Faculty of Computers & Information
Special Program in Software Engineering
Embedded and Realtime Systems
Instructor: **Dr. Anas Youssef**

Midterm Exam
Time: **1 hour**
Number of Pages: **1**
Total marks: **25**

Question 1: **(10 Marks)**

(a) List **four** different components that can be found in a typical embedded system.

(b) Describe by **sample applications** the difference between the following three types of events: **synchronous**, **asynchronous,** and **isochronous** events.

(c) Any embedded system requires what is called a **firmware**. What is the purpose of such software in any embedded system?

(d) Draw a simple diagram that shows the behavioral specification of **an embedded system that controls an elevator in a building** with at **least three different states**.

Question 2: **(15 Marks)**

(a) Briefly describe the meaning of **system synthesis** in the context of an embedded system design.

(b) Draw a diagram that shows a **hardware/software co-design methodology** for an embedded system.

(c) List **three different levels** of **hardware abstraction** that can be used when designing an embedded system.

(d) When designing an embedded system, the designer might be concerned with optimizing some design criteria like **schedulability**. Briefly describe what is meant by this design criteria?

(e) Suppose you want to design an embedded system with the objective to optimize the following three design criteria: **safety**, **reliability**, and **performance**. Each of the three design criteria is assigned a weight out of 10 as shown in **Table 1**. You also have two proposed design alternatives to design the system. The two design alternatives are named: DA1 and DA2. Each of the two design alternatives is assigned a score out of 10 for each design criterion as shown in **Table 2**. Construct a **design tradeoff spreadsheet** for the two proposed design alternatives and use this sheet to decide which of the two design alternatives will be more efficient to optimize the three required design criteria.

Table 1

Design Criterion	Weight (10)
Safety	8
Reliability	5
Performance	3

Table 2

Design Alternative/Design Criterion	Safety (10)	Reliability (10)	Performance (10)
DA1	5	9	6
DA2	8	3	2

Typical Embedded System consists of:

Sensors and actuators

Processor cores and Memory

Emulation and diagnostics and User Interface

Application-specific gates and Analog I/O

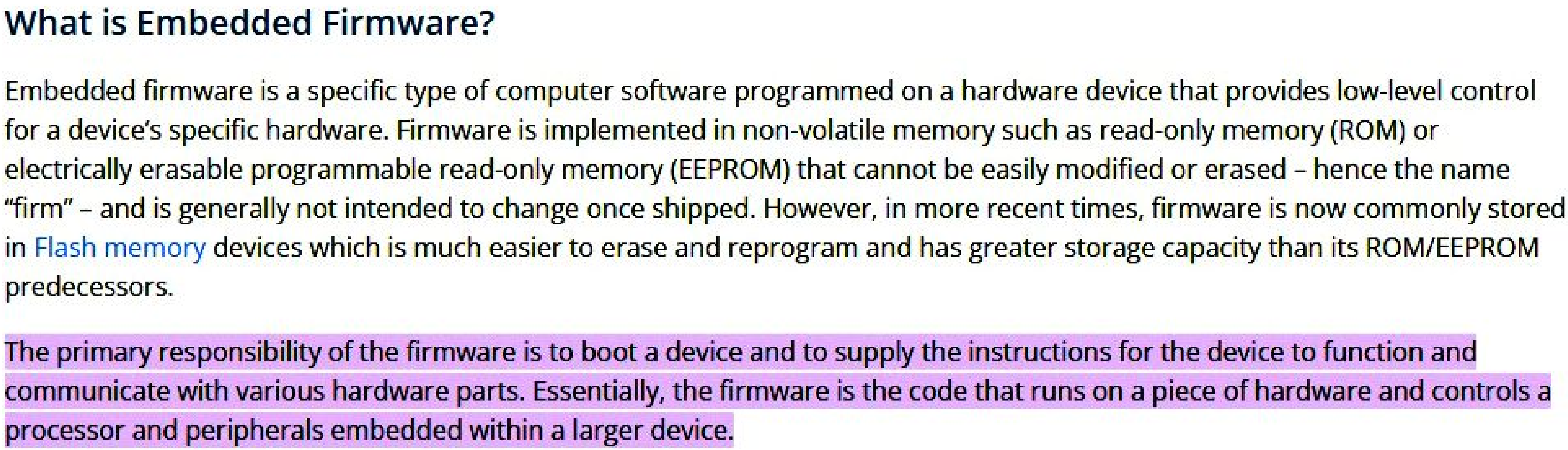
Software/Firmware and Power and Cooling

Describe by sample applications the difference between the following three types of events: synchronous, asynchronous, and isochronous events

Asynchronous events: entirely unpredictable, e.g. cell phone call arriving at a cellular base station

Synchronous events predictable events and occur with precise regularity, e.g. video streaming

Isochronous events occur with regularity within a given time window, e.g. audio data in a networked multimedia application must appear within a window of time when the corresponding video stream arrives

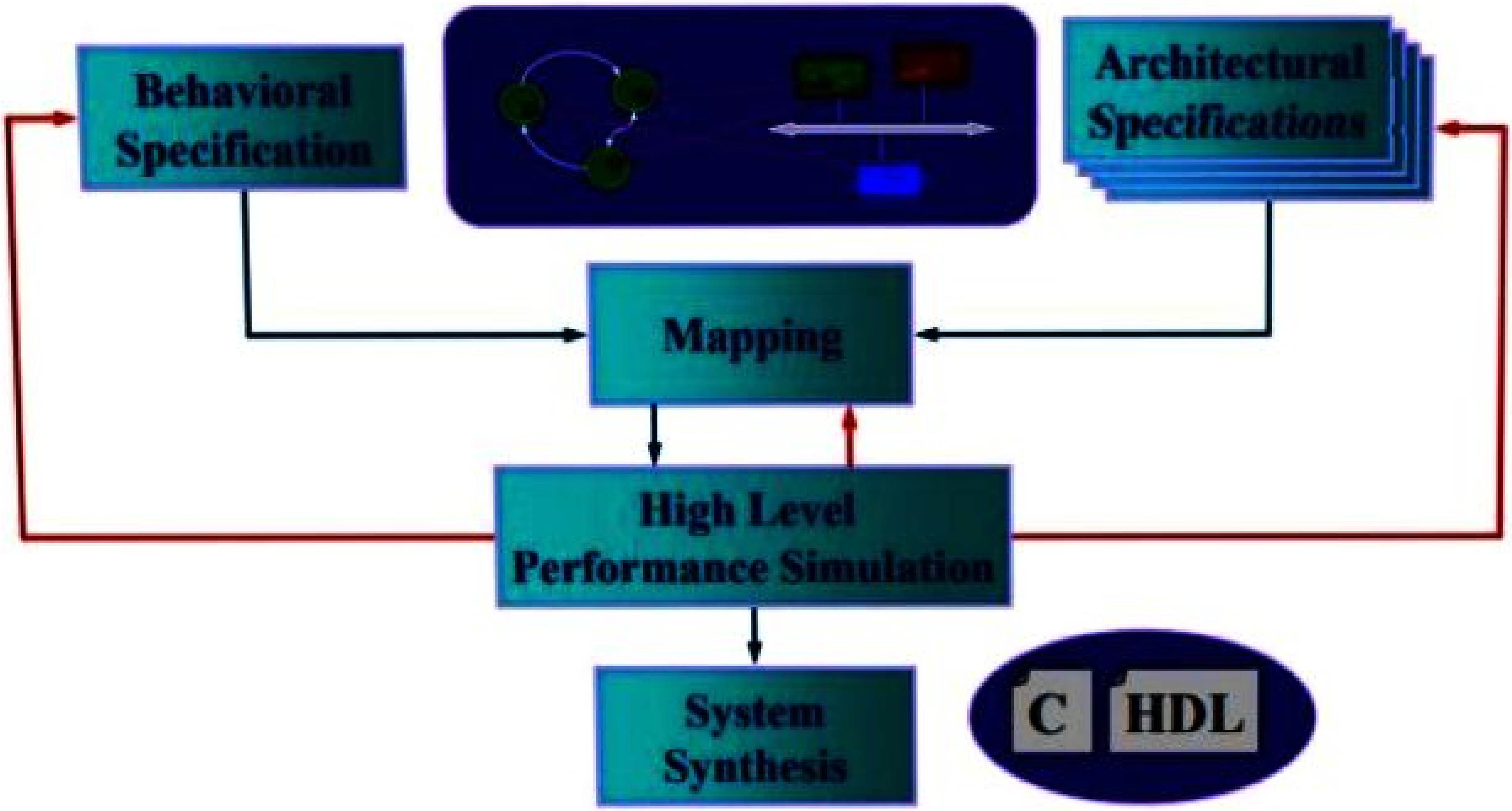


What is Embedded Firmware?

Embedded firmware is a specific type of computer software programmed on a hardware device that provides low-level control for a device's specific hardware. Firmware is implemented in non-volatile memory such as read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM) that cannot be easily modified or erased – hence the name "firm" – and is generally not intended to change once shipped. However, in more recent times, firmware is now commonly stored in Flash memory devices which is much easier to erase and reprogram and has greater storage capacity than its ROM/EEPROM predecessors.

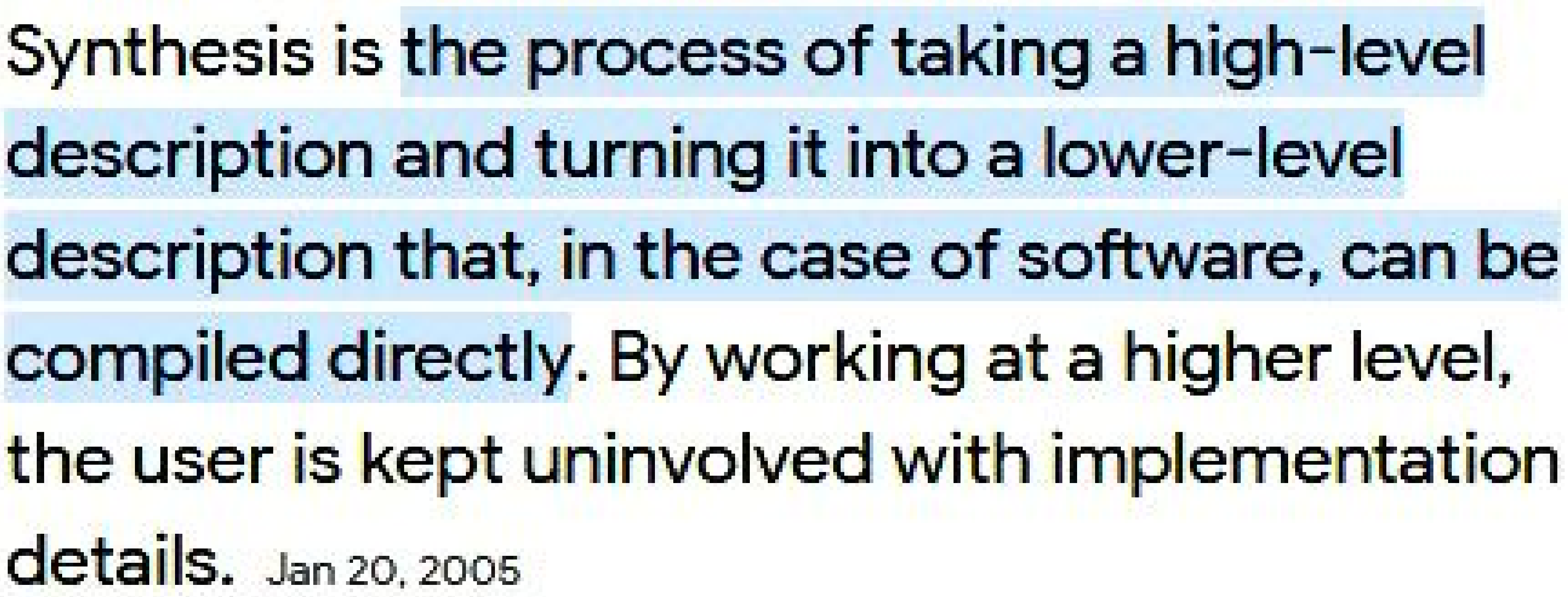
The primary responsibility of the firmware is to boot a device and to supply the instructions for the device to function and communicate with various hardware parts. Essentially, the firmware is the code that runs on a piece of hardware and controls a processor and peripherals embedded within a larger device.

Embedded System Design Behavior/Architecture Co-Design Methodology


Behavioral Specification
Architectural Specifications
Mapping
High Level Performance Simulation
System Synthesis
C
HDL

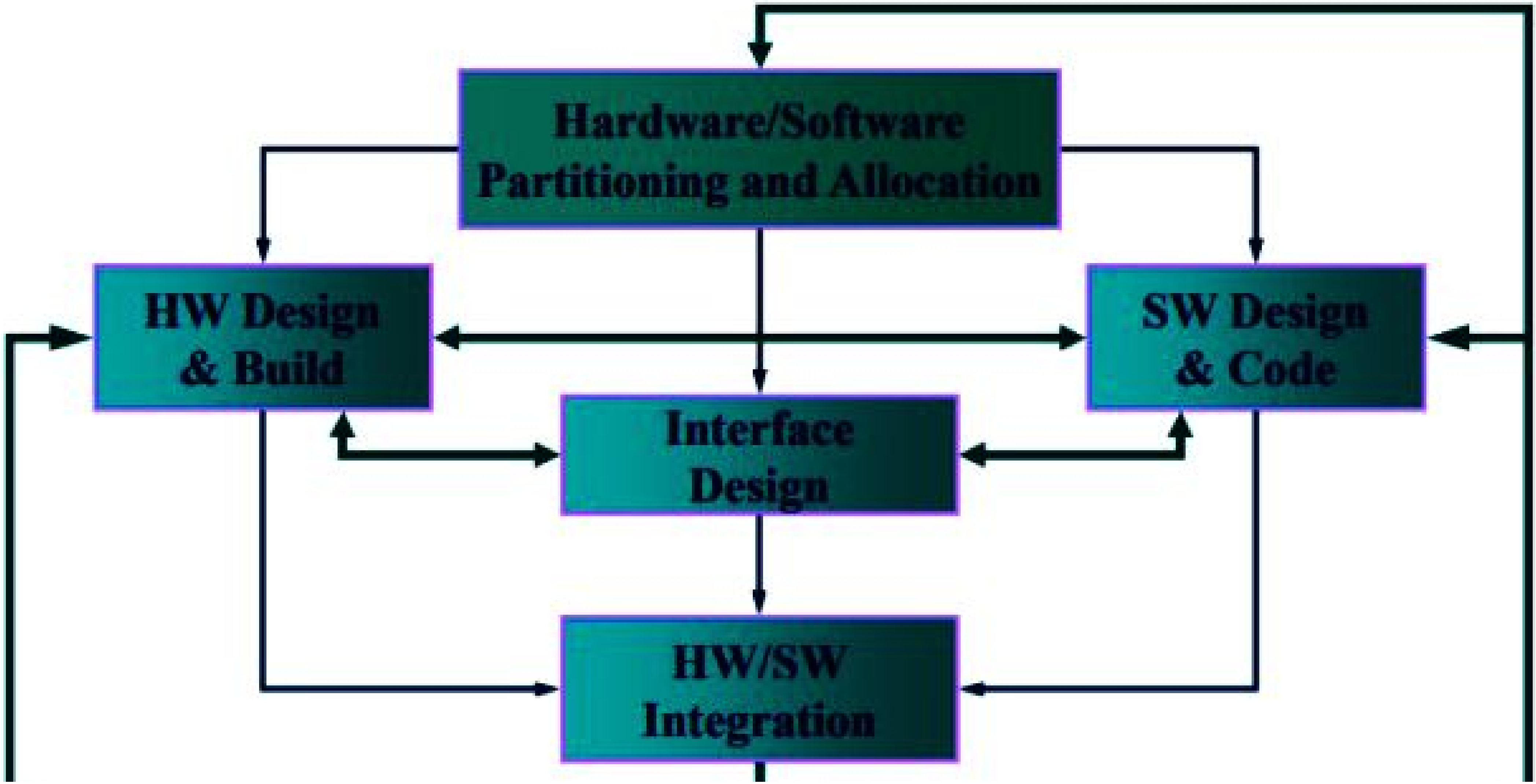
Behavioral Specification and Architectural Specification -> Mapping -> High Level Performance Simulation with the ability of returning back modifying specification -> System Synthesis (بناء السيستم كهاردوير)

HDL: Hardware Description Language



Synthesis is the process of taking a high-level description and turning it into a lower-level description that, in the case of software, can be compiled directly. By working at a higher level, the user is kept uninvolved with implementation details. Jan 20, 2005

Embedded System Design HW/SW Co-Design Methodology



Some Common Design Optimization Criteria

- Performance

  - Worst case

  - Average case

- Predictability

- Schedulability

- Throughput: number of tasks to be executed per second

  - Average: Average time to execute task

  - Sustained: not working average, but always working at peek potential (ثبات الاداء مع زيادة التاسكات)

  - Burst: القدرة على التعامل مع فترات الضغط المفاجئة زي وقت نزول النتيجة

- Reliability

  - With respect to errors or failures