**1. FireWire as an alternative to SCSI**

FireWire Serial bus: A high-speed alternative to SCSI and other small-system I/O interfaces. FW is defined by the IEEE standard 1394 for a high-performance serial bus.

Advantages: Very high speed, low cost and easy to implement, Also used in consumer electronics (Digital cameras and TVs) – transports video images coming from digitized sources, provides a single I/O interface with a simple connector – a single port.

FireWire configurations: Daisy-chain configuration with up to 63 devices connected to a single port, up to 1022 FireWire buses can be interconnected using bridges, automatic configuration, hot plugging – it is possible to connect or disconnect peripherals without switching off the CS or reconfiguring it, FireWire bus need not be a strict daisy chain, a tree-structured configuration is possible.

**2. RISC vs. CISC**

A CISC is a computer ISA in which each instruction can execute several low-level operations, such as a load from memory, an arithmetic operation, and a memory store, all in a single instruction. RISC represents a CPU design strategy emphasizing the insight that simplified instructions that "do less" may still provide for higher performance if this simplicity can be utilized to make instructions execute very quickly. CISC microprocessors aim to achieve speed by having such a rich instruction set that it takes fewer instructions to complete a task. RISC microprocessors require more instructions to complete a given task, but they focus on reducing the number of cycles it takes to complete an instruction. While CISC instructions vary in length, RISC instructions are all the same and can be fetched in a single operation.

**3. Speedup laws: Amdahl's law, Gustafson's law, Sun and Ni's law**

- If the purpose is to reduce the execution time of a fixed-workload problem, the scalability of the system can be defined as the fixed-load speedup, governed by the generalized Amdahl's law: Amdahl’s law says that we must not only reduce the sequential bottleneck, but also increase the average granularity in order to reduce the adverse impact of the overhead.

The performance of a parallel program is limited not only by the sequential bottleneck, but also by the average overhead.

- To achieve the best scaled speedup, fixed-memory scaling should be used, which is governed by Sun and Ni's law:

Generalizes Amdahl’s & Gustafson’s laws to maximize the use of both CPU & memory capacities. Especially for multicomputers – the total memory capacity increases linearly with the # of available nodes.

- Gustafson's fixed-time scaling is a compromise between fixed-load and fixed-memory scaling - guarantees Tpar of the scaled workload won't exceed Tseq of the original workload, barring any overhead: GUSTAFSON’S LAW: FIXED TIME

This concept achieves an improved speedup by scaling the problem size with an increase in machine size! Scaling for higher accuracy – finite-difference method – coarse grids (less accuracy) & finer grids (greater accuracy)

Gustafson’s law state that the fixed-time speedup is a linear function of n, if the workload is scaled up to maintain a fixed execution time.

**4. Reducing branch costs using dynamic hardware prediction**

The idea of dynamic prediction logic is fetching the most likely next instruction after a branch execution. Processors that use dynamic prediction keep a history for each branch and use it to predict future branches. These processors are correct in their prediction 90% of the time. This prediction helps the processor to overcome the stalls (in most cases - 90%)

**5. An interconnection network is characterized by its topology, routing algorithm, switching strategy and flow control mechanism. Discuss details:**

The topology is the physical interconnection structure of the network graph. Direct networks have a host node connected to each switch. Indirect networks have hosts connected only to a specific subset of the switches, which form the edges of the network. Hosts generate and remove traffic. Diameter of the network is the length of the maximum shortest path between any two nodes. Routing distance between a pair of nodes - the number of links traversed along the route. Average distance - over all pairs of nodes (the expected distance between a random pair of nodes). Separating edges - the graph is broken down into two disjoint graphs, Circuit switching - the path from the source to the destination is established and reserved until the message is transferred (used in phone systems). Packet switching - packets are individually routed from the source to the destination (typically allows better utilization of the network resources).

**6. Present the basic outline of the switch architecture for parallel computer interconnection networks**

The network is composed of links and switches. A link is a bundle of wires or fibers that carries an analog signal. A transmitter converts digital info at one end into an analog signal that is driven down the links and converted back into digital signals by the receiver at the other end. The physical protocol for converting between streams of digital symbols and an analog signal forms the lowest layer of the networks design. The transmitter, link and receiver collectively form a channel for digital information flow between switches attached to the link. The link-level protocol segments the streams of symbols crossing a channel into larger logical units, called packet or messages. Switches steer each unit arriving on an input channel to the appropriate output channel. Processing nodes communicate across a sequence of links and switches. The node-level protocol embeds commands for the remote CA within the packets or messages exchanged between the nodes to accomplish network transaction. A channel is a physical link between host or switch elements: it has a width w and a signaling rate f=1/T (for cycle time T), which together determine the channel bandwidth b=w\*f.

The amount of data transferred across a link in a cycle is called a physical unit or phit. Many networks operate asynchronously rather than being controlled by a single global clock. The network cycle time is defined as the time to transmit the smallest physical unit of information, a phit. For parallel architectures it is convenient to think about the processor cycle time and the network cycle time in common terms. The minimum unit of information that can be transferred across a link and either accepted or rejected is called a flow control unit, or flit (as small as a phit or as large as a packet).

**7. Present the standard benchmarks SPEC and TPC for performance estimation of computer systems.**

Business and TPC benchmarks – the most widely used commercial application benchmark; developed by the Transaction Processing Performance Council (database & transaction processing benchmarks). All the benchmarks measure performance in transactions per second, including a response time Requirement. TPC-W is a Web-based transaction benchmark that simulates the activities of a business-oriented transactional Web server exercising the database system as well as the underlying Web server software.

SPEC Benchmark Family – the speed metrics measure the ratios to execute a single copy of the benchmark, while the throughput (rate) metrics measure the ratios to execute multiple copies of the benchmark. The SPEC benchmarks are real programs, modified for portability and the role of I/O is minimized. The integer benchmarks vary from part of a C compiler to a VLSI place-and-route tool to a graphics application. The floating point benchmarks include code for quantum chromodynamics, finite-element modeling, and fluid dynamics. The SPEC CPU suite is useful for CPU benchmarking for both desktop systems and single-processor servers.

**8. Discuss the semantics and performance of the BSP model**

Bulk synchronous parallel model proposed for overcoming the shortcomings of the PRAM model, while keeping its simplicity.

A BSP computer consists of a set of n processor/memory pairs (nodes) interconnected by a communication network. A BSP program has n processes, each residing on a node. The basic time unit is a cycle (or time step). The program executes as a strict sequence of supersteps (computation, communication and barrier synchronization). The communication happens through message passing or shared variable. It accounts for all overhead except the parallelism overhead for process management.

**9.** **Present the structure and functioning of an instruction pipeline. Discuss the data dependencies and hazards causing bubbles within the pipeline**

An instruction pipeline is a technique used in the design of computers and other digital electronic devices to increase their instruction throughput (the number of instructions that can be executed in a unit of time). The fundamental idea is to split the processing of a computer instruction into a series of independent steps, with storage at the end of each step. This allows the computer's control circuitry to issue instructions at the processing rate of the slowest step, which is much faster than the time needed to perform all steps at once. The term pipeline refers to the fact that each step is carrying data at once, and each step is connected to the next.

For example, the RISC pipeline is broken into five stages with a set of flip flops between each stage.

 1. Instruction fetch

 2. Instruction decode and register fetch

 3. Execute

 4. Memory access

 5. Register write back

Pipeline data transfer – information is passed from one stage of the pipeline to the next. There are 2 methods for doing this:

Asynchronous method: a pair of “handshake signals”; provides greatest flexibility; the fastest pipeline; variable length FIFO buffers between stages

Synchronous method: one timing signal; limited by the slowest unit; staging latch between units; instead, path delays can be equalized

Data dependencies (read-after-write hazards, the result of the ith instruction is used as an operand of the (i+1)th instruction, types – true (flow) data dependency, antidependency (write-after-read hazard, an instruction writes to a location which has been read by a previous instruction), output dependency (write-after-write hazard) a sort of a resource conflict).

**10. Define the objectives and functional requirements of computer design.**

Computer design consists of instruction set architecture (ISA), organization and hardware. The instruction set architecture refers to the actual programmer - visible instruction set and serves as the boundary between the software and hardware. The term organization includes the high-level aspects of a computer’s design such as the memory system, the bus structure and the design of the internal CPU. Hardware is used to refer to the specifics of a machine, including the detailed logic design and the packaging technology of the machine.

**11. What is SCSI and how is it used in computer systems?**

SCSI - SMALL COMPUTER SYSTEM INTERFACE: An interface to external peripheral devices, first popularized on the Macintosh in 1984, now widely used on Windows/Intel systems & many workstations, standard interface for CD-ROM drives, audio equipment & external mass storage devices, uses a parallel interface with 8, 16 or 32 data lines.

SCSI is referred to as a bus, in fact devices are daisy chained together, each SCSI device has two connectors for I&O, all devices are chained together, one end of the chain is hooked into the host, all devices function independently and may exchange data with one another or with the host system, data is transferred in message packets.

Asynchronous transfer mode – RFQ/ACK handshake is required for every byte transferred. Synchronous transfer mode must be negotiated. Messages are exchanged between initiators and targets for the SCSI interface

Management (command complete, disconnect, initiator detected error, abort, synchronous data transfer).

The SCSI specification defines commands for direct and sequential access devices, printers, processors, CD-ROMs, scanners, optical memory devices, communication devices.

**12. Present the information fragmentation and encapsulation within the interconnection networks of a parallel computer.**

 

Flit - the minimum unit of information that can be transferred across a link and either accept or rejected is called flow control unit (as small as a phit or as large as a packet); Phit - the amount of data transferred across a link in a cycle is called physical unit

**14. Discuss the semantic and performance attributes of the PRAM model.**

The machine size can be arbitrarily large. The basic timestamp is called cycle. Within a cycle, each processor executes exactly one instruction for example Read, Compute, Write. All processors implicitly synchronize at each cycle. The synchronization overhead is assumed to be zero. The time complexity of most PRAM algorithms is expressed as a function of the size of the problem N and the machine size n .We assume that there are zero communication overhead and instruction – level synchrony.

Semantic attributes:

1. Homogeneity

PRAM(1) – SCSI, PRAM(n) – MIMD, SIMD - of each cycle all processors execute the same instruction,

SPMD - Single-Program-Multiple data – all processors execute the same program, parameterized by the processor index.

2. Synchrony: Synchronization at instruction level. At each cycle, all memory read op`s must be performed before processors may proceed. Real MIMD parallel computers are asynchronous. Each process executes at its own speed, independent of the speed of other processes.

3. Interaction mechanism: Processes interact through shared variables (or shared memory). An asynchronous MIMD machine interacting through shared variables is called a multiprocessor. An asynchronous MIMD machine interacting through message passing is called a multicomputer.

4. Address space: It has single address space. All processes have equal access time to all memory locations - UMA.

5. Memory model: EREW - Exclusive Read Exclusive Write (a memory cell can be read or written by at most one processor at a time). CREW - Concurrent Read Exclusive Write (at each cycle a memory cell can be read by multiple processors, but can be written by at most one processor. CRCW - Concurrent Read Concurrent Write

**16. Present cluster architecture and discuss availability support for clustering**.

Availability support – lots of redundancy of processors, memories, disks, I/O devices, networks, operating system images, etc.

* **Reliability –** measures how long a system can operate without a breakdown
* **Availability –** indicates the percentage of time that a system is available to the user (the percentage of uptime)
* **Serviceability –** how easy it is to service the system, including hardware and software maintenance, repair, upgrade, etc.
1. Shared-nothing architecture – nodes are connected through the I/O bus
2. Shared- disk architecture – for small-scale availability clusters in business applications
3. Shared-memory architecture – the SCI is connected to the memory bus of the nod via NIC module

**17. Present the architectural concept of a superscalar processor with reservation stations**

Each FU has one or more reservation stations. The reservation station holds:

- Instructions that have been issued and are awaiting an execution at a functional unit,

- The operands for that instruction if they have already been computed (or the source of the operands otherwise)

- The information needed to control the instruction once it has begin execution

The reservation stations buffer the operands of instructions waiting to issue, eliminating the need to get the operands from register (similar to forwarding). WAR hazards are avoided because an operand is already stored in reservation station even when a write to the same register is performed out-of-order

**21. How deadlocks are prevented in on interconnected network? How he the head-of-queue blocking is eliminated?**

Deadlocks are prevented via virtual channels. Eliminating head-of-queue blocking:

****

**22. Discuss the cache coherence problem in SMP. State the source of incoherence.**

Cache coherence refers to inconsistency of distributed cached copies of the same cache line addressed from the shared memory.

 Here are some issues:

- A read following a write to X by processor P will return the value written by P

- A read by processor P following a write to X by processor Q will return the value written by Q

- Writes to the same cache line X by different processors are always serialized to prevent the some order of memory events, seen by all processors.

 Sources of the incoherence:

1. The write of different processors into their cached copies of the same cache line in memory, asynchronously

2. Process migration among multiple processors without alerting each other

3. I/O operations bypassing the owners of cached copies

**23. What is ISA? Discuss details**

ISA comes from Instruction Set Architecture. It is one part from the computer design along with the organization and hardware.

ISA refers to the actual programmer-visible instruction set and serves as the boundary between the software and the hardware. An instruction set is a list of all the instructions, and all their variations, that a processor can execute. Instructions include:

 \* Arithmetic \* Logic instructions \* Data instructions \* Control flow instructions

**24. Present the isoperformance models of parallel computer systems**

- Isoefficency- characterizes system scalability E=f (w, n); if w fix efficiency to some constant and solve the efficiency equally for w, the resulting function is called the isoefficiency function of the system.

The isoefficioncy metric provides a useful tool to predict the required workload growth rate with respect to the machine size increase.

-Isospeed - a constant speed is preserved while scaling up both machine size and problem size at the same time. It characterizes system scalability.

-Isoutilization - predicts the workload growth rate with respect to the increase of utilization and it is consistent with execution time i.e., a more scalable system always has a shorter execution time.

**25. Define the terms computer architecture and implementation. Present Flynn's classification**

Computer architecture is defined as the attributes and behavior of computer as seen by a machine language programmer. This definition includes the instruction set, instruction formats, operation codes, addressing modes, and all registers and memory locations that may be directly manipulated by a machine language programmer. Implementation is defined as the actual hardware structure, logic, design, and data path organization of a particular embodiment of the architecture. The implementation of a machine has two components: organization and hardware. The implementation may encompass IC design, packaging, power and cooling.

Flynn’s classification:

- “Single Instruction Flow Single Data Flow” (SISD) comprises computers of von Newman’s architecture

- “Single Instruction Flow Multiple Data Flows” (SIMD) – single instruction flow is executed over multiple data flows on multiple processors

- “Multiple Instruction Flows Multiple Data Flows” (MIMD) encompasses multiprocessors and multi-computers.

- “Multiple Instruction Flows Single Data Flow” (MISD) – controversial

**26.** **What type of snoopy coherence protocols do you know? What is MESI?**

1. Write – invalidate: invalidates all other cached copies when a local cached copy is updated

2. Write – update: broadcasts the newly cached copy to update all other cached copies with the same line address

The MESI is a write-invalidate snoopy protocol. It keeps track of the state of a cache line, considering all read or write, cache hit or cache miss, and snoopy events detected on the bus. Every line in the data cache is always in one off the following 4 possible states:

Modified (M): the cache line is valid and it is not valid in any other cache. The memory line has not been updated yet.

Exclusive (E): the cache line is valid, and it is not valid in any other cache. The memory line has not been updated yet.

Shared (S): the line is valid, but it may also be valid in more than one remote cache or in the cache line in memory.

Invalid (I): an invalid state after reset, or the cache line has been invalidate by a write hit by another cache with the same address.

**27. Discuss the redundant components configurations.**

Redundant components configurations:

•Hot Standby – a primary component provides service, a redundant backup component is ready (hot) to take over when the primary fails (economical design – one standby component to back up multiple primary components)

•Mutual Takeover – all components are primary; 1 component fails – its workload is distributed to other components

• Fault-tolerant – N components deliver the performance of only 1 component

• Failover – the most important feature demanded in current clusters for commercial applications; when a component fails, the remaining system takes over the services originally provided by the failed component; provides failure diagnosis, failure notifications & failure recovery

• Failure diagnosis in a dual-network cluster: each node has a heartbeat daemon that periodically sends a heartbeat message to the master node through both networks - node failed, connection to a network failed

**28. Present single point of failure in computer cluster**

Single point of failure is a hardware or software component whose failure would bring the entire system down.

**29. Present the basic physical machine model of MIMD architecture class**

The MIMD architectural class encompasses 5 physical machine models: parallel vector processors, symmetric multiprocessors, massively parallel processors, distributed shared memory machines, and computer clusters (clusters of workstations).

**30. Discuss the factors determining the communication performance.**

**31. Describe the architecture of a super-server. Describe the architecture of megastation**

Cluster of homogeneous SMP servers is called a super-server. Super-servers combine message passing and the shared memory interaction mechanisms – hybrid programming models.

Mega-station is clustering of many workstations. We get a single system that is equivalent to one huge workstation.

**32.**  **Discuss the semantic and performance attributes of the phase-parallel model**

A parallel program is executed as a sequence of phases. The next phase cannot begin until all operations in the current phase have finished. There are three types of phases: parallelism phase (process management), computation phase, interaction phase (communication, synchronization or aggregation). The parallel model is closer to covering real machine/program behavior.

All types of overheads are accounted for the load imbalance overhead (the σ term), the interaction overhead (the t0 and tc terms), the parallelism overhead (the tp term)

**33.** **Discuss the purpose and the organization of the instruction window and the reorder buffer in contemporary processor architecture.**

There are 2 design types of superscalar processors. One of them is designed with specialized execution units. One of the units is namely instruction window. It is implemented in 2 ways – centralized window and distributed window.

The instructions are issued from the window when the operands of the instructions are available and the respective functional unit is free.

Instruction window contents:

|Instruction| Opcode | Register Destination Id| Operand 1| Register operand 1 ID| Operand2| register operand2 ID|

The main purpose of the instruction window is to keep the instructions ready to be used when needed just taken no need to wait for their fetching from memory.

Reorder buffer – consists of FIFO queue with entries that are dynamically located to instruction register results. When an instruction which writes to a register is decoded, an entry is allocated at the top of the queue. Instruction results are written to the reorder buffer entry.

 When the entry reaches the bottom of the queue, and the value has been written, it is transferred to the register file

**34. Present the essence of snoopy coherence protocols**

These protocols are important when we need to apply coherence control in writing of shared data, process migration, and I/O operations in a multi-processor environment.

Two classes of coherence protocols implemented with snoopy buses, monitoring the caching events across the bus

1. write–invalidate: invalidates all other cached copies when a local cached copy is updated

2. write–update: broadcasts the newly cached copy to update all other cached copies with the same line address.

**35.** **What do you know about clusters of SMP servers?**

Nowadays there is a visible industrial trend to cluster a number of homogenous SMP servers together as an integrated superservers.

I know about one superserver called SGI Power Challenge array.

The system links 2-8 SMP nodes, called POWER nodes, to form a cluster as a superserver. Each powernode is a Power Challenge SMP server with up to 36 MIPS R10000 processors and 16GB shared memory.

In total the cluster can have up to 128GB of main memory, more than 4 GB/s of sustained disk transfer capacity.

The nodes are interconnected by a crossbar HIPPI switch to support high bandwidth communication.

The system can be accessed via an Ethernet from Indy workstations (proizvoditel e na pc).

**36. Present the essence of hyper-threading and its implementation in Pentium 4 and Xeon.**

Intel’s hyper-threading Technology brings simultaneous multi-threading to the Intel architecture and makes a single physical processor appear as 2 logical processors with duplicated architecture state, but with shared physical execution resources. This allows 2 tasks (2 threads from a single application or 2 separate applications) to execute in parallel, increasing processor utilization and reducing the performance impact of memory latency by overlapping the memory latency of the one task with the execution of another. Hyper-threaded capable processors offer significant performance improvements for multi-threaded and multi-tasking workloads without sacrificing compatibility with existing software or single threaded performance.

XEON goals:

1. Minimizing the area cost of implementing Hyper-threading technology

2. Ensuring that when one logical processor is stalled other logical processor could continue to make forward progress

3. Allowing a processor running only one active software thread to run at the same speed on a processor with hyper-threading technology as on a processor without this capability. This means that partitioned resources should be recombined when only one software thread is active. All this is possible with the pipelining.

**37. How the components of the CPU are measured and modeled?**

**38. Summarize the standard desktop, server and transaction processing benchmarks for performance estimation of computer systems.**

DESKTOP BENCHMARKS

2 broad classes: CPU – intensive benchmarks & graphics – intensive benchmarks

The SPEC benchmarks are real programs, modified for portability and the role of I/O is minimized. The integer benchmarks vary from part of a C compiler to a VLSI place-and-route tool to a graphics application. The floating point benchmarks include code for quantum chromodynamics, finite-element modeling, and fluid dynamics. The SPEC CPU suite is useful for CPU benchmarking for both desktop systems and single-processor servers.

SERVER BENCHMARKS

Measurement SPECrate – a simple throughput-oriented benchmark is constructed by running multiple copies of each SPEC CPU benchmark on multiprocessors; SPEC offers SPECSFS (a file server benchmark) using a script of file server requests; it tests the performance of the I/O system (both disk & I/O) as well as the CPU; SPECWeb is a Web server benchmark that simulates multiple clients requesting both static & dynamic pages from a server, as well as clients posting data to a server.

TRANSACTION-PROCESSING (TP) BENCHMARKS

All the benchmarks measure performance in transactions per second, including a response time Requirement. TPC-W is a Web-based transaction benchmark that simulates the activities of a business-oriented transactional Web server exercising the database system as well as the underlying Web server software.

**39. Present the structure, the commands and the arbitration mechanism of the PCI bus. Discuss the advantages of multiple-bus hierarchies.**

Structure: PCI may be configured as a 32- or 64-biit bus

49 mandatory signal lines: System pins (clock & reset), address & data pins: 32 multiplexed lines, interface control pins: transaction timing & control, coordination off initiators & targets, arbitration pins: individual bus arbiter, error – reporting pins: parity, etc.

51 optional signal lines: Interrupt pins, cache support pins, 64-biit extension pins, boundary scan pins

Arbitration: Centralized synchronous scheme, each master has a unique REQ & GRANT signals – handshake to grant access to the bus, arbitration (for each transaction) algorithms alternatives: first-come-first served,, round robin, priority scheme), hidden arbitration.

Commands:

• Interrupt acknowledge (for interrupt controllers)

• Special cycle (initiator broadcasts a message to one or more targets)

• I/O read & I/O write (initiator I/O controller)

• Memory read (transfer of a burst of data, bursting ½ or less of a cache line, supports writing back a line to memory)

• Memory read line (bursting > ½ to 3 cache lines)

• Memory read multiple (bursting >3 cache lines)

• Memory write (at least 1 cache line)

• Configuration read & write (a master reads & updates configuration parameters of a PCI device – 256 internal registers)

• Dual address cycle (64 bits)

The bus length increases -> greater propagation delay.

**40. Make a comparison of store-and-forward and cut-through routing in interconnection networks of parallel computers. Discuss their advantages and disadvantages.**

Store-and-forward is a telecommunications technique in which information is sent to an intermediate station where it is kept and sent at a later time to the final destination or to another intermediate station. The intermediate station verifies the integrity of the message before forwarding it. In general, this technique is used in networks with intermittent connectivity, especially in the wilderness or environments requiring high mobility. It may also be preferable in situations when there are long delays in transmission and variable and high error rates, or if a direct, end-to-end connection is not available. No real-time services are available for this kind of networks.

In computer networking, cut-through switching is a switching method for packet switching systems, wherein the switch starts forwarding a frame (or packet) before the whole frame has been received, normally as soon as the destination address is processed. This technique reduces latency through the switch, but decreases reliability. In packet switched networks such as Ethernet, pure cut-through switching can only be used where the speed of the outgoing interface is less than or equal to the incoming interface speed. The main reason for not adopting cut-through routing is queuing backlog. Cut-through switching is very popular in InfiniBand networks, since these are often deployed in environments where latency is a prime concern, such as supercomputer clusters.

**41. Define the term superscalar processor. What kinds of instruction pipeline hazards do you know?**

Superscalar processors operate upon single operands such as integers; they execute more than one scalar instruction concurrently; require more than one instruction pipeline; fetch more than one instruction simultaneously. Design 1 – duplicate the pipelines - Pentium – pipeline V for simple instructions and pipeline U for complex (microcoded) instructions. Design 2 – integrated approach – superscalar design with specialized execution units: Instruction buffers – hold instructions waiting for their operands, linear pipelines – in-order issue of instructions, pipelines with specialized execution units – out-of-order issue of instructions – require in-order completion of instructions.

Resource conflicts (Concurrent requests of the same resource at the same time, resolved by duplicating the resource), Procedural dependencies (Execution order is unknown prior to instruction execution, main cause are branch instructions, strategies – duplicating instruction buffers, dynamic/static logic prediction, delayed branch instructions), Data dependencies (read-after-write hazard, the result of the ith instruction is used as an operand of the (i+1)th instruction, types – true (flow) data dependency, antidependency (write-after-read hazard, an instruction writes to a location which has been read by a previous instruction), output dependency (write-after-write hazard) a sort of a resource conflict).