CS501 - Advance Computer Architecture FAQs By www.virtualians.pk

Question: Explain Booth's Algorithm with example.

Answer:

Booth's multiplication can be performed using the following method:

Example:

Assume n=7 bits available. Multiply B=22=(0010110)2 by A=34=-(0100010)2 . First represent both operands and their negation in signed 2's complement:

22: 0010110, -22: 1101010

34: 0100010, -34: 1011110

Then carry out the multiplication

The product is given in signed 2's complement and its actual value is negative of the 2's complement:

BxA= -11110100010100 = -00001011101100= -74810

Example: 2 x 6

m=2,p=6;

m = 0010

p = 0000 0110

p

0000 0110 0 no-op

0000 0011 0 >> p

1110 0011 0 p = p - m

1111 0001 1 >> p

1111 0001 1 no-op

1111 1000 1 >> p

0001 1000 1 p = p + m

0000 1100 0 >> p

=12

Example: 2 x -3

m = 0010

p = 0000 1101

1110 1101 0 p = p - m

1111 0110 1 >> p

0001 0110 1 p = p + m

0000 1011 1 >> p

1110 1011 0 p = p - m

1111 0101 1 >>p

1111 0101 1 no-op

1111 1010 1 >>p

=-6

Also note that:

As the operands are in signed 2's complement form, the arithmetic shift is used for the right shifts above, i.e., the MSB bit (sign bit) is always repeated while all other bits are shifted to the right. This guarantees the proper sign extension for both positive and negative values represented in signed 2's complement.

When the multiplicand is negative represented by signed 2's complement, it needs to be complemented again for subtraction (when the LSB of multiplier is 1 and the extra bit is 0, i.e., the beginning of a string of 1's).

Question: What is diffrence between encoding and decoding?

Answer: Encoding:

In programming, to put something into code, which frequently involves

changing the form, for example, changing a decimal number to binary-

coded form.

In Computer Science, to convert (a character, routine, or program) into

machine language.

Decoding:

To convert from code into plain text.

Question: Differentiate between CPU register and Cache Memory.

Answer: CPU registers are memory slots on the actual processor, storing data

there gets rid of the overhead of retrieving the data from normal

memory. This memory is quite small compared to normal memory though so

only a few variables can be stored there.

Cache Memory is a special memory subsystem in which frequently-used data values are duplicated for quick access. A memory cache stores the contents of frequently-accessed RAM locations and the addresses where these data items are stored. When the processor references an address in memory, the cache checks to see whether it holds that address. If it does hold the address, the data is returned to the processor; if it does not, a regular memory access occurs. A cache is useful when RAM accesses are slow compared with the microprocessor speed, because cache memory is always faster than main RAM memory.

In Computer Science, a fast storage buffer in the central processing unit of a computer. In this sense, also called cache memory.

Question: What is the difference between compiler and assembler?

Answer: Compiler is a program that translates another program written in a high-level language into machine language so that it can be executed.

Assembler is a program operating on symbolic input data to produce the equivalent executable machine code.

Question: Sir what are the disadvantages of RICS machine architecture (as there are lot of disadvantages of CISC)?

Answer: There are many advantages of RISC also:

RISC supporters argue that it the way of the future, producing faster and cheaper processors - an Apple Mac G3 offers a significant performance advantage over its Intel equivalent.

Instructions are executed over 4x faster providing a significant performance boost! However, RISC chips require more lines of code to produce the same results and are increasingly complex. This will increase the size of the application and the amount of overhead required.

RISC developers have also failed to remain in competition with CISC alternatives. The Macintosh market has been damaged by several problems that have affected the availability of 500MHz+ PowerPC chips. In contrast, the PC compatible market has stormed ahead and has broken the 1GHz barrier.

Despite the speed advantages of the RISC processor, it cannot compete with a CISC CPU that boasts twice the number of clock cycles.

Question: Which one is better CISC or RISC?

Answer: RISC is comparatively better than CISC as it has reduced instruction sets. That is why it is used more now a days in computer systems.

Question: What is vex machine?

Answer: The Machine is a dual frequency-tripler circuit that uses crossover distortion. It generates the distortion of the wave in the sloped part of the cycle, instead of the peaks and valleys.

Question: Tell me about Behavioral RTL ? What are its functions ?

Answer: Register Transfer Level (RTL) is by far the most commonly used style of VHDL (Very High Speed Integrated Circuit Hardware Description Language) description. At this level, systems are described in terms of combinational and sequential functions at the behavioral level. The functions are often described using individual processes and interconnected using signals to form a dataflow. Typical functions included in an RTL description are Registers, Counters and State Machines along with combinational functions such as Multiplexers, Arithmetic Units and Decoders. The RTL style may include some structural information via the dataflow. However, the individual functional blocks are invariably described using behavioral constructs rather than instantiated gates and flip-flops. Many designs are made up from registers interspersed with combinational functions and together these form the data path. Transfers between registers and operations carried out by combinational functions are controlled by the controller or control path which is usually a behavioral description of a Finite State Machine.

The RTL level of description is possibly the most useful of all the alternative levels and is particularly relevant to those people involved with producing real hardware using VHDL, ie. users of logic synthesis, since this level is the accepted style for designs which are intended for synthesis. The majority of VHDL users are targeting synthesis tools and therefore it is probably true to say that the majority of VHDL descriptions are written in the RTL style.

Question: How can we utilize the functional parallelism ?

Answer: This kind of parallelism is based on different functional blocks in your application. The idea is simple: The application is split into separate processing units, that communicate with a fixed number other units in such a way that the output of one part serves as the input of another part. Thus we can visualize such a system as a set of nodes that are connected by pipes in which data only flow in one direction.

A good example is a compiler. Such a compiler may consist of a scanner, a parser and a code-generator. The output of the scanner is the input of the parser. The output of the parser is the input of the code-generator. And the output of the codegenerator is written to the disk.

Question: I want to know about PVM (PARALLEL VIRTUAL MACHINE) and MPI (MESSAGES PASSING INTER FACE).

Answer: PVM (Parallel Virtual Machine) is a software package that permits a heterogeneous collection of Unix and/or Windows computers hooked together by a network to be used as a single large parallel computer. Thus large computational problems can be solved more cost effectively by using the aggregate power and memory of many computers. PVM enables users to exploit their existing computer hardware to solve much larger problems at minimal additional cost. Hundreds of sites around the world are using PVM to solve important scientific, industrial, and medical problems in addition to PVM's use as an educational tool to teach parallel programming. With tens of thousands of users, PVM has become the de facto standard for distributed computing world-wide.

MPI-Message passing is a programming paradigm used widely on parallel computers, especially Scalable Parallel Computers (SPCs) with distributed memory, and on Networks of Workstations (NOWs). The MPI standard defines the user interface and functionality for a wide range of message-passing capabilities. The major goal of MPI, as with most standards, is a degree of portability across different machines. Another type of compatibility offered by MPI is the ability to run transparently on heterogeneous systems, that is, collections of processors with distinct

architectures. Another type of compatibility offered by MPI is the ability to run transparently on heterogeneous systems, that is, collections of processors with distinct architectures. It is possible for an MPI implementation to span such a heterogeneous collection, yet provide a virtual computing model that hides many architectural differences. One example is that MPI guarantees that the underlying transmission of messages is reliable. The user need not check if a message is received correctly.

Question: What is Process - Level Concurrent Execution ?

Answer: Modern operating systems allow several processes to execute concurrently. It is not difficult to imagine several independent processes each being concurrently executed statement-by-statement on a different machine.

On a single processor, the OS, to create the illusion of concurrency, switches a single processor among several processes, allowing it to run one for only a few thousandths of a second before moving on to another. When viewed by a human, these processes appear to run concurrently.

Thus in the example, the execution of the three processes is interleaved. As a result, the output on the screen is a mixture of `As' and `Bs'.

Question: What is difference between hardware debugger & software debugger?

Answer: Hardware Debugger is a graphical tool for detection, diagnostics and testing of custom hardware, designed to meet the needs of the hardware developer by enabling quick and easy testing of hardware without writing any code.

Hardware Debugger runs under Linux, Solaris, Windows 98, Me, NT, 2000 and Windows CE.

Software Debugger is a program designed to aid in debugging another program by allowing the programmer to step through the program, examine data, and monitor conditions such as the values of variables.

Question: What is difference between concurrent and sequential statements?

Answer: Concurrent is the characteristic of a computer operation in which two or more processes (programs) have access to the microprocessor's time and are therefore carried out nearly simultaneously. Because a microprocessor can work with much smaller units of time than people can perceive, concurrent processes appear to be occurring simultaneously but in reality are not.

Sequestial statements are formed or characterized by a sequence, as of units.

Question: What is FALCON-A ?

Answer: FALCON-A is an acronym for First Architecture for Learning Computer Organization and Networks (version A). The term Organization is intended to include Architecture and Design in this acronym.

Question: What is CPU register and processor register?

Answer: CPU registers are memory slots on the actual processor, storing data there gets rid of the overhead of retrieving the data from normal memory. This memory is quite small compared to normal memory though so only a few variables can be stored there. There is no such term as processor register. All registers are the part of processor.

Question: Explain the bench-mark programming ?

Answer: Benchmark programming is used to measure the performance of hardware or software. Its is a test program used to measure hardware or software performance. Benchmarks for hardware use programs that test the capabilities of the equipment; for example, the speed at which a CPU can execute instructions or handle floating point numbers. Benchmarks for software determine the efficiency, accuracy, or speed of a program in performing a particular task, such as recalculating data in a spreadsheet. The same data is used with each program tested, so the resulting scores can be compared to see which programs perform well and in what areas.

Question: Define MIPS computer?

Answer: MIPS is an acronym for millions of instructions per second. A common measure of processor speed. It is an old measure of a computer's speed and power, MIPS measures roughly the number of machine instructions that a computer can execute in one second. However, different instructions require more or less time than others, and there is no standard

method for measuring MIPS. In addition, MIPS refers only to the CPU speed, whereas real applications are generally limited by other factors, such as I/O speed.

Question: What difference between the logical address and effective address ?

Answer: Logical address is a virtual address which is a memory location accessed by an application program in a system with virtual memory such that intervening hardware and/or software maps the virtual address to real (physical) memory. During the course of execution of an application, the same virtual address may be mapped to many different physical addresses as data and programs are paged out and paged in to other locations.

Effective address is the operand or the destination address.

Question: How a debugger disassemble the machine language?

Answer: Debuggers offer more sophisticated functions such as running a program step by step ('single-stepping'), stopping ('breaking') (pausing the program to examine the current state) at some kind of event, and tracking the values of some variables. Debuggers have the ability to modify the state of the program while it is running, rather than merely to observe it.

Question: Why Data Dependence Distance is required for the Branch delays?

Answer: Designing a data forwarding unit requires the study of dependence distances. Without forwarding, the minimum spacing required between two data dependent instructions to avoid hazard is four. The load instruction has a minimum distance of two from all other instructions except branch. Branch delays cannot be removed even with forwarding.

Question: When CPU encounter an opcode of the instruction from the system memory, in what register does it hold that opcode?

Answer: When there are no queue of instructions , then it will not be stored in the memory , rather it will directly be processed. Otherwise , instruction register IR , is used to hold the current instruction.

Question: How are shift instructions useful? When do we use them?

Answer: SHIFT instructions move a bit string (or operand treated as a bit string) to the right or left, with excess bits discarded (although one or more bits might be preserved in flags). SHIFT instructions are useful in many ways. e.g., if you multiply two 16 bit numbers the result

will be of 32 bit number, so shift is used to adjust the result in the two 16 bit registers. Also, as to multiply or divide by 2, SHIFT Left or SHIFT Right instructions are used.

Question: What is Data bus, Address bus and control bus? How these three are interrelated with MAR and MBR with each other. please explain with diagram ?

Answer: A bus is essentially a shared highway that connects different parts of the system—including the microprocessor, disk-drive controller, memory, and input/output ports; and enables them to transfer information. The bus consists of specialized groups of lines that carry different types of information. One group of lines carries data; another carries memory addresses (locations) where data items are to be found; yet another carries control signals. Buses are characterized by the number of bits they can transfer at a single time, equivalent to the number of wires within the bus.

Address Bus is a hardware path usually consisting of 20 to 64 separate lines used to carry the signals specifying a memory location.

Control Bus is the set of lines (conductors) within a computer that carry control signals between the CPU and other devices. For example, a control bus line is used to indicate whether the CPU is attempting to read from memory or to write to it; another control bus line is used by memory to request an interrupt in case of a memory error.

The CPU can put an address in a special-purpose register known as the memory address register (MAR), and then use a control connection to the main memory. memory-buffer register (MBR) hold the content of memory operation (load or store). They play a vital role in fetch cycle of a unit. To start off the fetch cycle, the address which is stored in the program counter (PC) is transferred to the memory address register (MAR). The CPU then transfers the instruction located at the address stored in the MAR to the memory buffer register (MBR) via the data lines connecting the CPU to memory. This transfer from memory to CPU is coordinated by the control unit (CU). To finish the cycle, the newly fetched instruction is transferred to the instruction register (IR) and unless told otherwise, the CU increments the PC to point to the next address location in memory.

The illustrated fetch cycle above (Figure 3) can be summarised by the following points:

PC => MAR

MAR => memory => MBR

MBR => IR

PC incremented

After the CPU has finished fetching an instruction, the CU checks the contents of the IR and determines which type of execution is to be carried out next. This process is known as the decoding phase. The instruction is now ready for the execution cycle.

Question: Difference between Little Endian and Big Endian?

Answer: color="#000080">

Little Endian:

"Little Endian" means that the low-order byte of the number is stored in memory at the lowest address,

and the high-order byte at the highest address. (The little end comes first.) For example, a 4 byte LongInt

Byte3 Byte2 Byte1 Byte0

will be arranged in memory as follows:

Base Address+0 Byte0

Base Address+1 Byte1

Base Address+2 Byte2

Base Address+3 Byte3

Intel processors (those used in PC's) use "Little Endian" byte order.

Big Endian:

"Big Endian" means that the high-order byte of the number is stored in memory at the lowest address,

and the low-order byte at the highest address. (The big end comes first.) Our LongInt, would then be stored as:

Base Address+0 Byte3

Base Address+1 Byte2

Base Address+2 Byte1

Base Address+3 Byte0

Motorola processors (those used in Mac's) use "Big Endian" byte order.

Question: What is Binary Semaphore and the Queue Semaphore?

Answer: Binary Semaphore uses a boolean value as in case of flags. The value of the semaphore is tested, and if it is true, sets it to false, and if false, waits. In Queue Semaphore, the value of the semaphore is set in queue to be processed. A semaphore indicates to other potential users that a file or other resource is in use and prevents access by more than one user.

Question: What is memory mapped I/O ?

Answer: In memory mapped I/O a single bus is used for both I/O and memory. Memory mapped I/O is common is common in modern processors. It has two primary motivations: Data

transfer to and from the processor is standardized, and the number of connections to the processor chip or board is reduced.

Question: what is nested interrupt?

Answer: A system of interrupts that allows an interrupt handler to be interrupted is known as nested interrupts.

Question: What is an interrupt mask?

Answer: A collection of all I/O device interrupt enable bits is known as as a bit vector or Interrupt Mask.

Question: How the processor knows that it has executed the instruction,who generate the control signal for that. What is exectiopn?

Answer: Instruction Pointer contains the address of the next instruction to be

executed and once the instruction is executed, the next address is

shifted in IP, processor has no part in it.

Exception:

In programming, a problem or change in conditions that causes the

microprocessor to stop what it is doing and handle the situation in a

separate routine. An exception is similar to an interrupt; both refer

the microprocessor to a separate set of instructions.

Question: What is the meaning of ISA ?

Answer: This set of instructions or operations and the resources together form the instruction set architecture (ISA). It is the ISA, which serves as an interface between the program and the functional units of a computer, i.e., through which, the computer's resources, are accessed and controlled.

Question: What are the "Emulators " and what are the function of emulators . Is it hardware tool?

Answer: Emulators are designed to make one type of computer or component act as if it were another. By means of an emulator, a computer can run software written for another machine. In a network, microcomputers might emulate mainframes or terminals so that two machines can communicate. It can be a hardware or software.

Question: Why CISC processors are faster in execution as compared to RISC?

Answer: RISC architecture optimizes each of these instructions so that it can be carried out very rapidly, usually within a single clock cycle.

CISC architecture can be very powerful, allowing for complicated and flexible ways of calculating such elements as memory addresses. All this complexity, however, usually requires many clock cycles to execute each instruction.

So, RISC architecture is faster in execution.

Question: Sir, please tell me that what is the SRC NOTATION ?

Answer: SRC Notation can be written as:

• R[3] means contents of register 3 (R for register)

• M[8] means contents of memory location 8 (M for memory)

• A memory word at address 8 is defined as the 32 bits at address 8,9,10 and 11 in the memory. This is shown in the figure.

• A special notation for 32-bit memory words is

M[8]<31…0>:=M[8]ãM[9]ãM[10]ãM[11]

ã is used for concatenation.

Question: Sir what is an ISA ? What is ISA design and development ?

Answer: ISA is an acronym for Industry Standard Architecture. A bus design specification that allows components to be added as cards plugged into standard expansion slots in IBM Personal Computers and compatibles. Originally introduced in the IBM PC/XT with an 8-bit data path, ISA was expanded in 1984, when IBM introduced the PC/AT, to permit a 16-bit data path. A 16-bit ISA slot actually consists of two separate 8-bit slots mounted end-to-end so that a single 16-bit card plugs into both slots. An 8-bit expansion card can be inserted and used in a 16-bit slot (it occupies only one of the two slots), but a 16-bit expansion card cannot be used in an 8-bit slot.

ISA design is composed of Data types/sizes, instruction types, memory address modes, control flow instructions, instructions encoding etc .

Question: RISC and CISC architectures? what is the difference between these two?

Answer: RISC is an acronym for reduced instruction set computers. A microprocessor design that focuses on rapid and efficient processing of a relatively small set of simple instructions that comprises most of the instructions a computer decodes and executes. RISC architecture optimizes each of these instructions so that it can be carried out very rapidly, usually within a single clock cycle. RISC chips thus execute simple instructions more quickly than general-purpose CISC (complex instruction set computers) microprocessors, which are designed to handle a much wider array of instructions. They are, however, slower than CISC chips at executing complex instructions, which must be broken down into many machine instructions that RISC microprocessors can perform.

CISC is an acronym for complex instruction set computers. The implementation of complex instructions in a microprocessor design so that they can be invoked at the assembly language level. The instructions can be very powerful, allowing for complicated and flexible ways of calculating such elements as memory addresses. All this complexity, however, usually requires many clock cycles to execute each instruction.

Question: What is Parallel Architecture and what is relation between Language & Parallel Architecture?

Answer: The architecture of a computer system that can carry out several computational operations simultaneously. Parallel architectures have evolved from special-purpose machines to commodity servers.

Parallel Architecture can be hardware of software based, therefore, software based parallel architecture are related to language based.

Question: Tell me about Parallel Processing & ILP Processing? Tell me also about the difference between these two?

Answer: Parallel Processing is a method of processing that can run only on a type of computer containing two or more processors running simultaneously. Parallel processing differs from multiprocessing in the way a task is distributed over the available processors. In multiprocessing, a process might be divided up into sequential blocks, with one processor managing access to a database, another analyzing the data, and a third handling graphical output to the screen. Programmers working with systems that perform parallel processing must find ways to divide a task so that it is more or less evenly distributed among the processors available.

Instruction Level Parallel Processing utilizes the parallel execution of the lowest level computer operations; adds, multiplies, loads, and so on, to increase performance transparently.

ILP overlap individual machine operations (add, mul, load…) so that they execute in parallel, transparent to the user , speeds up execution.

Parallel Processing is having separate processors getting separate chunks of the program (processors programmed to do so) , non-transparent to the user, speed up and quality up.

Question: What are Basic ILP Classes ?

Answer: Instruction level parallelism is obtained primarily in two ways in uniprocessors: through pipelining and through keeping multiple functional units busy executing multiple instructions at the same time. When a pipeline is extended in length beyond the normal five or six stages (e.g., I-Fetch, Decode/Dispatch, Execute, D-fetch, Writeback), then it may be called

Superpipelined. If a processor executes more than one instruction at a time, it may be called Superscalar. These two techniques can be combined into a Super-Super architecture.

Question: I want to know about RAW , WAR & WAW Dependencies.

Answer: RAW (read after write) is an issue for pipelined processors.

B tries to read a register before A has written it and gets the old value. This is common, and forwarding helps to solve it.

WAW (write after write)

B tries to write an operand before A has written it.

After instruction B has executed, the value of the register should be B's result, but A's result is stored instead.

WAW (write after write)

B tries to write an operand before A has written it.

After instruction B has executed, the value of the register should be B's result, but A's result is stored instead.

This can only happen with pipelines that write values in more than one stage.

Question: Explain the bench mark programming ?

Answer: Benchmark program is used to measure the performance of hardware or software. Benchmark is termed as a test used to measure hardware or software performance. Benchmarks for hardware use programs that test the capabilities of the equipment, for example, the speed at which a CPU can execute instructions or handle floating-point numbers. Benchmarks for

software determine the efficiency, accuracy, or speed of a program in performing a particular task, such as recalculating data in a spreadsheet. The same data is used with each program tested, so the resulting scores can be compared to see which programs perform well and in what areas. The design of fair benchmarks is something of an art, because various combinations of hardware and software can exhibit widely variable performances under different conditions. Often, after a benchmark has become a standard, developers try to optimize a product to run that benchmark faster than similar products run it in order to enhance sales.

Question: How we identify the Address Mode of the Instructions ?

Answer: It is extremely easy to identify the addressing mode of the instruction. Just a little concentration is required to understand the concept. I shall use SRC ld (load) instruction to discuss the addressing modes. This instruction is also used in lectures.

ld instruction is used to load accumulator (register). Now one can load accumulator

1. by directly moving the data
2. by giving the address of memory location using [] (address operator)
3. by copying the data from some other register
4. by giving the address of memory location. This address will be provided by some other register
5. by using some constant to point the memory address where our desired data is lying
6. by some relative positioning.

On the basis of above given points, addressing modes are obtained.

When directly data is accessed, for example ld 123, then mode is immediate. (According to point 1)

When memory address is given like, ld[123], then mode is direct addressing.

There is also indirect addressing mode, ld[[123]]. Here data in memory at address 123 will be calculated by [123] and the resulting data will be calculated again by [result] to gain the final data to be moved in accumulator. (According to point 2)

When data is moved from some register, ld R2, then it is called register direct. (According to point 3)

When some register contains the address, ld [R2], then mode is register indirect. (According to point 4)

For ld[R1 + 12], the addressing mode is Displacement. (According to point 5)

Instruction jump 4 is example of relative addressing mode. Here directly address is given of memory. Jump 4 will take the control to memory address 4. (According to point 6)

Question: What are Dependencies Between the Instructions of a Program?

Answer: Several important aspects of software systems can be expressed as dependencies between their components. A special class of dependencies concentrates on the program text and captures the technical structure and behavior of the target system. The central characteristic making such program dependencies valuable in software engineering environments is that they can be automatically extracted from the program by applying well-known methods of programming language implementation. Dependencies between the instructions is caused as some instructions may not occur before some specific instruction.

Question: What is branching in advance architecture?

Answer: A sequence of program instructions to which the normal sequence of instructions give up control, depending on the value of certain variables. The instructions executed as the result of such a passing of control.

Question: What is data Parallelism?

Answer: Data parallelism involves performing a similar computation on many data objects simultaneously. This kind of parallelism is based on the need to process a lot of data in the same way. Well known examples are found in the area of image processing (raytracing, frame analysis), the queens problem discussed in a later chapter, database transaction systems etc. The common principle is that one program code treats different kinds of data on several independently processing nodes. The corresponding 'farm' software architecture usually consists of one 'master' process, distributing the data and several 'worker' processes that do the work.

Question: What are super scaler processors?

Answer: The name means these processors are scalar processors that are capable of executing more than one instruction in each cycle. The keys to superscalar execution are an instruction fetching unit that can fetch more than one instruction at a time from cache; instruction decoding logic that can decide when instructions are independent and thus executed simultaneously; and sufficient execution units to be able to process several instructions at one time. Note that the execution units may be pipelined, e.g. they may be floating point adders or multipliers, in which case the cycle time for each stage matches the cycle times on the fetching and decoding logic. In many systems the high level architecture is unchanged from earlier scalar designs. The superscalar designs use instruction level parallelism for improved implementation of these architectures.

Question: Tell me something about the Processing of Control Transfer Instructions .

Answer: In the binary file, there may be gaps in between the instructions due to the alignment constraints. Control of the program execution never reaches to such gaps. The flow of a program is affected by the control transfer instructions. We have divided the control transfer instructions under six categories, namely unconditional and conditional jump instructions, unconditional and conditional call (to a procedure) instructions and lastly unconditional and conditional return (from a procedure) instructions. The process of disassembly takes care of such instructions. An occurance of an instruction of such type is used to identify the address ranges that contains the code. Otherwise, disassembled instructions sequence might be completely wrong. Therefore, we need the information about all such instructions.

Instructions under each category can be found by a simple method. The method is based on the assumption that instructions are described in a hierarchical manner in the processor specification. If a complete instruction specification tree is made, then instruction of a category can be marked under a subtree i.e. an instruction is put under a particular category if the root node of the corresponding subtree is traversed during flattening of the instruction. If a processor specification is not written in this manner, then a little effort is needed to modify it.

Question: Tell me something about Pipelined Processors and its Types .

Answer: Pipelined Processors are consisting of the method of processing on a computer that allows fast parallel processing of data. This is accomplished by overlapping operations using a pipe(A portion of memory that can be used by one process to pass information along to another. Essentially, a pipe works like its namesake: It connects two processes so that the output of one can be used as the input to the other), or a portion of memory that passes information from one process to another.

Question: What is meant by Pipelining & Repilication ?

Answer: Pipelining is a method of fetching and decoding instructions (preprocessing) in which, at any given time, several program instructions are in various stages of being fetched or decoded. Ideally, pipelining speeds execution time by ensuring that the microprocessor does not have to wait for instructions; when it completes execution of one instruction the next is ready and waiting. In parallel processing, a method in which instructions are passed from one processing unit to another, as on an assembly line, and each unit is specialized for performing a particular type of operation.

Replication means to copy. In a distributed database management system, the process of copying the database (or parts of it) to the other parts of the network. Replication allows distributed database systems to remain synchronized.

Question: Tell me about VILW Architecture in detail.

Answer: Very-Long Instruction Word (VLIW) architectures are a suitable alternative for exploiting instruction-level parallelism (ILP) in programs, that is, for executing more than one basic (primitive) instruction at a time. These processors contain multiple functional units, fetch from the instruction cache a Very-Long Instruction Word containing several primitive instructions, and dispatch the entire VLIW for parallel execution. These capabilities are exploited by compilers which generate code that has grouped together independent primitive instructions executable in parallel. The processors have relatively simple control logic because they do not perform any dynamic scheduling nor reordering of operations (as is the case in most contemporary superscalar processors).

VLIW has been described as a natural successor to RISC, because it moves complexity from the hardware to the compiler, allowing simpler, faster processors.

The objective of VLIW is to eliminate the complicated instruction scheduling and parallel dispatch that occurs in most modern microprocessors. In theory, a VLIW processor should be faster and less expensive than a comparable RISC chip.

The instruction set for a VLIW architecture tends to consist of simple instructions (RISC-like). The compiler must assemble many primitive operations into a single "instruction word" such that the multiple functional units are kept busy, which requires enough instruction-level parallelism (ILP) in a code sequence to fill the available operation slots. Such parallelism is uncovered by the compiler through scheduling code speculatively across basic blocks, performing software pipelining, reducing the number of operations executed, among others.

VLIW has been perceived as suffering from important limitations, such as the need for a powerful compiler, increased code size arising from aggresive scheduling policies, larger memory bandwidth and register-file bandwidth, limitations due to the lock-step operation, binary compatibility across implementations with varying number of functional units and latencies. In recent years, there has been significant progress regarding these issues, due to general advances

in semiconductor technology as well as to VLIW-specific activities. For example, our tree-based VLIW architecture provides binary compatibility for VLIW implementations of varying width, and our VLIW compiler contains state-of-the-art parallelizing/optimizing algorithms.

Question: How a debugger disassemble the machine language? What is mean by functional unit ?What is source and destination operand?

Answer: Debuggers offer more sophisticated functions such as running a program step by step ('single-stepping'), stopping ('breaking') (pausing the program to examine the current state) at some kind of event, and tracking the values of some variables. Debuggers have the ability to modify the state of the program while it is running, rather than merely to observe it.

Functional Unit is an entity of hardware, software, or both, capable of accomplishing a specified purpose. It is a subsystem of the central processing unit of a computer. For example, arithmetic and logic unit, memory address register, barrel shifter, register file.

Source operand is an argument of an operator or of a machine language instruction which is used in the instruction operation.

In destination operand, the result is stored or where the resulting is pointing.

Question: What are FLYNN's Classification and Proposed Classification?

Answer: Flynn uses the stream concept for describing a machine's structure. A stream simply means a sequence of items (data or instructions). Four main types of machine organizations can be found:

SISD (Singe-Instruction stream, Singe-Data stream)

SISD corresponds to the traditional mono-processor (von Neumann computer). A single data stream is being processed by one instruction stream.

SIMD (Singe-Instruction stream, Multiple-Data streams)

In this organization, multiple processing units of the same type process on multiple-data streams . This group is dedicated to array processing machines. Sometimes, vector processors can also be seen as a part of this group.

MISD (Multiple-Instruction streams, Singe-Data stream)

In case of MISD computers, multiple processing units operate on one single-data stream. In practice, this kind of organization has never been used.

MIMD (Multiple-Instruction streams, Multiple-Data streams)

This last machine type builds the group for the traditional multi-processors. Several processing units operate on multiple-data streams.

There is no such specific parallel classification named as proposed classification. More likely, it is termed as the most recommended classification used for parallel architectures. The explicit architectures, for example, SIMD and MIMD are proposed as in reconfigurable architecture as they are prototyped more easily and ahead of their time.

Question: What is Programmed I/O?

Answer: Programmed I/O is used with devices that preform operations that take many instruction execution times and do not deliver or demand quantities of data larger than 1 word in a high-speed burst. Printers would fall into this category.

Question: What is Virtual Memory?

Answer: Virtual memory is a technique of using secondary storage such as disks, to extend the apparent size of physical memory. It permits each process to use the main memory as if it were the only user, and to extend the apparent size of accessible memory beyond its actual physical size.