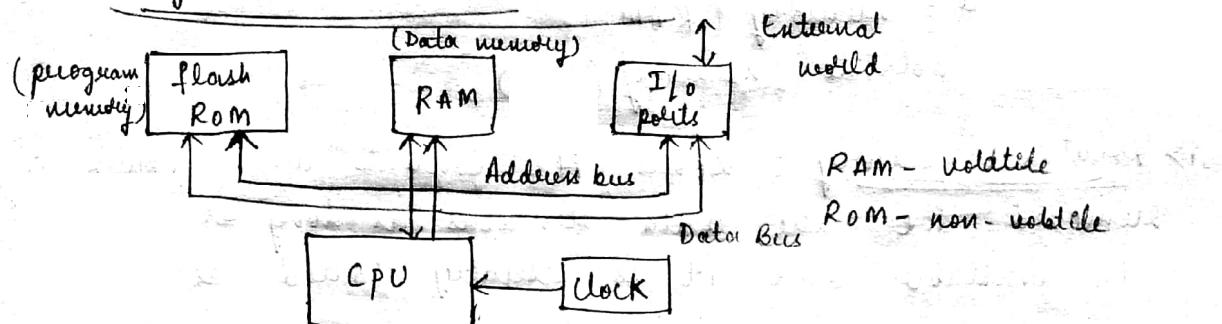


Module 1 : MSP430 ARCHITECTURE

Anatomy of a small microcontroller



Microcontroller is a general purpose programmable controller which consists of

- i) CPU - Central processing unit which includes
 - ALU - to perform arithmetic and logical operations
 - Registers needed for the basic operations of the CPU such as programme Counter (PC), Stack pointer (SP), Status register (SR) and general purpose registers.
 - Flag registers
 - Instruction decoder and other logic to control the CPU to handle reset and interrupt.

- ii) program memory (ROM) → required to store programs and constant data (non volatile)

- iii) Data memory (RAM) → required to store data and it is volatile.

- iv) Input and output port: Needed for digital communication with the external world

- v) Clock : Is needed to keep the entire system synchronized. Clock can be generated internally or from a crystal or an external source

- vi) Address bus and data bus : used to link the subsystems to transfer data and instructions.

The various other peripherals that can be included on to a microcontroller chip are

- vii) timer
- viii) Interrupt Controller
- ix) ADC
- x) DAC

- vii) Real time clock (RTC)
- viii) Communication interfaces

Memory :

Memory can be broadly classified into 2 types

- i) Volatile memory (RAM)
- ii) Non-volatile memory (ROM)

i) Volatile memory : Loses its contents when power is removed i.e. the data remains as long as power is available and it is usually called as RAM

ii) Non-volatile memory : Retains its contents even if power is used hence it is used to store programs and constant data.

ROM : Can be further classified into following types 10/8/17

i) Masked ROM :

The data are encoded into one of the masks used for photolithography and written into the IC during manufacture.

This memory is read only & it is used in the production of stable products because any change to the data requires new mask which is expensive.

ii) EPROM : Electrically programmable read only memory
This memory can be programmed electrically but it can't be erased, but the device must be exposed to UV light to erase its content.

Erasable devices come with the packages which has small quartz window to pass light.

iii) OTP : One time programmable

This is an EPROM in a package without a quartz window which means it can't be erased.

iv) Flash memory :

These devices can be programmed and erased electrically.
The difference between EEPROM (electrically erasable PROM) and flash is that individual bytes of EEPROM can be erased while flash can be erased in blocks.

A higher voltage is used to write into flash memory.

Ex: Memory Cards, USB.

RAM: It is a volatile memory.

RAM can be classified into 2 types namely

i) SRAM — Static RAM

ii) DRAM — Dynamic RAM

i) Static RAM (SRAM):

The contents of SRAM remains constant even without the clock provided power must be available.

ii) Dynamic RAM (DRAM):

The contents of ROM are lost when the clock stops even if power is available.

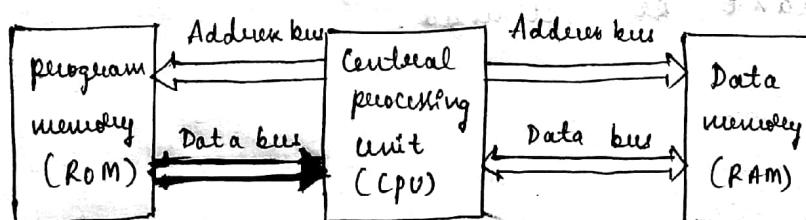
CPU ARCHITECTURES

CPU architectures are classified into 2 types.

i) Harvard architecture

ii) Von-Neumann architecture or Princeton architecture.

i) Harvard architecture:



* Separate program and data memory, each with its own address and data bus is the key feature of Harvard architecture.

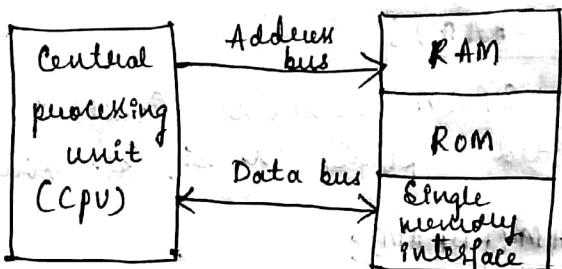
* It allows simultaneous access program and data memory i.e. CPU can read an operand from data memory, at the same time as it reads instruction from the program memory.

This system is faster.

* The 2 systems can be optimized separately.

Example: PIC16 has 8 bit wide data bus and
data memory
9 bit address bus
program memory has 14 bit data bus and
13 bit address bus

Q1) Princeton architecture or Von-Neuman architecture [18/14]



- * A single memory space is used to store both program and data with a single set of address and data bus.
- * Simultaneous access to data and program is not possible.
- * The system is slower.
- * Accidental overwriting of data and program may happen.
- * The system can't be optimized.

Ex: MSP430

Buses:

Bus is a collection of wires used to transfer multiple bit information.

Types of buses

- i) Data bus: It carries the data to be processed
- ii) It is bidirectional.

The size of the data bus indicate the size of the processor controller which in turn indicates the size of the ALU. i.e. the no. of bits the processor can handle

Address bus: It carries the address of the peripheral connected to CPU

The size of the address bus indicates the size of the memory or the addressing capability by the factor 2^n , where n is the no. of address lines

Ex: A processor or a controller which is 8 bit and has 12 address lines. Has a memory capacity of $2^{12} = 4\text{K bytes}$

$$2^5 = 32 \text{ bytes}$$

$$2^{10} = 1024 \text{ Bytes} = 1\text{KB} \quad (1\text{KB} = 1024 \text{ bytes})$$

$$2^{11} = 2048 \text{ Bytes} = 2\text{KB}$$

$$2^{12} = 4096 \text{ bytes} = 4\text{KB}$$

$$2^{13} = 8192 \text{ bytes} = 8\text{KB}$$

$$2^{14} = 16384 \text{ bytes} = 16\text{KB}$$

$$2^{15} = 32768 \text{ bytes} = 32\text{KB}$$

$$2^{16} = 65536 \text{ bytes} = 64\text{KB}$$

$$2^{20} = 1\text{MB}$$

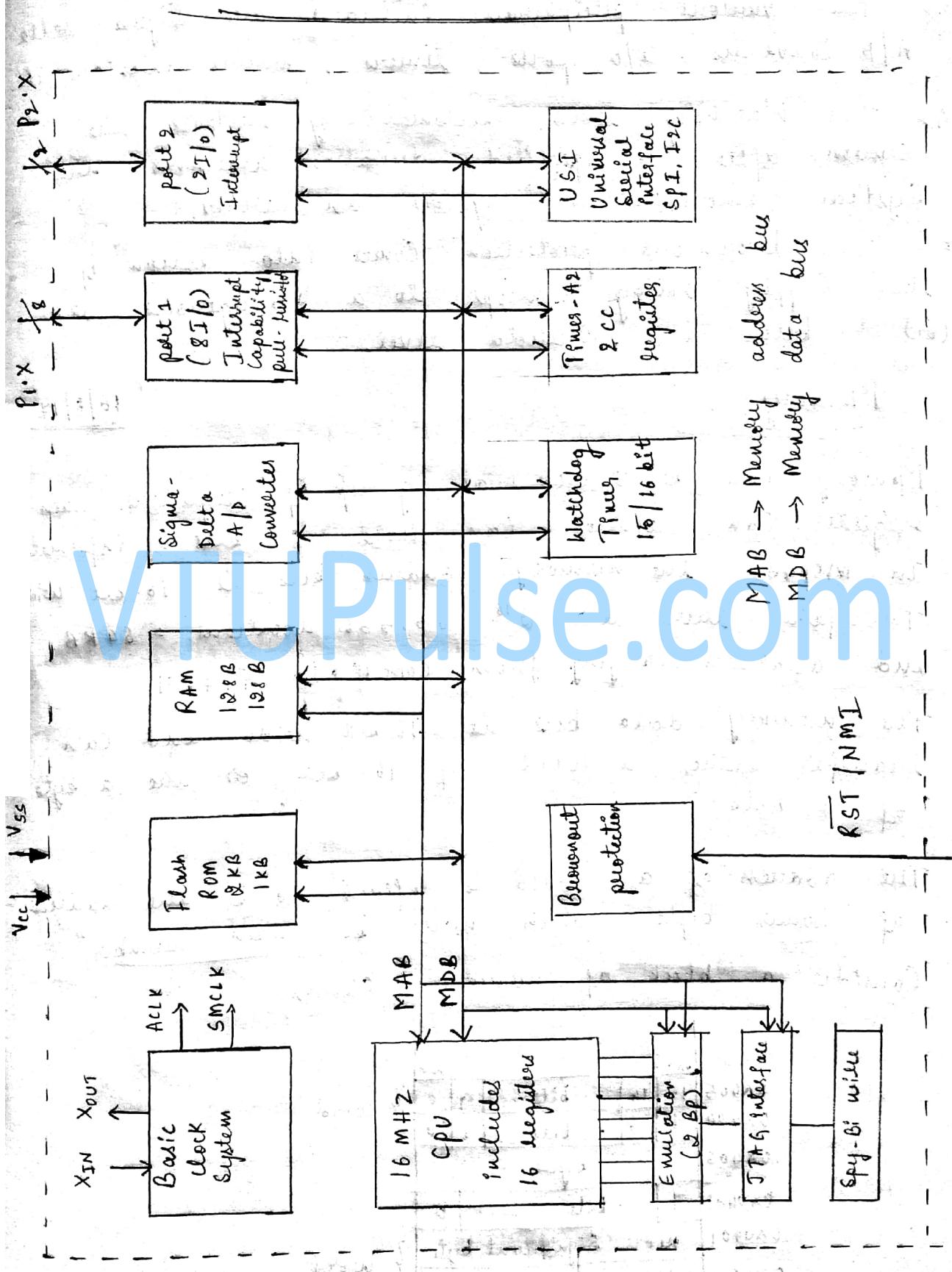
$$2^{30} = 1\text{GB}$$

$$2^{40} = 1\text{TB}$$

- * MSP - mixed Signal processor.
- * It is a 16 bit processor with a Von-Neuman architecture designed for low power applications.
- * It has a RISC (reduced instruction set CPU) Computer) CPU.
- * Both address bus and data bus are 16 bit wide
- * Registers in the CPU are 16-bit wide which can be interchangeably used to store data and addresses.
- * MSP 430X has extended registers and wider address bus that can handle upto 1MB of memory.
- * MSP430 has 16 registers in its CPU which enhances efficiency since they can be used for local variables, parameters passed for subroutines, either to store address or data. which is a typical feature of RISC System.
- * Several features of MSP430 make it suitable for low power and portable applications.
 - i) It is easy to put the device into a low power mode the mode is controlled by bits in the status register. MSP430 is a ~~Wakent~~ wakened by interrupt and switches automatically to its low power mode after handling the interrupt.
 - ii) There is a wide choice of clocks.
 - iii) Many peripherals can even without the CPU for most of the time.
 - iv) Many portable devices include LCD's which the MSP430 can ~~wake~~ directly.
 - v) Some MSP430's are application specific standard products. (ASSP)
- * Four families of MSP430 are available. The letter after MSP430 shows the type of memory F indicates flash memory and C indicates ROM. A second letter for ASSP shows the type of measurements, E indicates electricity

W indicates Wires, G indicates signal which needs gain
 Such as op-amp
 The last 3 digits identify the specific device.

Functional block diagram of Msp430 F2003 and F2013



- 1) The MSP430 has flash memory of 1KB in 14/8/17
MSP430F2003 and 2KB in MSP430F2013 and RAM of
128B in both the versions.
- 2) The operating voltage range is from 1.8V to 3.6V.
- 3) The various peripherals included are Sigma-delta
A/D converters, I/O ports, timers, serial interface etc.
- 4) The watchdog timer automatically reboots the
System after a specified duration whenever the
System encounters a hang (or) an error.
- 5) The blown out protection comes into action if
the supply voltage drops to a dangerous level
(or) below a threshold level.

Memory:

16/8/17

Memory can be viewed as group of registers, each register location can hold 8 bits of information.
In MSP430 the memory address bus is 16-bit wide.
Therefore there are $2^{16} = 65,536$ locations, = 64 KB.
and addresses ranging from 0000H to FFFFH.

The memory data bus is 16 bit wide and can transfer either a word of 16-bits or the a byte of 8 bits.

"The address of a word is defined to be the address of lower byte which must be even number"

Consider a block of memory shown

| m ₁₅ b | m ₁₄ b | ... bits ... | m ₉ b | m ₈ b | |
|-------------------|-----------------------|--------------|------------------|------------------|---|
| 0x405 | 15 | 14 | ... bits ... | 9 | 8 |
| 0x404 | 7 | 6 | ... bits ... | 1 | 0 |
| 0x403 | bytes | | | | |
| 0x0402 | 7 | ... bits ... | | 0 | |
| 0x0401 | more significant byte | | | | |
| 0x0400 | less significant byte | | | | |
| 0x03FF | word | | | | |
| | ↓ | word | | | |
| | word | | | | |
| | ↓ | word | | | |

The two bytes at 0x0400 and 0x0401 is a valid word with address 0x0400 which is an even no and may be fetched in a single cycle of the bus. The word at 0x0401 and 0x0402 can't be treated as a word as its address would be 0x0401 which is odd and invalid.

The MSP430 can't handle such misaligned words.

There are 2 ways in which the 2 bytes (lower and upper) of a word can be stored namely

Little endian and Big endian

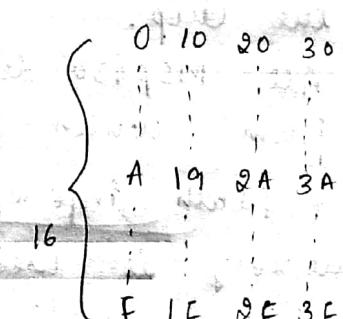
Little endian: In this the lower byte is stored in lower address and higher byte is stored in higher byte. It is used by MSP430.

Big endian: In this the lower byte is stored in higher address and higher byte is stored in lower address.

MEMORY MAP:

The organization of 64 KB memory of F2013 Controller is shown below in the memory map.

| Address | Type of memory |
|---------|---|
| 0xFFFF | Interrupt and test |
| 0xFFC0 | 64B Vector table |
| 0xFFB0 | Flash Code memory |
| 0xFF80 | 8KB (lower boundary varies) |
| 0xFFFF | |
| 0x1100 | |
| 0x10FF | Flash Information |
| 0x1000 | 256B memory |
| 0x0FFF | Bootstrap loader (not in F2xx) |
| 0x0C00 | |
| 0x0BFF | |
| 0x0280 | |
| 0x027F | RAM |
| 0x0200 | 128B (upper boundary varies) |
| 0x01FF | peripheral registers |
| 0x0100 | 256B with word access |
| 0x00FF | peripheral registers with |
| 0x0010 | 240B byte access |
| 0x000F | Special function registers (byte access) |
| 0x0000 | |



256 - FF

128 - FF

1) Special function registers (SFRs):

There are 16 bytes of SFRs used for enabling function of some modules, and enabling and signalling of interrupts from peripherals.

2) peripheral registers with byte access and word access:

These registers are used to provide communication between CPU and peripherals and they are grouped separately as byte and word access to avoid wasting of addresses.

If the words and bytes are mixed then the words must be aligned only at even addresses else it will lead to an error.

3) RAM : This memory is used for Variables. It always starts at address 0x0200 but upper limit may vary depending upon the type of RAM. BC02 F2013 has 128 bytes of RAM.

4) Bootstrap loader: Contains a program used to communicate using standard serial protocol with the Com port of PC. This is used to program the chip.

All MSP430 devices have bootstrap loader except F20XX from which it was omitted to improve security.

5) Flash information memory: A 256 bytes of flash memory is used for storage of non-volatile data which may include serial nos of to identify equipments or an address for a network.

i.e The variable that should be retained even when power is removed.

6) Flash Code memory: Used to store programs including the executable code and constant data. F2013 has 2KB of ROM and F2003 has 1KB of ROM.

1) Special function registers (SFRs) :

There are 16 bytes of SFRs used for enabling function of some modules, and enabling and signalling of interrupts from peripherals.

2) peripheral registers with byte access and word access :

These registers are used to provide communication between CPU and peripherals and they are grouped separately as byte and word access to avoid wasting of addresses.

When If the words and bytes are mixed then the words must be aligned only at even addresses else it will lead to an error.

3) RAM : This memory is used for variables. It always starts at address 0x0200 but upper limit may vary depending upon the type of RAM. B20 F2013 has 128 bytes of RAM.

4) Bootstrap loaders: Contain a program used to communicate using standard serial protocol with the Com port of PC. This is used to program the chip.

All MSP430 devices have bootstrap loaders except F20XX from which it was omitted to improve security.

5) Flash information memory: A 256 bytes of flash memory is used for storage of non-volatile data which may include serial nos of to identify equipments or an address for a network.

i.e The variable that should be retained even when power is removed.

6) Flash Code memory: Used to store programs including the executable code and constant data F2013 has 2KB of ROM and F2003 has 1KB of ROM.

7) Interrupt and Vector table:

Used to handle exceptions when normal operation of the processor is interrupted or when the device is reset.

The range of address can be extended from 64KB to 1MB in MSP430. Thus it requires 8 bit address line

CPU (Central processing unit)

17/8/17

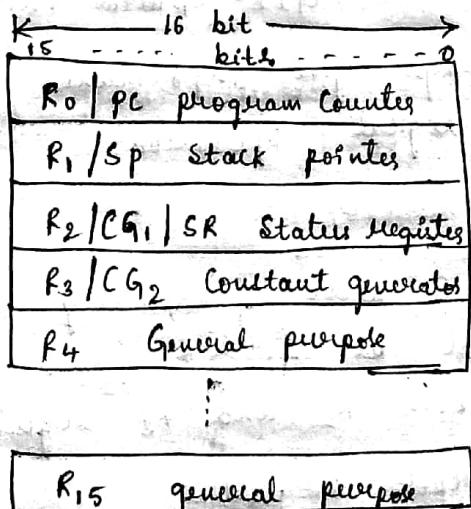
CPU executes the instructions stored in memory.

It includes the ALU which performs Computations (arithmetic and logical operations), memory in terms of registers and an instruction decoder to decode and execute the instructions.

The CPU can run at a maximum clock frequency of $f_{MCLK} = 16 \text{ MHz}$ in MSP430 F20xx family.

The CPU is built using static logic i.e. the CPU will retain its state until it is reset which is essential for low power operation.

MSP430 CPU has a set of 16 registers, R₀ - R₁₅ which are 16 bit wide and can hold a word of 16 bits as shown below.



The first four registers have dedicated functions while the remaining registers R₄ - R₁₅ are general purpose working registers.

1) program Counter : (pc)

It holds the address of next instruction to be fetched and executed.

The instructions ^{size} may be 1-3 words which must be aligned at even addresses only. Therefore lsb (least significant bit) is hard-wired / made zero.

2) Stack pointers : (sp)

Stack pointer points to the top of the stack.

Stack is a temporary storage area where data can be stored or retrieved in last in first out manner. When a subroutine is called, the CPU jumps to the subroutine, executes the subroutine code and then returns to the next instruction after call. The CPU must keep track of return addresses. Such addresses are stored in stack.

3) Status register (SR)

This register contains a set of flags (single bits) shown.

| | 15 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|----|------|------|---------|---------|-----|---|---|---|---|---|
| Reserved | V | SCG1 | SCG0 | OSC OFF | CPU OFF | GIE | N | Z | C | | |

Signed no.s - overflow

GIE - General interrupt enable.

The C, Z, N, V flags give information about the result of arithmetic or logical operation.

Constant generator

18/8/17

This provides the 6 most frequently used values so that they need not be fetched from memory whenever needed. It uses both R₂ and R₃ registers.

General purpose register

There are 12 general purpose registers R₄ - R₁₅. They may be used to store either address or data since both are 16-bit wide.

2) program Counter : (pc)

It holds the address of next instruction to be fetched and executed.

The instructions ^{size} may be 1-3 words which must be aligned at even addresses only. Therefore lsb (least significant bit) is hard-wired / made zero.

3) Stack pointers : (sp)

Stack pointer points to the top of the stack.

Stack is a temporary storage area where data can be stored or retrieved in last in first out manner. When a subroutine is called, the CPU jumps to the subroutine, executes the subroutine code and then returns to the next instruction after call. The CPU must keep track of return addresses. Such addresses are stored in stack.

3) Status register (SR) :

This register contains a set of flags (single bit) as shown.

| 15 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|---|------|------|---------|---------|-----|---|---|---|---|
| Reserved | V | SCG1 | SCG0 | OSC OFF | CPU OFF | GIE | N | Z | C | |

GIE - General interrupt enable.

The C, Z, N, V flags give information about the result of arithmetic or logical operation.

Constant generator

18/8/17

This provides the 6 most frequently used values so that they need not be fetched from memory whenever needed. It uses both R₂ and R₃ registers.

General purpose register

There are 12 general purpose registers R₄ - R₁₅.

They may be used to store either address or data since both are 16-bit wide.

* * Memory mapped input and output

Digital input and output operation takes place through pins. Called as ports. Each port can have upto 8 pins. MSPF2013 has 2 ports ICs namely port 1 and port 2, P1 and P2.

Port 2 has all 8 pins. P1.0 - P1.7 whereas port 2 has only 2 pins. P2.6 and P2.7.

Pins can be configured as either input or output pins, or some may generate interrupts when the voltage on the pin changes.

MSP430 uses memory mapped input and output where in a port appears to the CPU as memory registers. Called peripheral registers.

These registers can be read written and modified in the same way as registers in RAM.

To read the contents of port 1 the register P1IN should be read. To send or write to port 1 we need to write to the P1OUT register.

The three registers for port 1 are P1IN, P1OUT and P1DIR

port 1 input (P1 IN) :

Reading this register returns / gives the value on the pins of port 1. If port 1 is configured as input port. This register is read only and is volatile.

port 1 output (P1 OUT) :

Writing on to this register sends the value to the pins of port 1. If port 1 is configured as output port.

port 1 direction (P1 DIR) :

A bit of zero configures the pin as input which is default. Writing a one changes the pin to act as an output port pin.

A clock is a heart of every Synchronous digital system. The clock signal is a square wave whose edges triggers hardware so that the changes in the different components are synchronized.

To meet the demand for high performance and low power, modern microcontrollers have clocks with more than one source.

2 clocks with different specifications are needed to meet the requirements.

1) A fast clock: It will drive the CPU which can be started and stopped quickly to conserve energy but need not be accurate.

2) A slow clock: It runs continuously to monitor the real time and must consume less power and must be accurate.

Several types of Oscillators are used to generate clock signals namely

1) Crystal oscillator: provide accurate and stable frequency. Crystal for microcontroller can run at high frequency of few MHz. and to derive the main bus or a low frequency of $32,768\text{ Hz}$ (32 kHz) for a real time clock (RTC).

Disadvantages:

- * Crystals are expensive and delicate
- * Crystals draws large current at high frequencies
- * Crystals is an extra component and may need additional capacitors.
- * The crystal oscillators take a long time to start up and stabilize.

2) Resistor and Capacitor oscillators (R-C oscillators)

R-C oscillators provide low accuracy and low stability clock signals.

They are available at low cost

They start up quickly

The components may be external or integrated with the micro controllers.

The MSP430 looks into the conflicting demands of high performance and low power and precise frequency by using 3 internal clocks which can be derived from four sources.

The 3 internal clocks of MSP430 are

- i) Master clock (MCLK) → used by CPU and peripherals
- ii) Sub system master clock (SMCLK) → distributed to peripherals
- iii) Auxiliary clock (ACLK) → distributed to peripherals.

The MCLK and SMCLK both run in the MHz range whereas the auxiliary clock runs at a low frequency derived from a quartz crystal.

In the MSP430X1XX and MSP430F2XX may have

- i) An ACLK generated from a low frequency oscillator crystal typically at 32 kHz.
- ii) Both MCLK and SMCLK are supplied by internal digitally controlled oscillator (DCO) which runs at 0.8 MHz in MSP430X1XX family and 1.1 MHz at MSP430F2XX family.

A DCO starts very rapidly at full speed taking less than 11 sec in MSP430F2XX family which is essential for low power systems.

In the MSP430X4XX family

- i) An ACLK comes from a low frequency oscillator at 32 kHz
- ii) The MCLK and SMCLK are supplied from the DCO

which is controlled by frequency locked loop (PLL) which locks the frequency at 32 times the ACLK which is close to 1 MHz.

Exceptions & Interrupts and Reset

A program is executed in a sequence but there are 2 classes of exceptions to this rule namely.

Interrupts:

Interrupts are generated by hardware (or can be from a software also), which indicate an event has occurred and needs a response to the event.

Whenever an interrupt occurs the process stops current execution, stores information on to stack (contents of PC and SR). for it to resume later and then executes the ISR (Interrupt Service Routine).

On the completion of execution of the ISR it returns to its previous activity.

Interrupts are used to wake up the processor from a low power state.

The MSP430 uses Vector Interrupts where each ISR has ^{its own} address / vector which is stored at pre-defined address in a vector table in the memory address range (0xFFC0 - 0xFFFF). The vector table is at a fixed location but ISR can be located anywhere.

RESET:

Generated by hardware when an unusual activity happens and the normal operation can't continue a reset can be done.

A reset causes the device to restart from a well defined state.

CPU (in detail)

| | |
|----------------------------------|-----------------|
| R ₀ / pc | program counter |
| R ₁ / SP | Stack pointer |
| R ₂ / SR | CG ₁ |
| R ₃ / CG ₂ | |
| R ₄ | General purpose |
| | |

| | |
|-----------------|-----------------|
| R ₁₅ | General purpose |
|-----------------|-----------------|

PC:

pc holds the address of next instruction to be executed. Instruction size may be one to three words which must be aligned at even addresses only.
∴ LSB of pc is hardwired to zero.

The contents of pc are placed on to the address bus and the next instruction is fetched from this address. The value in the pc is automatically incremented by 2 after each fetch. So that it is ready for the next word. Instructions are executed sequentially unless there is an explicit jump.

Subroutines and interrupts also modify the program counter but in these cases the previous value is saved on the stack and restored later.

Stack pointers (SP):

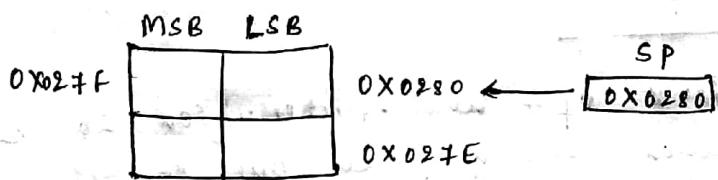
Stack is a temporary storage area of RAM. It is a LIFO data structure. Stack pointer points to the top of the stack. In MSP430 the stack is allocated at the top of RAM and it grows downwards towards lower addresses.

The least significant bit of the stack pointer is hardwired to zero in MSP430 which points to valid word.

Operation of stack

The stack is empty after the processor has been reset. The stack pointer should be initialized to the first address beyond the top of RAM which is i.e. 0x0280 in F2013.

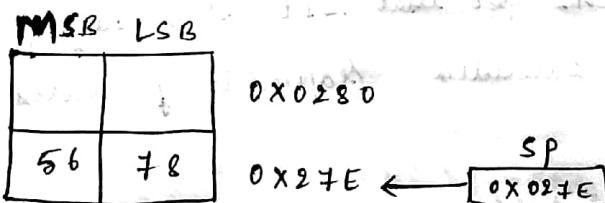
1) Stack after initialization:



2) Stack after execution of push.W #0x5678

The word 0x5678 has been pushed on to the stack.

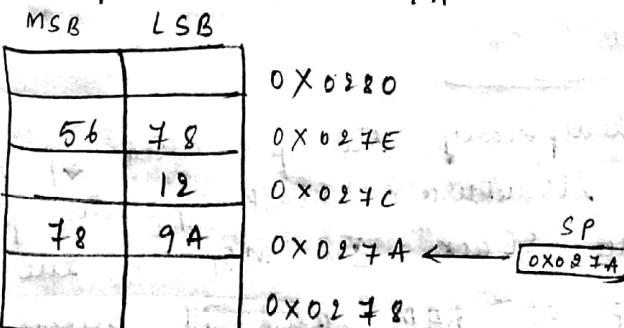
The value of the stack pointer is first decremented by 2 so that it points to the new location on the stack then the value is copied to this address.



push => decrement sp by 2, store the data.

pop => take the data increment sp by 2

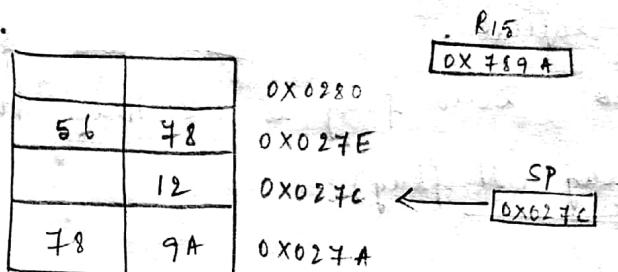
3) Stack after push.b #0x12 push.W #0x789A



The bytes 0x12 is written on to the stack it goes into the lower byte of the stack next word, where the upper byte is wasted.

4) Stack after pop in R15

The last push word is popped from the stack into R15 register when the contents of SP is incremented by 2



push operation is decrement SP by 2 and store the data and pop is take the data and increment SP by 2.

Status register (SR)

Status register also known as flag register. Each bit of Status register is called flag whose function can be put into 3 categories.

M The format of Status registers in MSP430 is as shown

| 15 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|---|------|------|---------|---------|-----|---|---|---|---|
| Reserved | V | SCG1 | SCG0 | OSC OFF | CPU OFF | GIE | N | Z | C | |

I 1) Result of arithmetic and logical operation.

i) Carry flag: The C flag is set whenever there is a carry from the most significant bit to indicate that the result of an arithmetic operation is too large than the space allocated i.e. the overflow has occurred.

Ex:

$$\begin{array}{r} 0x98H \rightarrow 1001\ 1000 \\ 0x85H \quad \quad \quad 1000\ 0101 \\ \hline \boxed{1}\ 1DH \quad \boxed{1}\ \overbrace{0\ 0\ 0\ 1}^{\text{Carry}} \overbrace{1\ 1\ 0\ 1}^{\text{Result}} \end{array}$$

The result is large to be held in a single byte the processor would put 1DH in destination and set the Carry flag to show that the result has overflow.

ii) Zero flag (Z flag): Zero flag is set when the result of an ~~selected~~ operation is zero.

Mainly used in compare instructions to check whether 2 nos are equal. The 2 nos are subtracted, if the result is zero. Zero flag is set. Hence it is used.

iii) Negative flag (N flag):

22/8/17

The negative flag is made equal to the most significant bit (msb) of the result which indicates the negative no. If the values are unsigned no, where 1 indicates -ve no and 0 indicates +ve no.

iv) Signed overflow flag (V): This flag is set when the result of a signed operation has overflow even though a carry may not be generated.

Eg:

$$\begin{array}{r} 77H \\ + 75H \\ \hline ECH \end{array}$$

msb

+ve -ve

| | | | | | | |
|-----|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| -ve | 1 | 1 | 0 | 1 | 1 | 0 |

E C

The range of nos for unsigned is 00H to FFH and the range of signed no is -80H to 7FH

II Enable Interrupts (GIE) General interrupt enable.

Setting GIE bit enables maskable interrupts provides that individual sources of interrupts are enabled. Clearing the bit disables all maskable interrupts.

The non-maskable interrupts can't be disabled by GIE

III Control of low power modes (CPU OFF, OSC OFF, SCG0, SCG1)

These bits control the mode of operation of IIC.

Setting these bits puts the device into one of its low power modes.

RESET :

A reset is a sequence of operation that puts the device into a well defined state from which the user program may start. This is necessary when power is first applied.

A reset is also generated if the device detects a serious fault in hardware or software from which the user program can't be expected to recover.

The MSP430 has 2 levels of reset depending on whether the reset was caused by hardware or software.

- i) power on reset (POR)
- ii) power up clear (PUC)

i) power on reset (POR)

A power on reset is generated by the following conditions related to hardware.

- * When the device is powered up the power on reset is trialed if the supply voltage drops to a low value at which the device may not function correctly i.e a brown out detection.
- * A low signal on ~~RESET~~ RST / NMI pin resets the device if the pin is configured for the reset function. The RST pin function is active by default.
- * A Supply voltage supervisor (SVS) is used in most of the MSP devices. It sets the SVSFG flag if the voltage falls below the program level and can reset the device optionally.

iii) power up clear (puc):

This always follows a POR. It is also generated when Software appears to be out of control as listed below

a) The watch dog timer overflows in the watch dog mode. WDT is made active by default and must be disabled or regularly cleared before it rolls over.

b) Whenever an attempt is made to write to WDTCTL (watchdog timer control register) with the incorrect password in the upper byte a reset occurs [correct password \Rightarrow WDTPW = 0x5A]

c) The registers for the flash memory controller (FCTLn) are protected by password 0xA5, to protect runaway software from corrupting the stored program.

d) puc is triggered if an attempt is made to fetch an instruction from the range of addresses reserved for peripheral registers or to read unimplemented memory.

Initial Conditions after reset:

Initial conditions for all registers and peripherals after a POR and PUC are

i) RST/NMI pin: The RST/NMI pin is configured for reset, it is also used for the JTAG interface in some devices like F2013

ii) Most I/O pins are configured as digital inputs few exceptions are, pins that are shared with the crystal in F2013

iii) peripheral modules and registers are initialized in different ways

a) RW=0 \rightarrow This means a bit can be read and written and is initialized to zero only after PUC

b) $\text{RW}-(0)$ → Means that a bit can be read and written and is initialized to zero only after a POR it retains its value through a PUC.

g) The status register is cleared, therefore the device operates at full power even though it might have been in low power mode before reset.

v) WDT starts in Watchdog mode which is essential as a safety feature. The WDT must be disabled or serviced before it times out and resets the chip.

vi) The PC is loaded with Reset Vector which is stored in the word at 0xFFFFE. This provides the address of first instruction to be executed.

It is important to identify the interrupt flag register. Source of Reset when debugging. Contains 1 (IFG1) Contains the following flags

a) WDT IFG (Watchdog timer interrupt flag):

This shows that the watchdog timed out as the security key was violated.

b) OFIG (Oscillator fault interrupt flag):

This indicates an oscillator fault (causes a non-maskable interrupt).

c) RST IFG (Reset interrupt flag): Indicates a reset caused by a low signal on RST/NMI pin.

d) POR IFG (power on reset interrupt flag): This flag is set for power on reset.

e) NMI IFG (Non-maskable interrupt flag):

This is set for non-maskable interrupt, flag

f) SVS FG (Supply voltage supervisor flag):

SVS FG can be checked in SVSCTL register which is initialized to zero after brownout reset.

Hardware Issues:

- 1) External connections to the reset pins:
The RST / NMI pin must not be left floating.
- 2) poor quality power supply must not be used.

Summary:

- * Block diagram of Computer System
- * Harvard v/s Von-Neumann.
- * Types of RAM and ROM → Memory
- * MSP430 - block diagram
- * Memory Map - mapped I/O - P1IN, P1OUT, P1DIR
- * CPU → PC, SP, SR, CG.
- * Clock generator → ACLK, MCLK, SMCLK
- * Exception - interrupt and reset
- * Reset - POR, PUC
- * MSP430 family