IQRA NATIONAL UNIVERSITY

DEFINING POSSIBILITIES, CREATING OPPORTUNITIES

Name : Sajawal Khan

Department : BS(CS)

Semester : 4th

ID # : 14756

Assignment No : 1st (Chapter# 01)

Subject : Computer Architecture

Submitted To :Muhammad Amin Sir

Q.1 Give answer to each of the following:

A. What are the four main functions of a computer?
Ans:

1.Data Processing:

Data may take a wide variety of forms, and the range of processing requirements is broad.

2. Data Storage:

The computer performs a long-term data storage function. Files of data are stored on the computer for subsequent retrieval and update.

3. Data Movement:

When data are moved over longer distances, to or form a remote device, the process is known as “data communication”.

4. Control:

Within the computer, a control unit manages the computer’s resources and orchestrates the performance of its functional parts in response to instructions.

B. Figure 01 shows the IBM zEnterprise EC12 Core layout. Briefly explain the function of each sub-area.

Ans:

- **ISU (instruction sequence unit):**
  Determines the sequence in which instructions are executed in what is referred to as a superscalar architecture.
- **IFU (instruction fetch unit):**
  Logic for fetching instructions.
- **IDU (instruction decode unit):**
  The IDU is fed from the IFU buffers, and is responsible for the parsing and decoding of all z/Architecture operation codes.
- **LSU (load-store unit):**
  It is responsible for handling all types of operand accesses of all lengths, modes, and formats as defined in the z/Architecture.
- **XU (translation unit):**
  This unit translates logical addresses from instructions into physical addresses in main memory. It contains TLB used to speed up memory access.
- **FXU (fixed-point unit):**
  The FXU executes fixed-point arithmetic operations.
- **BFU (binary floating-point unit):**

The BFU handles all binary and hexadecimal floating -point operations, as well as fixed-point multiplication operations.

- **DFU (decimal floating-point unit):**
  The DFU handles both fixed-point and floating-point Operations on numbers that are stored as decimal digits.
- **RU (recovery unit):**
  The RU keeps a copy of the complete state of the system that includes all registers, collects hardware fault signals.
- **COP (dedicated co-processor):**
  The COP is responsible for data compression and encryption functions for each core.
- **I-cache:**
  This is a 64-kB L1 instruction cache, allowing the IFU to prefetch instructions before they are needed.
- **L2 control:**
  This is the control logic that manages the traffic through the two L2 caches.
- **Data L2:**
  A 1-MB L2 data cache for all memory traffic other than instructions.
- **Instr-L2:**
  A 1-MB L2 instruction cache.

C. Discuss the IAS operation using the flowchart in Figure 02.

Ans:

The IAs operates by respectively performing as instruction cycle. Each instruction cycle consists of two sub-cycles.

Fetch Cycle:-

The opcode of next instruction is loaded into the IR and the address portion is loaded into the MAR. This instruction may be taken from the IBR, or it can be obtained from memory by loading a word into the MBR, and then down to the IBR, IR and MAR.

Execute Cycle:-

The control circuitry interprets the Opcode & executes the instruction by sending out the appropriate control signals to cause data to be moved or an operation to be performed by the ALU.

D. For each of the following examples, determine whether this is an embedded system, explaining why or why not.

a. Are programs that understand physics and/or hardware embedded? For example, one that uses finite-element methods to predict fluid flow over airplane wings?

Ans:

No. These programs are never considered to be embedded because they are not an integral component of a larger system.

b. Is the internal microprocessor controlling a disk drive an example of an embedded system?

Ans:

Yes, regardless of what the disk drive is used for. The software (firmware, actually) within the disk drive controls the HDA (head disk assembly) hardware and is hard real time as well.

c.I/O drivers control hardware, so does the presence of an I/O driver imply that the computer executing the driver is embedded?

Ans:

No, Input-Output drivers do not represent the embedded system

d.Is a PDA (Personal Digital Assistant) an embedded system?

Ans:

Yes, PDA is an embedded system because it is just like a personal computer in hand.

e. Is the microprocessor controlling a cell phone an embedded system?

Ans:

Yes, the firmware in the cell phone is controlling the radio hardware.

f. Are the computers in a big phased-array radar considered embedded? These radars are 10-story buildings with one to three 100-foot diameter radiating patches on the sloped sides of the building.

Ans:

Yes, these computers were generally some of the most powerful computers available when the system was built, are located in a large computer room occupying almost one whole floor of a building and may be hundreds of meters away from the radar hardware. However, the software running in these computer controls the radar hardware; therefore, the computers are an integral component of a large system.

g.Is a traditional flight management system (FMS) built into an airplane cockpitconsidered embedded?

Ans:

If the FMS is not connected to the avionics and is used only for logistics computerizations, a function readily performed on a laptop, then the FMS is clearly not embedded.

h.Are the computers in a hardware-in-the-loop (HIL) simulator embedded?

Ans:

Yes, both in the simulator, and in the thing being tested in the HIL simulator. Hardware is being controlled on both sides.

i.Is the computer controlling a pacemaker in a person's chest an embedded computer?

Ans:

Yes, in this case of the "system" is the combination of the pacemaker and the person's heart.

j. Is the computer controlling fuel injection in an automobile engine embedded?

Ans:

Yes, it is part of a large system, the engine, and it is directly monitoring and controlling the engine through special hardware.

Q.2 Write a note on each of the following:

A. Main structural components of a computer.

Ans:

There are four main structural components:

1. **Central processing unit (CPU):** Controls the operation of the computer and performs its data processing functions; often simply referred to as **processor**.
2. **Main memory:** Stores data.
3. **I/O:** Moves data between the computer and its external environment.
4. **System interconnection:** Some mechanism that provides for communication among CPU, main memory, and I/O.

B. Key characteristics of a planned computer family.

Ans:

The characteristics of a family are as follows:

- **Similar or identical instruction set:** In some cases, the lower end of the family has an instruction set that is a subset of that of the top end of the family. This means that programs can move up but not down.
- **Similar or identical operating system:** The same basic operating system is available for all family members.
- **Increasing speed:** The rate of instruction execution increases in going from lower to higher family members.
- **Increasing number of I/O ports:** The number of I/O ports increases in going from lower to higher family members.
- **Increasing memory size:** The size of main memory increases in going from lower to higher family members.
- **Increasing cost:** At a given point in time, the cost of a system increases in going from lower to higher family members.

C. Stored program computer.

Ans:

A fundamental design approach first implemented in the IAS computer is known as the *stored-program concept*. This idea is usually attributed to the mathematician John von Neumann.

The first publication of the idea was in a 1945 proposal by von Neumann for a new computer, the EDVAC (Electronic Discrete Variable Computer).

In 1946, von Neumann and his colleagues began the design of a new stored-

Program computer, referred to as the IAS computer, at the Princeton Institute for Advanced Studies.
It consists of
■■ A **main memory**, which stores both data and instructions.
■■ An **arithmetic and logic unit (ALU)** capable of operating on binary data.

D. Moore's law.
Ans:

The famous Moore's law, which was propounded by Gordon Moore, cofounder of Intel, in 1965 [MOOR65]. Moore observed that the number of transistors that could be put on a single chip was doubling every year.
The pace slowed to a doubling every 18 months in the 1970s but has sustained that rate ever since.
The consequences of Moore's law are profound:

1. The cost of computer logic and memory circuitry has fallen at a dramatic rate.
2. Because logic and memory elements are placed closer together on more densely packed chips, the electrical path length is shortened, increasing operating speed.
3. The computer becomes smaller, making it more convenient to place in a variety of environments.
4. There is a reduction in power requirements.
5. With more circuitry on each chip, there are fewer interchip connections.

Q.3 Differentiate each of the following:

A. Computer organization and computer architecture.
Ans:

- **Computer architecture** refers to those attributes of a system visible to a programmer or, put another way, those attributes that have a direct impact on the logical execution of a program. A term that is often used interchangeably with computer architecture is instruction set architecture (ISA).
- **Computer organization** refers to the operational units and their interconnections that realize the architectural specifications. Examples of architectural attributes include the instruction set, the number of bits used to represent various data types (e.g., numbers, characters), I/O mechanisms, and techniques for addressing memory.

B. RISC and CISC.
Ans:

- The current x86 offerings represent the results of decades of design effort on **complex instruction set computers (CISCs)**. The x86 incorporates the sophisticated design principles once found only on mainframes and supercomputers and serves as an excellent example of CISC design.
- An alternative approach to processor design is the **reduced instruction set**

computer (RISC). The ARM architecture is used in a wide variety of embedded systems and is one of the most powerful and best-designed RISC-based systems on the market. In this section and the next, we provide a brief overview of these two systems.

C. Microprocessors and Microcontrollers.

Ans:

- **Microprocessor** chips include registers, an ALU, and some sort of control unit or instruction processing logic. As transistor density increased, it became possible to increase the complexity of the instruction set architecture, and ultimately to add memory and more than one processor.
- A **microcontroller** is a single chip that contains the processor, non- volatile memory for the program (ROM), volatile memory for input and output (RAM), a clock, and an I/O control unit. The processor portion of the microcontroller has a much lower silicon area than other microprocessors and much higher energy efficiency.

D. Cortex-A, Cortex-R, and Cortex-M.

Ans:

- The **Cortex- A** and **Cortex- A50** are application processors, intended for mobile devices such as smartphones and eBook readers, as well as consumer devices such as digital TV and home gateways (e.g., DSL and cable Internet modems). These processors run at higher clock frequency (over 1 GHz), and support a memory management unit (MMU).
- The **Cortex- R** is designed to support real- time applications, in which the timing of events needs to be controlled with rapid response to events. They can run at a fairly high clock frequency (e.g., 200MHz to 800MHz) and have very low response latency.
- **Cortex-M** series processors have been developed primarily for the microcontroller domain where the need for fast, highly deterministic interrupt management is coupled with the desire for extremely low gate count and lowest possible power consumption.

Q.4 Solve each of the following:

A. Given the memory contents of the IAS computer shown below,

Address Contents

1. **08A 010FA210FB**
2. **08B 010FA0F08D**
3. **08C 020FA210FB**

a. show the assembly language code for the program, starting at address 08A.

Ans:

1. Here is a simple way to understand this problem:

Contents are divided up into two 5 bit instructions, LH and RH

LH instruction = 010FA

opcode = 01

address = 0FA

RH instruction = 210FB

opcode = 21

address = 0FB

Since this is in hexadecimal form you have to convert the numbers to binary form: (use the IAS instruction set)

LH instruction:

01 = 00000001 = LOAD M(X)

M(X) refers to the memory address location 0FA

The first 5 bits of 08A should read - LOAD M(0FA)

RH instruction:

21 = 00100001 = STOR M(X)

M(X) refers to the memory address location 0FB

The second 5 bits of 08A should read - STOR M(0FB)

Finally the assembly language code for 08A 010FA210FB is

LOAD M(0FA)

STOR M(0FB)

2. Here is a simple way to understand this problem:

Contents are divided up into two 5 bit instructions, LH and RH

LH instruction = 010FA

opcode = 01

address = 0FA

RH instruction = 0F08D

opcode = 0F

address = 08D

Since this is in hexadecimal form you have to convert the numbers to binary form: (use the IAS instruction set)

LH instruction:

01 = 00000001 = LOAD M(X)

M(X) refers to the memory address location 0FA

The first 5 bits of 08B should read - LOAD M(0FA)

RH instruction:

0F = 00001111 = JUMP + M(X,0:19) refers to the memory address location 08D

The second 5 bits of 08B should read – JUMP + M(08D,0:19)

Finally the assembly language code for 08B 010FA0F08D is

LOAD M(0FA)

JUMP + M(08D,0:19)

3. Here is a simple way to understand this problem:

   Contents are divided up into two 5 bit instructions, LH and RH

   LH instruction = 020FA

   opcode = 02

   address = 0FA

   RH instruction = 210FB

   opcode = 21

   address = 0FB

Since this is in hexadecimal form you have to convert the numbers to binary form: (use the IAS instruction set)

LH instruction:

02 = 00000010 = LOAD - M(X)

M(X) refers to the memory address location 0FA

The first 5 bits of 08C should read – LOAD - M(0FA)

RH instruction:

21 = 00100001 = STOR M(X)

M(X) refers to the memory address location 0FB

The second 5 bits of 08C should read - STOR M(0FB)

Finally the assembly language code for 08C 020FA210FB is

LOAD - M(0FA)

STOR M(0FB)

b. Explain what this program does?

Ans:

1. In 08A address, the M(0FA) transfer to the accumulator and tansfer contents of accumulator to memory location 0FB.

2. In 08B address, the M(0FA) transfer to the accumulator and take next instruction from left half of M(08D).

3. In 08C address, the –M(0FA) transfer to the accumulator and tansfer contents of accumulator to memory location 0FB.