Chapter 8: CPU and Memory Design Enhancement and Implementation

1) CPU architecture is defined by the basic characteristics and major features of the CPU. “CPU architecture” is sometimes called

a) architecture design

b) structural organization

c) instruction set architecture

d) CPU design and organization

2) The use of fixed-length, fixed-format instruction words with the op code and address fields in the same position for every instruction would allow instructions to be fetched and decoded

a) independently.

b) dependently and in parallel.

c) independently and in serial.

d) independently and in parallel.

3) There are several factors that determine the number of instructions that a computer can perform in a second. Which of the following is NOT a factor?

a) Word size

b) Clock speed

c) Instruction format - fixed or variable

d) Number of steps required by each instruction type

4) The \_\_\_\_\_\_\_\_\_\_ must be designed to assure that each step of the instruction cycle has time to complete before the results are required by the next step.

a) ALU

b) clock cycle

c) Control Unit

d) instruction pointer

5) The fetch unit portion of the CPU consists of an instruction fetch unit and an instruction \_\_\_\_\_\_\_\_\_\_\_\_ unit.

a) decode

b) translate

c) decipher

d) conversion

6) The \_\_\_\_\_\_\_\_\_\_\_ unit contains the arithmetic/logic unit and the portion of the control unit that identifies and controls the steps that comprise the execution part for each different instruction.

a) fetch

b) decode

c) execution

d) conversion

7) Overlapping instructions—so that more than one instruction is being worked on at a time—is known as the

a) conveyor belt method.

b) pipelining method.

c) assembly line method.

d) accelerator method.

8) Instruction reordering makes it possible to provide parallel pipelines, with duplicate CPU logic, so that multiple instructions can actually be executed

a) sequentially.

b) consecutively.

c) simultaneously.

d) very fast in serial operation.

9) Which of the following is not a specific execution unit?

a) steering unit

b) LOAD/STORE unit

c) integer arithmetic unit

d) floating point arithmetic unit

10) A(n) \_\_\_\_\_\_\_\_\_\_\_\_\_ processor is one that can complete an instruction with each clock tick.

a) linear

d) direct

c) scalar

d) express

11) There are a number of difficult technical issues that must be resolved to make it possible to execute multiple instructions simultaneously. One of the most important of these is

a) Instructions completing out of order.

b) Instructions that have floating point operations.

c) Instructions that can be serialized.

d) Instructions that require the same number of CPU cycles complete.

12) Out-of-order instruction execution can cause problems because a later instruction may depend on the results from an earlier instruction. This situation is known as a \_\_\_\_\_\_\_\_\_\_ or a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

b) risk, reliance

a) hazard, reliance

d) risk, dependency

c) hazard, dependency

13) CPUs can actually search ahead for instructions without apparent dependencies, to keep the execution units busy. Current Intel x86 CPUs, can search \_\_\_\_\_\_\_\_\_\_\_ instructions ahead, if necessary, to find instructions available for execution.

a) five to ten

b) ten to twenty

c) twenty to thirty

d) fifty to one hundred

14) Branch instructions must always be processed ahead of subsequent instructions. Conditional branch instructions are more difficult than unconditional branches. These types of dependencies are known as control dependencies or sometimes as \_\_\_\_\_\_\_\_\_\_\_\_\_\_ or branch dependencies.

a) flow

b) decision

c) qualified

d) provisional

15) Some systems provide a small amount of dedicated memory built into the CPU that maintains a record of previous choices for each of several branch instructions that have been used in the program being executed to aid in determining whether a branch is likely to be taken. What are the contents of this memory called?

a) look-ahead table

b) branch history table

c) branch prediction table

d) future speculation table

16) What are the slowest steps in the instruction fetch-execute cycle?

a) Slowest steps are those that require memory access.

b) Slowest steps involve incrementing the instruction pointer.

c) Slowest steps are those that require special integer register access.

d) Slowest steps are those that require floating point register access.

17)What is the major drawback of Dynamic RAM (DRAM)?

a) cost

b) capacity

c) data loss

d) memory latency

18) Which of the following is a commonly used approach for improving performance of memory?

a) Doubling the capacity of memory.

b) Using DRAM instead of SDRAM.

c) Compressing instructions and data in RAM.

d) Widening the system bus between memory and the CPU.

19) Another method for increasing the effective rate of memory access is to divide memory into parts, called, \_\_\_\_\_\_\_\_\_\_\_\_\_ so that it is possible to access more than one location at a time.

a) block separation

b) high-low separation  
c) memory interleaving

d) wide-path separation

20) Each block of cache memory provides a small amount of storage, perhaps between 8 and 64 bytes, also known as

a) a cache hit.

b) niche cache.

c) a cache line.

d) a small block cache.

21) What does "locality of reference" mean?

a) most memory references occur concurrently

b) most memory references will pull data of numeric type

c) most memory references will be accessed in a predictable order

d) most memory references are confined to one or a few small regions of memory

22) The locality of reference principal makes sense because most of the instructions being executed at a particular time are

a) register-to-memory type instructions.

b) math and logical type instructions.

c) control and branch type instructions.

d) part of a small loop or a small procedure or function.

23) Cache memory hit ratios of \_\_\_\_\_\_\_\_ percent and above are common with just a small amount of cache.

a) 30

b) 60

c) 80

d) 90

24) When a cache miss occurs, however, there is a time delay while new data is moved to the cache. The time to move data to the cache is called \_\_\_\_\_\_\_\_\_\_\_\_\_ time.

a) stall

b) backup

c) write-through

d) cache back

25) Which of the following is most likely:

a) L1 cache has 32KB and L2 cache has 1MB

b) L1 cache has 1MB and L2 cache has 32KB

c) L1 cache has 32KB and L2 cache has 32KB

d) L1 cache has 1MB and L2 cache has 1MB

26) A part of main memory can be allocated to store several adjoining blocks of disk memory. If the requested data is in \_\_\_\_\_\_\_\_\_ then no disk access is necessary.

a) disk cache

b) cache blocks

c) read once cache

d) buffer disk cache

27) Instructions, fetched from memory, are \_\_\_\_\_\_\_\_\_\_\_\_\_within the instruction unit, to determine the type of instruction that is being executed. This allows branch instructions to be passed quickly to the branch processing unit for analysis of future instruction flow.

a) partially decoded

b) partially executed

c) completely decoded

d) completely executed

28) In a superscalar CPU, the instruction unit has a(n) \_\_\_\_ to hold instructions until the required type of execution unit is available.

a) pipeline

b) assembly unit

c) instruction set

d) cache memory

29) Computers that have multiple CPUs within a single computer, sharing some or all of the system's memory and I/O facilities, are called\_\_\_\_\_\_\_\_\_\_\_\_\_\_, or sometimes tightly coupled systems.

a) bundled systems

b) simultaneous systems

c) multiprocessor systems

d) compound processor systems

30) Under ideal conditions, each CPU processes its own assigned sequence of program instructions

a) independently of other CPUs.

b) partially sharing the workload with other CPUs.

c) without interrupting the other CPUs.

d) by sharing L1 cache between other CPUs.

31) Each CPU in the processor, within a single integrated chip, is called a \_\_\_\_\_\_\_\_\_\_\_

a) core.

b) CPU unit.

c) control unit.

d) Independent Processor Chip (IPC).

32) Which of the following is not an advantage of adding more than one CPU processor within a single integrated chip?

a) Relatively inexpensive.

b) Reduce resource conflicts.

c) Programs can be divided into independent pieces and run separately.

d) Reduces power consumption, heat, and stress but gives equivalent processing power.

33) What is a "thread"?

a) The same segment of code used by many programs.

b) Independent segments of programs available to be executed in parallel.

c) The set of all variables that are used by all programs in execution.

d) Shared allocation of cache memory used by programs available to be executed.

34) In Symmetrical Multiprocessing (SMP) each CPU has

a) identical access to memory.

b) identical access to the I/O and memory.

c) identical access to the operating system, I/O and memory

d) identical access to the operating system, and to all system resources, including memory.

35) Simultaneous thread multiprocessing (STM) is also known as \_\_\_\_\_\_\_\_\_\_\_

a) hyperthreading.

b) superthreading

c) expert threading

d) concurrent threading

Chapter 8 Discussion Questions

1) What are common characteristics included in the instruction set architecture (ISA)?

Sol: From the text: “These characteristics include such things as the number and types of registers, methods of addressing memory, and basic design and layout of the instruction set.”

2) What are three important CPU architectural families, today?

Sol: From the text: “At present, important CPU architectural families include the IBM mainframe series, the Intel x86 family, the IBM POWER/PowerPC architecture, the ARM architecture, and the Oracle SPARC family.”

3) Having large numbers of specialized instructions inhibited efficient organization of the CPU. Why?

Sol: From the text: “Specialized instructions were used rarely, but added hardware complexity to the instruction decoder that slowed down execution of the other instructions that are used frequently.

4) What is the benefit of having fixed-length instructions over variable-length instructions?

Sol: From the text: “The use of fixed-length, fixed-format instruction words with the op code and address fields in the same position for every instruction would allow instructions to be fetched and decoded independently and in parallel. With variable-length instructions it is necessary to wait until the previous instruction is decoded in order to establish its length and instruction format.”

5) Describe the technique called code-morphing.

Sol: From the text: “Furthermore, a technique called code-morphing can be used to translate complex variable-width instruction words to simpler fixed-width internal equivalents for faster execution. This technique allows the retention of legacy architectures while permitting the use of modern processing methods. Modern x86 implementations use this approach.”

6) Summarize how separating the fetch-execution cycle into different fetch and execution units improves performance.

Sol: From the text: “Implementation of the fetch-execute cycle is divided into two separate units: a fetch unit to retrieve and decode instructions and an execution unit to perform the actual instruction operation. This simple reorganization of the CU and ALU components allows independent, concurrent operation of the two parts of the fetch-execute cycle.”

7) Summarize how pipelining improves performance.

Sol: From the text: “The model uses an assembly line technique called pipelining to allow overlapping between the fetch-execute cycles of sequences of instructions. This reduces the average time needed to complete an instruction.” Or “Pipelining allows several instructions to be ‘in flight’ at the same time.”

8) Summarize the performance benefits of using separate execution units for different types of instructions.

Sol: From the text: “The model provides separate execution units for different types of instructions. This makes it possible to separate instructions with different numbers of execution steps for more efficient processing. It also allows the parallel execution of unrelated instructions by directing each instruction to its own execution unit.”

9) Do pipelining and superscalar processing techniques affect the number of clock cycles of any individual instruction? Explain your answer.

Sol: No they do not. From the text: “It is important to remember that pipelining and superscalar processing techniques do not affect the cycle time of any individual instruction. An instruction fetch-execute cycle that requires six clock cycles from start to finish will require six clock cycles whether instructions are performed one at a time or pipelined in parallel with a dozen other instructions. It is the average instruction cycle time that is improved by performing some form of parallel execution.”

10) Superscalar processing complicates the design of a CPU considerably. There are a number of difficult technical issues that must be resolved to make it possible to execute multiple instructions simultaneously. What are the most important issues?

Sol: From the text:

Problems that arise from instructions completing out of order

Changes in program flow due to branch instructions

Conflicts for internal CPU resources, particularly general-purpose registers

11) How does the cache controller know when to use cache or go to main memory?

Sol: From the text: “Every CPU request to main memory, whether data or instruction, is seen first by cache memory. A hardware cache controller checks the tags to determine if the memory location of the request is presently stored within the cache.”

12) Caching recently used data and instructions is an important performance enhancement. Summarize the benefits cache.

Sol: selected from the subsection on Cache Memory: Main memory access is fast, but still slower than the speed of the CPU. By storing recently used data and instruction in cache, the time required to fulfill a memory request is decreased, thereby increasing overall performance. This improvement is significant because cache hit ratios are commonly over 90%.

13) If a cache block is altered, what methods are used to update main memory? Of the various methods, which is faster?

Sol: from the text: “Two different methods of handling the process of returning changed data from cache to main storage are in common use. The first method, write through, writes data back to the main memory immediately upon change in the cache. This method has the advantage that the two copies, cache and main memory, are always kept identical. Some designers use an alternative technique known variously as store in, write back, or copy back. With this technique, the changed data is simply held in cache until the cache line is to be replaced.”

Sol: from the text: “The write back method is faster…”

14) Summarize the principle of “locality of reference”?

Sol: From the text: “The locality of reference principle states that at any given time, most memory references will be confined to one or a few small regions of memory.”

15) Increasing the number of CPUs is usually an effective way to improve performance although, as the number of CPUs increases, the value of the additional CPUs diminishes. Why is that?

Sol: From the text: “…because of the overhead required to distribute the instructions in a useful way among the different CPUs and the conflicts among the CPUs for shared resources, such as memory, I/O, and access to the shared buses. With the exception of certain, specialized systems, there are rarely more than sixteen CPUs sharing the workload in a multiprocessing computer; more commonly today, a multiprocessor might consist of two, four, or eight core CPUs within a single chip.”

Solutions

|  |  |  |
| --- | --- | --- |
| Problem | Answer | Section in text / comments |
| 1 | c | Section 8.1 CPU Architectures |
| 2 | d | Section 8.1 CPU Architectures |
| 3 | a | Section 8.1 CPU Architectures |
| 4 | a | Section 8.2 CPU Features and Enhancements |
| 5 | c | Section 8.2 CPU Features and Enhancements |
| 6 | b | Section 8.2 CPU Features and Enhancements |
| 7 | c | Section 8.2 CPU Features and Enhancements |
| 8 | c | Section 8.2 CPU Features and Enhancements |
| 9 | a | Section 8.2 CPU Features and Enhancements |
| 10 | c | Section 8.2 CPU Features and Enhancements |
| 11 | a | Section 8.2 CPU Features and Enhancements |
| 12 | c | Section 8.2 CPU Features and Enhancements |
| 13 | b | Section 8.2 CPU Features and Enhancements |
| 14 | a | Section 8.2 CPU Features and Enhancements |
| 15 | b | Section 8.2 CPU Features and Enhancements |
| 16 | a | Section 8.3 Memory Enhancements |
| 17 | d | Section 8.3 Memory Enhancements |
| 18 | d | Section 8.3 Memory Enhancements |
| 19 | c | Section 8.3 Memory Enhancements |
| 20 | c | Section 8.3 Memory Enhancements |
| 21 | d | Section 8.3 Memory Enhancements |
| 22 | d | Section 8.3 Memory Enhancements |
| 23 | d | Section 8.3 Memory Enhancements |
| 24 | a | Section 8.3 Memory Enhancements |
| 25 | a | Section 8.3 Memory Enhancements |
| 26 | a | Section 8.3 Memory Enhancements |
| 27 | a | Section 8.4 The Complete Modern Superscalar CPU |
| 28 | a | Section 8.4 The Complete Modern Superscalar CPU |
| 29 | c | Section 8.5 Multiprocessing |
| 30 | a | Section 8.5 Multiprocessing |
| 31 | a | Section 8.5 Multiprocessing |
| 32 | b | Section 8.5 Multiprocessing |
| 33 | b | Section 8.5 Multiprocessing |
| 34 | d | Section 8.5 Multiprocessing |
| 35 | a | Section 8.5 Multiprocessing |