Design Verification

Ryan Hou

2025-02-18

Table of contents

Preface

Notes on Design Verification. The notes will mainly cover verification concepts, Verilog and SystemVerilog, and UVM. Pre-req: digital logic design fundamentals (see my Digital Logic Design notebook). Common interviews can be found in the Questions chapters.

Resources

Some relevant resources:

- ChipVerify
- Verification Guide
- Verification Academy
- HDLBits
- BSG SystemVerilog Coding Standards

UVM Specific Resources: - ClueLogic

Textbooks:

- Book 1

For playing around with simulations:

- EDA Playground

List of interview questions: - Verification Guide - ASIC Verification Interview Questions - Verification Guide - SOC Verification Interview Questions - Verification Guide - UVM Interview Questions - Verification Guide - SystemVerilog Interview Questions - ChipVerify - Verilog Interview Set 1 - More can be found on their website - ChipVerify - SystemVerilog Interview Set 1 - More can be found on their website - ChipVerify - UVM Interview Set 1 - More can be found on their website - NAND LAND - FPGA Interview Questions

1 Introduction

1.1 Perspective

Note 1: Definition - Some definition

Term is defined as blah blah blah...

This note does ...

1.2 High Level Ideas

2 Summary

In summary...

3 Questions - Fundamentals of Verification

Key concepts:

- How does verification fit in the product devlopment stack/timeline?
- What teams do verification engineers work with and in what way?
- Come up with a verification plan given a module and their specs/info
- Writing constraints

3.1 Verification Concepts

Q: What is the way to start verifying a design after finishing designing?

Q: What are the steps to verify a design?

Q: What conversations would you have while working with the designer?

Q: After functional simulation and no more bugs, will there be more bugs that will be caught in the emulation / gate level simulation stages?

Q: How would you verify an asynchronous design?

Q: CDC Question: What is the way to get a data input from an asynchronous signal?

Q: Imagine a case where your coverage events are not correct. Top bins are not being hit. Why might this be the case? Brainstorm possible reasons why top bins are being hit?

A: Possible reasons include driver issue, constraint being wrong

Q: Who do DV teams interact with?

Q: What are differences between SoC level verification vs IP/Component level verification?

Q: Design sign-off: when and how?

Q: There are always logical bugs found post-silicon. What to do about those?

Q: Coverage - FSM, toggle, branch, code, functional

3.2 Test Plans

Q: Given a DUT block diagram and timing diagram, try to come up with a verification plan to verify it

Q: Come up with a verification plan for priority arbiter with ack. What test cases would you run? What is your test plan? How would you structure your testbench?

Q: Given a round robin arbiter, where each input to the round robin arbiter is a FIFO, how would you verify this module?

Q: Verification plan - table of contents

3.3 Constraints

Q: You have 4 animal types: Dog, Cat, Sheep, Horse. Each dog can eat exactly 4 apples, each cat can eat exactly 3 apples, each sheep can eat exactly 1 apple, and each horse can eat exactly 10 apples. There are a total of 1024 apples. Write a constraint such that all animal types in total will eat more than 16 apples.

4 Questions - Logic Design & Basics of Digital Systems

Key concepts:

- Basics of logic design
- Basics of digital systems (bits/bytes, hex/bin/dec, throughput/latency, metrics, conversions, etc)
- FIFOs (synchronous and asynchronous)
- Arbiters
- Static timing analysis
- FSMs
- FPGA related roles will ask FPGA-specific optimizations/concepts as well

4.1 Digital System Basics

Q: Calculation of gigabits and speed

Q: How do you tell if your system is little endian or big endian?

Q: Constraint question: You have a total memory size of 4096 bytes. Each page can be 32, 64, 128 bytes. Each page must be aligned. Write a constraint / function that generates a random range that satisfy this constraint

4.2 FIFOs

Q: Asynchronous FIFO qustion (with dual port RAM): gray code? FIFO depth is 10, read/write pointers is 4 bits wide. What should the initial value of the read/write pointers be?

Q: How would you implement this: An asynchronous FIFO, with 8bit read port and 16 bit write port, and the pointers are encoded in gray code

4.3 Static Timing Analysis

Q: Static timing analysis - given a logic circuit and some delay values, what will the setup and hold time be? How about the maximum clock frequency (given some constraints, circuit, etc)?

Q: (not really STA but timing): How do you delay a signal by N cycles?

4.4 Sequential Logic Design

Entries inserted into a buffer. You have to choose the oldest request among those that are ready. Optimize for power

Q: Regarray vs Flops. Indirect branches

4.5 Logic Design - General

Q: Design a circuit to divide the clock frequency by 4 (f/4) - in sychronous way and asynchronous way

Q: Design a circuit to divide the clock frequency by 3 - 50 % duty cycle

Q: How do you build a larger priority encoder from smaller priority encoders (16:4 from 4:1)?

Q: How do you design an edge detector? How about an edge detector that also detects glitched edges? (normal edge detector circuit does not detect if the transition is from x->x'->x)

Q: How are latches different from flip-flops?

Q: What are the different types of latches? What about flip-flops?

4.6 FPGAs

Q: How do FPGAs work? How are they programmable?

Q: How to fix setup and hold time in an FPGA?

Q: What are some data structures used in an FPGA?

Q: How is coding Verilog on FPGA different from ASIC?

5 Questions - Computer Architecture

Key concepts:

- In-order 5 stage pipeline design
- Out-of-order CPU design
- Memory hierarchy - caches
- Memory coherence (MSI, MESI, MOESI, etc)
- Memory consistency ()

5.1 ISA

Q: ISA has a set of compressed instructions. Should find the instruction using LSB. Each inst can take one or two lines based on the type. The section selected can start from an inst or an half inst of an uncompressed.

5.2 In-Order CPU

Q: Explain how a 5-stage pipeline works

Q: Why don't we make in-order pipelines deep? Why not make pipelines of anything in general deep (general tradeoffs)?

Q: What are the hazards in an in-order pipeline? How are they detected? What causes them? How does moving towards OoO help with each?

5.3 OoO CPU Design

Q: LSQ. How does it work

Q: (in your project), how did you decoder work?

Q: (in your project), how did you try to optimize your design?

Q: (in your project), what are the sizes of your reservation station, reorder buffer, store queue, other buffers, etc? How did you find that it is the best size?

Q: (in your project) Explain the register path of your project, from the free list to back

Q: What are the differences/tradeoffs between a distributed reservation station and a centralized reservation state?

Q: Explain Tomasulo

Q: What are the different types of hazards in an OoO design? How are they detected and dealt with?

5.4 Memory Architecture, Caches

Q: How would you determine if the requested data is available in the cache?

Q: What LRU policy do you use? (in project setting or for specific scenario)

Q: Assume a scenario where a cache line gets evicted and immediately the CPU Wants the same cache line. What should you do?

Q: What type of cache did you use? (for project setting or specific scenario)

Q: NUMA vs UMA. Differences and tradeoffs?

Q: Virtual memory, and TLBs. How do they work, what are they for?

Q: Aliasing of memory addresses in a cache. How to solve aliasing and conflict misses?

Q: What are the different types of cache misses?

Q: What are the different types of cache writeback policies and eviction policies?

Q: You have a 16-way cache. How would you speed it up?

Q: In cache design, explain the differences/tradeoffs between DM vs FA vs SA. What situations would you use them in?

Q: True LRU vs Pseudo LRU

Q: True LRU: How many bits to implement?

Q: Given an age vector for a LRU cache and a valid/invalid vector, how would you choose which way to evict?

Q: Prefetcher and I-cache logics

Q: (for your project) implementation details on your Load Store Queue

Q: Come up with cache access patterns in which FA would perform better than DM. In general: come up with a cache access pattern that performs better for each of DM, FA, SA

Q: Write back vs Write through, MRU vs LRU. (and what did you do in your project and how you went about choosing this?)

Q: Non-blocking D-cache and how you implement it?

Q: Virtual memory, VIPT. Explain

5.5 Branch Prediction

Q: Perceptron, what is it? Equation? How do you implement it in hardware?

5.6 Memory Coherency

Q: What is memory coherence and why is it needed?

Q: MSI protocol. What is it, how does it work?

Q: MSI, MESI, MOESI: why was each extra state added?

Q: Draw the state diagram for MESI

5.7 Memory Consistency

Q: Consistency basics, what is memory consistency and why is it needed?

Q: Given a sequence of instructions on 2 cores, which values are possible/not possible - how would you check if a barrier is working?

6 Questions - Coding (Non-HDL)

Common “software” coding questions, concepts, and ideas asked in a design verification interview:

- Bit manipulation
- String parsing/manipulation
- Basic recursive algorithms and their iterative counterparts
- Object-oriented design (polymorphism, inheretence, encapsulation, abstraction)
- Various algorithmic concepts (DFS, BFS, TSP)
- Parallelism - multi-threading (fork, join), writing parallelised algorithms
- Basic data structures and their implementation
- Basic array leetcode problems
- Mostly in C++ or Python (or another choice of scripting language)

6.1 Bit Manipulation

Q: Bit manipulation leetcode questions

6.2 Strings

Q: String matching regex

Q: Parse a string in Python: Given a .txt file, parse the file and print the frequency of each word, printed by ordering of most frequent first.

6.3 Data Structures

Q: Code up a class for a stack, including functions for push and pop

Q: Write C++ code to move value to the front of an array

Q: Write C++ code to find the second minimum of an array

Q: Two ints are represented as a linked list. Create a new linked list that is the sum of these two integers

Q: How do you build a queue from a stack? How do you build a stack from a queue?

6.4 Recursion/Iteration

Q: Write a function to calculate the factorial of a given number. Recursively and iteratively

Q: Write a function to output the fibonacci sequence recursively. How about iteratively? What are the differences (especially in terms of complexity)?

Q: Deleting objects at the end after execution (freeing up dynamic memory): use linked list, vector, or a counter

6.5 Object-Oriented Design

Q: What is polymorphism? What is inheritance?

- LinkedIn Post - Polymorphism in DV Interview

6.6 Others

Q: Write the pseudocode for how you would go about solving the traveling salesman problem

Q: Write code for a card game in C++: Black jack - shuffle, deal, and check the sum is equal to a value. If greater then lose. If rounds are complete, then highest wins

Q: Write code to perform matrix multiplication in C++

Q: Write C++ code to implement LRU. What is the time complexity?

Q: Loop interchange - to improve cache hit rate

Q: Compilers - register renaming

Q: How would you code Fibonacci in assembly?

6.7 Scripting

Q: In Python, parse a file line-by-line and finding the minimum time for a particular bucket id given this file structure: TASK_ID | STATUS | COMMAND_LINE | BUCKET_ID | TIME (HH:MM:SS)

7 Questions - Verilog

Key concepts

- Blocking vs Non-blocking
- How will given code synthesize
- Coding up FIFOs, arbiters, FSMs

Q: Given some Verilog code (blocking vs non-blocking assignment), what will these synthesize to?

Q: Code fizz buzz in Verilog

Q: How does polymorphism work in SystemVerilog?

Q: How do virtual interface work in SystemVerilog?

Q: What do clocking blocks do in SystemVerilog? What are they?

Q: Differences between Verilog and SystemVerilog?

Q: Code a sequence (pattern) detector state machine. How would you verify this state machine?

Q: SV data structures: queues, maps, associative array, dynamic array

Q: Write the code for a FIFO in SystemVerilog

Q: SystemVerilog Fork, Join(). Given some example code with these, what would happen to a task (e.g. a given task could take forever)? What could you do to fix it (e.g. by writing a task to print ERROR if a task doesn't return by TIMEOUT seconds)?

Q: Traffic light signaling problem in Verilog (state machine). Use counter and output color depending on range using assign statements

Q: Given an FSM state diagram, code it in SystemVerilog

Q: Logic vs wire vs bit in Verilog/SystemVerilog

Q: Write the RTL to generate a signal that is clock for 3 cycles then 0 for 3 cycles

Q: Given 3 enums, constrain 32 bits to be within these 3 enums. Then constrain their distribution

Q: SystemVerilog Assertions. Write assertions for Req and Ack with the given spec:

- Ack should assert within 5-10 cycles after Req asserts
- Ack should deassert within 5-10 cycles after Req desserts
- No Ack should assert before Req
- No Ack should deassert before Req deasserts

Q: CAM & TCAM in SystemVerilog. Modeling with 2D/3D array vs associative array (key as data, addr as value), and pros and cons of these approaches

Q: Write a module to convert binary to gray code (how about vice versa)?

7.1 Online References

- Blind Post - Design Verification Interview

8 Questions - UVM

Q: What UVM phase is different from the others, and why?

Q: UVM Polymorphism question, child handle

References