VTI1E2 – APLIKASI MIKROKONTROLER dan ANTARMUKA©

SEMESTER GANJIL – KURIKULUM 2020

Denny Darlis S.Si., M.T. - 13770026

Program Studi D3 Teknologi Telekomunikasi

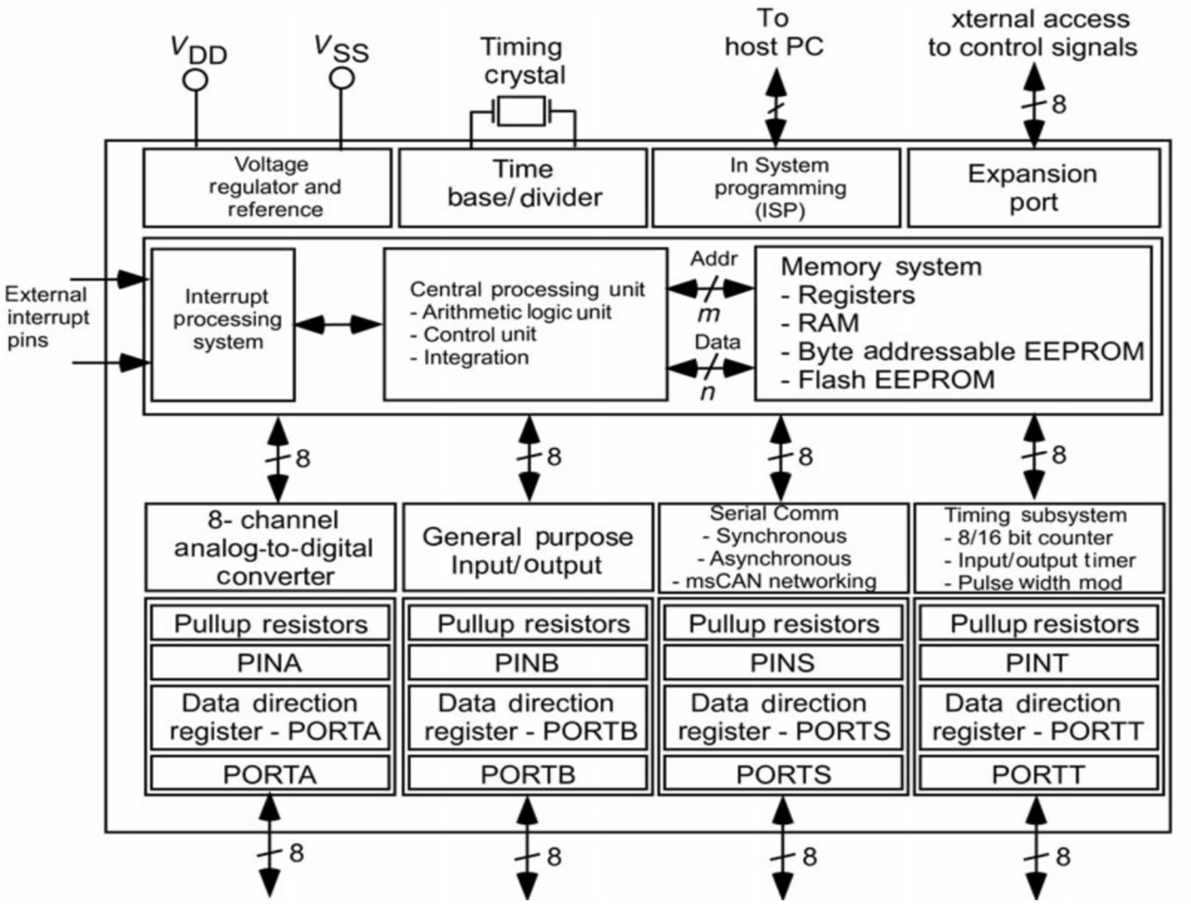
Fakultas Ilmu Terapan - Universitas Telkom

Ikhtisar

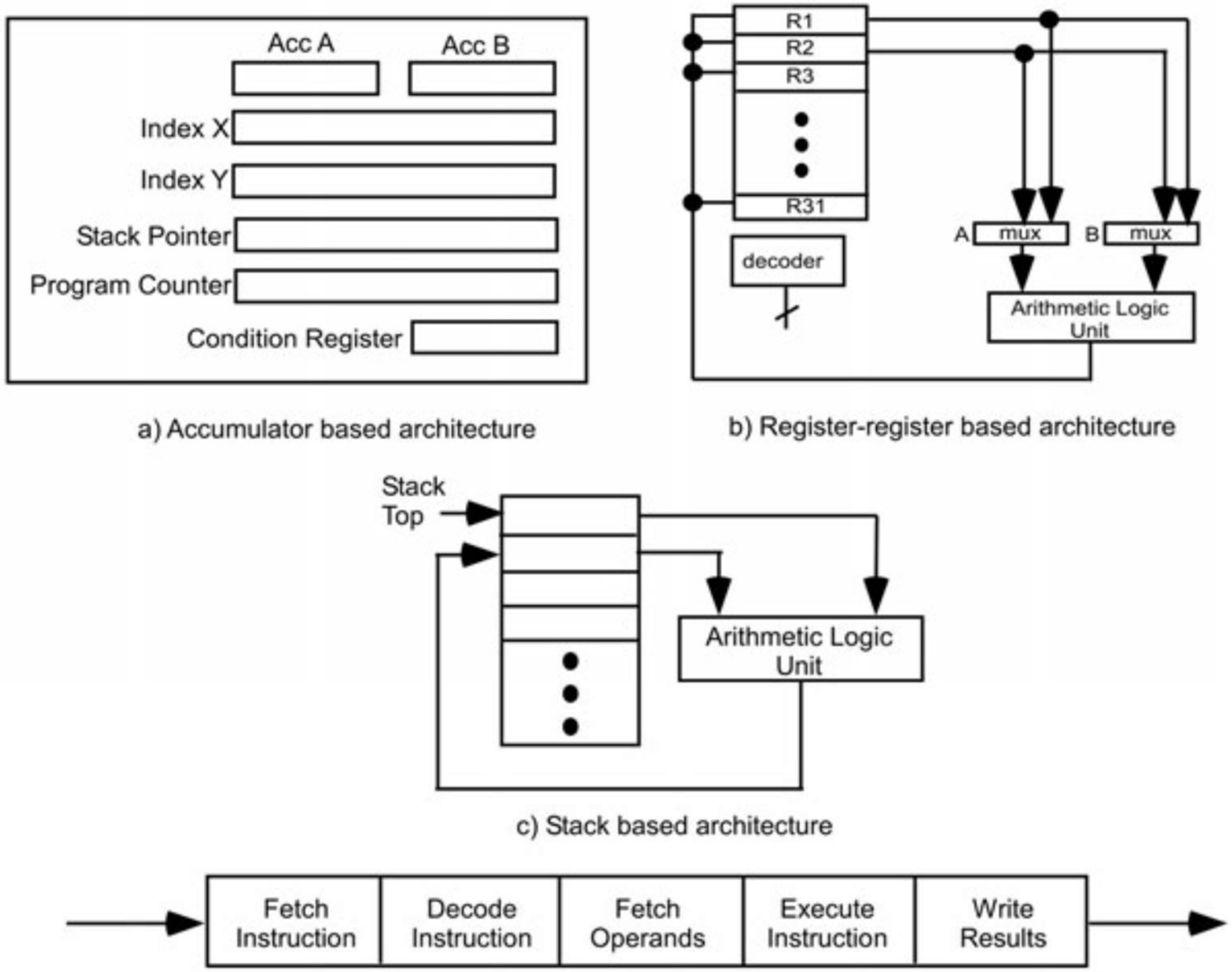
- Overview
- CPU Architecture
- Implementation and Testing Tools
- Software Development Process
- RISC versus CISC Instruction Set
- REGISTER SET
- BUS STRUCTURE
- MEMORY
- TIME BASE
- Timing Subsystem
- PORT SYSTEMS
- ANALOG-TO-DIGITAL CONVERTERS
- COMMUNICATION SYSTEMS
- INTERRUPT SYSTEM
- SPEED
- CUTTING EDGE TECHNOLOGY

Overview

- Most microcontrollers are equipped with the subsystems shown in the figure.
- The ports are used to provide access to the microcontroller to the outside world. Typically,ports are bidirectional and also have alternate functions such as analog-to-digital conversion, serial communications, and a flexible timing system.
- Microcontrollers are also equipped with a complement of different memory components.
- Normal microcontroller operation can be interrupted by an external event using the external interrupt pins. This allows the microcontroller to respond to high priority events.
- The microcontroller is programmed via In System Programming (ISP) features using a host personal computer.
- The time base for the microcontroller is provided by an external crystal oscillator or resonator



CPU Architecture



a) Accumulator based architecture

b) Register-register based architecture

c) Stack based architecture

d) Pipeline architecture

- *Accumulator-based architecture:* In an accumulator-based architecture (a), instructions begin and end in specially designated registers called accumulators (A and B). Typically, an operation is performed in which one operand is found in an accumulator and the other is fetched from memory. The result is then placed in the accumulator. This architecture tends to run slower than the other two configurations since operands must be continually fetched from memory. Typically, the memory runs at a slower speed than the main processor so the processor must slow down to accommodate an operand fetch from memory. An accumulator-based architecture has the ability to execute fairly complicated instructions. The architecture may also be modified such that one operand is located in a register and the other is found in memory.

- *Register-based architecture:* In a register-based architecture, both operands are stored in registers that are typically collocated with the central processing unit. The result of a given operation is also stored in a register. Since the CPU and the registers operate at the same speed, the processor does not have to slow down to read or write operands. Register contents are read from and written to memory using a background operation.
- *Stack-based architecture:* In a stack-based architecture, both operands and the operation to be performed are stored on the stack. The result is then placed back on the stack. The stack may be based in dedicated registers or may be a special portion of random access memory.

- *Pipeline architecture:* A pipeline-based microcontroller architecture has the general form illustrated in Figure (d). The architecture consists of separate hardware subsystems called stages to fetch an instruction from memory, decode the instruction, fetch instruction operands from memory or registers, execute the instruction, and then write the results back to memory. Each stage is simultaneously processing a different instruction such that the overall result is that an instruction completes execution on every clock cycle. For example, in a five-stage pipeline, five instructions are simultaneously being processed through the pipeline each at a different stage. Typically, instructions in a pipeline processing system are simple instructions easily implemented within a single stage. More complex instructions are built up from these small instruction building blocks.

Implementation and Testing Tools

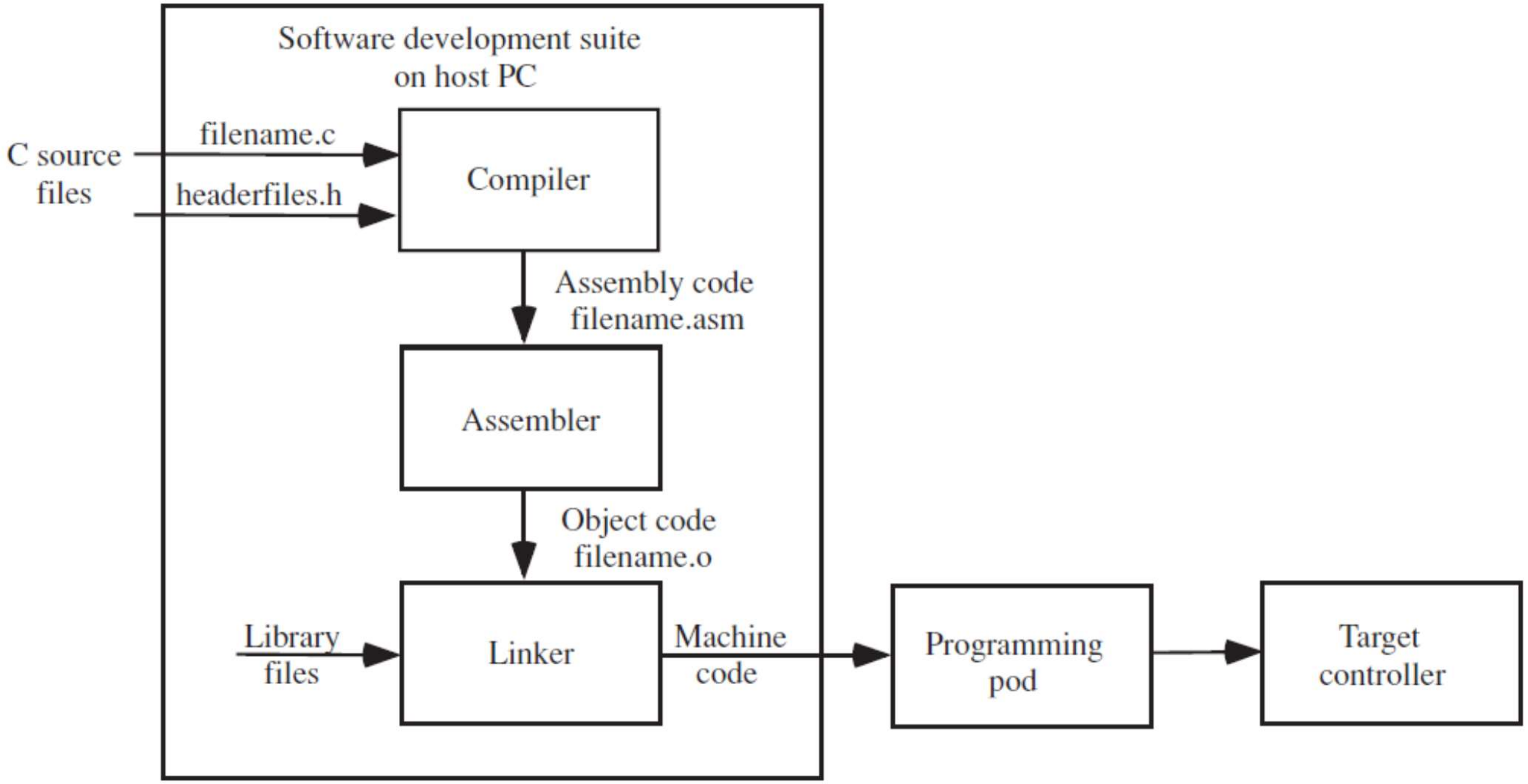
To aid in the design and development process, there are a number of support tools commercially available to make the process easier. We provide a brief definition of some of these design tools.

- *Assembler:* An assembler converts a microcontroller control algorithm written in assembly language to processor specific machine code.
- *Compiler:* A compiler translates a high-level language such as C first into assembly language and then into processor specific machine code.
- *Emulator:* An emulator is a software program that simulates the operation (emulates) of a hardware microcontroller. The emulator is used frequently during system development and testing.

- *Programmer:* Usually microcontroller algorithms are programmed in a host personal computer (PC) using an assembler or compiler. The resulting machine code is downloaded from the host PC to the microcontroller via a hardware programmer. For some microcontrollers this is simply a serial RS-232 compatible cable. For other microcontrollers it consists of a programming pod which converts the machine code to appropriate programming signals to program the memory resident within the microcontroller.
- *Logic analyzer:* A logic analyzer is a piece of test equipment that allows the simultaneous viewing of multiple channels of signals from a microcontroller. It is especially useful for examining the timing relationships between related microcontroller signals.

- *Oscilloscope:* An oscilloscope normally can display two to four channels of data simultaneously. It is especially useful for examining analog signals.
- *In system programming (ISP):* Some microcontrollers are equipped with ISP capability. This means that a program can be downloaded into the microcontroller while the microcontroller is resident within its systems.

Software Development Process

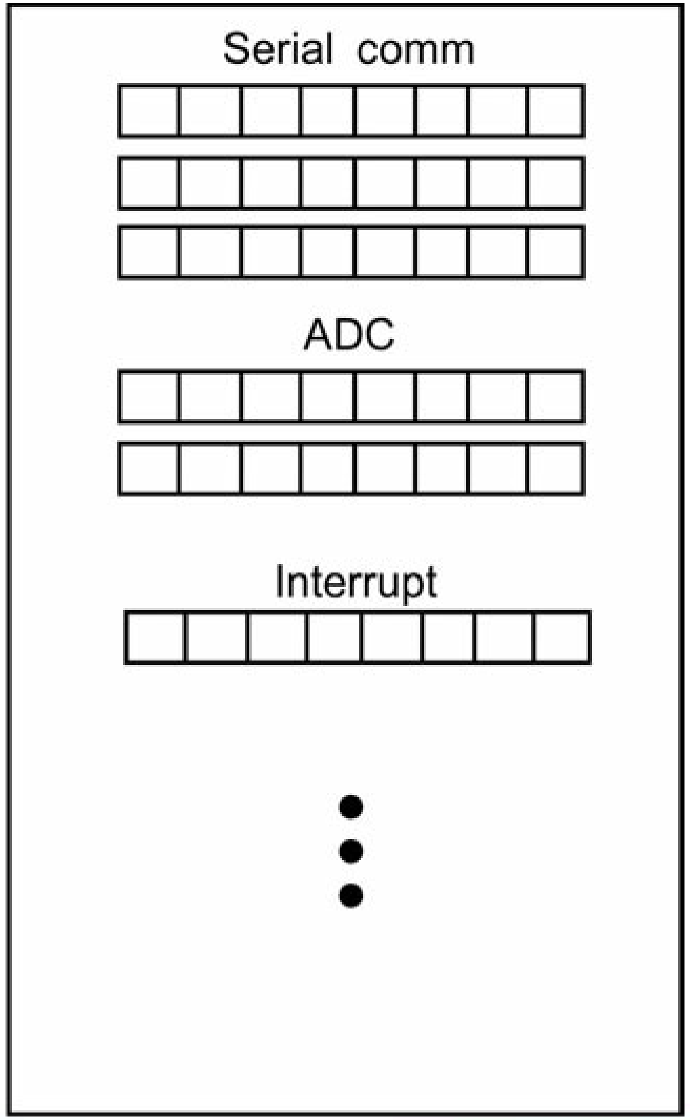


RISC versus CISC Instruction Set

- Reduced Instruction Set Computer (RISC) processor as its name implies has a complement of simple building block instructions. More complex instructions are built up from the basic instructions in the RISC processor. RISC-based instruction architectures lend themselves to systems with less complex CPU architectures.
- CISC-based architecture has a complement of fuller feature, more complex instructions than the RISC-based architecture.
- It is difficult to predict whether a given program will be more efficiently coded with a RISC- or CISC-based instruction set. It largely depends on how well the specific algorithm matches the feature set of a given processor.
- As a system designer you need to be intimately familiar with the hardware and software architecture of a given microcontroller, particularly if you will be coding the system using an assembly language.
- However, if you will be programming using a high-level language such as C, knowledge of some of the lower level architectural details are not required as long as you have a thorough understanding of the microcontroller subsystems at the register level.

REGISTER SET

- Most microcontrollers have a complement of registers designated as the register set. The register set is the interface between you the user and the different subsystems aboard the microcontroller.
- Each register consists of a collection of flip-flops.
- Each flip-flop can either be set to a logic one or logic zero. Each flip-flop can be viewed as a software configurable switch. Sending a logic one to the flip-flop asserts or turns on the switch while sending a logic zero to the flip-flop de-asserts or turns the flip-flop off.



REGISTER SET

- The flip-flops are categorized by subsystem as shown in Figure. Usually all registers associated with a given subsystem are grouped together. To configure a specific subsystem the system designer will determine the appropriate setting for each bit within the register. The function of each register and each register bit are carefully defined in the specific microcontroller's documentation.
- All bits within a register are programmed simultaneously by setting the name of the register (as defined in the compiler definitions) to the desired bit pattern.
- For example, to set a register called SCIBaudRate to the binary value 1010 1110 the following command may be used:

  SCIBaudRate = 0xAE; //0x is used to designate a hexadecimal number
- It is important to note that compilers provide a complement of header files. These header files contain the "personality data" for a given microcontroller. Specifically, they provide a link between the name of a specific register used within a program and its location within the microcontroller.

BUS STRUCTURE

- Different subsystems within a microcontroller are connected with several different buses. A bus is a collection of parallel conductors that have a similar function. Most microcontrollers are equipped with an address bus, a data bus, and a control bus.
- The **address bus** provides a connection between the central processing unit and the memory subsystem aboard the microcontroller. The number of conductors in the address bus sets the upper limit of memory locations that may be linearly addressed by the microcontroller.
- The first address in the memory subsystem will be all zeroes while the final address will be all logic ones. The number of individually addressable memory addresses may be determined by evaluating $2^{\textit{address lines}}$ *=addressable locations*.

BUS STRUCTURE

- For example, a microcontroller equipped with a 16-bit address bus is capable of addressing 65 536 (64 kB) separate locations. The first address in this memory space is (0000)16 while the last address in this space will be (*FFFF*)16.
- To expand the span of addressable memory locations, some microcontrollers employ a paged memory addressing scheme. In paged addressing, a memory system is subdivided into memory pages. Memory page length is usually some smaller block of memory such as a 4 kB page. Many 4 kB pages of memory can be assigned to the same 4,096 addresses in the linearly addressable memory space. To select a specific memory page for access additional addressing or select bits are required.

BUS STRUCTURE

- The **data bus** as its name implies is used to route parallel data about different subsystems within the microcontroller. Microcontrollers are commonly available with data bus widths of 4, 8, 16, or 32 bit.
- The width of the data path generally determines the size of a data argument that the microcontroller can process.
- For example, the largest unsigned integer that may be stored in a microcontroller with an eight-bit data path is 255. It should be emphasized that a 32-bit microcontroller is not a better microcontroller than its four-bit counterpart.
- Recall that the primary objective of the embedded system designer is to choose the most economical microcontroller that accomplishes the requirements of a specific design.
- For example, a four-bit microcontroller may be an ideal choice to host the control algorithm for the irrigation (water sprinkler) system used to water your lawn. However, a 32-bit processor may be required to host the control features required of a cellular phone.

BUS STRUCTURE

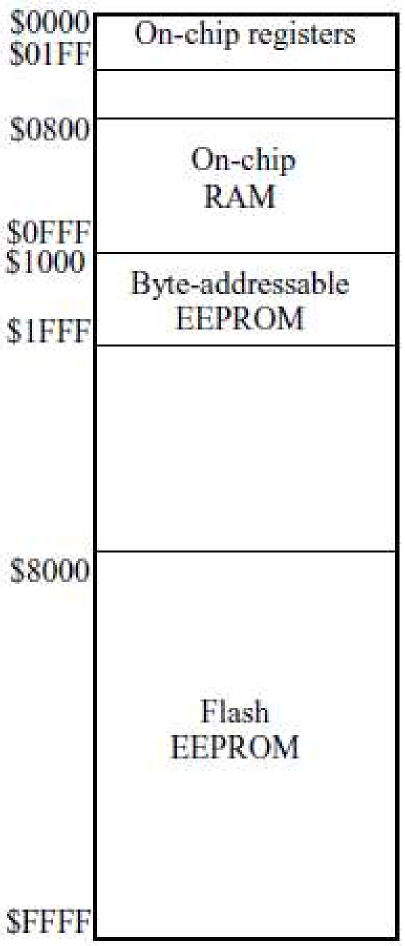
- Microcontrollers are equipped with paths to send and receive a collection of control signals designated as the **control bus**.
- These signal lines carry control signals to different subsystems throughout the microcontroller. Most of the control signals are internal to the microcontroller integrated circuit (chip). However, they are often conveniently provided at a microcontroller port so that external components may be added to the processor.
- Control signals are issued by the CPU in response to program instructions to insure the instruction is properly executed.

MEMORY

- As previously mentioned the number of uniquely addressable memory locations in a microcontroller is determined by the width of the address bus. This span of addressable memory usually contains several different types of memory including Static Random Access Memory (SRAM), byte-addressable Electrically Erasable Programmable Read Only Memory (EEPROM), and bulk programmable Flash EEPROM.
- To keep a track of the memory locations in use and the type of memory present within the system, a visual tool called a memory map is employed.
- The memory map provides the size in bytes of each memory component and its start and stop address within the memory system.
- A sample memory map is provided in Figure. Note that there are portions of the memory map not in use. These open spaces are provided for system expansion.

MEMORY

- The following memory components are shown in Figure and are commonly available in most microcontrollers:
- *RAM:*RAM memory is volatile. That is, if the microcontroller loses power, the contents of RAM memory are lost. It can be written to and read from during program execution.
- It is typically used during system development to store a program. Once development is complete, the completed program is stored in nonvolatile memory such as Flash EEPROM. During program execution, RAM is used to store global variables, support dynamic memory allocation of variables, and to provide a location for the stack.



$0000
$01FF
On-chip registers

$0800
$0FFF
On-chip
RAM

$1000
$1FFF
Byte-addressable
EEPROM

$8000
$FFFF
Flash
EEPROM

MEMORY

- *Byte-addressable EEPROM:*This type of memory is used to permanently store and recall variables during program execution. It is especially useful for logging system malfunctions and fault data during program execution. It is also useful for storing data that must be retained during a power failure but might need to be changed periodically.
- Examples where this type of memory is used are found in applications to store system parameters, electronic lock combinations, and automatic garage door electronic unlock sequences.
- *Flash EEPROM:* Bulk programmable Flash EEPROM is used to store programs. It can be erased and programmed as a whole. Some microcontroller systems provide a large complement of both RAM and Flash EEPROM. Therefore, a system program can be developed in RAM and then transferred to Flash EEPROM when complete.

MEMORY

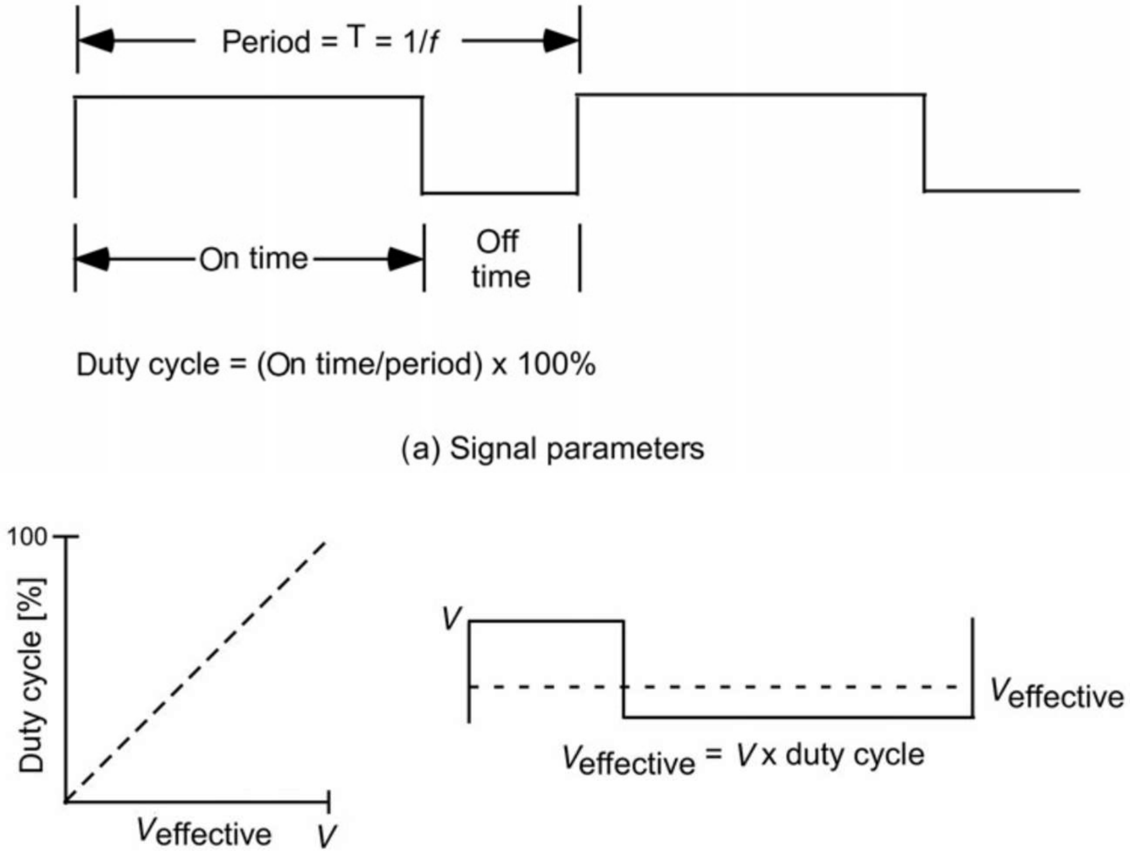
- Other microcontrollers provide only a large Flash EEPROM and a smaller RAM component.With this memory configuration, system development takes place in Flash EEPROM. Flash EEPROM is typically programmed using In System Programming (ISP) techniques. That is, a host PC is connected via a cable to a microcontroller while it is resident within its application circuit. The host PC downloads the program to the microcontroller.

TIME BASE

- As previously mentioned a microcontroller is a complex synchronous state machine that responds to program instructions in a predictable fetch-decode-execute sequence.
- The speed at which a microcontroller sequences through its actions is controlled by an external time base.
- The time base may be provided by an external quartz crystal, a programmable oscillator, a ceramic resonator, or an internal time base.
- Some microcontrollers are designed to operate at a set specific frequency such as 8 MHz while others have been flexibly designed to operate at a wide range of frequencies.
- Currently, microcontrollers are available that operate up to approximately 50 MHz. Frequently a microcontroller is employed to control a slow (relative to the microcontroller)mechanical system. In this type of application a slow time base may be required.
- A 32.768-kHz time base is frequently used in systems that track 24 h clock time. Some microcontrollers are equipped with an internal time base.

Timing Subsystem

- The main clock source is routed throughout the microcontroller to provide synchronicity to the subsystems. Most microcontrollers are also equipped with a timing subsystem.
- To better understand the features of this subsystem, we review common terminology associated with signal timing illustrated in Figure.
- *Frequency:* Signal frequency is the number of cycles per second completed by a repetitive signal. It is expressed in units of Hertz (Hz).



(a) Signal parameters

(b) Pulse width modulation (PWM) concepts

Timing Subsystem

- *Period:* The period is the time increment in seconds required for a repetitive signal to complete a single cycle. The period is the reciprocal of the frequency ($T = 1/ f$).
- *Duty cycle:* The duty cycle indicates the percentage of time for which the signal is active in a single period.
- *Pulse width modulation (PWM):*PWM signals are frequently used to control motor speed. The digital PWM signal is converted to an effectiveDC value by the mechanical inertia of the motor as well as the low pass filter characteristics of the inductance inherently present in a motor. Note in Figure how various effective DC voltages can be delivered to a load by simply adjusting the duty cycle of the digital PWM signal.

Applications

- Most microcontrollers are equipped with a multichannel timing system.
- The channels within the timing system may be configured to
  - Measure parameters of input signals such as period and duty cycle.
  - Generate precision output pulses or repetitive signals.
  - Count incoming pulses present in an input signal.
  - Generate PWM signals.

PORT SYSTEMS

- Microcontrollers are equipped with a series of ports to provide access to the world beyond the microcontroller. Frequently these ports are organized as eight-bit input/output ports as shown.
- Usually a port register is equipped with an accompanying data direction register.
- This is used to set the direction (input or output) for a given port pin.
- Although the ports are used by the input and output of digital signals, many have alternate functions such as analog-to-digital conversion, serial communication, and network interfacing.
- Even when equipped with alternate functions, often there are not enough external pins to provide access to all microcontroller features.

PORT SYSTEMS

- To alleviate this challenge, some microcontroller ports are equipped with time multiplexed capability. This simply means that the function of a given port alternates at a prescribed interval.
- For example, expansion ports provided to route the data and address ports outside the microcontroller may employ time multiplexed features.
- To deinterleave the time multiplexed port data, external data latches are required. A common latch used for this application is the 74HC573 (octal, three-state, D-type transparent latch).
- Expansion ports are also provided to route control signals from inside the microcontroller to external components.

ANALOG-TO-DIGITAL CONVERTERS

- Many microcontrollers are equipped with analog-to-digital conversion (ADC) subsystems.
- This subsystem converts continuously varying analog signals from the outside world into a binary representation suitable for use by the microcontroller.
- These converters commonly have 8–10-bit resolution.
- Therefore, a continuous signal is converted to a series of digital snapshots of the analog signal.
- As the system designer we must determine how often to initiate an ADC conversion for a given application.

COMMUNICATION SYSTEMS

- When we consider communication between two systems, the taxonomy consists of parallel and serial communications.
- Simply put, a parallel communication method utilizes multiple channels, bus wires, to send and receive multiple streams of data simultaneously, compared to a serial communication method where only a single stream of data is sent and received at a time.
- The most obvious advantage of a parallel communication method over a serial communication method, provided that the communication rate is equal, is the speed of data transfer.
- Using the multiple number of connections, the same amount of data can be sent and received quicker, proportional to the number of connections, than a single connection used in a serial communication method.

COMMUNICATION SYSTEMS

- The disadvantage of a parallel communication method is the hardware and software cost to enable the fast data transfer.
- Typically, parallel communication techniques are used for short distance communication within and outside of a microcontroller.
- For a long distance communication, a serial communication technique is used to send and receive data.

Serial Communications

- There are two different types of serial communications available: synchronous communication and asynchronous communication.
- The key challenge in serial communications is to maintain synchronization between the transmitter and the receiver.
- The asynchronous communication method uses a start and stop bit protocol to synchronize a transmitter and receiver.

COMMUNICATION SYSTEMS

- The start and stop synchronization bits are embedded in each transmitted signal byte.
- The advantage of the asynchronous communication is that it is inexpensive, but the disadvantage is that the data transmission rates are typically slower than a synchronous serial communication system due to its overhead (start and stop bits).
- Asynchronous serial communication subsystems are referred to as the UART (universal asynchronous receiver transmitter) or the SCI (serial communications interface).
- The synchronous serial communication uses a synchronized clock to send and receive each bit.

COMMUNICATION SYSTEMS

- The synchronous serial communication receiver is much more simple compared to an asynchronous serial communication receiver module.
- It may be viewed as a synchronous 16-bit shift register with an eight-bit half residing in the transmitter and the other eight-bit half residing in the receiver.
- The disadvantage of the synchronous serial communication compared to the asynchronous communication method is the task of synchronizing the transmitter and receiver clocks.
- This is usually accomplished with an additional clock line linking the transmitter and receiver.
- For a long distance communication, the synchronous serial communication technique is not recommended.
- For a short distance communication, the synchronous communication system can considerably improve the bit throughput over the asynchronous communication system. A synchronous serial communication system is referred to as the SPI (serial peripheral interface).

Terminology

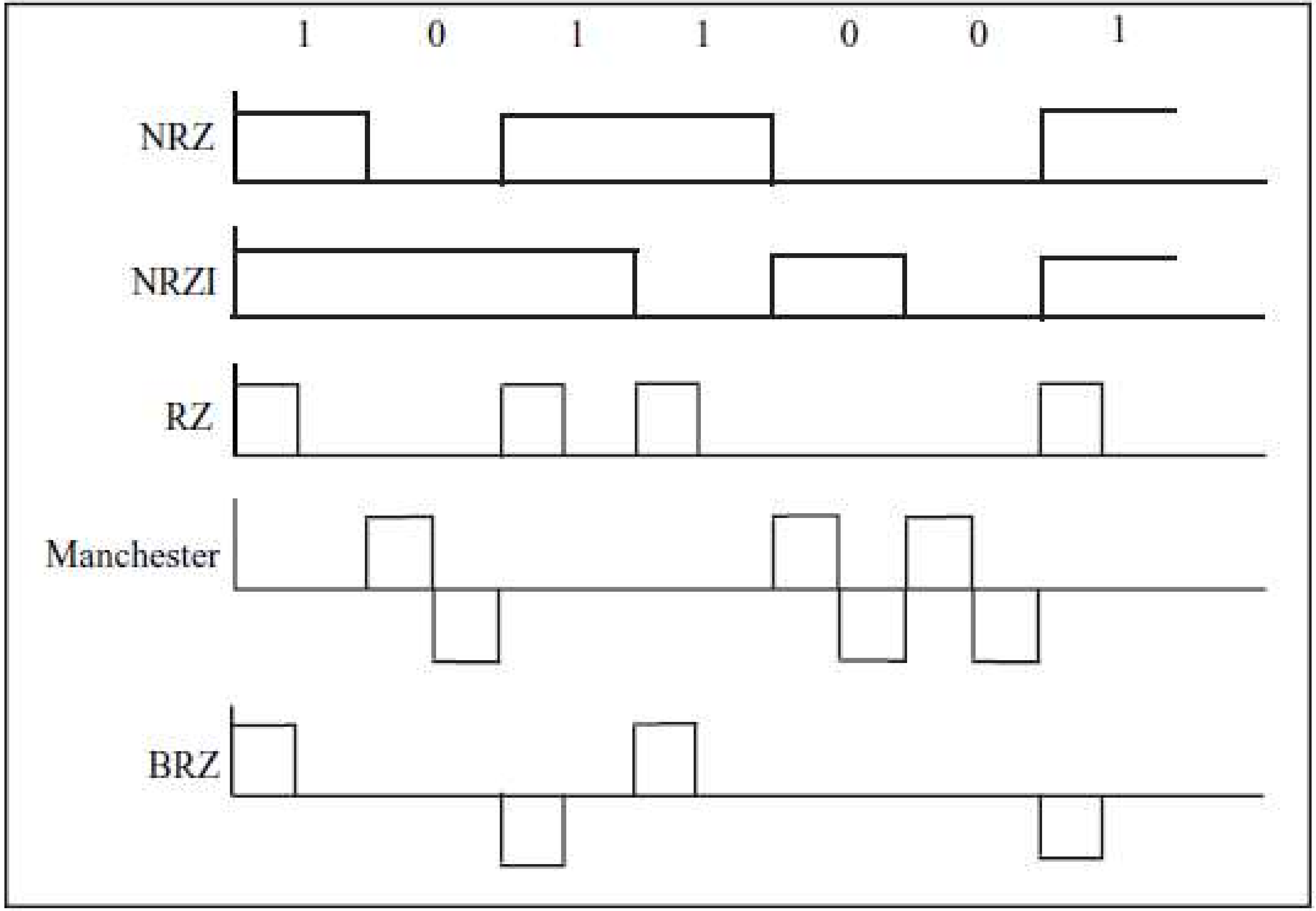
- To better understand serial communication concepts, some terminology must be first introduced.
- Space constraints do not allow us to include an exhaustive list of terms associated with serial communication techniques.
  - *Simplex mode:* In this mode, the serial communication is accomplished by transmitting data in one direction at a time.
  - *Duplex mode:* In this serial communication mode, data can be transmitted and received from both ends of the communication link at the same time.
  - *BAUD rate:* The rate of bits sent or received. It describes the number of bits communicated per second.

- *ASCII code:*The American Standard Code for Information Interchange code is used in communication to encode alphabets, numbers, punctuation, and control characters using a seven-bit representation. ASCII is a subset of the international Unicode standard.
- *Bit time:* The time required to transmit or receive a single bit.
- *Serial line code:* A specific encoding mechanism used to transmit and receive information.

Serial Communication Signals

- consider five different serial communication line codes. It is important that both the transmitter and receiver use a common line code, BAUD rate, and parity setting for proper communications.

- The different
  in Figure.
  - *Non-return-z*
    one is repres
    bit zero is rep
  - *Non-return-z*
    code: Bit cha
    voltage high
    represented a
  - *Return-zero* (
    represented v
    one is repres
    the first half
    bit time is rep
    zero.



Bit one is
zero and bit
oltage one for
nd the rest of
voltage level
me.

line code:
voltage zero
by either
w for the
he rest of the
voltage zero.
one alternates
voltage low.

Handshake Mechanisms

- To ensure the robust serial communication with minimal errors, several handshake mechanisms have been developed.
- For asynchronous serial communications, each data frame is constructed with start bits, data bits, a parity bit, and stop bits. The start bits inform a receiver that a data frame has arrived. Typically, one or two bits are used as the start bits.When the transmitter is idle it typically provides a logic high idle signal on the serial communication output pin.
- A start bit is typically a logic low. The transition from idle (logic high) to a start bit (logic low) allows the receiver to detect the start of a new incoming data frame.
- Data bits can be either eight or nine bits. The ASCII format uses seven bits to represent a character. The eighth bit is used for parity. A nine-bit format is used when the data to be transmitted is eight bits in width (e.g. the output from an eight-bit analog-to-digital converter).

COMMUNICATION SYSTEMS

- In this case, the ninth bit in the data portion of the frame is the parity bit.
- The parity bit improves the accuracy of serial communication by providing the receiver the ability to detect the presence of a single bit error in transmission.
- There are two different types of parity bits: an even parity bit and an odd parity bit. When an even parity bit mode is used, the parity bit is used to make the number of logic high data bits and the parity bit to be an even number.
- When an odd parity bit mode is used, the total number of logic high bits in the data bits and the parity bit must be an odd number. If a receiver detects a parity error, it may notify the transmitter to resend the data. If more robust error correction capability is required, additional parity bits may be used.
- Similar to the start bits, the stop bits inform a receiver of the end of a data frame. Start, data, parity, and stop bits are used by a receiver to ensure correct reception of a frame and synchronization between transmitter and receiver.

RS-232 Protocol

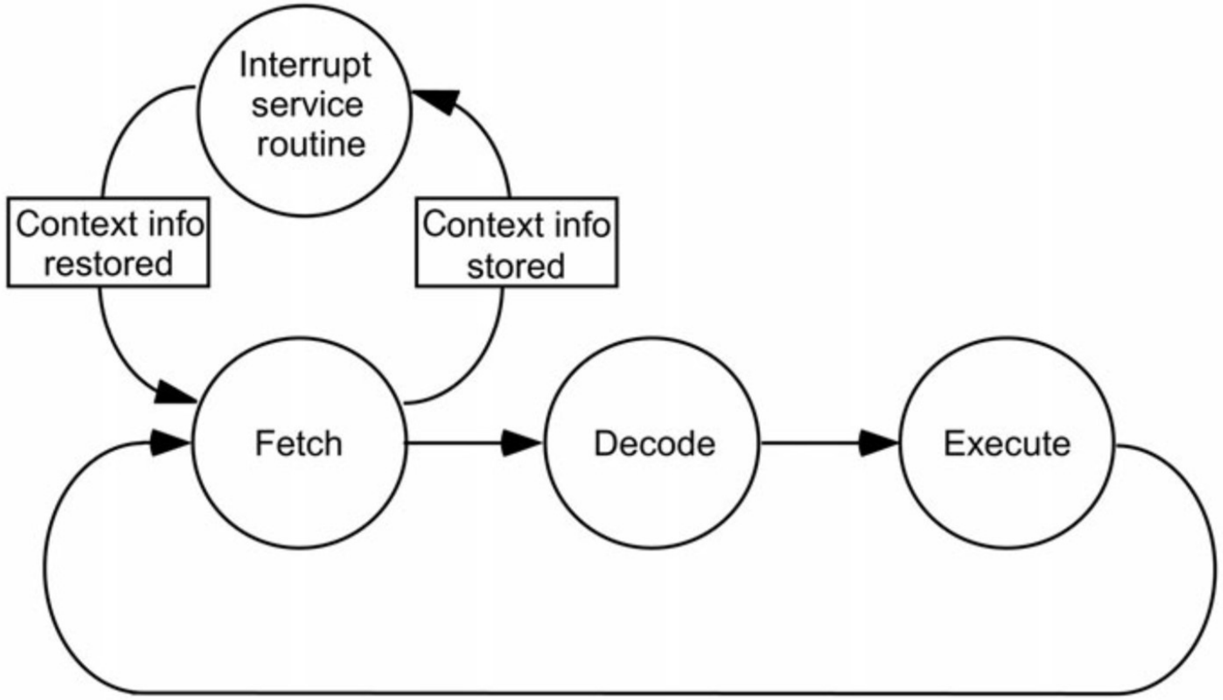
- Many peripheral devices are equipped to communicate with an RS-232 compatible interface. The “RS-232” is an Electronic Industry Association (EIA) standard formally designated EIA-232-D.
- This standard specifies the different aspects of a serial communication interface including
  - electrical specifications,
  - functional signal specification,
  - mechanical specification, and
  - procedural specifications.
- A variety of chips are available to translate microcontroller compatible signals to RS-232 compatible signals.

COMMUNICATION SYSTEMS

- The RS-232 standard represents a logic high with a −10 VDC level and a
- logic low with a +10 VDC level. These chips are equipped to provide interfacing for a twoway (transmit and receive) communication system.
- That is, the output serial bit stream from a microcontroller to a RS-232 compatible peripheral device must be converted to a RS-232 signal.
- The serial bit stream returning from the peripheral device must be converted from the RS-232 format back to microcontroller compatible signal levels.

INTERRUPT SYSTEM

- A program consists of a number of program steps that are executed in sequence.
- The normal execution of a program step follows the fetch, decode, execute sequence as shown in Figure.
- Sometimes this normal sequence of events must be interrupted to respond to high priority faults and status both inside and outside the microcontroller.



INTERRUPT SYSTEM

- When these higher priority events occur, the microcontroller must temporarily suspend normal operation and execute event specific actions.
- These actions are commonly called an interrupt service routine.
- Once the higher priority event has been serviced, the microcontroller returns and continues processing the normal program.
- It must be emphasized that although the interrupt temporarily suspends the operation of the normal sequence of events, an orderly transition to and from the interruptmust be provided.
- This is accomplished by the use of a stack. A stack is a temporary storage location set aside as a portion of RAM memory.

INTERRUPT SYSTEM

- When an interrupt occurs the microcontroller will finish executing the current instruction.
- It will then place context information, key register values and the return address to the normal program, on the stack.
- The interrupt service routine (ISR) will then be executed. The ISR consists of the microcontroller activity required by the interrupt.
- When the ISR is finished, control will return to the program that was executing when the interrupt occurred.
- The context information previously stored on the stack will be used to restore the controller to its preinterrupt configuration.

INTERRUPT SYSTEM

- There is a wide variety of interrupts available with most microcontrollers. Some are generated by hardware or software malfunctions. Others are available to the system designer.
- Two very useful interrupts are the real time interrupt (RTI) and the external interrupt request (IRQ).
- The RTI interrupt provides a regular, periodic interruption in the main program flow. This is useful for updating a real time clock or checking critical system status such as the system battery level.
- The IRQ is an external controller hardware pin. When this pin is asserted an interrupt is generated and the associated interrupt service routine is executed. This is a useful method of allowing an external system hardware component to alert the microcontroller that a malfunction has occurred.
- For example, if the microcontroller is controlling an access gate and something is preventing the gate from opening, the microcontroller needs to be alerted of this condition.
- If not, the microcontroller will continue to issue an open control signal to the gate’s motor and cause the motor to be damaged.

SPEED

- A novice system designer might believe that the best microcontroller is the one that has the fastest operating speed and equipped with the most number of features.
- As mentioned earlier, it is the system designer’s responsibility to find the most economical microcontroller with the proper features for a given application.
- Microcontrollers are available in a wide range of operating speeds. In general, you should use the lowest acceptable speed for a given application.
- This is because that the power consumption of a microcontroller is directly proportional to its operational speed. Since many microcontroller applications are battery powered, conserving power and hence extending battery life is essential

CUTTING EDGE TECHNOLOGY

- • *FPGA and microcontroller combined processors:* It is becoming common place to develop a design employing both programmable logic hardware and microcontrollers in the same system. Several microcontroller manufacturers have come out with joint development kits. It is up to the system designer to implement system functions in the most appropriate portion of the system.
- • *Microcontrollers on the Internet:* The Internet is everywhere! Some microcontroller manufacturers have taken advantage of this by providing the capability to directly interface their microcontroller to the Internet. This allows a system design to employ the Internet infrastructure to connect distributed microcontrollers. This would be well suited for applications such as providing security and safety monitoring within a large building or complex with existing Internet infrastructure. Microcontroller-based data collection nodes can be distributed throughout a building complex and then relay the collected information via the Internet infrastructure to a central processor.
- • *Data links:* As mentioned earlier, microcontrollers may be networked together in a simple but powerful configuration. These microcontroller networks have become common place in a wide variety of commercial and military products.
- • *USB controllers:* The universal serial bus (USB) has become a common method of connecting peripheral devices to a PC. Several manufacturers have provided USB support for their microcontrollers. This allows for the easy interface of a microcontroller to a PC for data exchange and peripheral development.

Referensi

1. Steven F. Barrett and Daniel J. Pack, ”Microcontrollers Fundamentals for Engineers and Scientists”, 2006, Morgan & Claypool

Ada pertanyaan?

Tugas 2

1. Carilah papan pengembangan atau sistem minimum rangkaian mikrokontroler AVR dan Espressif untuk masing-masing kelompok dan buatlah slide presentasi untuk menerangkannya!

Semoga Bermanfaat dan Terima Kasih atas Perhatiannya