2. Total Overhead Time ($t_{overhead}$) The summed time that all cores spend waiting at any synchronization point (e.g., time spent trying to acquire a mutex, waiting at a barrier).

$$Overhead\ Ratio\ =\ \frac{\Sigma T_{overhead\ per\ core}}{T_{total} \times Number\ of\ Cores} \times\ 100\%$$

Performance Benchmarks: Quantitative evaluation of latency, throughput, and utilization.

Innovation: Flexibility of design (parameterizable architecture, scalable to more cores).

Documentation: Clarity of explanations, reproducibility, and depth of analysis.

Rules and Team Composition :

Team will consist of **8 members**

(At max 1 PhD/PG, 1 Y23 B.Tech/BS, 3 Y24 B.Tech/BS, 3 Y25 B.Tech/BS)

Note :

(1) The problem statement is under a professor guidance

(2) Partial submissions are allowed

For Any Queries, The Pool Captains and PS Leads are encouraged to ask in the Discord channel.