1. **Describe the main Hw and Sw solutions to optimize the power consumption.**

- Hardware:
  - Leakage Power
    - Multi-VT: The leakage of a transistor increases with the decrease in Vt, thus its possible to use high Vt cells when there aren't timing restrictions and then use low Vt cells when the timing requirements are high.
    - Power Gating: If there's no voltage applied to block there will be no current. So, it's possible to complete turn of the supply voltage to a block to reduce its static consumption.
    - Long Channel Devices: With a higher L, the devices present a lower leakage, however there are also downsides to this approach since the increase in L means an increase in area and thus capacitance, and reduction in the current. So, we reduce the driving strength but reduce the leaked power.
  - Dynamic Power:
    - Frequency Control by Clock Gating: On a hardware level its possible to just shut off the clock, thus not having losses.
    - Voltage control by Variable Supply Voltage: Its possible to reduce the dissipated power by reducing the voltage excursion that the output has to do by simply reducing the supply voltage.
  - On a system/block level the layout of the ASIC/SoC can also be optimized to reduce the power dissipation, such as by guaranteeing the needed clearance for power tracks.
- Software (at OS level):
  - Use Advanced Configuration and Power Interface (ACPI), so part of the power management goes to the operating system. Thus, it can have several power modes that can dynamically control the Operating Point (OP) which is the pair of the supply voltage and the clock frequency, which can be controlled independently. These power modes are different power states, like sleep, power or performance. Those state can be applied to global system (G-State), system-wide (S-State), CPU (C-State), performance (P-State), devices (D-State)
  - On a higher level one can implement power-aware algorithms and efficient coding practices.

2. **Compare microcontrollers and general-purpose microprocessors in terms of application fields, cost, computing power, power consumption, Software Development Toolkits**
    - Application Fields:
        - GPP – Very generic architecture, so very versatile. Used in low end ES when energy and performance are not crucial. Might be used when a prototype done with a MCU is ready for production. Will require a PCB or SoC to integrate all the peripherals. Limited interaction with the environment.
        - MCU – A more specific processing unit, which aims to tackle simple problems, which don't require top computing power, but have constraints on the use of resources and development time. Easier to develop a prototype and working example. Useful because there's no integration to be one, since its already a SoC. Built to interact with the environment.
    - Cost:
        - GPP – Higher price
        - MCU – Lower price.
    - Computing Power:
        - GPP – It has more computing power but it more power hungry.
        - MCU – It has less computing power but is more specific and low power.
    - Power Consumption:
        - GPP – Are typically more power hungry, however using RISC can become a low power solution.
        - MCU – Usually done to be low or ultra-low power with very power management and adequate usage of power states.
    - Software Development Toolkits:
        - GPP – The programming is done in assembly. Either in RISC or CISC.
        - MCU – Programming is done in C, and only something in assembly, and there are several SDK, ranging from simple C compilers (like Arduino IDE) to complete integrated framework for performance analysis and configuration management (like STMCube).

3. Describe the design flow of an embedded systems based on COTS + PCB.

- The first step of a PCB-based design is the choice of the COTS.
  - Several types of components which can be passive, power supply, converters, filters, electro-optical components, RF components, displays, sensors and digital components.
  - They also have different packaging which alters the mounting, positioning of the pins and material of the package, and therefore the design choices.
  - Should consider effectiveness, pin-count, pitch and thermal conductivity when choosing the package.
- Design the physical support.
  - After choosing the components, one must design the PCB, composed of 3 main elements, the conductors, the insulators and the glue. In the end a board is a stack of layers of insulator, where their faces host the wires, separated by layers of pre-peg
- Send the board to production and wait.
- Test the board.
  - After the realization of the board, it is necessary to do electrical tests to verify if the contact points are connected or isolated.
  - Can be done by a bed-of-nails, flying-probe or X-Ray inspection
- Mounting
  - With a board which is function one must solder and mount all the components. For small boards and limited volumes, the mounting is carried out manually. For volume and/or high complexity, specific equipment is used.
- Iterate if needed.

4. **Design Space Exploration: which are the main problems to be solved? What is the meaning of optimal and robust optimization? What is the concept of fidelity of the evaluation metrics?**

- Main problems to be addressed:
    - Hardware platforms are used by vendors as reference designs for family of applications, allowing for design time customization targeted to a specific application, pre-verified configurable IP that are instantiated and sized to meet application-specific constraints and allows for low-risk deployment while meeting time-to-market constraints. Thus, DSE is the tunning process aimed at solving these problems.
- Meaning of optimal and robust optimization:
    - While optimizing its desirable to find the global minimum, thus making the optimization optimal. If a local minimum is found instead, the system could still be further optimized. However, due to the fact that in reality most of the optimization problems don't have fixed paraments, but stochastic ones, the optimization should be robust to the point where that, even with the inherent randomization and stochasticity the system is still optimized. Note that finding the best robust solution doesn't mean that the best global solution has been found. To address such a problem usually a Multi Objective Optimization Algorithm is needed.
- The concept of fidelity of evaluation metrics:
    - The fidelity of an estimation is the percentage of correctly predicted comparisons between design implementations.

5. **Describe the main mechanisms for the interfacing**

- I/O Addressing – A MCU communicates using some of its pins.
    - Port-Based (parallel): Processor has one or more N-Bit ports, and its software reads and writes a port just like a register
    - BUS-Based (serial): Processor has address, data and control ports that form a single bus, the communication protocol (like I2C or CANBUS, is built into the processor, a single instruction carries out the read or write protocol on the bus. Can be either standard I/O or memory mapped I/O
        - Memory-mapped I/O: Peripheral register occupy addresses in same address space as memory
        - Standard I/O: Additional PIN M/IO indicates whether a memory or peripheral access.
- Interrupts: The processer polls an interrupt pin. If any peripheral has data to send, the Int will be read, and an Interrupt Service Routine will run, after suspending the current program. Useful to avoid polling all the peripherals.7
    - Fixed Interrupt: The address of the ISR is fixed and built into the microprocessor.
    - Vectored Interrupt: The peripheral must provide the address.
    - Interrupt address table: A compromise between fixed and vectored interrupts, using an interrupt pin and one interrupt acknowledge pin and a table in memory holding all the ISR address. The peripheral provides the index of the table instead of the ISR address.
- Direct Memory Access (DMA): Data is sent directly to the memory using a DMA controller which is independent of the CPU.

6. **Main characteristics of the protocols used to interface peripherals in microcontroller-based systems**
7. **Comparative analysis of the main characteristics, role and application fields of the protocols I2C and SPI**
8. **Explain the possible use of WCET (Worst Case Execution Time) analysis and the difference w.r.t. pWCET (probabilistic WCET)**
9. **Describe the main differences between scheduling and real-time scheduling of tasks, with a description and comparison between EDF and RMS scheduling**
10. **Describe the main source of unpredictability in the execution of code and their possible solution (or at least mitigation)**

11. Describe the different phases and accuracy during the different steps in the verification of an embedded system

- Requirement Verification: Low-level conceptual accuracy.
- Modelling and Simulation: High accuracy if models are representative of the real system.
- Code-Level Verification: High accuracy for logical and functional errors.
- Hardware-in-the-Loop Testing: Very high accuracy in terms of hardware and software integration.
- System-Level Verification: High accuracy for overall functionality and real-world performance.
- Field Testing: Very high accuracy for real-world behaviour but may expose unforeseen issues.
- Regression and Maintenance Testing: High accuracy for ensuring that changes don't break functionality.

12. **Which are the main differences between an ideal and a real sensor? Describe the main characteristics of a MEM sensor and its potential benefits over conventional technologies.**

- On an ideal sensor typical we model it as device which generates an electric signal proportional to a physical measurement, being that acceleration, distance, pressure whatever. A real sensor will never be only proportional, it might have offsets, it might be non-linear, and the proportionality only applies for a certain interval of its working range, thus a good conditioning chain is needed to make sure that those non-idealities are accounted for.
- Regarding MEMS, they are micro, usually done by either being bulk or surface micromachining, so they are easily integrated with already existing technologies and product chains. They are electro and mechanical, so they are transductors or actuators that transform one of these signals into the other. They present several potential benefits since their small size allow for the mass production on such devices, lowering their cost. Besides it allows for a better integration, since the smaller they are the more sensors its possible to pack in a system allowing for more reliability and diverse measurements from the same system.

13. Describe the main aspects to be taken into account to create a toolchain to analyse power and thermal aspects of an embedded multiprocessor.

- Such a toolchain will need to have two main parts which work together, the power analysis and the thermal analysis. It should take into account both average and peak power as both are relevant for energy and thermal optimization and should analyse how the temperature affects the power consumption and vice-versa.
    - Power: To analyse the power, a robust tool will survey all the power saving opportunities, from the System level to the physical, passing through the behavioural, RTL and Logic levels. Such a tool will give more relevance to where more savings can be achieved, namely at the system level guaranteeing that both HW and SW are optimized. Will also have to look at how the power states are used.
    - Thermal: Regarding the thermal part, the tool should consider the power analysis done, relate it with the physical design of the system and estimate the temperature

14. Describe the main steps in the verification of Hw-Sw embedded systems.

- Requirement Verification: The designers should make sure that both the HW and SW requirements are well defined and guarantee that their interaction is has expected
- Modelling and Simulation: Create high level models of both the HW and the SW and verify if they are according to the defined requirements. Again, guarantee that the combined hardware-software model also accurately models their expected interaction.
- Hardware Verification:
    - RTL Testing and simulation.
    - Hardware emulation, like post synthesis simulation.
    - Post fabrication testing, verifying if the developed hardware was produced correctly. Can be done using scan chains.
- Code-Level (SW) Verification:
    - Test individual units of code.
    - Test the integration of those units.
    - Eventually test if new software doesn't break old one.
    - Test if the SW meets the RT constraints.
    - Test other requirements, such as memory and resource usage.
- Co-Verification and HW-SW Integration Testing: To verify if the complete system works as intended, for instances at critical boundaries such as peripheral I/O and DMA controllers and that both meet the timing requirements.
- System level Validation: To test the complete system under real world operational conditions.
- Performance testing and optimization: If needed further testing can be done with the intent to optimize the system.
- Regression and Maintenance Testing: At any update of either HW or SW testing should be done to guarantee that no functionality was damaged by the update.

15. Role and management of timers and watchdog in embedded systems based on microcontrollers

- A timer is a specialized type of clock to measure time intervals. Has two main operating modes, either as a Counter or a Timer. A timer in counter mode will store the number of times a particular event or process occurred in each time interval with respect to a clock signal. Can be used to improve performance, reduce power or simplify the design by replacing repetitive or looping CPU operation with a simple timer or counter interrupt, can be used to generate PWM to provide which can be used for instance in the control of motor.
- The watchdog is a periodic timer that's used to monitor the application execution evolution. Configured once at booth to trigger an interrupt or reset at a fixed time interval, thus it must be cleared before it expires to prevent triggering. It's installed in a system to detect any software anomalies
- Both are peripherals with stringent restrictions on the clock that they are provided with since as noted both handle some sort of time keeping. Both present several working modes Timer – (Timer or Counter) and Watchdog – (windowed or non-windowed) and these modes can be configured by setting their registers appropriately

16. Describe the mechanisms and requirements to implement over the air firmware update (FOTA), focusing also on the benefits, costs and associated risks.

- FOTA allows to partially or fully update the firmware on a device wirelessly. Thus, it allows to address bugs and security vulnerabilities, allows for a shorter time to maker since the products can be shipped faster and its possible to delay low priority features and roll them out later.
- To implement FOTA the device needs wireless connection and there needs to be a firmware server. When a new update is rolled out, the firmware images are encrypted and sent to the firmware update server, from there the end-device queries the firmware update server and fetches new firmware image, afterwards on the device itself the package will be decrypted, validated an applied.
- This process has 3 major challenges
    - Memory: The update needs to be stored after the downloading and there should be always a fallback, in case there's a need to rollback some update.
    - Communication: Theres need for the wireless connection and the software will be sent in discrete packets, thus the packetizing, the packet structure and the protocol must all be accounted for in the SW design
    - Security: The software should be well encrypted and secured. So, there should be authentication, confidentiality and integrity should be guaranteed while sending the file.
- Regarding the cost, it is cheaper, since a total product recall is way more costly than implementing the FOTA. However, this is not an excuse to produce not accurately verified systems.

17. Describe the main sources of power consumption of a hardware platform and the main solution to mitigate such problem.

- The primary source of dynamic power consumption is the switching power, so the power required to charge and discharge the output capacitance on a gate. The main solution to reduce these losses is to reduce the supply voltage, since the relation between the dissipated power and the voltage is quadratic. Its possible to do so to different logic blocks, according to performance requirements.
- Pdyn = Nb * C * f * V^2
    - Pdyn = Dynamic Power
    - Nb = Number of switching bits
    - C = Capacitance
    - f = Switching frequency
    - V = Supply voltage

18. What can be done, at the software and operating system levels, to face with the problem of power consumption

- At the OS level one can set operating points (ie. The pair of optimal supply voltage and clock frequency for a system/block given usage). Doing so can be done by using ACPI (Advanced Configuration Power Interface) a low-level software that moves the power management under the control of the operating system and allows to set the power sates for:
    - Global system states
    - System-wide sleep states
    - CPU level power/sleep/performance states
    - Device level power/sleep states
- Some of these states can be implemented by using DVFS: Dynamic Voltage/Frequency Scaling
- Additionally, it's also possible to implemented power aware algorithms which allow for lower power consumption.

19. Which are the goals of the ACPI standardization and the main status that can characterize the different part of a computing platform.

- The goal of the ACPI (Advanced Configuration and Power Interface) standardization was to present an open-source platform-independent standard that allows for power management, auto configuration (plug and play functionality) and status monitoring.
- The main status that can characterize the different parts of a computing platform are
    - Global system States
    - System-wide sleep States
    - CPU level power/sleep/performance States
    - Device level power/sleep States