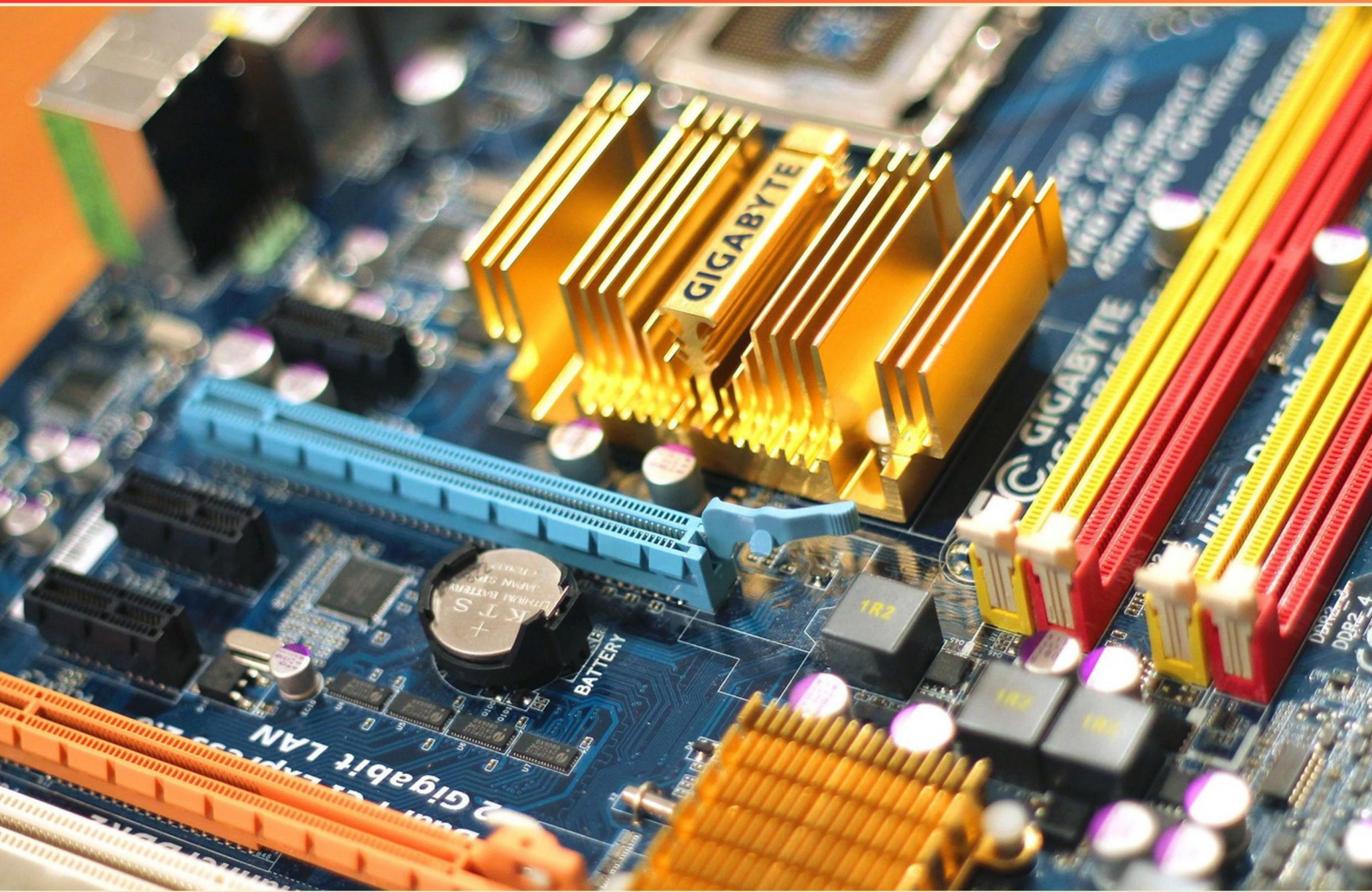


WESTERN GOVERNORS UNIVERSITY (WGU) ICSC3120 C952 COMPUTER ARCHITECTURE PRACTICE EXAM (SAMPLE)

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.

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!

Questions

SAMPLE

- 1. What role does the Special Function Unit (SFU) serve in shading operations?**
 - A. It primarily handles memory allocation
 - B. It computes accurate interpolations for pixels
 - C. It connects the processor to storage
 - D. It executes vertex movements
- 2. What does B.GE indicate when executing a branch?**
 - A. Branch if greater than
 - B. Branch if less than
 - C. Branch if less than or equal to
 - D. Branch if greater than or equal to
- 3. How many bits are utilized for addressing in ARMv8 virtual memory?**
 - A. 32 bits
 - B. 48 bits
 - C. 64 bits
 - D. 16 bits
- 4. Why is a page defined as a fixed-size block in virtual memory systems?**
 - A. To simplify memory allocation and management
 - B. To ensure optimal page retrieval speeds
 - C. To allow support for variable-sized applications
 - D. To minimize storage costs on disks
- 5. What does a very large-scale integrated (VLSI) circuit contain?**
 - A. Tens to hundreds of transistors
 - B. Thousands of transistors
 - C. Hundreds of thousands to millions of transistors
 - D. Only a single transistor
- 6. In the context of branch prediction, what does a correlating predictor do?**
 - A. Combines execution times for branches
 - B. Utilizes global data only
 - C. Integrates local behavior with global trends
 - D. Records execution failures

7. In branching operations, what does the B.LT instruction check for?
- A. Greater than
 - B. Less than
 - C. Equal to
 - D. Not equal to
8. Which instruction set architecture is characterized by operations only between registers?
- A. Accumulator architecture
 - B. Load-Store architecture
 - C. Complex Instruction architecture
 - D. High-level programming architecture
9. What aspect of SIMD is critical for its functionality?
- A. Independent data streams must be operated on
 - B. Identical operations must be applied to each data stream
 - C. It requires multiple instruction types
 - D. It utilizes single-thread processing
10. Which LEGv8 field represents the first register source operand?
- A. rm
 - B. shamt
 - C. rn
 - D. rd

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What role does the Special Function Unit (SFU) serve in shading operations?

- A. It primarily handles memory allocation
- B. It computes accurate interpolations for pixels**
- C. It connects the processor to storage
- D. It executes vertex movements

The Special Function Unit (SFU) plays a crucial role in shading operations by computing accurate interpolations for pixels, which is vital for the rendering of graphics. In graphics processing, shading involves determining the color and brightness of pixels based on various factors such as light sources, textures, and the geometry of objects being rendered. By using interpolations, the SFU can accurately calculate how light interacts with surfaces at various angles and distances, leading to more visually appealing and realistic images. This is especially important in scenarios where detailed shading effects, such as reflections or specular highlights, are required. The SFU is specialized for these types of computations, allowing the main processor to focus on other tasks, thereby improving the overall performance and efficiency of the rendering pipeline. This specialization helps ensure that graphics applications can handle complex calculations involving shading without causing bottlenecks in performance.

2. What does B.GE indicate when executing a branch?

- A. Branch if greater than
- B. Branch if less than
- C. Branch if less than or equal to
- D. Branch if greater than or equal to**

B.GE stands for "Branch if Greater than or Equal to." This condition checks the result of a previous comparison operation, specifically within a signed integer context. When the condition is met (meaning the first operand is greater than or equal to the second operand), the program control is directed to the specified target address. This branch instruction is commonly used in assembly language programming and is part of various instruction sets in computer architecture. By directing program execution based on the condition of the comparison, B.GE can help implement control flow, such as loops and conditional statements, which are fundamental to programming logic. Understanding this instruction is essential as it influences how decisions are made during execution based on arithmetic evaluations, ensuring that algorithms can effectively repeat or skip sections of code depending on data values.

3. How many bits are utilized for addressing in ARMv8 virtual memory?

- A. 32 bits
- B. 48 bits**
- C. 64 bits
- D. 16 bits

In the ARMv8 architecture, the standard implementation uses a 48-bit virtual address space for addressing. This allows for a significantly larger number of addressable locations compared to smaller bit-length addressing schemes. The 48-bit addressing capability is particularly advantageous as it enables efficient memory management and a vast range of possible memory addresses, accommodating the needs of modern applications that require large amounts of memory. While some other architectures or older versions may utilize 32-bit or 64-bit addressing, in the case of ARMv8, 48 bits are specified as the amount used for virtual address space. This is a design choice that aligns with typical application demands and helps facilitate features such as advanced virtual memory management, including paging and other methods of translating logical addresses to physical memory locations.

4. Why is a page defined as a fixed-size block in virtual memory systems?

- A. To simplify memory allocation and management**
- B. To ensure optimal page retrieval speeds
- C. To allow support for variable-sized applications
- D. To minimize storage costs on disks

A page is defined as a fixed-size block in virtual memory systems primarily to simplify memory allocation and management. By using fixed-size pages, the operating system can efficiently track memory usage, allocate memory space, and manage the loading and swapping of pages between physical memory and disk. This uniformity allows for easier implementation of data structures like page tables, which map virtual addresses to physical addresses. It eliminates the complexity of dealing with variable-sized memory segments, making the allocation and deallocation processes more straightforward and reducing fragmentation. In addition to ease of management, fixed-size pages help in speeding up performance because each page is a known size. This consistency aids the operating system in predicting how much physical memory will be required for a given number of pages and helps in optimizing the use of caching and buffering strategies, although that is not the primary reason the fixed size was established. The other options, while they may touch on valid aspects of memory management, do not correctly highlight the primary rationale behind the choice of fixed-size pages. For instance, ensuring optimal page retrieval speeds is a secondary result of using fixed-size pages but is not the foundational reason for their establishment. Similarly, the support for variable-sized applications and minimizing storage costs on disks do not serve as valid justifications for defining

5. What does a very large-scale integrated (VLSI) circuit contain?

- A. Tens to hundreds of transistors
- B. Thousands of transistors
- C. Hundreds of thousands to millions of transistors**
- D. Only a single transistor

A very large-scale integrated (VLSI) circuit is characterized by the integration of hundreds of thousands to millions of transistors onto a single chip. This technology allows for the miniaturization of electronic components and the creation of complex circuits that can perform various functions, such as processing data, managing memory, and driving displays. The high density of transistors enables the development of advanced applications in computing, telecommunications, and consumer electronics, making VLSI a critical component in modern electronic design. The ability to fit so many transistors into a compact space drastically enhances performance and efficiency, leading to smaller and more powerful devices. This scalability is one of the core principles behind the rapid advancements in technology in recent decades.

6. In the context of branch prediction, what does a correlating predictor do?

- A. Combines execution times for branches
- B. Utilizes global data only
- C. Integrates local behavior with global trends**
- D. Records execution failures

A correlating predictor in branch prediction is designed to enhance the accuracy of predicting whether a branch will be taken or not by considering the behavior of previously executed branches along with their global context. This means it assesses both the local history of a specific branch (i.e., how that branch has behaved in the past) and the global trends of other branches in the program. By integrating local behavior with global trends, a correlating predictor can make more informed decisions based on patterns that might not be apparent when looking at a branch in isolation. For example, certain branches may have a tendency to behave similarly when influenced by the outcomes of nearby branches. This creates a more dynamic and effective method for predicting branch outcomes compared to using only local history or relying solely on global data. This predictor type ultimately leads to improved performance in processors by reducing the number of mispredicted branches, which can significantly impact execution speed and efficiency.

7. In branching operations, what does the B.LT instruction check for?

- A. Greater than
- B. Less than**
- C. Equal to
- D. Not equal to

The B.LT instruction is a branch operation commonly found in assembly language and low-level programming that checks for a condition of "less than." When this instruction is executed, it evaluates the result of a previous operation by comparing two values, typically those found in registers or memory locations. If the first operand is indeed less than the second, the program control branches to the specified target address. This operation is essential for implementing decision-making and control flows in programs, allowing for loops and conditional execution based on value comparisons. The "LT" in B.LT stands for "less than," which directly indicates the specific comparison being checked when the instruction is executed. Understanding this functioning is crucial for effectively working with conditional branches in computer architecture.

8. Which instruction set architecture is characterized by operations only between registers?

- A. Accumulator architecture
- B. Load-Store architecture**
- C. Complex Instruction architecture
- D. High-level programming architecture

The instruction set architecture characterized by operations only between registers is the Load-Store architecture. In this type of architecture, computations are performed exclusively on values within registers. Data must first be loaded from memory into registers before any operations can be executed, and once operations are completed, results must be stored back to memory. This design leads to a simplified instruction set, where instructions focus on moving data to and from memory, while performing arithmetic or logical operations solely on the contents of registers. The emphasis on register-to-register operations enhances performance and reduces memory bandwidth usage because it minimizes the number of direct interactions with slower memory systems. In contrast, an Accumulator architecture allows operations to involve an accumulator register, which can perform operations with memory directly. Complex Instruction architecture includes many types of operations and does not restrict operations solely to registers, allowing for more varied instruction types. High-level programming architecture is not a recognized category of instruction set architecture, focusing instead on abstract concepts rather than the specifics of register operations.

9. What aspect of SIMD is critical for its functionality?

- A. Independent data streams must be operated on
- B. Identical operations must be applied to each data stream**
- C. It requires multiple instruction types
- D. It utilizes single-thread processing

The critical aspect of SIMD (Single Instruction, Multiple Data) is that it applies identical operations to multiple data streams simultaneously. This is fundamental to its design, as SIMD architecture is built to execute the same instruction across a set of data elements effectively and efficiently. By allowing multiple data points to be processed at the same time, SIMD can significantly enhance performance for operations that involve large datasets, such as in graphics processing and scientific computations. The strength of SIMD lies in its ability to leverage the parallelism inherent in data processing tasks, meaning that a single instruction being executed repeatedly over a collection of data promotes speed and efficiency. This parallel processing capability makes it well-suited for applications where operations can be standardized across multiple items, contributing to more optimal use of resources and reduced execution time.

10. Which LEV8 field represents the first register source operand?

- A. rm
- B. shamt
- C. rn**
- D. rd

In the context of LEV8 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.

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