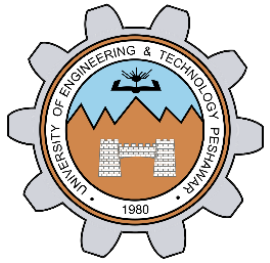
LAB 1:

INTRODUCTION TO MSP43O MCU AND INSTALLATIONS OF HARDWARE AND SOFRTWARE TOOLS:

Engr. Shahzada Fahim Jan

No Cell Phones During Lecture



MOTOROLA SOLUTIONS

Embedded Software Engineer

Motorola Solutions 3.2 37 reviews · View all jobs

Bibra Lake, Perth WA

Engineering - Software (Information & Communication Technology)

Full time

Embedded Systems Engineer

TheDriveGroup View all jobs

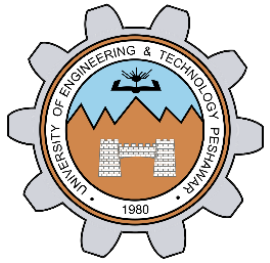
Sydney NSW

Electrical/Electronic Engineering (Engineering)

Full time

$130k to $160k + super

Requirements … ?????



MOTOROLA SOLUTIONS

Embedded Software Engineer

Motorola Solutions 3.2 37 reviews · View all jobs

Bibra Lake, Perth WA

Engineering - Software (Information & Communication Technology)

Full time

Embedded Systems Engineer

TheDriveGroup View all jobs

Sydney NSW

Electrical/Electronic Engineering (Engineering)

Full time

$130k to $160k + super

- Strong programming skills in C/C++, and familiar with development tools (e.g. Keil).
- Hands-on experience with serial communication protocols (UART, SPI, I2C) and hardware debugging tools.
- Familiar with circuit design tools like Altium Designer, or similar software.
- Strong problem-solving abilities.

PREREQUISITE SUBJECTS:

- **CIRCUIT AND SYSTEMS**
  - You must have knowledge of resistor, capacitor and inductor
  - Operational amplifier
  - Working with oscilloscope is Compulsory
- **DIGITAL LOGIC DESIGN**
- **COMPUTER ORGANIZATION AND ARCHITECTURE**
- **C PROGRAMMING**

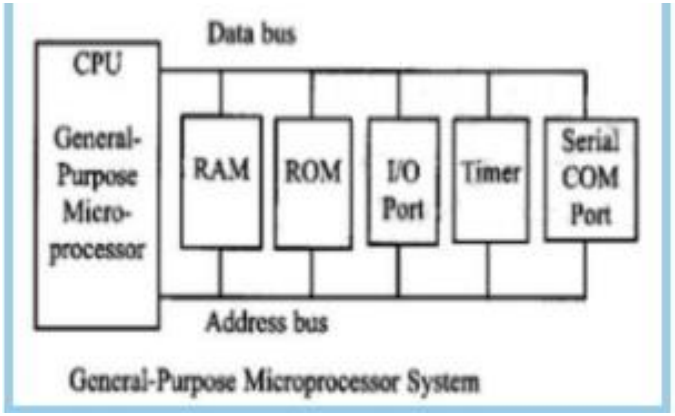
Quiz No: 1

- **Write Your Name, Reg No and Section**

Bin	Dec	Hex
		0xAA
		0x55
	10	
	16	
1111 0000		

Microcontroller and Microprocessor.

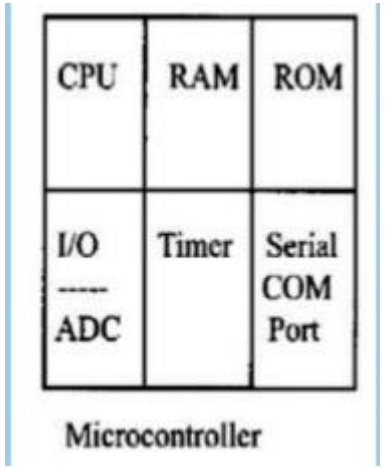
- Microprocessor contains no ROM, no RAM, and no I/O Ports on the chip itself.
- Commonly referred to as general-purpose microprocessors
- We can add external memory, I/O or timer to it



General-Purpose Microprocessor System

Microcontroller and Microprocessor.

- On chip RAM, ROM, I/O ports and timer
- Cannot add external memory, I/O or timer.
- Ideal for many applications in which costs And space are critical.
- Less power consumption



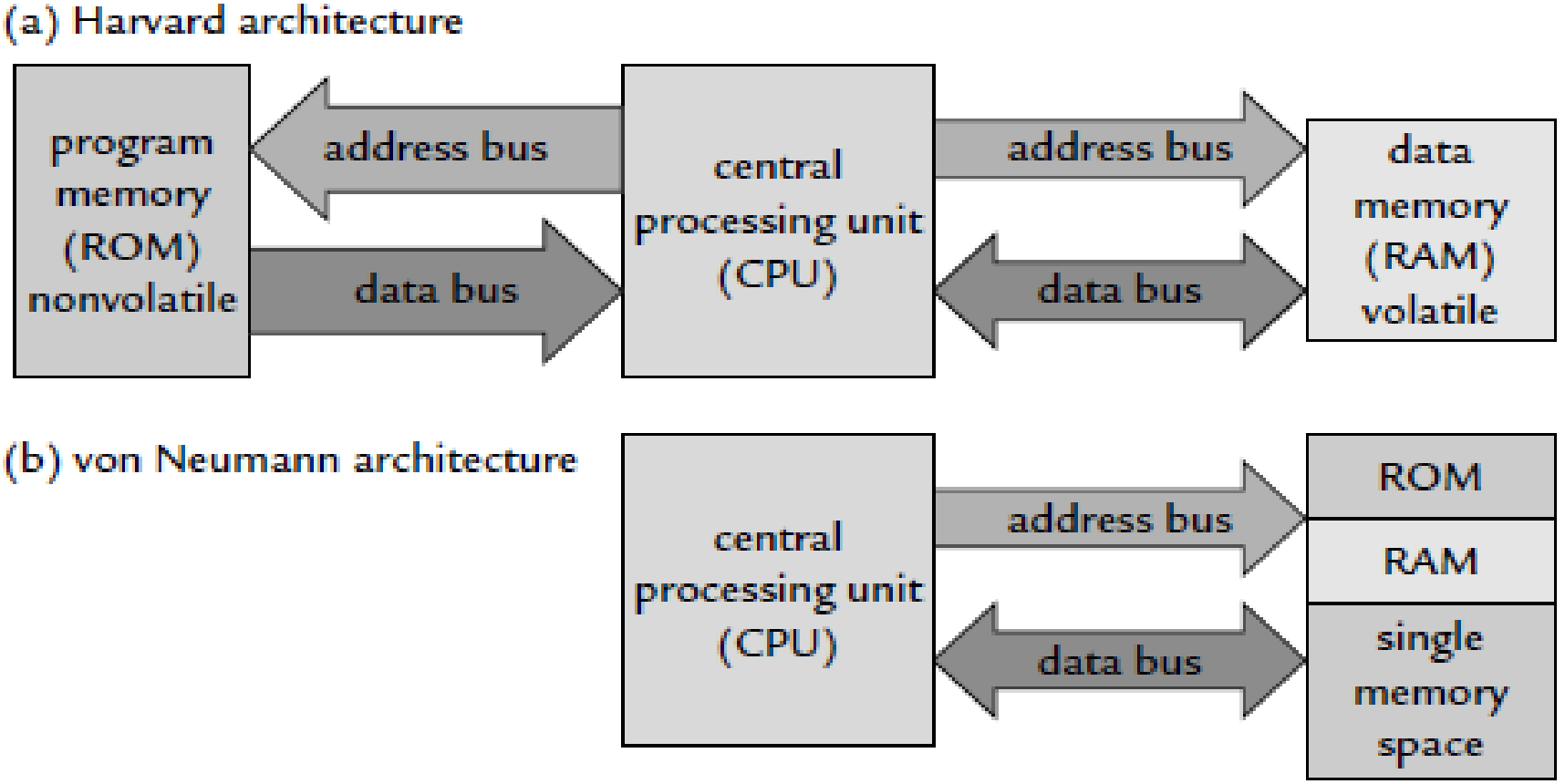
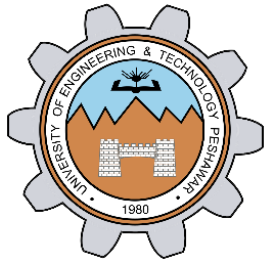


Figure 1.3: Harvard and von Neumann architectures for memory.



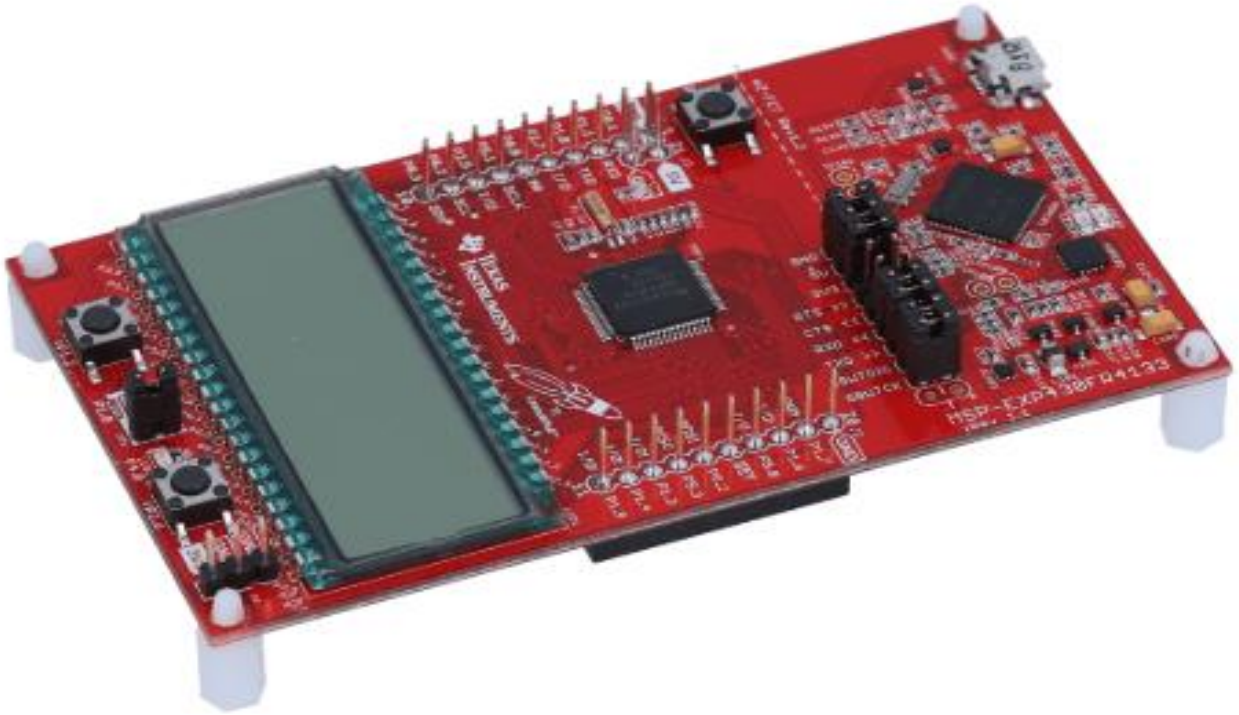
Harvard Architecture

- The volatile (data) and nonvolatile (program) memories are treated as separate systems.
- each with its own address and data bus.
- It allows simultaneous access to the program and data memories

von Neumann Architecture

- There is only a single memory system in the von Neumann or Princeton architecture.
- This means that only one set of addresses covers both the volatile and nonvolatile memories.
- The architecture is intrinsically less efficient because several memory cycles may be needed to extract a full instruction from memory.
- However, the system is simpler and there is no difference between access to constant and variable.

MSP430 MICROCONTROLLER



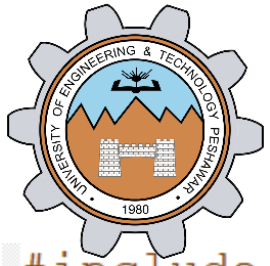
MSP430FR4133

MSP430 MICROCONTROLLER

- The MSP430 was introduced in the late 1990s.
- It is a particularly straightforward 16-bit processor with a von Neumann architecture, designed for low-power applications. The CPU is often described as a reduced instruction set computer (RISC).
- Both the address and data buses are 16 bits wide.
- The registers in the CPU are also all 16 bits wide and can be used interchangeably for either data or addresses.
- it can address only 2^16 = 64KB of memory.
- The MSP430 has 16 registers in its CPU, which enhances efficiency because they can be used for local variables, parameters passed to subroutines, and either addresses or data.

MSP430 MICROCONTROLLER

- Several features make the MSP430 suitable for low-power and portable applications:
- The CPU is small and efficient, with a large number of registers.
- It is extremely easy to put the device into a low-power mode.
- The mode is controlled by bits in the status register.
- The MSP430 is awakened by an interrupt and returns automatically to its low-power mode after handling the interrupt.



PROGRAMMING MSP430FR4133

```
#include <msp430fr4133.h>
int main(void) {
    WDTCTL = WDTPW | WDTHOLD;    // Stop watchdog timer
    PM5CTL0 &= ~LOCKLPM5;// Disable the GPIO power-on default high-impedance mode
                         // to activate previously configured port settings
    P1DIR |= 0x01;               // Set P1.0 to output direction

    for(;;) {
        volatile unsigned int i;   // volatile to prevent optimization

        P1OUT ^= 0x01;             // Toggle P1.0 using exclusive-OR

        i = 100000;                // SW Delay
        do i--;
        while(i != 0);
    }

    return 0;
}
```

TASKS:

TASK1:

1. Write C program for Msp430 which toggle P1.0 or any other Pin of Msp430 MCU.

TASK2:

2. Write C program for Msp430 which toggle P4.0 for msp430fr4133 or any other Pin of Msp430 MCU.

TASK 3

3. Write C program for Msp430 which toggle P5.1 for msp430fr4133 or any other Pin of Msp430 MCU.

TASK 4

4. Write C program for Msp430 which toggle P1.1 for msp430fr4133.