**1.Introduction to MSP430**

The MSP430 (Mixed Signal Processor ) was a board family of LowPower 16–bit microcontroller from Texas Instruments.

The MSP430 (Mixed Signal Processor ) was introduced in the late 1990s, although its ancestry goes back to the 4-bit TSS400.

It is a particularly straight forward 16-bit processor with a von Neumann architecture, designed for low-power applications.

Ultra low power optimization extends battery life and multiple low power modes of operation

The CPU is often described as a reduced instruction set computer (RISC) “Reflections on the CPU and Instruction Set.

Currently at least 40 flavors available

The family of devices ranges from

1k ROM – 60k ROM 128 bytes RAM – 2K RAM Price : $ - $10

The device split into 3 families :

MSP430x3xx is a basic unit

MSP430x1xx is more feature rich family

MSP430x4xx is similar to 1xx with built in LCD driver

**Part Numbering Convention**

Part numbers for MSP430 devices are determined based on their capabilities.

All device part numbers follow the following template:

**MSP430MtFaFbMc**

*Example:*

*MSP430F435*

*F- Flash memory device*

*43- LCD* controller, a hardware UART,

5- as 16 kb of code memory, and 512 bytes of RAM.

2.Where Does MSP430 Fit

The MSP430 is the simplest microcontroller in TI’s current portfolio. Its more powerful siblings include the TMS470, which is based on the 32/16-bit ARM7, and the C2000, which incorporates a digital signal processor.

It supports 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.

This makes the MSP430 simpler than an 8-bit processor with 16-bit addresses.

It can address 216 = 64KB of memory.

MSP430X has extended register and wider address bus that handle up to 1MB of memory

In many ways, the MSP430 fits between traditional 8- bit and 16-bit processors.

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.

Its more powerful siblings include the TMS470, which is based on the 32/16-bit ARM7, and the C2000, which incorporates a digital signal processor

2.1.Features make the MSP430 suitable for low-power :

The CPU is small and efficient, with a large number of registers.

Easy to put the device into a low-power mode: No special instruction is needed: 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.

There are several low-power modes, depending on how much of the device should remain active and how quickly it should return to full-speed operation.

There is a wide choice of clocks: low-frequency watch crystal runs continuously at 32 KHz and is used to wake the device periodically.

The CPU is clocked by an internal, Digitally Controlled oscillator (DCO), which restarts in less than 1s in the latest devices.

Therefore the MSP430 can wake from a standby mode rapidly, perform its tasks, and return to a low-power mode.

A wide range of peripherals is available, many of which can run autonomously without the CPU for most of the time.

Many portable devices include liquid crystal displays, which the MSP430 can drive directly.

Some MSP430 devices are classed as application-specific standard products (ASSPs) and contain specialized analog hardware for various types of measurement.

Currently 4 – Families of MSP430 available letter after MSP430 shows type of memory

F- FLASH

C – ROM

For ASSPs type of measurements intended

E – Electricity

W- for Water

G – for Signal that require gain provided by operational Amplifier

**a ) MSP430x1xx:**

* + Provides a wide range of general-purpose devices from simple versions to complete systems for processing signals.
  + Peripherals include a hardware multiplier, which can be used as a rudimentary digital signal processor.
  + Packages have 20–64 pins.

**MSP430F2xx:**

A newer, general-purpose family introduced in 2005.

Its CPU can run at 16 MHz, double the speed of earlier devices, while consuming only half the current at the same speed.

Some come in 14-pin packages, including a traditional plastic dual-in-line (PDIP)

Package is – 14pin requires low frequency clock

Pull-up or pull-down resistors are provided on the inputs to reduce the number of external components needed. There are many options for analog inputs. Even the smallest, 14-pin devices offer a 16-bit sigma–delta ADC.

**MSP430x3xx:** The original family, which includes drivers for LCDs.

**MSP430x4xx:** Can drive LCDs with up to 160 segments. Many of them are ASSPs, but there are general-purpose devices as well.

Their packages have 48–113 pins, many of which are needed for the LCD.

**MSP430X:** The original MSP430 architecture, extended to give the MSP430X in 2006, mainly so that it can address extra memory but with other improvements as well.

The devices are included in the MSP430F2xx and MSP430F4xx families with nothing in their part number to distinguish them.

The CPU is a MSP430x if there is more than 64KB of memory.

***3. Architecture: CPU and Memory***

The MSP430 utilizes a 16-bit RISC architecture, which is capable of processing instructions on either bytes or words.

The ’430 family. It consists of a 3-stage instruction pipeline, instruction decoding, a 16-bit ALU, four dedicated-use registers, and twelve working (or scratchpad) registers.

The CPU is connected to its memory through two 16-bit busses, one for addressing, and the other for data.

All memory, including RAM, ROM, information memory, special function registers, and peripheral registers are mapped into a single, contiguous address space

**CPU Features**

***The ALU***

The ’430 processor includes a pretty typical ALU (arithmetic logic unit). The ALU handles addition, subtraction, comparison and logical (AND, OR, XOR) operations. ALU operations can affect the overflow, zero, negative, and carry flags.

***Working Registers***

*12 working register of 16-bit ( R14- R15) used for register mode operation****.***

Any variable which is accessed often should reside in one of these locations, for the sake of efficiency.

Any of these registers for any purpose, either data or address.

*R4 and R5 reserve for debug information.*

*R8, R9, R15 used for code extensive operations*

***Constant Generators***

R2 and R3 function as constant generators,

so that register mode may be used instead of immediate mode for some common constants. (R2 is a dual use register. It serves as the Status Register, as well.)

Generated constants include some common single-bit values (0001h, 0002h, 0004h, and 0008h), zero (0000h), and an all 1s field (0FFFFh). Generation is based on the W(S) value in the instruction word, and is described by the table below.

***Program Counter***

The Program Counter is located in R0. Since individual memory location addresses are 8-bit, but all instructions are 16 bit, the PC is constrained to even numbers (i.e. the LSB of the PC is always zero).

One exception to this rule of thumb is the implementation of a switch, where the code jumps to a spot, dependent on a given value. (I.e., if value=0, jump to location0,

if value=1, jump to location1, etc.)

***Status Register***

The Status Register is implemented in R2, and is comprised of various system flags.

The flags are all directly accessible by code, and all but three of them are changed automatically by the processor itself. The 7 most significant bits are undefined. The bits of the SR are:

**The Carry Flag (C)**

Location: SR(0) (the LSB)

Function: Identifies when an operation results in a carry. Can be set or cleared by software, or automatically.

1=Carry occurred

0=No carry occurred

**The Zero Flag (Z)**

Location: SR(1)

Function: Identifies when an operation results in a zero. Can be set or cleared by software, or automatically.

1=Zero result occurred

0=Nonzero result occurred

**The Negative Flag (N)**

Location: SR(2)

Function: Identifies when an operation results in a negative. Can be set or cleared by software, or automatically. This flag reflects the value of the MSB of the operation result (Bit 7 for byte operations, and bit 15 for word operations).

1=Negative result occurred

0=Positive result occurred

**The Global Interrupt Enable (GIE)**

Location: SR(3)

Function: Enables or disables all maskable interrupts. Can be set or cleared by software, or automatically. Interrupts automatically reset this bit, and the retie instruction automatically sets it.

1=Interrupts Enabled

0=Interrupts Disabled

**The CPU off bit (CPUOff)**

Location: SR(4)

Function: Enables or disables the CPU core. Can be cleared by software, and is reset by enabled interrupts. None of the memory, peripherals, or clocks are affected by this bit. This bit is used as a power saving feature.

1=CPU is on

0=CPU is off

**The Oscillator off bit (OSCOff)**

Location: SR(5)

Function: Enables or disables the crystal oscillator circuit. Can be cleared by software, and is reset by enabled external interrupts.

OSCOff shuts down everything, including peripherals. RAM and register contents are preserved. This bit is used as a power saving feature.

1=LFXT1 is on

0=LFXT1 is off

**The System Clock Generator (SCG1,SCG0)**

Location: SR(7),SR(6)

Function: These bits, along with OSCOff and CPUOff define the power mode of the device.

**The Overflow Flag (V)**

Location: SR(8)

Function: Identifies when an operation results in an overflow. Can be set or cleared by software, or automatically. Overflow occurs when two positive numbers are added together, and the result is negative, or when two negative numbers are added together, and the result is positive. The subtraction definition of overflow can be derived from the additive definition.

1=Overflow result occurred

0=No overflow result occurred

***Stack Pointer***

The Stack Pointer is implemented in R1. Like the Program Counter.

The LSB is fixed as a zero value, so the value is always even.

The stack is implemented in RAM, and it is common practice to start the SP at the top (highest valid value) of RAM.

The push command moves the SP down one word in RAM (SP=SP-2), and puts the value to be pushed at the new SP. Pop does the reverse.

Call statements and interrupts push the PC, and return statements pop the value from the TOS (top of stack) back into the PC.

4.Key Features of MSP430

with a set of intelligent peripherals like I/O, Timers ADC, DAC, flexible clock and USCI

low cost

lowest power consumption

Ultra low power optimization extends battery life

multiple low power modes of operation

* Extensive interrupt capability relieves need for polling
* Prioritized nested interrupts
* Seven source-addressing modes
* Four destination-addressing modes
* Only 27 core instructions and

24 Emulated Instructions

* Large register file
* Efficient table processing
* Fast hex-to-decimal conversion

MSP430 requires

0.1 μ A for RAM data Retention,

0.8 μ A for RTC mode operation

250 μA /MIPS for active mode operation.

• Low operation voltage (from 1.8 V to 3.6 V).

• Zero-power Brown-Out -Reset (BOR)

**5.Architecture - MSP430**

