MICROCONTROLLERS ATMEL ATMEGA16

Submitted by:

GURSIMRAN SINGH
IST YEAR
CSE
M.NO – 9915326619
simar.i3r@gmail.com

INDEX

Preface

Making the report was a great experience. Learning from the internet and then documenting it, was really very experiencing and during the process I learnt a lot of new things (both Tech and non Tech) some of which I would like to share over here.

I got some new ideas like retaining the data of the SEARCH KEYWORDS used as if somebody else is dissatisfied or if sometime later one has to review his report he will not waste his precious time in searching the irrelevant keywords and at the same time will surely benefit from the relevant sites and search keywords as content on the internet keeps on updating and new content will be found using that previous keywords most probably.

As a lot of time was spent in finding the meaning of some common terms that are very essential in understanding this report, so I feel including the TERMINOLOGY in a separate section. These both include things I know before and that I didn ’ t know and found using Google and also wiki.

Two fonts have been used in this report. Things that are taken from authentic sources like data and other authentic material which I felt should be kept unaltered and unedited are written in VERDANA. Other text items that have been written by me are in GOTHIC.

Also I learnt some MSWord shortcuts like:

- ^+k for bookmarking
- ^+I for *italics*
- ^+B for **bolding**
- ^+[for dec font size
- ^+] for increasing font size

were extremely USEFUL AND helped alot in managing things quickly n easily.

MICROCONTROLLER FOR SATELLITE

What is a microcontroller?

Microcontroller is in contrast to microprocessors used in personal computers. While one is used in multi purpose computers to perform a variety of tasks by handling complex instructions, microcontrollers is less complex and has less processing speed and is used in embedded systems where the purpose is very confined and specific.

At the same time there is a similarity that both performs calculation (ie arithmetic and logical) and is considered as the brain of the computer and handles all the processing related work.

What is the need of microcontroller in a satellite?

As on board computer is obviously the necessary component of the satellite. It is quite clear that any fatal error in the OBC will render the whole satellite USELESS!

Precautions taken while choosing a microcontroller

The fact that the computer has to operate in space implies that the choice of microcontroller is very specific and has to be carefully chosen. Design of the OBC also has to be done in a space operating demand.

- In case of a system error, one can ’t reset the system manually in space as done easily on earth. So special circuits called WATCHDOG has to be used in this context.
- BIT FLIPS in the memory due to radiation in space has to carefully managed otherwise software may do unpredictable things. Techniques such as EDAC have to carefully implemented in software as well as hardware to handle these BITFLIPS.
- Necessary UART, A/D and D/A for communication on earth and handling the data in various forms.

- Real-Time clock is used to schedule tasks and the software + hardware (system) has to be successfully implemented to get REAL-TIME responses as and when required.
- The ACTUATORS must be supplied with necessary PWM.
- POWER CONSUMPTION(low required)
- TEMPERTURE RANGE (-40 to 80 (in C)
- SCALABLE FREQUENCY (so conserve power when not required)

Also while choosing the microcontroller, we encounter a fight between REALIBILTY and NEW TECH. Actually the processors that has been successfully implemented and tested in earlier satellite projects(that takes 2 to 3 years duration) are backlogged in this emerging new technology world. In this large time span new technology with newly added features that emerge seems to be very tempting but reliability is equally important.

THE ATMEL AVR ATMEGA16

A Brief About ATMEL corporations

The Atmel AVR ATmega16 microcontroller is developed by ATMEL corporations. An Atmel corporation is a very reputed industry standard and holds expertise in microcontrollers production.

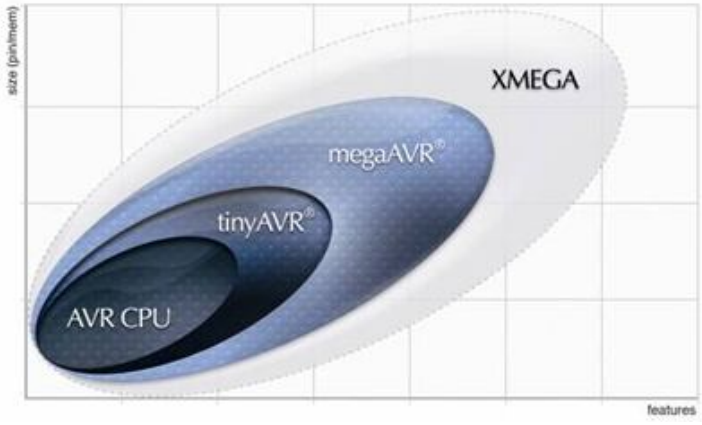
Atmel corporations has been making a variety of low power consuming and fast microprocessors that are being used worldwide for variety of purposes in embedded systems.

For the simplicity and easy of users Atmel has characterized its processors in different categories (in account their memories or some added features like LED or some other specific features) intended for different types of users for their specific purposes.

All these categories use the same basic ***AVR CORE that*** is a modified Harvard architecture 8-BIT RISC based, developed **by atmel in 1996. As all these microcontrollers use same AVR technology these have the same basic CPU core.**

Different Product Families

An AVR Flash microcontroller IS a large family of processors that share a SINGLE CORE ARCHITECTURE. This makes it easy to code these microcontrollers by understanding the basic CORE architecture which is common in all the product families.



- TinyAVR — the ATtiny series
- MegaAVR — the ATmega series
- XMEGA — the ATxmega series
- Atmel At94k FPSLIC

ATmega16 - THE ATMEGA SERIES MICROCONTROLLER

Atmega16 is of the type mega AVR and is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Developed for applications that need to store a large amount of program code, megaAVR offers substantial program and data memories, and performance approaching 1 MIPS per MHz. Better yet, megaAVR delivers the power of self-programmability for fast, secure, cost-effective remote upgrades. {4}

The ATmega16 is the big brother of the ATmega163. In addition to the ATmega163 features, it contains more clock modes, JTAG, OCD, and USART, It also have an additional PWM channel.There are 40 pins (20 each size as shown in figure) DIP. Also named as AVR AEmega328-BIT Microcontroller (nomenclatured as 32 registers, single byte(8-BIT) in size. {5}

BRIEF IDEA ABOUT THEIR NOMENCLATURE

Here 16 is amount in kB of the programmable flash memory.Also written as ATmega328-BIT Microcontroller indicates 32 registers which are 8-bit long.

ATMEL AVR CPU CORE

What is Atmel AVR?

TO start a way off with these microprocessors is really complicated and confusing. As I mentioned earlier in this report that all these microcontrollers has the same CPU core structure and uses the same programming technique and has same number of registers, interrupts ect.

So starting with the in-depth understanding of the 'Atmel Atmega16' is not at all possible without the understanding of the AVR technology of the Atmel corporations.

The **AVR** is a Modified Harvard architecture 8-bit RISC *(here 8-bit means size of register i.e single byte register)* single chip microcontroller (μC) which was developed by Atmel in 1996**.** The AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to One-Time Programmable ROM, EPROM, or EEPROM used by others. {5}

Why TO Use AVR Technology?

Variety Of Configuration

- Multifunction, Bi-directional General Purpose I/O ports with configurable, built-in pull-up resistors.
- Multiple Internal Oscillators, including RC oscillator without external parts.
- Internal, Self-Programmable Instruction Flash Memory up to 256 KB (384 KB on XMega).
    - In-System Programmable using serial/parallel low-voltage proprietary interfaces or JTAG
    - Optional Boot Code Section with Independent Lock Bits for Protection

- On chip debugging (OCD) support through JTAG or debugWIRE on most devices
- Internal Data EEPROM up to 4 KB
- Internal SRAM up to 8 KB (32 KB on XMega)
- 8-Bit and 16-Bit Timers
- Analog Comparator
- 10 or 12-Bit A/D Converters, with multiplex of up to 16 channels
- 12-bit D/A Converters
- A variety of serial interfaces, including
    - I²C Compatible Two-Wire Interface (TWI)
    - Synchronous/Asynchronous Serial Peripherals (UART/USART) (used with RS-232, RS-485, and more)
    - Serial Peripheral Interface Bus (SPI)
    - Universal Serial Interface (USI) for Two or Three-Wire Synchronous Data Transfer
- Brownout Detection
- Watchdog Timer (WDT)
- Multiple Power-Saving Sleep Modes
- Lighting and motor control (PWM Specific) Controller models
- CAN Controller Support
- USB Controller Support
    - Proper Full-speed (12 Mbit/s) hardware & Hub controller with embedded AVR.
    - Also freely available Low-speed (1.5 Mbit/s) (HID) bitbanging software emulations
- Ethernet Controller Support
- LCD Controller Support
- Low-voltage Devices Operating Down to 1.8v (to 0.7v for parts with built-in DC-DC upconvertor)
- picoPower Devices
- DMA controllers and "Event System" peripheral communication.
- Fast Cryptography support for AES and DES

HIGH PERFORMANCE

- True RISC architecture

- True single cycle execution
- One MIPS per MHz
- 32 general purpose registers
- Harvard architecture

With only a one-stage pipeline, the AVR brings fast, single-clock-cycle-per-instruction execution to the table - just as you would expect from any true RISC architecture machine.AVR Flash microcontrollers operate with clock rates up to 20 MHz, achieving close to 20 MIPS. With 32 general purpose registers, the AVR delivers unmatched performance and flexibility, especially when you program in high-level languages, like C, Pascal or Basic. {1}

LOW POWER CONSUPTION

Features such as SCALABLE FREQUENCY are used in almost all AVRs so that these do the most optimal use of power which is most critical and important in satellite systems. When the full processing speed is not needed in such situations processor may save power by coming into less power state.

- 1.8 to 5.5V operation
- A variety of sleep modes
- Fast wake-up from sleep modes
- Software controlled frequency
- Single cycle and high code density {1}

TEMPERATURE RANGE

In space the temperature variation is such that one side of the satellite that faces sun is at 200C and the other side may be at say -100C. So the temp resistant components should be used. Amtel AVR manufactures components that can work in a wide range of temperature range. So this is best suited for satellite design.

DEVICE ARCHITECTURE

MAIN POINTS IN BRIEF

Flash, EEPROM, and SRAM are all integrated onto a single chip, removing the need for external memory. Program instructions are stored in non-volatile Flash memory. Although they are 8-bit MCUs, each instruction takes one or two 16-bit words.

There is no provision for off-chip program memory; all code executed by the AVR core must reside in the on-chip flash.

The AVRs have 32 single-byte registers and are classified as 8-bit RISC devices. In most variants of the AVR architecture, the working registers are mapped in as the first 32 memory addresses (0000_{16}-$001F_{16}$) followed by the 64 I/O registers (0020_{16}-$005F_{16}$).Actual SRAM starts after these register sections (address 0060_{16}). (Note that the I/O register space may be larger on some more extensive devices, in which case the memory mapped I/O registers will occupy a portion of the SRAM address space.)

In most variants of the AVR architecture, this internal Electrically Erasable Programmable Read Only Memory (EEPROM) memory (which is semi conductor non volatile) is not mapped into the MCU's addressable memory space. It can only be accessed the same way an external peripheral device is, using special pointer registers and read/write instructions which makes EEPROM access much slower than other internal RAM.

Since the number of writes to EEPROM is not unlimited — Atmel specifies 100,000 write cycles in their datasheets — a well designed EEPROM write routine should compare the contents of an EEPROM address with desired contents and only perform an actual write if contents need to be changed.

Atmel's AVRs have a two stage, single level pipeline design. This means the next machine instruction is fetched as the current one is executing. Most instructions take just one or two clock cycles, making AVRs relatively fast among the eight-bit microcontrollers.

The AVR family of processors was designed with the efficient execution of compiled C code in mind and has several built-in pointers for the task.

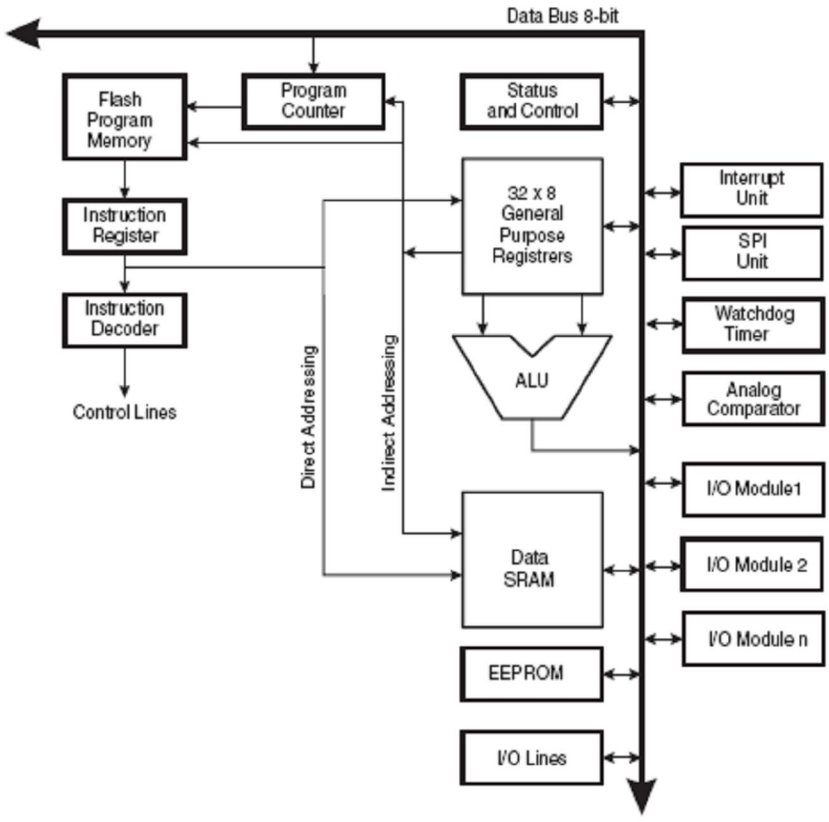
The AVR line can normally support clock speeds from 0-20 MHz, with some devices reaching 32 MHz. Lower powered operation usually requires a reduced clock speed.

All operations (excluding literals) on registers R0 - R31 are single cycle, the AVR can achieve up to 1MIPS per MHz. Loads and stores to/from memory take 2 cycles. Branching takes 3 cycles. {5}

AVR CPU CORE ARCHITECTURE

The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 3. Block Diagram of the AVR MCU Architecture



ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock

cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – Arithmetic, Logical, and Bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. {2}

HARVARD ARCHITECTURE

In order to maximize performance and parallelism, the AVR uses Harvard architecture which supports separate memories and buses for program and data. {2}

SINGLE LEVEL PIPELINING

Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. {2}

REGISTERS

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit **X**-, **Y**-, and **Z**-register. The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation. Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address

space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

The Status Register contains information about the result of the most recently executed arithmetic instruction. {2}

MEMORY

The program memory is In-System Reprogrammable Flash memory. Program Flash memory space is divided in two sections, the Boot program section and the Application Program section. Both sections have dedicated **Lock bits** for write and read/write Protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section. {2}

INTERRUPTS

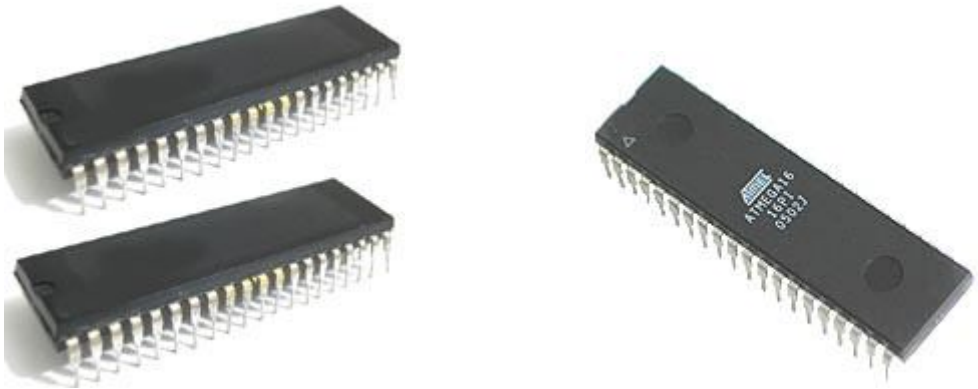
There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding Interrupt Flag. The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags.

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. {2}

AVR ATmega328-BIT Microcontroller / AVR ATmega16 Microcontrollers.

COMMON FEATURES OF ATmega SERIES IN A GLIMPSE

- 4–256 kB program memory
- 28–100-pin package
- Extended instruction set (Multiply instructions and instructions for handling larger program memories)
- Extensive peripheral set



Atmel AVR ATmega16 Microcontroller – DIP(dual in-line package**)**

CONFIGURATION

Memory	
Flash Memory	16 kB

EEPROM Data Memory	512 B
SRAM Data Memory	1024 B
General Purpose Registers (Accumulators)	32

MCU Specific

Clock Frequency	0 – 16 MHz
Supply Voltage	4.5 – 5.5 V
Sleep Modes	6
Hardware Multiplier	Yes
I/O Pins	32
On Chip Oscillator	Yes
Interrupts	20
Interrupts, External pins	3
Brown-out Detection	Yes
Power-on Reset	Yes
Fully Static Operation	Yes
On-Chip Debug support via JTAG port	Yes
IEEE 1149.1 (JTAG) Boundary Scan	Yes

Timers / Counters

Timer/Counters (8-bit)	2
Watchdog Timer with On-chip Oscillator	Yes
Real Time Counter	Yes
Timer/Counters (16-bit)	1
Pulse Width Modulator	4 ch

Analog I/O

Analog Comparator	Yes
Analog-to-Digital Converter (10-bit)	8 ch
Analog Gain Stage	2 ch

Programming Modes

In-System Programming via SPI Port	Yes
High Voltage Parallel Programming (12V)	Yes
Self-Programming via on-chip Boot Program	Yes
In-System Programming via JTAG port	Yes

Serial I/O

Full Duplex Serial Peripheral Interface (SPI)	Yes
2-wire Serial Interface (I2C compatible)	Yes
Full Duplex USART	1

MAIN FREATURES

- High-performance, Low-power AVR® 8-bit Microcontroller

- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution.
  - 32 x 8 General Purpose Working Registers.
  - Fully Static Operation.
  - Up to 16 MIPS Throughput at 16 MHz.
  - On-chip 2-cycle Multiplier.

- High Endurance Non-volatile Memory segments
  - 16K Bytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1K Byte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits

- In-System Programming by On-chip Boot Program

- True Read-While-Write Operation
  - Programming Lock for Software Security

- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface

- Peripheral Features

- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels
- 8-channel, 10-bit ADC
- 8 Single-ended Channels
- 7 Differential Channels in TQFP Package Only
- 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF

- Operating Voltages
  - 4.5 - 5.5V for ATmega16

- Speed Grades
  - 0 - 16 MHz for ATmega16

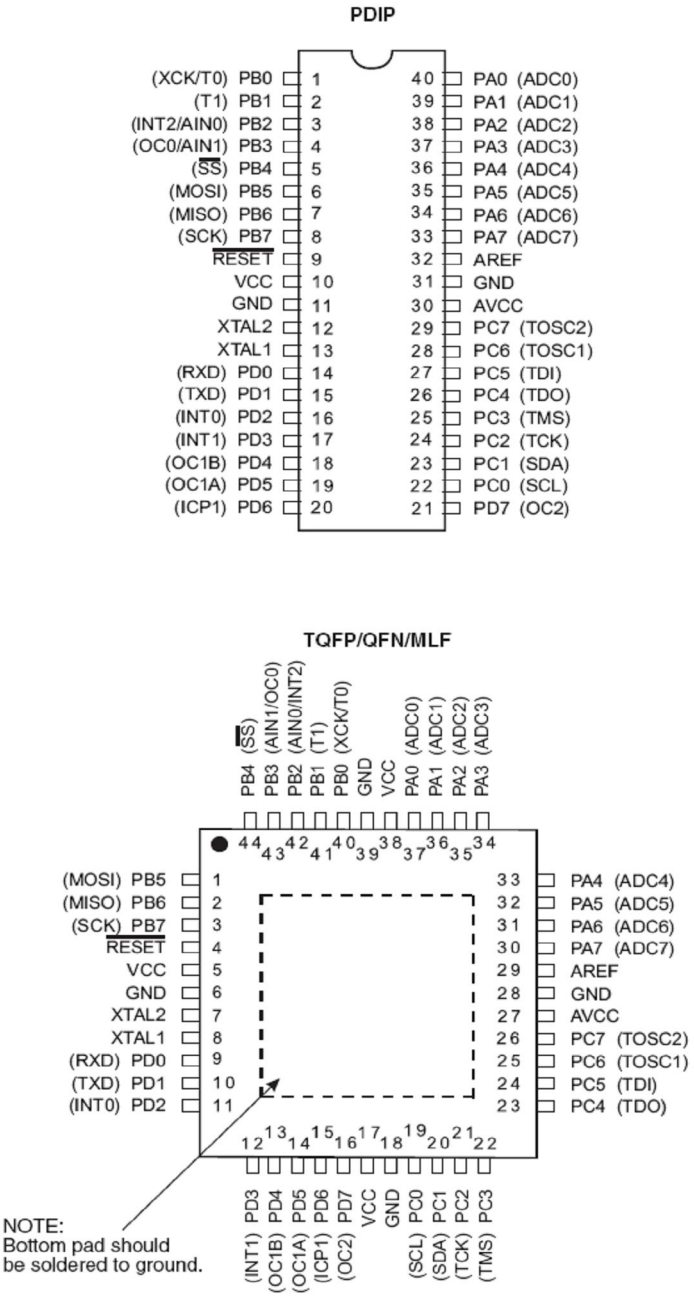
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
  - Active: 1.1 mA
  - Idle Mode: 0.35 mA
  - Power-down Mode: < 1 Ma

{2}

Pin Configurations

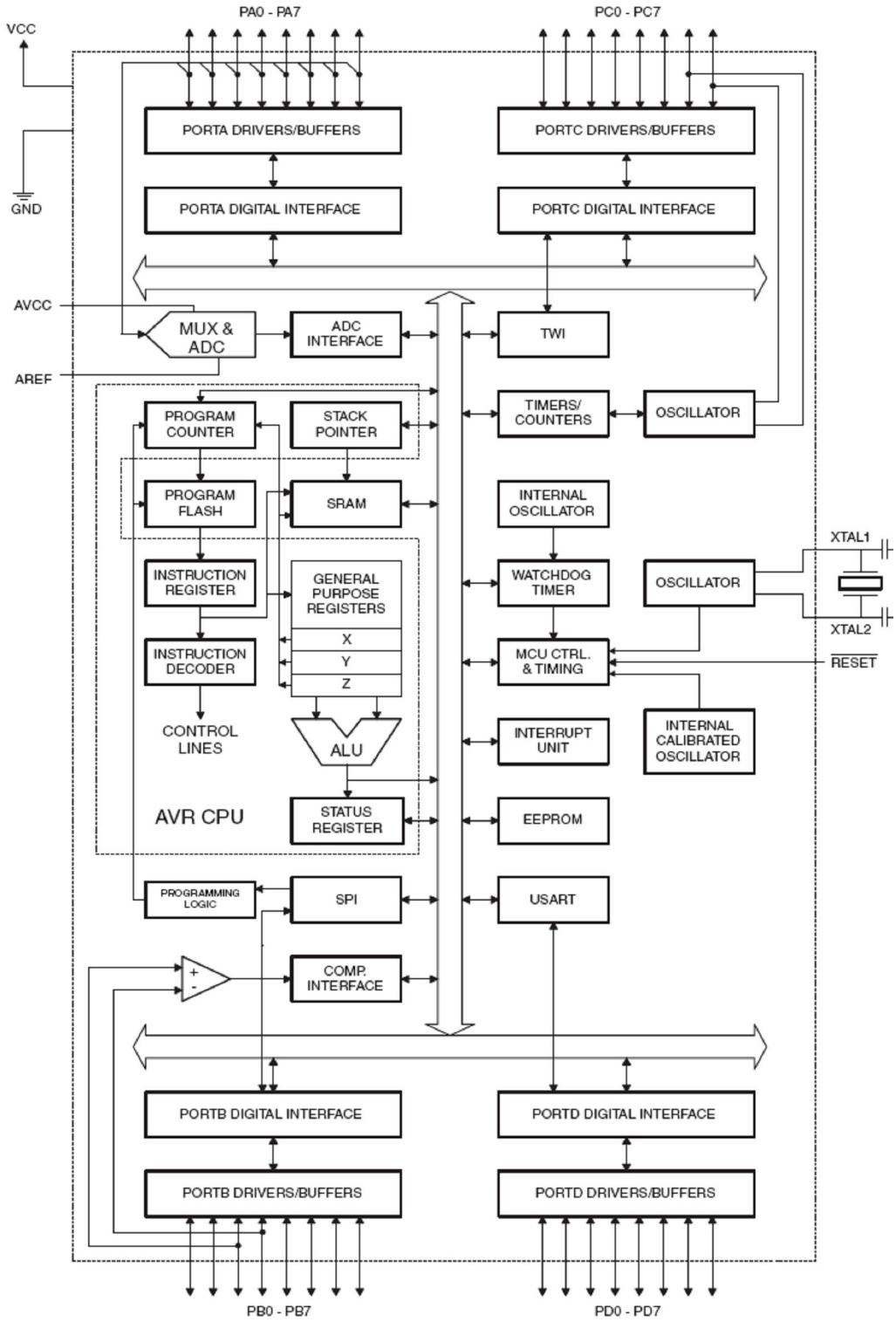
As AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. {2}

Figure 1. Pinout ATmega16



Block Diagram

Figure 2. Block Diagram



MEMORY

This section describes the different memories in the ATmega16. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega16 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

SRAM Data Memory

The first 96 locations address the Register File and I/O Memory, and the next 1024 locations address the internal data SRAM. The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment.

EEPROM Data Memory

The ATmega16 contains 512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

In-System Reprogrammable Flash Program Memory

The ATmega16 contains 16K bytes On-chip In-System reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 8K x 16. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section. The Flash memory has an endurance of at least 10,000 write/erase cycles. {2}

TIMERS/COUNTERS

CPU Clock – clkCPU

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

I/O Clock – clkI/O

The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted.

Flash Clock – clkFLASH

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

Asynchronous Timer

Clock – clkASY

The Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external 32 kHz clock crystal. The dedicated clock domain allows using this Timer/Counter as a real-time counter even when the device is in sleep mode.

ADC Clock – clkADC

The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results. {2}

POWER FEATURES

- Idle Mode
- ADC Noise
- Reduction Mode
- Power-down Mode
- Power-save Mode
- Standby Mode
- Extended StandbyMode

CONCLUSION

By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications. The ATmega16 AVR is supported with a full suite of program and system development tools including C compilers, macro assemblers, program debugger / simulators, in-circuit emulators, and evaluation kits.

Amtel ATmega16 microcontroller is suitable for use in a very sensitive system like satellite. All the features are good enough and if used with suitable and efficient software will provide the required real-time, bug and error free stable self sufficient system.

TERMINOLOGY

EMBEDDED SYSTEM

An **embedded system** is a computer system designed to perform one or a few dedicated functions,[1] often with real-time computing constraints. It is *embedded* as part of a complete device often including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, is designed to be flexible and to meet a wide range of an end-user's needs. Embedded systems control many of the common devices in use today.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. {8}

RISC/CISC

CISC – Complex Instruction Set Computer
One of the primary goal of the CPU designers was to provide more and more instructions in the instruction set of a CPU to ensure that CPU supports more function directly.Ofcourse every additional instruction in the instruction set of the CPU requires the necessary hardware circuitary to handle that instruction adding more circuitary hence inc the complicity of he CPU.

RISC – Reduced Instruction Set Computer.
In 1980s, some CPU designers realised that many instructions supported by CISC-based CPU are rarely used.Hence, an idea involved that the design complexity of a CPU can be reduced greatly by implementing the bare minimum and the frequently used instructions ones in the hardware circuitary (diff logic gates) of the CPU substituting the software for it. These systems use less power and are faster.

INSTRUCTION CODE

Every CPU has a built in ability to execute a set of machine instructions called its instruction code like to add, sub, mul ect. Most

CPUs has 200 or more instruction code .The machine language supported by the CPU is based on the set of instuctions supported by the CPU in the instruction code.
Ref-Chapter7 Computer Fundamentals Pradeep K Sinha

ARM

The **ARM** is a 32-bit reduced instruction set computer (RISC) instruction set architecture (ISA) developed by ARM Limited. It was known as the **Advanced RISC Machine**, and before that as the **Acorn RISC Machine**. The ARM architecture is the most widely used 32-bit ISA.[1][2] It is implemented in many microprocessors and microcontrollers for embedded systems. Because many ARM processors consume little power, they are dominant in the mobile electronics market, where low-power consumption is a critical design goal.
As of 2007, about 98 percent of the more than a billion mobile phones sold each year use at least one ARM processor {7}

HARVARD ARCHITECTURE/Von Neumann architecture

The original Harvard architecture computer, the Harvard Mark I, employed entirely separate memory systems to store instructions and data. The CPU fetched the next instruction and loaded or stored data simultaneously and independently. This is by contrast with a Von Neumann architecture computer, in which both instructions and data are stored in the same memory system and (without the complexity of a cache) must be accessed in turn. The physical separation of instruction and data memory is sometimes held to be the distinguishing feature of modern Harvard architecture computers. With Microcontrollers (entire computer systems integrated onto single chips), the use of different memory technologies for instructions (e.g. Flash memory) and data (typically read/write memory) in von Neumann machines is becoming popular. The true distinction of a Harvard machine is that instruction and data memory occupy different address spaces. In other words, a memory address does not uniquely identify a storage location (as it does in a von Neumann machine); you also need to know the memory space (instruction or data) to which the address belongs. {9}

8086 MICROPROCESSOR

8086 includes family of microprocessors from INTEL which includes commonly used PENTIUM series and others. These processors uses same basic ARCHITECTURE as well as BACKWARD COMPATIBILTY with 8086 so instruction code, registers are the same with some added features.

BUSSES

These are cables used to carry data from different parts of motherboard or some embedded system. Like from CPU to RAM ect . Different types of common bussses includes Data Bus, Address Bus, PCI bus ect.

TPU

Time Processing Unit or TPU for short is a sophisticated timer. In addition to counting down, the TPU can detect input events, generate output events, and perform other useful operations.

MICROCONTROLLERS

Microcontroller (also microcontroller unit, MCU or µC) is a small computer on a single integrated circuit consisting of a relatively simple CPU combined with support functions such as a crystal oscillator, timers, watchdog, serial and analog I/O etc. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM.Microcontrollers are designed for small or dedicated applicationsThus, in contrast to the microprocessors used in personal computers and other high-performance or general purpose applications, simplicity is emphasized. They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. {10}

Interrupt Latency

When an electronic device causes an interrupt, the intermediate results (registers) have to be saved before the software responsible for handling the interrupt can run. They must also be restored after that software is finished. If there are more registers, this saving and restoring process takes more time, increasing the latency. Ways to reduce such context/restore latency include having relatively few registers in their central processing units (undesirable because it slows down most non-interrupt processing substantially), or at least not having hardware save them all (hoping that the software doesn't then need to compensate by saving the rest "manually"). Another technique involves spending silicon gates on "shadow registers": one or more duplicate registers used only by the interrupt software, perhaps supporting a dedicated stack.

Firmware

Firmware is software in the microcontrollers or microprocessors that are substitute for hardware eg that there is a fn that first add then subtract and the multiply, to get such a function we have to add new gates in the CPU or that we could have a softwate as a instruction code in the ROM of CPU that is included in its firmware.

DIP-Dual-in line package also known as DIL package

In microelectronics, a **dual in-line package (DIP)**, sometimes called a **DIL** package, is an electronic device package with a rectangular housing and two parallel rows of electrical connecting pins. The pins are all parallel, point downward, and extend past the bottom plane of the package at least enough to be through-hole mounted to a printed circuit board (PCB), i.e. to pass through holes on the PCB and be soldered on the other side. DIP is also sometimes considered to stand for *dual in-line pin*, in which case the phrase "DIP package" is non-redundant. {11}

PROM

There are two types ROM 1.Manufacturar programmed 2.User Programmed. A manufacturer programmed ROM is one in which data is BURNT by the manufacturer eg system boot programme.A user programmed ROM is one in which user can load and store read-only programmes and data eg micro programmes.

EPROM

Erasable pogrammable read only memory

In EPROM it is possible to erase information stored in EPROM chip and the chip can be reprogrammed.R&D personal use the micro programmes to test the efficiency of a computer system with new programmes

Two types … … …

1. UVEPROM(Ultra voilet EPROM) – in this stored information is erased by exposing the chip to ultra voilet light
2. EEPROM - *Electrically erasable programmable read only memory*

In this the stored information is erased by using high voltage electric pulses.

SRAM

Static random access memory is in contrast to DRAM(dynamic RAM) . Has following characteristics

- 1 It does not need to be periodically refreshed as dynamic RAM
- 2 Exhibits **Data reminance (but still is volatile in nature)**
- **3 Expensive, faster and less power hungry which makes it ideal for embedded systems**
- 4 Less dense than DRAM not to be user for high capacity applications

REGISTERS

In computer architecture, a **processor register** is a small amount of storage available on the CPU whose contents can be accessed more quickly than storage available elsewhere. Most, but not all, modern computers adopt the so-called load-store architecture. Under this paradigm data is shuffled from subordinated memory- be it L# cache or RAM- into registers, crunched therein by running

instructions from the instruction set, then transferred out. A common property of computer programs is locality of reference: the same values are often accessed repeatedly; and holding these frequently used values in registers improves program execution performance.

Some common regisers are
1. Memory Address register
2. Memory Buffer Register
3. Instruction register
4. I/O register
ect {12}

Wikipedia- processor registers

PULL UP REGISTERS

These are used in electronic logic circuits to ensure that inputs to logic system settle at expected logic levels if external devices are disconnected. It is required by **I2C.**

ISP

In-system programming is the ability to be programmed while installed in a complete system rather than require the chip to be programmed prior to installing it into the system.
Communication with the programmer is via a serial protocol. Most use a variant of JTAG protocol, other may use propietary protocol or protocol defined by older standards.
wiki

LOCK BITS

It is a technology in Atmel AVRs which when activated locks the bits so that no one can steal the original code working in the flash memory of the embedded system.

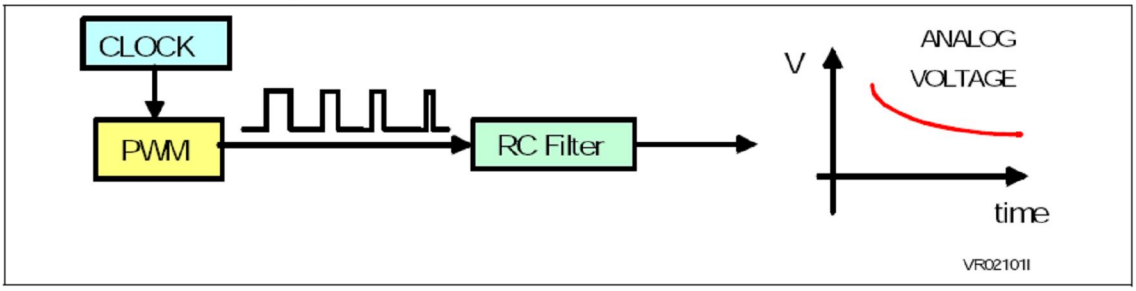
JTAG

It is a protocol used between programmer and embedded system having insystem protocol.
Refer ISP.

PWM

Pulse Width Modulation (block makes it possible for the CPU to control power converters, resistive loads, motors, etc., without using lots of CPU resources in tight timer loops. Often used as a digital-to-analog conversion technique. A pulse train is generated and regulated with a low-pass filter to generate a voltage proportional to the duty cycle.

Figure 10. PWM Principle



Pulse Accumulator

A pulse accumulator is an event counter. Each pulse increments the pulse accumulator register, recording the number of times this event has occurred.

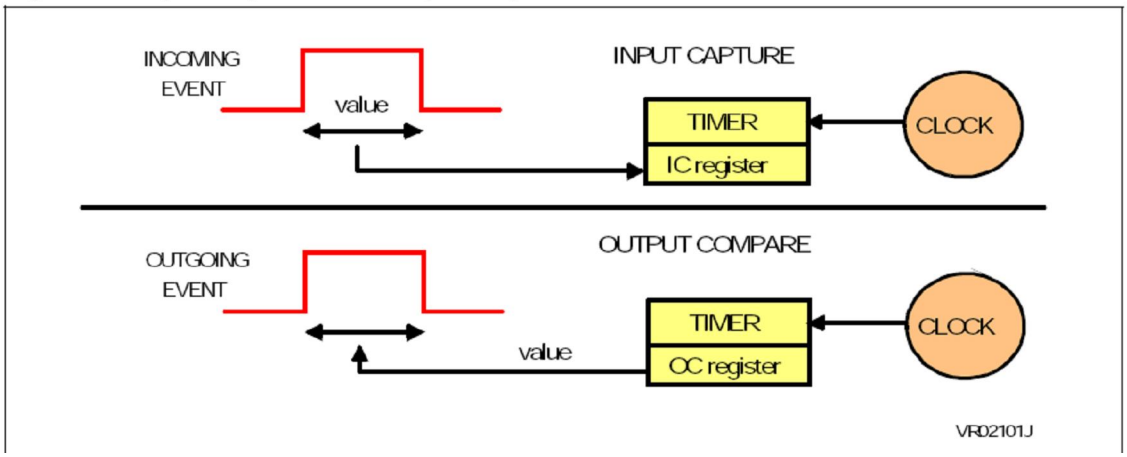
Input Capture

Input Capture can measure external frequencies or time intervals by copying the value from a free running timer into the input capture register when an external event occurs.

Output Compare

Output Compare can time an external event by sending a value stored inside the output compare register.

Figure 11. Input Capture and Output Capture Timer Functions

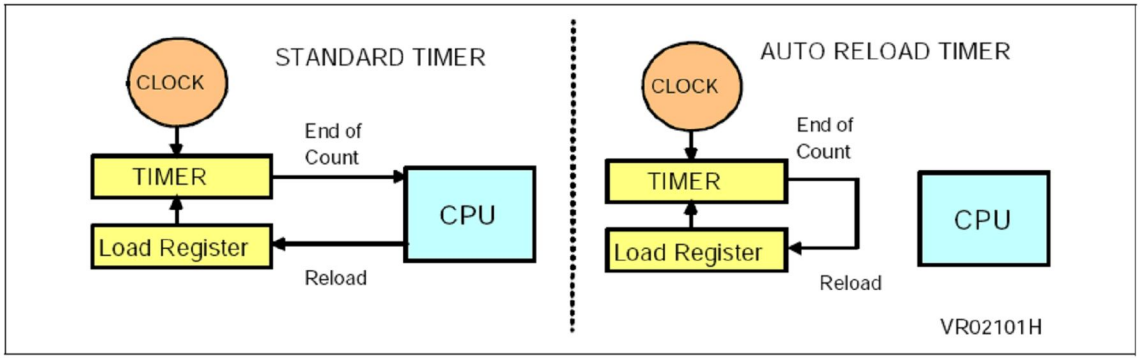


Watchdog timer

A watchdog timer provides a means of graceful recovery from a system problem. This could be a program that goes into an endless loop, or a hardware problem that prevents the program from operating correctly. If the program fails to reset the watchdog at some predetermined interval, a hardware reset will be initiated. The bug may still exist, but at least the system has a way to recover. This is especially useful for unattended systems.

Auto Reload Timer Compared to a standard timer, this timer automatically reloads its counting value when the count is over, therefore sparing a waste of CPU resource.

Figure 9. Standard Timer and Auto-Reload Timer Principle



Comparator

One or more standard comparators can sometimes be placed on a microcontroller die. These comparators operate much like standard comparators however the input and output signals are available on the microcontroller bus.

BUS –I2C Inter-Integrated Circuit Bus

The I2C bus is a simple 2 wire serial interface developed by Philips. It was developed for 8 bit applications and is widely used in consumer electronics, automotive and industrial applications. In addition to microcontrollers, several peripherals also exist that support the I2C bus. The I2C bus is a two line, multi-master, multi-slave network interface with collision detection. Up to 128 devices can exist on the network and they can be spread out over 10 meters. Each node (microcontroller or peripheral) may initiate a message, and then transmit or receive data. The two lines of the network consist of the serial data line and the serial clock line. Each node on the network has a unique address which accompanies any message passed between nodes. Since only 2 wires are needed, it is easy to interconnect a number of devices.

UART

A UART (Universal Asynchronous Receiver Transmitter) is a serial port adapter for asynchronous serial communications. It makes it possible to receive and transmit data over a serial line with very little load on the CPU.

USART

A USART (Universal Synchronous / Asynchronous Receiver Transmitter) is a serial port adapter for either asynchronous or synchronous serial communications. Communications using a USART are typically much faster (as much as 16 times) than with a UART.

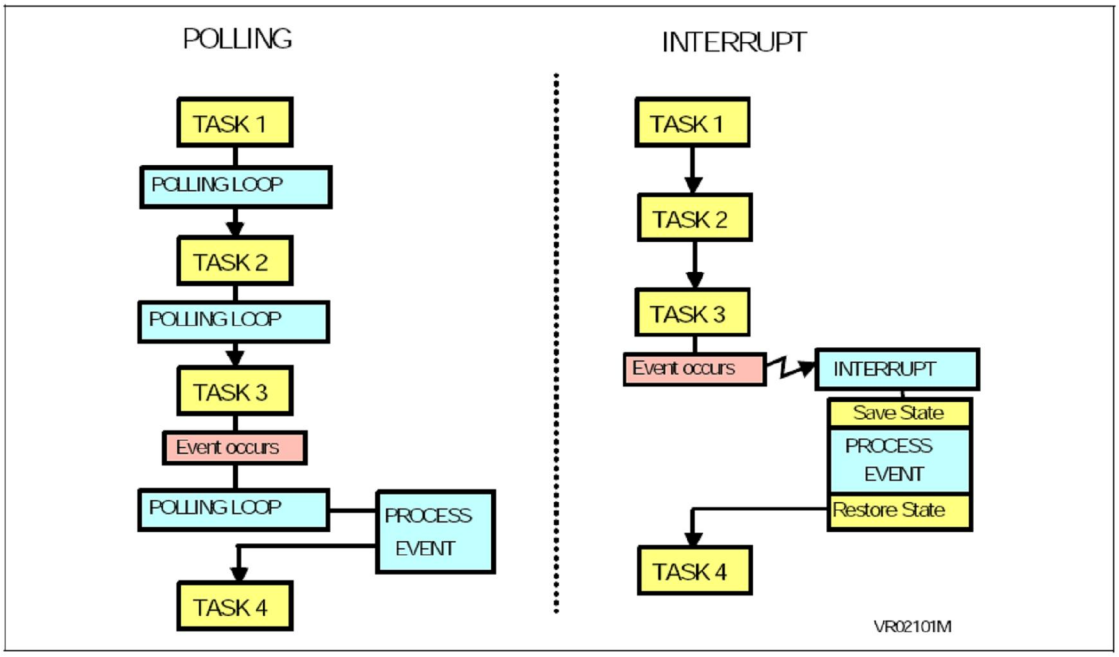
POLLING

Polling is what you have to do if your microcontroller does not have interrupts or if what you want to do is not time critical. It is a software technique whereby the controller continually asks a peripheral if it needs servicing. The peripheral sets a flag when it has data ready for transferring to the controller, which the controller notices on its next poll. Several peripherals can be polled in succession, with the controller jumping to different software routines, depending on which flags have been set.

INTERRUPT

Rather than have the microcontroller continually polling - that is, asking peripherals (timers / UARTS / A/Ds / external components) whether they have any data available (and finding most of the time they do not), a more efficient method is to have the peripherals tell the controller when they have data ready. The controller can be carrying out its normal function, only responding to peripherals when there is data to respond to. On receipt of an interrupt, the controller suspends its current operation, identifies the interrupting peripheral, then jumps to the appropriate interrupt service routine. The advantage of interrupts, compared with polling, is the speed of response to external events and reduced software overhead (of continually asking peripherals if they have any data ready). {12}

Figure 14. Polling versus Interrupt



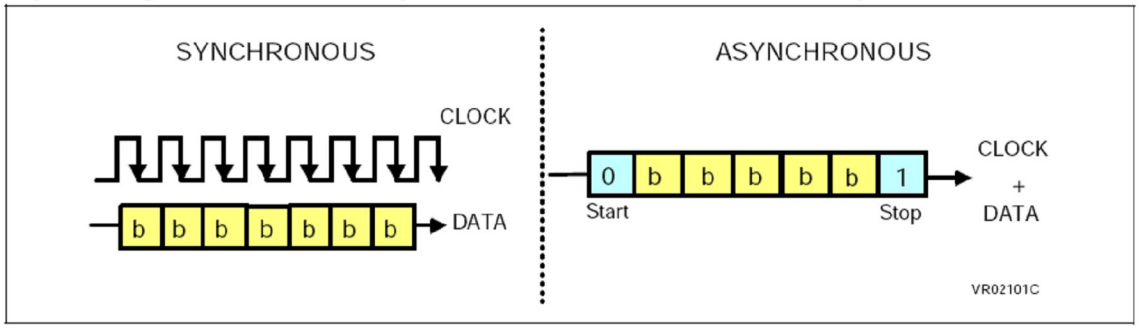
FLASH MEMORY

Flash is an electrically erasable and programmable memory. It can be used instead of ROM to store program instructions and data. A key parameter of Flash memory is its endurance or the number of times it can be erased or reprogrammed. Depending on the technology used, flash endurance can be either 100 or 300,000 program/erase cycles.

SERIAL INTERFACE

Serial interface are used to exchange data with the external world. Many microcontrollers have both synchronous and **synchronous** communications peripherals built in. Usually, an asynchronous interface is called a serial communication interface (**SCI** or **UART**) while the synchronous interface is called a serial peripheral interface (**SPI**). A typical SCI application is to connect a PC for debugging purpose while a typical SPI application is to connect an external EEPROM. A synchronous bus includes a separate line for the clock signal which simplifies the transmitter and receiver but is more susceptible to noise when used over long distances. *With an asynchronous bus the transmitter and receiver clocks are independent, and a resynchronization is performed for each byte at the start bit.*

Figure 3. Synchronous and Asynchronous Communication Principles



Parallel Device

Device that communicates with binary data by sending the bits that represent each character simultaneously along a set of seprate data lines unlike a serial device.

Real Time

Generally used to describe system that guarantee a response to an external event within a given time.

REAL-TIME SYSTEM

Programs that respond to events in the world as they happen. Eg- an automatic pilot program in an aircraft must respond instantaneously in order to correct directions from its course.

SCALABLE FREQUENCY

It is a feature in power efficient microcontrollers by which it can slow down and consume less power while still awake listening to interrupts and doing light processing when requirement is not very intense thus lowering its speed.

BROWNOUT DETECTION

Monitors supply voltage during operation, is nothing more than a COMPARATOR.

CAN CONTROLLER

Controlled area network (CAN-BUS) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other within a vehicle without a host computer.

CPU CLOCK

Operations are usually synchronized by a built in electronic clock that emits millions of regularly spaced electric pulses per second known as clock cycle. Instructions are fetched decoded and executed at proper intervals at intervals timed by a specific no of clock cycles.

CLOCK SPEED

Number of pulses produced per second. One clock is the time it takes to perform one operation such as moving a byte of data from one memory location to another.

OCD

On chip debugging

MISP

Millions of Instructions per second

Search keywords and sites visited explicitly

In the vast internet there is lots of information
Searching in this internet is now not at all easy so by this someone may use only relevant searches..
This can be used to duplicate the entire process of learning and making the report.

(g) – represents searches in google
(w) represents searches in wikipedia
(b) – represent searches in bing

1. Text which are **bold** are relevant
2. Text which are both **bold** and *italics* are most relevant
3. Text items which are none are less or of no relevance

(b) ana university + cubesat
(g) nokia 6600 + processor
(g) arm microcontroller
(w) ATmel
(w) microcontroller
(g) cubesat + onboardcomputer
(w) AVR 8bit RISC
(w) Real time os
(w) Embedded linux
(w) ATmega
(g) AAUSAT
(g) DTUSAT
(g) ANUSAT
(g) jugnu + iit kanpur
WikiBooks Embedded systems

www.kanda.com/products/atmel/atmega16.html
www.evilbitz.com – (related to jugnu IIT KANPUR sat less relevant)
www.avrfreaks.com
www.captain.at/electronics/atmel-programmer
www.atmel.com

References

1. http://www.atmel.com/dyn/products/Product_card.asp?part_id=2010 – PRODUCT CARD

2. http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf - PDF - ATmega16

3. On Board Computer for picoSatellite – PDF document

4. http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf - QUICK REFRENCE GUIDE

5. http://en.wikipedia.org/wiki/Atmel_AVR - ATMEL_AVR

6. http://www.avrfreaks.net/index.php?module=Freaks%20Devices&func=displayDev&objectid=56 – AVR FREAKS

7. http://en.wikipedia.org/wiki/Arm_architecture - ARM ARCHITECTURE

8. http://en.wikipedia.org/wiki/Embedded_system - EMBEDDED SYSTEM

9. http://en.wikipedia.org/wiki/Modified_Harvard_architecture - HARVARD ARCHITECTURE

10. http://en.wikipedia.org/wiki/Microcontroller - MICROCONTROLLER

11. http://en.wikipedia.org/wiki/Dual_in-line_package DUAL_IN_LINE PACKAGE

12. http://en.wikipedia.org/wiki/Processor_register - REGISTER

13. http://www.cubesat.auc.dk/dokumenter/software.pdf - AAU CUBESAT

14. www.st.com/stonline/books/pdf/docs/4966.pdf - MICROCONTROLLER MADE EASY

15. http://www.futurlec.com/Atmel/ATMEGA16.shtml

16.www.kanda.com/products/atmel/atmega16.html

17.http://en.wikipedia.org/wiki/Real_time_os - REAL TIME OS

18.http://www.aero.iitb.ac.in/pratham/prodoc.php - PRATHAM IIT BOMBAY

19.http://www.kanda.com/largeimage.php?id=715 – IMAGE

20.WikiBooks - EMBEDDED SYSTEMS

21.COMPUTER FUNDAMENTALS - PK SINHA