**Embedded System** - Any system which contains a computer as a part of the overall system.

**Microprocessor** - has the entire CPU on one chip. Memory and I/O are on separate chips. Memory is generally RAM

**Microcontroller** - has CPU + Some memory + I/O on the same chip. (May have extra memory on a separate chip). Microcontrollers generally possess different types of memories like RAM , ROM, EPROM , EEPROM , FLASH PROM etc

**Assembly Language Pros:** - Close control over hardware - Optimized programs with minimum memory and execution time

**Cons:** - Some knowledge of H/W required - Language is machine specific, detailed and not easy to learn - Code not portable to other microcontrollers - Difficult to code complex scientific calculations

**High-Level Pros**: - Does not require much H/W knowledge - Easier to learn - Code is not machine specific and thus portable to other microcontrollers - Can handle complex scientific calculations with greater ease

**Cons**: - Code not optimized – occupies greater memory space and takes longer to execute

**Memory Types:** RAM (Random Access Memory) ROM (Read Only memory) PROM (Programmable ROM) EPROM (Erasable PROM) EEPROM (Electrically Ersable PROM) FLASHPROM

**RAM**: Read / Write. Contents lost if power removed. w/battery back-up is **Non-volatile RAM** **(NVRAM)**

**SRAM** **(Static RAM)** does not need any refresh, is faster and more expensive

**DRAM (Dynamic RAM)** needs constant refresh without which contents will be lost, is slower and less expensive

**ROM** - contents are programmed in the factory and cannot be changed later - Contents safe from corruption except under ionizing radiation.

**PROM** - programmed by purchase once read many times. **EPROM** - can be erassed. **EEPROM** – erased by applying voltage

**FLASHPROM** – errase sectors of memory

**Internal Main Oscillator (IMO)** which is at 24 MHz is also known as the **CPU CLK.**

**Pulse Width Modulation** is a technique in which digital waveforms of prescribed period and duty cycle are generated.

**GPIO** (**General Purpose Input Output)** – pins on the board.

**Interrupts -** stops main code and runs a different code. **Posted** – processor checks bit to see if interrupt active. Must be cleared. **Pending** - can wake processor. Must be cleared. **Active** - if Global Interrupt Enable (GIE) bit is set. Active interrupt goes off periodically.

**Internal Registers: Accumulator (A) - Index(X) - Program Counter (PC) - Stack Pointer (SP) - Flags (F).** All **8 bits.** PC is **16 bits.**

**Address Spaces: ROM (4x64 bytes) – RAM (2kB ) – Registers (2x256 bytes).**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Argument** | **OPCode** | **Cycle** |
| **JMP** | **Expr** | **0x8 FFF** | **5** |
| **JZ** | **Expr** | **0xA FFF** | **5** |
| **JNZ** | **Expr** | **0xB FFF** | **5** |
| **JC** | **Expr** | **0xC FFF** | **5** |
| **JNC** | **Expr** | **0xD FFF** | **5** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Argument** | **OPCode** | **Cycle** |
| **TST** | **[Expr],expr** | **0x47** | **9** |
| **TST** | **X+expr, expr** | **0x48** | **9** |
| **TST** | **Reg[expr],expr** | **0x49** | **9** |
| **TST** | **reg[x+expr],expr** | **0x4A** | **10** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Argument** | **OPCode** | **Cycle** |
| **INC** | **A** | **0x74** | **4** |
| **INC** | **X** | **0x75** | **4** |
| **INC** | **[expr]** | **0x76** | **7** |
| **INC** | **[x+expr]** | **0x77** | **8** |

**Include "memory.inc"**

**export bShadow**

**area MyArea(RAM)**

**bShadow: BLK 1**

**area text(ROM,REL)**

**export \_main**

**loop:**

**.terminate: jmp .terminate**

**#include <m8c.h>**

**#include "PSoCAPI.h" // PSoC API definitions for all User Modules**

**BYTE bShadow;**

**void main()**