**LINUX EXERCISE (LAB 03)**

**How to submit your assignment**

Check the requirements of the lab 01.

Exercise 01

1. Review the types of memory used in embedded systems, state their advantages and disadvantages, and provide one practical application for each type of memory.

ROM (Read-Only Memory): This type of memory is non-volatile and is used to store the firmware or the code that runs the system. The contents of ROM cannot be modified once it has been programmed.

RAM (Random Access Memory): This type of memory is volatile and is used to store temporary data and program code that is being executed by the system. The contents of RAM are lost when the power is turned off.

Flash Memory: This is a type of non-volatile memory that can be programmed and erased electrically. It is commonly used to store the operating system, application code, and user data.

EEPROM (Electrically Erasable Programmable Read-Only Memory): This type of memory is non-volatile and can be electrically programmed and erased. It is commonly used to store small amounts of data that need to be retained even when power is removed.

PROM is a type of non-volatile memory that is programmed at the factory. Once programmed, the contents of PROM cannot be changed. PROM is useful for storing data or code that needs to be retained even when the power is turned off, but does not need to be updated.

EFROM is a type of non-volatile memory that can be programmed and erased electrically. Unlike EEPROM, EFROM can be programmed only once and cannot be reprogrammed. It is commonly used in applications where the data or code needs to be retained even when the power is turned off, but does not need to be updated.

Compare:

1. ROM:

Advantages:

* Non-volatile memory, which means the contents are retained even when the power is turned off.
* Reliable, because the contents cannot be changed or corrupted.
* Low power consumption.

Disadvantages:

* Once programmed, the contents cannot be changed, which limits the flexibility of the system.
* Can be more expensive than other types of memory.

1. RAM:

Advantages:

* Volatile memory, which means the contents can be changed or updated easily.
* Fast access times, making it useful for storing frequently accessed data.
* Low cost compared to other types of memory.

Disadvantages:

* Contents are lost when the power is turned off.
* Can be more susceptible to corruption or data loss.

1. EFROM:

Advantages:

* Non-volatile memory, which means the contents are retained even when the power is turned off.
* Can be programmed electrically, which makes it easier to use than PROM.
* Low power consumption.

Disadvantages:

* Can be more expensive than other types of memory.
* Can be programmed only once, which limits the flexibility of the system.

1. PROM:

Advantages:

* Non-volatile memory, which means the contents are retained even when the power is turned off.
* Reliable, because the contents cannot be changed or corrupted.

Disadvantages:

* Once programmed, the contents cannot be changed, which limits the flexibility of the system.
* Can be more expensive than other types of memory.

1. EFPROM:

Advantages:

* Non-volatile memory, which means the contents are retained even when the power is turned off.
* Can be programmed and erased electrically, which makes it more flexible than PROM.
* Low power consumption.

Disadvantages:

* Can be more expensive than other types of memory.
* Limited number of write and erase cycles.

1. Flash Memory:

Advantages:

* Non-volatile memory, which means the contents are retained even when the power is turned off.
* Can be reprogrammed, which makes it more flexible than other types of non-volatile memory.
* High storage density and low cost compared to other types of non-volatile memory.

Disadvantages:

* Limited number of write and erase cycles.
* Slower access times than RAM.

Example:

ROM:

* Mask ROM: a type of ROM that is programmed at the factory during the chip fabrication process.

RAM:

* SRAM (Static Random Access Memory): a type of RAM that stores each bit using a flip-flop circuit, which provides fast access times but requires more power than other types of RAM.

EFROM:

* One-Time Programmable (OTP) Memory: a type of EFROM that can be programmed electrically only once.

PROM:

* EPROM (Erasable Programmable Read-Only Memory): a type of PROM that can be erased using UV light and reprogrammed multiple times.

EFPROM:

* EEPROM (Electrically Erasable Programmable Read-Only Memory): a type of EFPROM that can be erased and reprogrammed electrically.

Flash Memory:

* NOR Flash: a type of flash memory that provides fast random access times and is commonly used for code storage.

1. Compare microprocessors and microcontrollers and provide a few examples of their use in embedded systems.

Compare:

* A microprocessor is a central processing unit (CPU) that is designed to perform a range of computing tasks. It requires external components, such as memory, input/output interfaces, and other peripherals, to be connected to it in order to function as a complete system. Microprocessors are often used in applications where a high degree of flexibility is required, and the ability to perform a wide range of computational tasks is necessary.
* A microcontroller, on the other hand, is a complete computer system on a single chip. It typically includes a CPU, memory, input/output interfaces, and other peripherals, all integrated into a single chip. Microcontrollers are often used in applications where a high degree of control is required, and the ability to interface with external components and sensors is necessary.

Example:

1. Microprocessors:

Intel Core i5: used in desktop and laptop computers, where high performance and flexibility are required.

1. Microcontrollers:

Arduino: used in hobbyist and educational projects, such as robotics, home automation, and IoT devices.

1. Find some examples of input/output methods for embedded systems.

Example 1: Programmed I/O

An embedded system that includes a temperature sensor connected to an I/O port. The CPU needs to read the temperature data from the sensor and store it in memory. In this case the CPU is directly controlling the transfer of data between the I/O device (temperature sensor) and the memory. It starts by sending commands to the I/O to initiate the data transfer and the I/O port signals the CPU when the data is ready to be read. The CPU then reads the data and acknowledges the transfer to the I/O port.

Example 2: Interrupt-driven I/O

An embedded system with a serial communication port (UART) connected to a GPS module. The GPS module is continuously sending data to the UART, and the CPU needs to read and process the data as it arrives. In this case, the CPU is not actively controlling the transfer of data between the GPS module and memory. Instead, the UART is continuously receiving data and signaling the CPU with an interrupt when new data is available. When the CPU receives the interrupt, it reads and processes the data, then resumes its previous task.

1. Research a real-world embedded system and list its components.

Example: digital camera

1. Microcontroller or microprocessor: The brain of the camera that controls all the functions and processes the data from other components. Examples include the ARM Cortex processors and Atmel AVR microcontrollers.
2. Image sensor: Captures images in the form of electrical signals. Common types of image sensors used in cameras are CCD (charge-coupled device) and CMOS (complementary metal-oxide-semiconductor).
3. Flash memory: Stores the captured images and other data, such as camera settings and firmware. Flash memory is non-volatile, which means it retains data even when the camera is turned off. Examples of flash memory used in cameras include SD (secure digital) cards and CompactFlash cards.
4. Input/Output ports: Allow the camera to connect to other devices, such as a computer or printer. Common I/O ports include USB (Universal Serial Bus) and HDMI (High-Definition Multimedia Interface).
5. Image processing software: The software that runs on the microcontroller to process the image data captured by the image sensor. This software may include algorithms for noise reduction, color correction, and image compression.

It typically uses programmed I/O and interrupt-driven I/O methods to transfer data between the camera's components and memory.

Programmed I/O is used when the CPU directly controls the transfer of data between the camera's components and memory. For example, when the user presses the camera's shutter button, the CPU executes code that controls the camera's lens and image sensor to capture an image. The CPU then transfers the image data from the image sensor to the camera's memory using programmed I/O.

Interrupt-driven I/O is used when the CPU needs to respond to events from the camera's components. For example, when the image sensor has captured a full image, it sends an interrupt request (IRQ) to the CPU to indicate that new data is available. The CPU then executes an interrupt service routine (ISR) to transfer the image data from the image sensor to the camera's memory