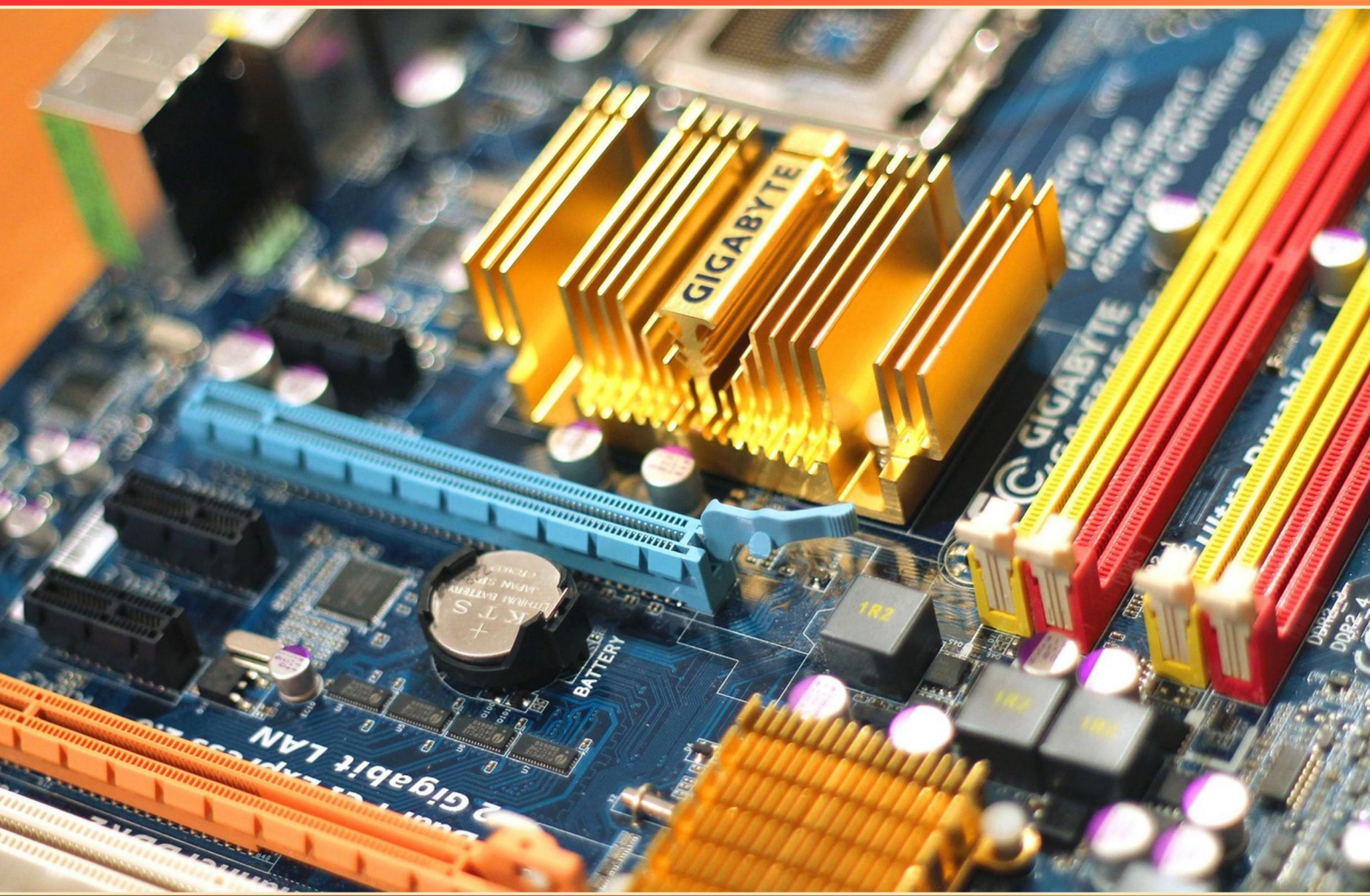
WESTERN GOVERNORS UNIVERSITY (WGU) ICSC3120 C952 COMPUTER ARCHITECTURE PRACTICE EXAM (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://wgu-icsc3120-c952.examzify.com

Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright 1
Table of Contents 2
Introduction 3
How to Use This Guide 4
Questions 5
Answers 8
Explanations 10
Next Steps 15

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

SAMPLE

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

SAMPLE

Questions

SAMPLE

1. What does RAID 6 offer that RAID 5 does not?

A. Increased speed

B. Better disk capacity

C. Fault tolerance for two drive failures

D. Lower hardware cost

2. In the context of finite-state machines, what does a next-state machine determine?

A. The output based on external signals

B. The immediate input requirements of a logic gate

C. The next state based on the current state and inputs

D. The final output after computation

3. What characterizes the Allocate state in memory management?

A. Data is exclusively written to a cache

B. A read operation has been completed

C. A write to the main memory is currently occurring

D. Processor is idle and not performing operations

4. What does sign-extending a data item entail?

A. Replicating the low-order sign bit

B. Increasing the size by adding zeros

C. Replicating the high-order sign bit

D. Reducing the size of data item

5. What situation does aliasing refer to in virtual memory?

A. Multiple physical addresses accessing a single object

B. Two addresses accessing the same object

C. Virtual addresses being mapped incorrectly

D. A memory leak during process execution

6. What operation does the instruction MVZ perform?

A. Move variable with zero

B. Move value with zero

C. Move wide with zero

D. Move word with zero

SAMPLE

7. Which of the following best describes a combinational element in computer architecture?

A. A memory storage device

B. An operational element like an AND gate

C. A signal controller

D. A processing unit

8. What is the purpose of an address in computer architecture?

A. To define operations to be performed

B. To delineate the location of a specific data element within memory

C. To store data temporarily

D. To access the CPU cache

9. What is the formula for CPU time?

A. (Instructions) + (CPI) + (Clock Cycle Time)

B. (Instructions) x (CPI) x (Clock Speed)

C. (Instructions) x (CPI) x (Clock Cycle Time)

D. (Clock Speed) / (CPI) + (Instructions)

10. Which LEGv8 field represents the first register source operand?

A. rm

B. shamt

C. rn

D. rd

SAMPLE

Answers

SAMPLE

1. C
2. C
3. B
4. C
5. B
6. C
7. B
8. B
9. C
10. C

SAMPLE

Explanations

SAMPLE

1. What does RAID 6 offer that RAID 5 does not?

A. Increased speed

B. Better disk capacity

C. Fault tolerance for two drive failures

D. Lower hardware cost

RAID 6 offers fault tolerance for two drive failures, which distinguishes it from RAID 5. In RAID 5, data is striped across multiple drives along with parity information distributed among those drives. This configuration allows for the failure of a single drive without losing any data. However, if a second drive fails during the time it takes to replace the first, data may be lost. RAID 6 improves upon this by adding an additional layer of parity. It requires a minimum of four drives and can tolerate two simultaneous drive failures. This increased redundancy ensures that data remains accessible even if two drives fail at the same time, significantly enhancing data protection and reliability for critical systems. The other options do not accurately describe the key advantage of RAID 6 over RAID 5. RAID 6 does not inherently provide increased speed compared to RAID 5; in fact, the additional parity calculations may slightly reduce performance. While RAID 6 does provide better data protection, it doesn't offer better disk capacity than RAID 5, as the extra parity means less usable space for data. Additionally, the requirement for an extra parity block results in a potentially higher hardware cost than a RAID 5 configuration.

2. In the context of finite-state machines, what does a next-state machine determine?

A. The output based on external signals

B. The immediate input requirements of a logic gate

C. The next state based on the current state and inputs

D. The final output after computation

The next-state machine determines the next state based on the current state and inputs by utilizing a defined set of rules or transitions. In a finite-state machine (FSM), the system maintains a current state and changes to a new state based on external inputs and its present condition. This transition fundamentally defines the behavior of the FSM as it processes inputs. In this context, the next-state machine is crucial because it is responsible for reading the current state and the accompanying inputs to decide what the subsequent state will be. This mechanism allows for the implementation of various digital systems, whether they're simple controllers or complex computing processes. Understanding this function is vital for anyone studying computer architecture and FSMs, as it underpins the way these systems operate and respond to changes in their environment.

3. What characterizes the Allocate state in memory management?

A. Data is exclusively written to a cache

B. A read operation has been completed

C. A write to the main memory is currently occurring

D. Processor is idle and not performing operations

The Allocate state in memory management is characterized by a process that is actively managing memory resources. This includes the condition where a write operation is ongoing in the system. When a write to the main memory occurs, it typically means that data is being transferred from a cache or registers to the main memory, thus marking the Allocate state where resources are being set up or managed. While the read operation is a part of memory access processes, it does not indicate the specific Allocate state, which is more focused on the actions taken to create, modify, or assign memory resources. In contrast, being idle or having completed a read operation does not reflect the active management of memory resources that is characteristic of the Allocate state. The active allocation of memory indicates engagement with memory processes, emphasizing ongoing changes or updates, rather than a completed task or inactivity. Thus, the essence of the Allocate state pertains to current operations that are integral to memory handling, especially writes to main memory.

EXAMPLE

4. What does sign-extending a data item entail?

A. Replicating the low-order sign bit

B. Increasing the size by adding zeros

C. Replicating the high-order sign bit

D. Reducing the size of data item

Sign-extending a data item involves replicating the high-order sign bit, which is the leftmost bit that indicates the sign of a number in a signed binary representation. When a smaller signed integer is converted to a larger size, such as from 8 bits to 16 bits, the sign bit is extended to maintain the correct numerical value. This process ensures that negative numbers remain negative and positive numbers remain positive after the data type expansion. For example, if you have an 8-bit signed integer with a binary value of 11111101 (which represents -3 in two's complement), when sign-extending to 16 bits, the resulting value would be 11111111 11111101. Here, the sign bit (1) is replicated into the higher-order bits to keep the intended value intact. This is crucial in computing systems to ensure that arithmetic operations and other computations yield the expected results, especially when transitioning between data sizes.

5. What situation does aliasing refer to in virtual memory?

A. Multiple physical addresses accessing a single object

B. Two addresses accessing the same object

C. Virtual addresses being mapped incorrectly

D. A memory leak during process execution

Aliasing in virtual memory refers to the situation where two different addresses refer to the same object in memory. This can occur in various contexts, such as when multiple pointers or references correlate to the same data structure, leading to discrepancies in how that data is accessed or modified. When two addresses access the same underlying memory location, any changes made through one address will be reflected when accessed through the other. This phenomenon can be particularly important in scenarios involving shared memory, caches, or even in data structures when not properly managed, as it can lead to unexpected behaviors or bugs in programs. The other options touch on related concepts but do not accurately encapsulate the core definition of aliasing. For instance, while multiple physical addresses accessing a single object may suggest a similar idea, it does not explicitly convey the notion of different references pointing to the same data at the same abstraction level. Mismanaged virtual address mapping could lead to incorrect access but is distinct from aliasing itself. Lastly, a memory leak relates to memory that is no longer accessible but has not been released, which is another issue altogether not tied to the idea of aliasing.

SAMPLE

6. What operation does the instruction MVZ perform?

A. Move variable with zero

B. Move value with zero

C. Move wide with zero

D. Move word with zero

The instruction MVZ typically refers to "Move Wide with Zero." In computer architecture, this operation is often associated with moving wider data types (e.g., 32-bit or 64-bit) into a register while ensuring that the higher bits are set to zero. This is particularly common in certain assembly languages or architectures where registers have varying sizes, and a move operation might need to clear out the upper bits to ensure that only the desired data is retained in the lower bits after the operation. The term "wide" in this context emphasizes that the operation is dealing with larger data sizes, and setting the upper bits to zero is important to prevent any unintended sign extension or data corruption when manipulating numbers. This operation is useful when performing arithmetic or logical operations where the data's integrity must be maintained across different register sizes or operations. Therefore, the choice referring to the move operation that deals with 'wide' data types while zeroing out higher register bits aligns with the function of the MVZ instruction.

7. Which of the following best describes a combinational element in computer architecture?

A. A memory storage device

B. An operational element like an AND gate

C. A signal controller

D. A processing unit

A combinational element in computer architecture is defined primarily by its function in processing and output generation. It refers to components like logic gates, which generate outputs based solely on the current inputs they receive, without any memory or feedback from previous inputs. An operational element such as an AND gate exemplifies this behavior perfectly; the output reflects the logical operation being performed directly based on the present input signals. In this context, the other choices do not fit the definition of a combinational element. A memory storage device involves storing and retrieving data, which requires maintaining state; therefore, it operates as a sequential element rather than combinational. A signal controller may manage signals and timing within a system but does not inherently perform logical operations based only on current inputs. Lastly, a processing unit generally includes both combinational and sequential elements, as it processes data and may rely on stored information for its operations, thus straddling the boundary between both types of architecture.

8. What is the purpose of an address in computer architecture?

A. To define operations to be performed

B. To delineate the location of a specific data element within memory

C. To store data temporarily

D. To access the CPU cache

The purpose of an address in computer architecture is to delineate the location of a specific data element within memory. Each data element stored in a computer's memory has a unique address, which provides a way for the CPU or other components to access and manipulate that piece of information. Addresses serve as pointers that allow the system to identify where particular data is stored, enabling efficient retrieval and storage during program execution. Understanding this concept is crucial because it illustrates how memory management and data access are fundamental aspects of computer architecture. The correct interpretation of an address is central to how programs interact with memory hardware, optimizing the performance of applications by ensuring that the processor can quickly locate and access the required data without ambiguity.

SAMPLE

9. What is the formula for CPU time?

A. (Instructions) + (CPI) + (Clock Cycle Time)

B. (Instructions) x (CPI) x (Clock Speed)

C. (Instructions) x (CPI) x (Clock Cycle Time)

D. (Clock Speed) / (CPI) + (Instructions)

The formula for CPU time is derived from how long it takes to execute a given number of instructions, considering both the number of clock cycles required for each instruction and the duration of each clock cycle. The correct answer indicates that to calculate CPU time, one must multiply the number of instructions by the cycles per instruction (CPI), and then multiply that result by the clock cycle time. To elaborate, here's how the components fit together: - The number of instructions indicates how many separate tasks the CPU needs to execute. - The CPI is a metric that tells us how many clock cycles on average it takes to execute each instruction. - The clock cycle time is the duration of one cycle of the clock, reflecting how quickly the CPU can process instructions. When you multiply these together, you get the total CPU time needed to execute all the instructions, as it shows how many total cycles are needed (number of instructions times CPI) and then converts that cycle count into time (by multiplying by clock cycle time). Thus, this approach effectively measures the total time taken for the processor to complete the instructions in a program. Considering these factors, the formulation in the correct option perfectly captures the relationship between these variables in determining the CPU time.

10. Which LEGv8 field represents the first register source operand?

A. rm

B. shamt

C. rn

D. rd

In the context of LEGv8 assembly language, the instruction format typically dictates how various fields in the instruction are utilized. In the case of the first register source operand, the field that represents this is known as "rn." The "rn" field specifically identifies the first source register, which is essential for operations that involve two registers. This source register is often paired with another register (represented by the "rm" field) to perform computations or logical operations, while the result is usually stored in another designated register (typically indicated by the "rd" field). Understanding the role of the "rn" field is crucial for interpreting instructions and understanding how data flows within the CPU during execution. This knowledge is key when considering instruction types, operand fetching, and the overall architecture of the processor's operations.

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://wgu-icsc3120-c952.examzify.com

We wish you the very best on your exam journey. You've got this!

SAMPLE