FE/EIT Review

Computers

Instructor: Russ Tatro

References

John A. Camara, Electrical Engineering Reference Manual, 6th edition, Professional Publications, Inc, 2002.

John A. Camara, Practice Problems for the Electrical and Computer Engineering PE Exam, 6th edition, Professional Publications, Inc, 2002.

National Council of Examiners for Engineering & Surveying, Principles and Practice of Engineering, Electrical and Computer Engineering, Sample Questions and Solutions, NCEES, 2001.

National Council of Examiners for Engineering & Surveying, Fundamental of Engineering, Supplied-Reference Handbook, NCEES, 2008.

References

Michael A. Lindeburg, PE, FE Review Manual, Rapid Preparation for the General Fundamentals of Engineering Exam, 2nd Edition, Professional Publications, 2006.

Michael A. Lindeburg, PE, FE/EIT Sample Examinations, 2nd Edition, Professional Publications, 2006.

Introduction

Section IV. Computers includes:

A. Terminology
memory types, CPU, baud rates, Internet

B. Spreadsheets
addresses, interpretation, "what if," copying formulas

C. Structured programming
assignment statements, loops and branches, function calls

Computer Hardware

The computer hardware topic can be divided into the following sections:

- Computer architecture
- Microprocessors
- Control of computer operation
- Computer memory
- Parity
- Input/Output devices
- Random secondary storage devices
- Sequential secondary storage devices
- Real-time and batch processing
- Multitasking and Time-sharing
- Background and foreground processing
- Teleprocessing
- Distributed systems and Local-Area networks

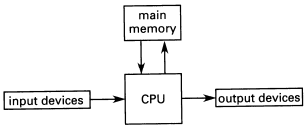
Computer Architecture

All digital computers contain three main components:

A central processing unit - CPU

Main memory

External (peripheral) devices



Microprocessors

A microprocessor is a CPU on a single chip.

It is common to combine several CPUs onto a single *die* as in dual-core, quad-core, or eight-core.

Other chips both external to the microprocessor and on the chip itself will be required to perform all the computer functions.

CPUs consist of an arithmetic and logic unit (ALU), accumulators, registers, stacks and a control unit.

The ALU executes commands and manipulates data.

The accumulators hold data and instructions for further manipulation in the ALU.

Registers are used for temporary storage of instructions or data. Types include *program counter* (PC), *instruction register* (IR) and *stacks*.

Control of computer operation

The user interface and basic operation of a computer are controlled by the *operating system* (OS) also know as the *monitor program*.

The OS manages the memory, schedules processing operations, accesses to peripheral devices, communicates with the user by way of mouse, keyboard and displays, and resolves conflicting requirements for resources.

A low-level term for the control of devices is *basic input/output system* (BIOS) usually contained in *firmware* on the *motherboard*.

A hardware portion of the computer is called the *interrupt* signals that signals the CPU that a program, device or condition requires attention.

Computer memory

Computer memory consists of many equally sized storage locations.

Each storage location has an associated address.

The total number of address can be described in several ways:

- bit – binary digit
- nibble – 4 bits
- words – 8, 16, and 32 bits
- the number of storge locations is always a multiple of two - n^x

The physical hardware implementing the memory is often used to name the memory – DRAM, SDRAM, ROM, PROM, EPROM, FLASH, …

DRAM – direct memory addressing by a peripheral device – fast

Parity

Parity is a technique used to ensure that the bits within a memory byte are correct.

For every 8 data bits – a ninth bit is added to create a *frame*.

Even or Odd parity refers to the setting of the 9^{th} bit such that the frame contains either an even or odd number of “one” bits.

Checksum is a term often used to describe the process verifying data integrity.

Input/Output devices

Devices that feed data into the computer for receive data from the computer are known as *input/output* (I/O) *devices*.

The *parallel interface* has as many separate wires in the interface as there are bits (typically 7, 8 or 9) in the code representing a character.

The *serial interface* sends data bits one at a time through a single line in the interface cable.

The *universal serial bus* (USB) is a standard that has become very popular in computer interfaces.

The flow of data is controlled by both hardware and software methods. This is known as *handshaking*.

Random secondary storage devices

Random access storage devices included magnetic and optical disk drives.

They are random access because individual records can be accessed without having to read through the file sequentially.

Magnetic disk drives (hard drives) are composed of thin spinning magnetic media platters. Data on the surface is organized into tracks, sectors and cylinders.

Common measures of merit for a hard drive include capacity (GB), access time (average seek time, track to track seek time, and rotational delay), and areal density (how tightly packed are the magnetic bits).

Sequential secondary storage devices

Magnetic tape units are sequential devices.

To arrive at a specific record, the tape must be read from the beginning to that record.

Since the decline of hard drive prices, there has been less use of sequential storage devices.

One application for such a device is the aircraft "black box". A heat resistant material continually loops in the recorder for a pre-set amount of time.

Real-time and batch processing

Batch processing – programs are grouped into efficient categories and then run without user interaction. This is the mainframe "payroll" model.

Real-time (interactive) processing is when a program runs when it is submitted to the CPU, often with user interaction.

Multitasking and Time-sharing

When the hardware is quick enough, several tasks or users may use the same CPU simultaneously.

In a *virtual machine* (VM) several users may appear to have their own computer but are running on a shared resource.

Thus multitasking is when multiple tasks can be performed simultaneously.

Time-sharing is when a set of resources is time divided among several users quickly. The users may not notice any delays and seem to have a dedicated computer.

True multitasking must have sufficient resources (such as multiple CPU cores) so that multiple programs really do run at the same time.

Background and foreground processing

A user sitting at a computer and interacting with a program is an example of *fore-ground processing*. All computer resources are optimized to respond quickly to the user.

A non-time critical task may operate in the background without user interruption as long as sufficient computer resources exist for the fore-ground application to run unaffected.

Most operating systems now frequently perform background tasks such as software updates, disk integrity monitoring, or *widgets* to aid user productivity.

Teleprocessing

Teleprocessing is the access of a computer from a remote station. Think internet.

There are three classes of communication lines:

- narrow-band (used infrequently in remote sensors)
- voice-grade (standard phones)
- wide-band

Cycling redundancy checking (CRC) is a method to detect errors in data transmission.

Distributed systems and local-area networks

Distributed systems historically consisted of a main computer with various remote terminals.

With the creation of relatively cheap microcomputers, this evolved into a configuration where many computers are linked together by some communication system.

A local-area network (LAN) is a system where computers are connected both with wires and wirelessly to share files, resources such as printers and other functions.

WiFi, WAN, LAN, IP, Nodes, Data encryption and such are common terms used in today's interconnected network topologies.

Short 5 minute break

Computer Software

The computer software topic can be divided into the following sections:

- Character coding
- Program design
- Flowcharting symbols
- Low-level languages
- High-level languages
- Relative computational speed
- Structure, Data typing, and Portability
- Structured Programming
- Spreadsheets
- Fields, Records, and File types
- File Indexing
- Sorting, Searching, Hashing
- Database structures, Hierarchical and relational data structures
- Artificial Intelligence (AI)

Character coding

Alphanumeric data - characters that can be printed or displayed.

Think of the alphabet and most symbols

Control characters

tab, carriage return, form feed,….

ASCI – American Standard Code for Information Interchange

a 7 bit code which allows 128 different combinations

an 8th bit is not standardized but frequently used

EBCDIC – Extended Binary Coded Decimal Interchange Code

8 bits per character which allows 256 different combinations
often converted to Hexadecimal (Hex) for readability
Hex is a base 16 system

Program design

A *program* is a sequence of computer instructions that performs some function.

The program implements an *algorithm*.

Human readable instructions are called *source code* or *source code statements*.

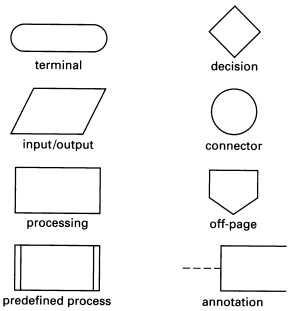
Compiler – translates *source code* into machine readable code also called an *executable program*.

Software – an executable program stored on disk or tape.

Firmware – an executable program stored on a read only memory (ROM) or other non-removable non-volatile memory.

Flowcharting symbols

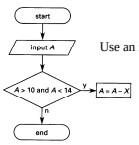
A flowchart is a step-by-step drawing representing a specific procedure or algorithm.



Example - Flowcharting symbols

Flowchart the following procedure: If A is greater than 10 and if A is less than 14, subtract x from A. If A is less than 10 or if A is greater than 14, exist the program.

All flow charts start with a terminal and end with a terminal.



Use an input symbol to enter A

Use a decision symbol the compare A to preset values.

Use a process symbol to perform the math.

Low-level languages

Low-level languages include machine language and assembly language.

Machine language are instructions immediately understood by the computers CPU.

Machine language is the CPUs native language.

Assembly language uses mnemonic codes to specify operations.

These mnemonics are more human friendly but require decoding by an *assembler* before the CPU can understand the instruction.

Portions of other programs or function libraries are combined together by a *linker*.

A *loader* places the program into the computer's memory and then the CPU can run the program.

High-level languages

High-level languages more closely resemble English.

Thus high-level programming is easier to learn, to code and to debug.

An *interpreter* or *compiler* translates the high-level program into machine language for a specific processor system.

Relative computational speed

Certain languages are optimized to execute faster on specific CPUs.

Exceptions and deviations abound! But in general:

Assembly language is the fastest

Then:

- compiled
- pseudo-compiled
- interpreted programs.

Structure, Data typing and Portability

Structured language – subroutines and procedures have only one entry point and one return point.

Contrast with the BASIC language unlimited GOTO statements and returns from anywhere in a GOSUB subroutine.

Data types typical fall into integer, and real numbers. Later in database structures there will be many more data types.

Portability – a higher-level language is written so that with the appropriate interpreter – the program can run on different hardware platforms.

Structured programming

Structured programming follows a *top-down* path.

The main program is best characterized as a series of calls (references) to other subprograms by use of For/Next, Do/While, Do/Until commands (loops).

Local variables are available only within a subprogram.

Global variables are available to all subprograms and the main program.

Spreadsheets

Spreadsheets combine a visible layout of data with a reference library of mathematical commands.

Microsoft Excel is the most widely used spreadsheet.

You have undoubtedly used a spreadsheet and done some programming in Excel.

Remember the row/column addressing and avoid recursive addressing.

Fields, Records, and File types

An assemblage of fields is a record.

for example, a person's age, name, residence and so on…

A group of records is a file.

a collection of various people's data

The data may be stored in a sequential or random file.

Sequential - the data must be read from beginning to end – think of old magnetic tape.

Random - the data can be read any any order – think of a hard drive.

File Indexing

An index is a key or keyword that is similar to the index of a book.

You can also think of an index as being a playlist.

The physical storage of the data will usually be in some efficient storage order.

The index will point to that data location for quick data retrieval.

Sorting, Searching, Hashing

Sorting routines place data is some specific order such as ascending or descending numeric or alphabetical order.

There are many sorting algorithms such as *successive minima*, *quicksort*, or *heap sort* with characteristics useful in relation to how the data is already sorted and stored.

Sorts can take up to n^2 operations.

Searching is an operation that tries to locate a specific data record. The number of operations to find a specific record is on the order of log(n).

Hashing is a technique where the record location is determined from a numeric key that was used to place the record in a specific order in the first place.

Database structures, Hierarchical and relational data structures

A database is a collection of indexed files.

A relational database is a database where certain business rules are implemented to relate data fields such as insurance carrier to a specific patient.

In a relational system - data is divided into logical subcategories which limits redundant information.

A hierarchical database contains records in an organized and structured format. A search must first find the record in accordance with the structure method.

In a hierarchical database all records are self contained in that no references to other data are allowed within the record.

Artificial Intelligence (AI)

Artificial intelligence infers that the program can "learn" from the acquisition of data.

New data can be absorbed and organized in ways the lead to logical reasoning and allow for the response to inquiries.

AI software has been implemented in expert medical programs and aerospace autopilots.

As yet, AI is implemented for a specific function and is not a "human like" intelligence, but can associate a large number of records into a form that results in a dialog.

FE/EIT Review

Computers

Instructor: Russ Tatro