CHETAN SOMANA

7093879657 chetan.somana@gmail.com Hyderabad

Seasoned ASIC Verification Engineer with 2+ years of experience. I thrive in fast-paced environments where work culture and creativity are paramount. Aware of RTL to GDS II flow. Excellent coding / debugging skills in SV, UVM using the tool interactive debugger. Good knowledge in digital electronics, data structures, scripting.

WORK EXPERIENCE (2 Years)

Cyient LTD

APR 2022 – Present

ASIC Design Verification Engineer

AUTOMOTIVE PMIC (NXP Semiconductors CWF)

Worked with NXP from verifying the design starting with scratch to the metal layer tape out.

> Gained experience in verifying the automotive device, functional safety rated at ASIL D.

> Hands on experience of 80% of pmic block like bucks, LDO, I/O's, Fault Monitoring, Voltage Monitoring, System monitoring, Fault detection, Always on supply, Thermal management.

VPLANS

> Proactively engaged with the product definer, architects, designers and verification leads to have clear verification scope understanding.

> Made vplan with summarized diagrams, tables, waveforms to save time during the review.

> Initiated internal discussion with the team to gain different perspectives of other verification engineers.

Test development

> Developed the mindset of reusing the existing code than to reinvent wheel for better future reusability and to save time.

> Developed reusable sequences.

> Followed Constraint random verification.

> Developed writing assertions which mimics the design.

Functional Coverage & Code coverage

> Independently worked on complete functional coverage closure.

> Improved the code coverage of specific blocks analysing block, expression, toggle, FSM areas.

Tools used for verification: Cadence based Simvision, vManager, Virtuoso

Project maintenance tools: Bitbucket, Jira, Doors, Confluence, pipa, Git

Maintenance work explored:

> Creating dashboard/ queries in Jira to get holistic view of the tickets flagged during verification.

> Creating Documentation in confluence to explaining the flow of creating coverage for the register model.

> Maintaining the main branch of GIT version control while team lead is away. Able to deliver successful release tag to design verification team and as well as AMS team.

System Bus Communication

Responsible for enhancing monitor, scoreboard codes by decreasing the numbers of transactions used to verify and associative array rather than queues. Developed assertions, byte strobe testcase. Debugged the clock driving generation logic.
Tools used: Questa Sim

AXI4 LITE

Created the necessary environment with appropriate configuration, interface, and packages. Responsible for writing logic for on the fly reset. Developed driver which supports in order transaction flow. Learned how to run regression with vmanager. Developed my own script which compiles and elaborates all my testcases once and simulates each testcase with random seed.
Tools used: Cadence Xcelium, Simvision, IMC, Vmanager

QSPI

Responsible for writing the driver logics in STR mode which supports single, dual, and quad transfer along with DTR mode which supports single, dual, and quad transfer. Written monster random test case which randomizes all the features included. Developed my own script which renames folder name with a new name. Able to change the contents of the file with the new name.
Tools used: Cadence Incisive, Simvision, IMC

I2C

Responsible for driving the data through driver and collecting the data through Monitor for in house developed I2C model. Written test cases to generate 2^7 addresses and 100KHz, 1MHz frequencies.
Tools used: Cadence Incisive, Simvision, IMC

Risetime Semiconductors

Aug 2021 – MAR 2022

Design Verification Engineer

16*16 router using UVM

Checking the same functionality mentioned below with UVM. Learned using factory, we can create UVM components like driver, monitor, agent, environment. Checked if the components are registered by printing the hierarchy. Passed the values from different scope through UVM_CONFIG. Made the connection to the DUT using virtual interface.
Tools used: VCS, Verdi

16*16 router using SV

Built a testbench for 16*16 router which sends a packet from input port to output port. All the information about the port address and data is encapsulated in a packet class and is randomized. Driver is used to drive the packet object to DUT and a monitor is used to sample and is given to scoreboard to verify. Accomplished 100% functional coverage. Learned SV constructs real time use case.
Tools used: VCS, Verdi

EDUCATION

Bachelor of Engineering / CGPA 7.29
Methodist College of Engineering and Technology

I have invested 4 years of my engineering in coding and electronics. I was fascinated by the LEDs which glow on billboards, so, I made 4*4*4 LED cube. LEDs are controlled by Arduino uno. By this I can dump program which turn the cube into different patterns. I am also an intermediate programmer in Java, Python. Using OpenCV library, I have built a motion tracker which replicates hand movement on paint.

12 High School / 94%
Mac Arthur High School

During my High School, I made new friends from distinct cultures as USA is highly diverse. This helped me to have more interactions with people having different perspectives.

10 SSC / CGPA 9.2
Bhashyam High School

I have developed various interests during my school days. Geography, Finance, Biology, Math are few of them. These interests helped me to be a participant in every conversation people have.

SKILLS

System Verilog
TCL interactive debugger
Assertions
UVM
Assertions
Micro architecture design
Leadership
Problem-solving
Rational Thinker
Coding in c & python
Linux
ASIL D Functional Safety

Online Certified courses

>**Cadence RTL to GDS II Flow v5.0: (completed)**

During the course I've learned the other end spectrum of the verification. We started with a simple 8 bit sequential counter. Done RTL simulation, coverage analysis, logic synthesis, floor planning, power planning, placement, cts, routing, sta, gls, gdsII.
Got familiarized with Genus, Lec, Innovus, tempus tools.

>**Cadence Jasper formal-fundamentals-v2212 (ongoing)**

During the course I've developed mindset to avoid assertion liveness. Instead use auxiliary code and split into multiple properties

---

Books / Texts

>*Computer Architecture: A Quantitative Approach* Fifth Edition by John L. Hennessy (Author), David A. Patterson (Author)
>SystemVerilog Assertions and Functional Coverage by Ashok B. Mehta