MODULE 1 Explain the basic concepts and structure of Embedded systems

Embedded Systems – Definition, Comparison with general purpose computers – Classify embedded systems with different criteria, Applications and Purpose.
Building blocks of an Embedded Systems – Core of Embedded System, Classification of Memory, Memory selection for embedded systems, Role of Sensors, actuators, I/O subsystem, communication interface, Embedded Firmware and other components.
Characteristics and Qualities of Embedded System – Characteristics, Quality Attributes,
AVR Microcontroller Architecture - Factors to be considered in selection – Simplified view of AVR microcontroller, ATMega32 - Registers, Data Memory, AVR Status register, Program Counter and Program ROM space, I/O ports, Registers associated with I/O ports.

M1.01 Explain the features of embedded systems.
M1.02 Illustrate the building blocks of embedded systems.
M1.03 Interpret the AVR architecture.

EMBEDDED SYSTEM

- it is an electronic/electromechanical system designed to perform a specific funcion.
- It is a combination of both hardware and firmware.
- It is used in fields like- household appliances, telecommunications, medical equipment etc.

COMPARISON

General purpose	Embedded system
Combination of generic h/w and os for executing a variety of applications	Combination of special purpose hardware and embedded os for executing a specific applications
Contains general purpose os	May or may not contain an os
Can reinstall os and add or remove user application	S/w are pre-programmed. There may be exception
Selection process based on performance	Selection process based on application
Response are not time critical	Certain category are time-critical
Need not be deterministic	Is deterministic(everything that happens must happen as it does and could not have happened any other way)

CLASSIFICATION

1. based on generation, ES are

- **first generation**-4 and 8 bit microprocessor were used. Hardware is simple and firmware is in assembly code. e.g. digital telephone keypad, stepper motor control unit.
- **Second generation**-16 bit microprocessor and 8/16 bit microcontroller. Instruction set are more complex. Embedded os are used. e.g.data aquisition system
- **third generation**- 32 bit processor and 16 bit microcontrollers are used. Instructions are more complex. Instruction pipeline evolved.e.g.robotics, media, industrial process control
- **Fourth gneration**- high performance. Real time embedded os used.e.g. smartphone, mobile internet devices

2. based on complexity and performace, ES are

- **small scale ES**- simple, not time critical, low performance,low cost, 8/16 bit microprocessor/microcontrollers.may or may not contain an OS. e.g. electronic toy
- **Medium scale ES**,- slightly complex, medium performance, low cost, 16/32 bit microprocessor/microcontroller.OS is either general purpose or real time is present.
- **Large scale Es-** highly complex, high performance, 32/64 bit RISC processor/ controller.may contain multiple processor. RTOS is used.

3. based on deterministic behaviour(means execution behaviour)

- Hard ES-a system have strict execution time limit
- Soft ES- no mandatory requirement of completing the deadline for every task.

4. based on triggering

- event triggered-a processing activity is iniiated as a consequence of occurence of a single event
- time triggered- executes one or more set of tasks according to a predetermined and set task schedule.

PURPOSE

1. **Data Collection/Storage/Representation**-data collection is usually done for storage, analysis, manipulation and transmission. Data refers text,voice,image,video, electrical signals and any other measurable quantities.e.g.digital camera.
2. **Data Communication-** transfer data from one sysem to remotely located system. e.g.wireless router.
3. **Data Processing**-ES can be used for various kinds of data processing. e.g. digital hearing aid
4. **Monitoring-** ES designed for monitoring purpose. e.g.ECG
5. **Control-**ES with control functionalities control over some variables according to the changes in input variables.e.g.AC
6. **Application Specific User Interface**- ES with application specific user interface such as buttons, switches, keypads, lights, bells, display units etc. e.g. mobile phone.

MAJOR APPLICATION AREAS OF ES

1) consumer electronics-camcoders, cameras
2) house hold appliances- TV, washing machine, fridge
3) home automation and security system-AC, fire alarm
4) automotive industry-engine control, ignition system
5) telecom-celphone,telephone witches
6) computer peripherals- printer, scaner
7) computer networking system-roter, switch, hub
8) healthcare-Eeg,ECG
9) measurement and instrumentation-digital multimeter, CRO
10) banking nad retail-ATM, currency counter
11) card readers-barcode, handheld devices etc.

Typical ES

elements of an ES

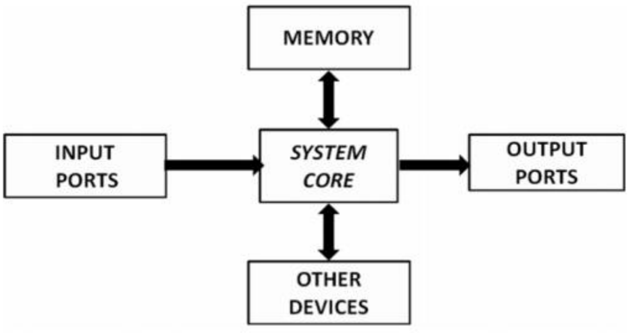


Figure 2.0 : Elements of an Embedded System

- system core-microprocessor/controller/field prgrammable gate array(fpga)/digital signal processor. Processor performs the predefined operation with the help of firmware.
- output ports- actuators or other devices that regulate the physical varible.e.g,LCD,buzzer
- input ports-sensors , keyboards, pushbutton switches.
- Memory- it is ROM. It is for holding the control algorithm and other important configuration details.

CORE OF ES

thecore of the ES cosist of the following categories.

1. General Purpose And Domain Specific Processors- these are microprocessor/microcontroller/digital signal processor
2. Application Specific Integrated Circuits
3. Programmable Logic Devices
4. Commercial Off-The-Shelf Components

General Purpose And Domain Specific Processors

microprocessor

- isa silicon chip representing CPU
- it perform arithmetic and ogic operations
- different generation of processors are:-

Intel 4004	4 bit processor speed- 740khz used in olden calculators	1k data memory 12 bit PC 4k program memory 16 4 bit registers 46 instructions
Intel 4040	4 bit processor	1k data memory 12 bit PC 8k program memory 16 4 bit registers 60 instructions interrupt capabilities
Intel 8008	4 bit processor	1k data memory 14 bit PC 8k program memory 16 4 bit registers 60 instructions
Intel 8080	8 bit processor used in industrial control and embedded applications	16 bit address bus 16 bit PC 7 8 bit register
Intel 8085	8 bit processor	2 new instructions 3 interrupt pins seria I/O clock generator and bus controller circuits supply=5v
Now processors have- 16/32/64 bit 2mhz to 2.4ghz speed high performance and low cost companies are- intel, amd,freescale,IBM,TI, cyrix,hitachi,NEC,LSI logic etc.		

Microcontroller

- it is highly integrated chip
- contains CPU, ram,registers, onchip memory, I/o ports, timers and interrupt controller circuts.
- First microcontroller-TMS1000
- intel first microcontroler is intel 8048
- other microcontrollers are- 8051. it put under MCS-51 family.it is general purpose microcontroller
- PIC- controller from microchip technologies
- AVR- designed for automotive domain

Difference Between Microprocessor And Microcontroller

microprocessor	microcontroller
Silicon chip contains CPU	Highly integrated chip contains CPU,Ram, registers, onchip memory,i/o ports,timer and interrupt controller
It requires the combination of other chips fo functioning	It doesn’t require external chips for its functioning
General purpose processor	Domain specific processor

doesn’t contain built in I/O port	Contain multiple built in i/o ports
Limited power saving options	Lot of power saving features

Digital Signal Processor

- powerful specific purpose 8/16/32 bit microprocessor
- designed for performing high speed computational operations
- e.g. audio video signal processing, telecommunication and multimedia appliations
- incoporate
  - program memory- memory for storing program
  - data memory- for storing data and signal
  - computational engine- perform signal processing
  - i/o unit- capturing signas to be processed and delivering the process

processors have RISC and CISC architecture

RISC	CISC
Reduced instruction set computer	complex instruction set computer
Lesser no.of instructions	greater no.of instructions
Instruction pipelining	No Instruction pipelining
Operations are performed on regiters only	Operations are performed on regiters or memory depending on the instruction
Large no. Of registers	Limited no. Of registers
More code to execute a task	less code to execute a task
Single, fixed length instructions	Variable length instructions
Hardvard architecture	Hardvard architecture or von neumann architecture
e.g. Atmel AVR	e.g.8051

Processor architecture design

Hardvard	Von Neumann(Princeton)
Separate bus for instuction and data fetching	Single shared bus for instuction and data fetching
High performance	low performance
High cost	cheaper
No memory alignment problems	Allows self modifying codes
No chances for corruption of program memory	Chances for corruption of program memory

Little Endian Processor	**Big Endian Processor**
Lower order byte of data stored in lowest memory address. Highr order address at highest address	Higher order byte of data stored in lowest memory address and lower order byte at highet address

Application Specific Integrated Circuits(Asics)

- it is a microchip designed to perform a specific or unique application
- it integrates several functions into a single chip and reduce system development cost
- proprietary products

Programmable Logic Devices

- they can be re-configured to perform any nmber of functions at any time
- customers can change the circuitary at any designphase

- e.g. network router, DSL modem, DVD player
- PLDs are two types
  - field programmable gate array(highest no.of logic gates)
  - complex programmable logic devices(smaller amount of logics upto 10,000 gates)

Commercial Off-The-Shelf Component(Cots)

- these are designed to provide easy integration and interoperability with the existing system.
- They are readily available in the market
- no need to design any module ourself
- no operational and manufacturing standards
- e.g. remote controlled toy car

MEMORY

- used for storing data
- memory inside the processor is known as on chip memory
- memory outside the processor is known as off chip memory
- different types of memory used in ES.
- Depending on fabrication, erasing and programming techniques, the memories are:-

masked ROM-

- one time programmable device
- use hardwired technology for storing data
- low cost for high volume production
- not possibe to alter the data
- factory programmed according to the data provided by the end user.

PROM/OTP

- it is programmed by a programmer.
- Can not be reprogrammed
- low cost production
- widely used for production of ES
- not used for development process

EPROM

- can reprogram the same chip
- programmed by the programmer by applying a high voltage
- it contains a quartz crystal window for erasing data.
- If the window is exposed to ultraviolet rays for a fixed duration, the entire memory will be erased.
- Reprogramming is a tedious and time cosuming process.

EEPROM

- memory can be altered by using electrical signals at the register/byte level
- capacity is limited

FLASH

- variation of EEPROM
- large capacity
- the erasing of memory can be done sector/ page level
- does not affect the other pages
- each page should be erased before reprogramming

NVRAM(non volatile RAM)

- it is a RAM with battery backup
- memory and battery arebpacked together in a single package.

RAM

- data memory or working memory of controller or processor.
- We can access data directly
- it is volatile
- 3 types

SRAM

- made of transistors
- doesn’t require refreshing
- low capacity
- expensive
- fast operation

DRAM

- made of capacitor
- require refreshing
- high capacity
- less expensive
- slow in operation

NVRAM

- it is a RAM with battery backup
- memory and battery arebpacked together in a single package.

MEMORY SELECTION FOR ES

selection of memory is

- based on type and application
- system requirements
- type of the processor used for designed
- on chip memory(RAM/ROM) is sufficient or external memory is needed.

SENSORS AND ACTUATORS

- sensors are the trasducer device that converts energy from one form to another for any measurement or control purpose
- actuators converts signals to corresponding physical action(motion). It act as an output device.
- These two are act as a i/o subsystems in embedded system

I/O SUSYSTEMS

some of the sensors and actuators used in embedded system are

1. **LED**

used as an indicator for the status of various signals or limitations. It is a P-N junction diode and contains anode and cathode. For proper working anode connected to +ve terminal and cathode to -ve terminal.

2**. 7 segment LED display**

it contains 8 LEDs . 7 are used for displaying alpha numeric characters and 1 is used for representing decimal point. It is available in two configuration- common anode and common cathode.

3.**Optocoupler**

combines an LED and a phototransistor in a single package.it is used for supressing interference in data communication circuit isolation,high voltage separation etc. It can be used in either input or output circuits.

4.**Stepper motor**

A stepper motor is designed to break up a single full rotation into a number of much smaller (and essentially equal) part rotations. For practical purposes, these can be used to instruct the stepper motor to move through set degrees or angles of rotation. widely used in printer/fax, electronic products and robot controls.

5.**Relay**

The relay is the device that open or closes the contacts to cause the operation of the other electric control. It works on the principle of an electromagnetic attraction.

6**.Piezo buzzer**

a piezo buzzer is **a type of electronic device that's used to produce a tone, alarm or sound**. It's lightweight with a simple construction, and it's typically a low-cost product.

7.**Push button switch**

A push button switch is a mechanical device used to control an electrical circuit in which the operator manually presses a button to actuate an internal switching mechanism.

Comes in two configuration- push to make and push to break.

8.keyboard

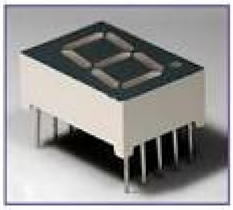
Keypad is an analog switching device which is generally available in matrix structure. It is used in many embedded system application for allowing the user to perform a necessary task. A **matrix keypad** is consists of an arrangement of switches connected in matrix format in rows and columns.

piezo buzzer

relay

stepper motor

7 sement display



push button

keyboard

Programmable peripheral interface

- used for extending the i/o capabilties of microprocessor/ microcontroller.
- e.g.8255 A. It is a 8 bit microprocessor/microcontroller
- 8255A supports 24 i/o pins

COMMUNICATION INTERFACES

- it is used for communicating with various subsystems of the embedded system and with the external world.
- Communication interface can be viewed in two different perspectives- device/board level(onboard communication) and product level(external communication interface).

Onboard communication interface:

- communication between various components within an embedded product.
- e.g. serial interfaces like I2C,SPI,UART and parallel interfaces.
- **I2C**- pronounced I square C. I2C bus comprise of two bus- serial clock line(scl) and serial data line(sdl).scl generate synchronous clock and sdl transmit serial data.the device connected to I2C bus act as master and slave.it supports 3 different data rate- 100kbps(standard mode), 400kbps(fast mode) and 3.4mbps(high speed mode)
- **SPI**(serial peripheral interface)-it is a synchronous bidirectional four wire serial interface bus. It works on the principle of shift register. It does not support an acknowledgement mechanism. It supports multiple master and slave devices. Master have the main control and slave follow the master.
- **UART**(universal Asynchronous receiver transmitter)- data transmission is in the form of asynchronous serial transmission. For proper communication the transmit line of sending device should be connected to receiving line of the receiving device.
- **Parallel interface**:- the communication through the parallel bus is controlled by control signal interface between device and the host. It offers high speed for data transfer.

External communication interfaces:

- communication between the ES with external world. Various interfces for external communication:-
- RS232-full duplex,wired, asynchronous serial communication interface. It supports two types of connectors-DB 9 and DB25.it is a point to point communication interface.
- USB- high speed wired serial bus for datacommunication.it transmit data in packet format. The host start the communication.it has the ability to supply power to the connecting device.
- Infrared (IrDA)-serial, half duplex, line of sight based wireless technology for data transmission.it supports point to point and point to multipoint data communication.
- Bluetooth -low cost, low power, short range wireless communication for data and voice. It uses 2.4ghz radio frequency spectrum fo communication.

- Wifi- wireless fidility. It supports IP based communication Or it intended for network communication.it supports data range from 1mbps to 150mbps.

EMBEDDED FIRMWARE:

- refers the program instructions or configuration setting embedded into the program memory of an embedded system.
- There are various methods to develop an embedded firmware. Some of them are:
  1. write the program in high level language.
  2. write the program in assembly language.
- After developing the program, it should convert into processor understandale machine code. This processing is called hex file creation.
- For hex file creation, can use a cross compiler included in IDE or utilities supplied by the processor/controller vendor.
- There are two approaches for software design- first, infinite loop or superloop- control runs from top to bottom and jumps back to the top of the program.
- Second, splitting the functions into task and run the task using a scheduler.

OTHER SYSTEM COMPONENTS

The essential circuits for the proper functioning of processor/ controller are:

1.**Reset Circuit:**

- it ensures the device is not operating a voltage that is not guaranteed to operate.
- The reset signal brings the internal register and different hardware systems of pocessor/controller to a known state and start the firmware execution.

2. **Brown-Out Protection Circuit:**

- it prevents the processor/controller from unexpected program execution behaviour
- e.g. in the case of battery powered device, the processor behaviour may not be predictable if the supply voltage is below the recommende operating voltage.

3. **Oscillator Unit:**

it is for producing the neccessary clock signals.

4. **Real Time Clock(Rtc)**:

- it holds the information like current time, date , month, year and day of the week,etc. And supplies timing reference to the system.

5. **Watchdog Timer**

- it monitor the firmware execution and reset the system proessor/ controller when the system hangs up.
- It is a hardware timer.
- It increments or decrements a conter with each clock pulse and generate a reset signal that reset the processor if the count become zero or highest value.
- If the firmware execution doesn’t complete due to malfactioning with in the time required by the watchdog timer to reach the value, the counter will generate a reset pulse and this will reset the processor.

CHARACTERISTICS OF AN EMBEDDED SYSTEM

1. **Application And Domain Specific**:- embedded system designed to perform a specific task and can not replace for a particular domain to another.
2. **Reactive And Real Time**:-Es produce changes in output in response to changes in input and the system should respond to request or task in a known amount of time.
3. **Operates In Harsh Environment**
4. **Distributed**:ES may be a part of a large system. Many no. Of such distributed ES form a single large embedded control unit.
5. **Small Size And Weight**
6. power **Concerns:Es** should be designed in such a way as to minimize the heat dessipation.

QUALITIES OF ES

Operational Quality Attributes

1. **Response -**it is a measure of quickness of the system. it gives an idea about how fast the sysytem is tracking the changes in input variables
2. **Throughput-**it is the measure of efficiency of a system.it can be defined as the rate of production.
3. **Reliability-**it is a measure of what is the percentage of susceptibility of the system to failures.
4. **Maintainability-**it deals with the support and maintenance to the end user. It can be divided into two- preventive maintenance(scheduled or periodic maintenance) and corrective maintenance(maintenance to unexpected failures).
5. **Security-**it is a measur of protection of data from unauthorised disclosure(confidentiality), unauthorised modification(integrity) and unauthorised users(avilability).
6. **Safety-** it deals with the possible damages that can happen to the operators.

Non -operational Quality Attributes

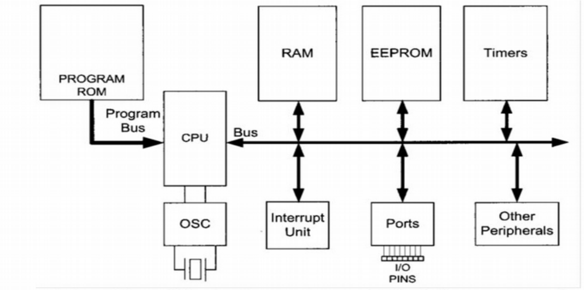
1. **Testability And Debugability**-it deals with how easily one can test the design and application. Debugging are two types- hardware debugging and firmware debugging. This means check the error.
2. **Evolvability**-it refered as non heritable variation
3. **Portability**- embedded product functions various environment.
4. **Time To Prototype And Market**-time to market the product is a critical fator in the success of commercial embedded product.
5. **Per Unit And Total Cost-**budget and the total system cost should be proprly balanced to provide a marginal profit.

AVR MICROCONTROLLERS

CRITERIA FOR CHOOSING A MC

1. whether 8 bit, 16bit or 32 bit microcontroller can best handle the computing needs of the task most effectively
2. what is the highest speed that the microcontroller supports.
3. packaging is important in terms of space, assembling and prototyping the end products
4. power consumptions
5. the amount of RAM and ROM on the chip
6. number of i/o pins and timer on the chip
7. easy of upgrade to higher performance or lower power consumption version.
8. cost per unit
9. how easy it is to develop products around it.
10. it is ready available in needed quantities both now and in the future.

SIMPLIFIED VIEW OF AVR MICROCONTROLLERS



The Simplified view of an AVR Microcontroller

program ROM

- used to store program
- size is 8 Mbytes
- it can vary from 1k to 256k

data RAM

- used for storing data
- aximum size 64kb

EEPROM

- store critical data that does not need to be changed very often

AVR Family

AVRs are generally classified into 4 groups:

1. **Classic**
2. **Mega**
3. **Tiny**
4. **Special Purpose**

Classic AVR (AT90Sxxxx)

- This is the original AVR chip, which has been replaced by the newer AVR chips.
- Example: AT90S2313, AT90S2323 etc.

Mega AVR (ATmegaxxxx)

- These are powerful micro controllers with more than 120 instructions and lots of different peripheral capabilities. These are widely used.

Charactristics are:

- Program Memory : 4k to 256k bytes
- Package : 28 to 100 pins
- Extensive peripheral set
- Extended instruction set: They have rich instruction sets.
- Example: ATmega8, ATmega16, ATmega32, ATmega64 etc.

Tiny AVR (ATtinyxxxx)

- Have less instructions and smaller packages .
- Can design systems with low costs and power consumptions .
- Characteristics are
- Program Memory : 1k to 8k bytes
- Package : 8 to 28 pins.
- Limited peripheral set.
- Limited instruction set: The instruction sets are limited. Some of them do not have the multiply instruction.
- Example: ATtiny13, ATtiny25, ATtiny44 etc..

Special Purpose AVR

- ICs of this group can be considered as a subset of other groups, but their special capabilities are made for designing specific applications.
- Some of the special capabilities/ examples are: USB controller, CAN controller, LCD controller, Zigbee, FPGA, Ethernet controller and Advanced PWM.

AVR -ATMega 32

AVR Data Memory

In AVR micro controller there are two kinds of memory space:

- Code Memory Space
- Data Memory Space.

Our program is stored in code memory space,whereas data memory stores data.

The data memory is composed of three parts:

- GPRs(General Purpose Register)
- I/O Memory
- Internal SRAM

The diagram below shows Data Memory for the AVRs:

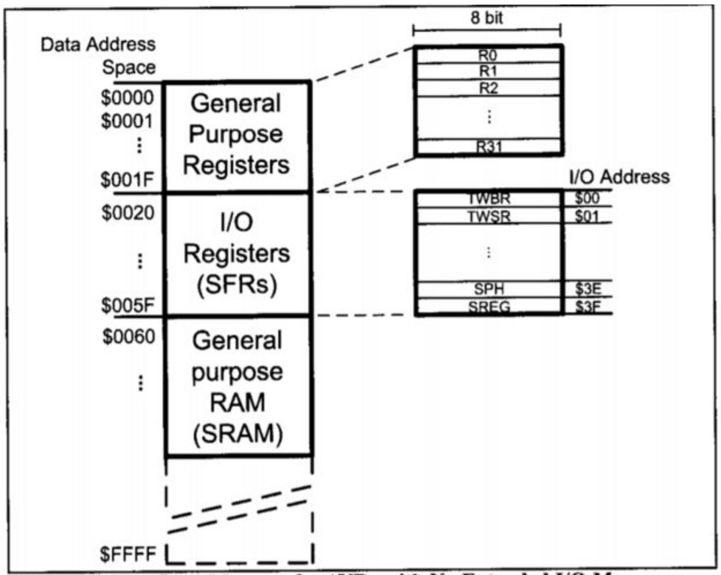


Figure 2-3. The Data Memory for AVRs with No Extended I/O Memory

GPRs

- Registers are 8 bit.
- There are 32 general purpose register. They are R0-R31.
- They can be used by all arithmetic and logic instructions.
- GPRs uses 32 bytes of data memory space .
- They always take the address location of $00-$1F in data memory space.

Input Output Memory

- The I/O memory is dedicated to specific functions such as **status register, timers, serial communication, I/O ports, ADC** and so on.
- The function of each I/O memory location is fixed by the CPU designer.
- I/O memory is made of **8-bit registers**.
- The number locations in the data memory depends on the pin numbers and peripheral functions supported by the chip.
- AVRs have at least **64 bytes of I/O memory location**.
- This 64 byte section is called **Standard I/O memory**.
- Also called **SFR (Special Function Register)** because of its its specific function.
- More than 32 I/o pins, tere is also an extended I/o memory.

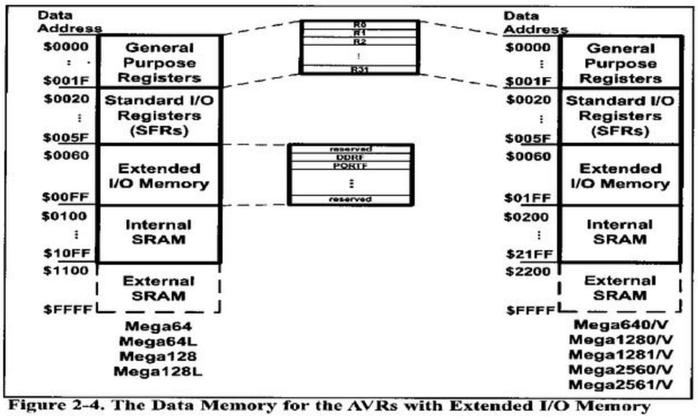


Figure 2-4. The Data Memory for the AVRs with Extended I/O Memory

Internal Data SRAM

- Internal data SRAM is widely used for storing data and parameters by AVR programmers and C Compilers.
- Each location of the SRAM can be accessed directly by its addresses.
- Size of SRAM can vary from chip to chip.

Formats Avr Data Format Representation

- The AVR microcontroller has only one data type.
- It is 8 bits, and the size of each register is also 8 bits.
- It is the job of the programmer to break down data larger than 8 bits (00 to OxFF or 0 to 255 in decimal) to be processed by the CPU.
- There are four ways to represent a byte of data in the AVR assembler.
- The numbers can be represented in AVR assembler in 4 ways-hex, binary, decimal, or ASCII formats.

Hex numbers

- There are two ways to show hex numbers:
    - Put 0x (or 0X) in front of the number like this: LDI R16,0x99
    - Put$ in front of the number, like this: LDI R22, $ 99

Binary numbers

- There is only one way to represent binary numbers in an AVR assembler.
- Put 0b infront of the binary number
- LDI Rl6,0bl0011001   ;R16 = 10011001 or 99 in hex
- Can use 0b or 0B.

Decimal numbers

- To indicate decimal numbers in an AVR assembler we simply use the decimal and nothing before or after it.

  LDI   R17,   12

ASCII characters

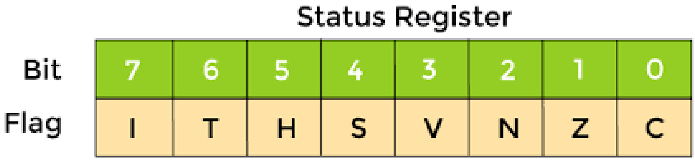
- To represent ASCII data in an AVR assembler we use single quotes as follows:

  LDI R23,’2’

  to represent a string, double quotes are used

AVR Status Register

- AVR has a flag register to indicate arithmetic conditions such as carry bit.
- The **flag register in AVR is called as Status registe**r.
- Status register is an **8-bit** register.
- Fig. 8 bits of Status register in an AVR



Status Register

Bit	7	6	5	4	3	2	1	0
Flag	I	T	H	S	V	N	Z	C

- The bits C, Z, N, V, S and H are called *conditional flags*, meaning that they indicate some conditions that result after an instruction is executed.

1. Carry flag (C)
    - This flag is set(c=1) whenever there is a carry out from the D7 bit.
    - This flag bit is affected by 8 bit addition or subtraction.
    - Used to detect errors in unsigned arithmetic operation.
2. Zero flag (Z)
    - Shows the result of an arithmetic or logic operation.
    - If the result is zero, then Z=1 and If the result is not zero, then Z=0.
3. Negative flag (N)
    - Reflects the result of an arithmetic operation.
    - If the D7 bit of the result is zero, then N=0 and the result is positive.

- If the D7 bit of the result is one, then N=1 and the result is negative.

4. Overflow flag (V)
    - Is set whenever the result of a signed number is too large, causing the high order bit to overflow into sign bit.
    - Used to detect errors in signed arithmetic operation.
5. Sign bit (S)
    - Is the result of Exclusive OR ing of N flag and V flag.
    - If v=0,it shows that overflow has not occurred and the S flag will be the same as D7.i.e
        - v=0 D7=0 N=0 S=0 result is positive
        - v=0 D7=1 N=1 S=1 result is negative
    - if V=1, it shows that overflow occurred and s flag will be opposite to D7.
        - V=1 D7=0 N=0 S=1 result is negative
        - v=1 D7=1 N=1 S=0 result is positive
6. Half carry flag (H)
    - If there is a carry from D3 to D4 during an ADD or SUB operation, this bit is set, otherwise it is cleared.
    - This flag bit is used by instructions that perform BCD arithmetic.
    - In some microprocessor this is called AC flag (Auxiliary Carry flag).
7. Bit Copy Storage (T)
    - Acts as source and destination for the Bit Copy Storage instructions.
8. Global Interrupt Enable (I)
    - Activates or deactivates the interrupts.

Program counter in the AVR

- The most important register .
- The program counter is used by the CPU to point to the address of the next instruction to be executed.
- As the CPU fetches the opcode from the program ROM,the program counter is incremented automatically to point to the next instruction.
- a 14-bit program counter can access a maximum of 16K $(2^{14} = 16K)$ program memory locations.
- The ATmega64 has a 15-bit program counter, so it has 32K locations.
- The program counter in the AVR family can be up to 22 bits wide.
- This means that the AVR family can access program addresses 000000 to $3FFFFF, a total of 4M locations.

ROM memory map in the AVR family

- No member of the AVR family can access more than 4M words of opcode because the program counter in the AVR can be a maximum of 22 bits wide (000000 to $3FFFFF address range).
- The AVR microcontroller wakes up at memory address 0000 when it is powered up., when the AVR is powered up, the PC (program counter) has the value of 00000 in it.
- Starting address of program ROM is 00000 and the last location can be different depending on the size of the ROM on chip.
- ATMega 32 , 32k onchip ROM. Code addressing range- 00000-03fff. ROM organization- 16Kx2 bytes.

ATTiny25	2k	00000-003ff	1kx2 bytes
ATMega8	8k	00000-00fff	4kx2bytes
ATMega64	64k	00000-07fff	32kx2

ATMega128	128k	00000-0ffff	64kx2
ATMega256	256k	00000-1ffff	128kx2

problem :

- find the ROM memory address of each of the following AVR chips.
  - ATTiny25 with 2k
    ATMega64 with 64k
    ATMega16 with 16kb
    solution:
    ATmga16 with 16kb
    total no. of bytes= 16*k(1k=1024 bytes)=16*1024=16384 bytes
    each address location in AVR is 2 bytes. So total memory location=16384/2=8192
    so starting address is fixed, i.e. 0000
    8192 in hex=2000 (divide the number by 16)
    so the location is 1fff(2000-1)

do it yourself

I/O PORT

◆ In the AVR family, there are many ports for I/O operation. A total of 32 pins are set aside for the 4 ports PORTA, PORTB, PORTC & PORTD.

◆ To use any of these ports, it must be programmed.

◆ In addition to simple I/O each port has other functions such as ADC, timers, interrupts & serial communication.

◆ Each port has three I/O registers associated with it.

◆ They are designated as PORTx, DDRx & PINx.

◆ Eg:- For port B, we have PORT B, DDRB & PINB

◆ DDR(Data Direction Register), PIN (Port Input Pin)

◆ Each I/O registers have 8 bit wide and each port has a maximum of 8 pins.

HOW TO ACCESS I/O REGISTERS ASSOCIATED WITH THE PORTS

DDRx register

To work the DDRX register an output port set DDRX register 1.

- Eg:

✓ For this the command is

. DDRB= OXFF;

. ✓ Then the PORT B act as a output register, then we can output data to the AVR I/O pins.

- To make a port an input port, we must put zeros into the DDRX register.
- Eg:

```
DDRB=0X00;
```

PIN register

- To read the data present at the pins, we should read the PIN register.
- To bring the data into CPU from pins , we read the contents of the PINxregister.
- To send data out to pins, we use PORTx register.

PORT register

PORT A

- It has 8 pins (PA0-PA7)
- To make PORT A an output port, DDRA must be set
- To make PORT A an input port, DDRA must be cleared

PORT B

- It has 8 pins (PB0-PB7)
- To make PORT B an output port, DDRB must be set
- To make PORT B an input port, DDRB must be cleared

PORT C

- It occupies a total of 8 pins(PC0-PC7)
- to use the pins of pairs of PORT C as input port, DDRC must be cleared
- To use the pins of PORT C as output port DDRC must be set

PORT D

- It occupies a total of 8 pins PD0-PD7
- To use the pins of PORTD as input or output ports each bits of the DDRD register must be cleared or set respectively.
- to use the pins of pairs of PORT D as input port, DDRD must be cleared
- To use the pins of PORT D as output port DDRD must be set