

가온칩스 이력서

가온칩스를 지원한 이유와 입사 후 회사에서 이루고 싶은 꿈을 서술하시기 바랍니다.

AI의 폭발적인 성장을 체감하며 딥러닝 모델을 학습했고, AI 알고리즘을 서비스화하는 Python 개발자로 커리어를 시작했습니다. 현업에서 더 좋은 성능을 내기 위해 알고리즘을 개선하고 데이터를 튜닝하는 노력을 거듭했습니다. 하지만 소프트웨어 레벨의 최적화만으로는 성능 향상에 명확한 한계가 있음을 깨달았고 좋은 칩을 구현해내는 일이야말로 AI 시대의 핵심 역량임을 깨달았습니다.

가온칩스는 특정 분야에 국한되지 않고 AI·자율주행·통신용 SoC 설계를 하며 급격히 성장하고 있습니다. 이는 엔지니어로서 '성장과 경험'을 동시에 이룰 수 있는 최고의 환경이라 확신합니다. 저는 이러한 산업 변화의 중심에서 SoC 설계의 통합적 사고를 가진 엔지니어로 성장하고 싶어 지원했습니다.

가온 칩스에 입사를 하게 된다면 5개월간 OJT 와 학습 관련된 복지 제도를 적극 활용하고 싶습니다. 실제 현업 환경에서 다양한 IP와 SoC 아키텍처를 다루며 검증 및 설계 역량을 성장시킬 수 있는 기회이기 때문입니다.

단기적으로는 ASIC 설계 전 과정을 완전히 이해하는 설계 엔지니어가 되고, 장기적으로는 신규 IP 개발과 저전력 SoC 아키텍처 최적화를 주도하는 리더로 성장하고자 합니다.

끊임없이 배우고, 오래 함께 성장하는 사람을 찾는 가온칩스의 인재상처럼, 저는 꾸준한 학습과 실무 몰입을 통해 회사의 핵심 설계 인력으로 자리 잡고 싶습니다.

지원한 직무와 관련하여 본인이 가진 역량에 대하여 작성해 주시기 바랍니다.

저는 SoC 설계 전 과정에 필요한 논리적 설계력과 툴 운용 능력, 시스템 통합 역량을 함께 보유하고 있습니다. SSDEC 반도체 설계 전문가 과정을 통해 설계-검증-합성-타이밍 분석 까지의 전체 Flow를 실무 수준으로 경험했습니다.

첫째, Front-End에서 Back-End까지의 공정 이해입니다. 12주간 SystemVerilog 기반 RTL 설계, Synopsys VCS/Verdi를 활용한 Simulation 및 Debugging, Design Compiler를 이용한 논리합성, PrimeTime 기반 STA(정적 타이밍 분석)까지 실제 ASIC 설계 플로우를 툴 중심으로 수행했습니다. 특히 Timing Closure 과정에서 Path Delay 분석

을 통해 회로 병목을 직접 개선하며, 단순한 도구 활용을 넘어 '설계-검증-합성' 간 연계 구조'를 체득했습니다.

둘째, IP 설계 및 시스템 통합 능력입니다. 현재 진행 중인 'RISC-V 기반 Smart Door Lock SoC' 프로젝트에서는 AXI Bus Interconnect 구조를 직접 설계하고, GPIO, I2C, SPI IP를 RTL로 구현하여 시스템에 통합했습니다. 각 IP를 AXI4-Lite 및 AXI4 Bus에 연동하며 Address Mapping, Handshake Protocol, Timing Violation 등을 점검했습니다. 이 경험을 통해 복잡한 SoC 구조에서도 논리적 데이터 흐름과 모듈 간 연결 관계를 설계 수준에서 조정할 수 있게 되었습니다.

셋째, 협업 및 코드 품질 관리 역량입니다. 개발자로 근무할 당시, Git을 통해 Branch를 관리하며 다양한 병합 충돌(Merge Conflict)을 해결한 경험이 있어, 팀 기반 설계 환경에서도 구조적 협업을 수행할 수 있습니다.

이처럼 이론적 이해를 기반으로 실무 Tool과 System Integration 경험을 갖춘 점은, 가온 칩스의 고성능 SoC 설계 프로젝트에서도 빠르게 적용하고 기여할 수 있는 강점이 될 것입니다.

지원한 직무를 수행하기 위한 노력 및 직무 경험에 대해 구체적으로 작성해주시기 바랍니다.

비전공자라는 한계를 극복하기 위해, 저는 "모르는 것을 끝까지 이해하고 구조화하는 학습 방식"으로 성장해왔습니다.

학부 시절 C프로그래밍, 알고리즘, 컴퓨터비전 과목을 통해 로직 중심 사고력과 메모리 제어 개념을 쌓았고, 이를 바탕으로 SSDEC 과정에서 RISC-V와 주변장치 간 인터페이스 시뮬레이션을 수행했습니다.

특히 GPIO와 I2C 통신을 구현하며 비정상 입력 시나리오를 직접 만들어 버스 충돌·타이밍 오류를 검증했습니다. 단순히 동작 확인에 그치지 않고, 오류의 원인을 논리적 레벨에서 추적해 수정하는 과정에서 Verification의 본질을 체득했습니다.

이후 FPGA 실습에서는 Vivado 환경에서 각 모듈의 Timing Report를 분석하고, 병목 구간을 최적화하며 "동작하는 설계에서 효율적인 구조로 개선"하는 경험을 했습니다. 이를 통해 단순 구현이 아닌 설계의 '품질'을 고민하는 태도를 익혔습니다.

또한 협업 프로젝트 중에는 갈등을 체계적으로 해결하기 위해 서면 피드백 방식과 개선 중심 커뮤니케이션 규칙을 직접 도입했습니다. 이는 설계자에게 필수적인 논리적 사고와 원활한 협업 능력을 동시에 키워준 경험이었습니다.

가온칩스 입사 후에는 OJT 과정을 통해 FPGA 수준에서 배운 구조적 사고를 ASIC 단계로 확장하고, 전력·성능·면적의 균형을 고려한 SoC 설계자로 성장하겠습니다.

본인이 생각하고 있는 자신의 성격의 장단점에 대하여 기술하여 주시기 바랍니다.

저는 논리적인 소통을 통해 문제를 해결하는 성격을 가지고 있습니다. 갈등이 발생했을 때 감정적으로 반응하기보다는 원인을 구조적으로 분석하고 해결 프로세스를 설계하려는 습관이 있습니다.

졸업 프로젝트 당시 아이디어 회의 중 팀원 간 갈등이 심화된 적이 있었습니다. 저는 회의 구조를 개선하기 위해 Google Docs 기반 서면 피드백 시스템을 제안하고, 롤링페이퍼 방식을 도입해 서로 칭찬 또는 비판을 하되, 비판 시에는 반드시 개선점을 작성하도록 했습니다. 또한 최종 아이디어는 Google Form을 활용한 다수결 투표로 결정했습니다. 그 결과 감정이 완화되고 회의 효율성이 크게 향상되었습니다.

이 경험은 단순한 협업을 넘어 “소통 구조를 설계할 줄 아는 사람”으로 성장하는 계기가 되었습니다.

반면 제 단점은 완벽하게 준비해야만 안심하는 신중함입니다. 하지만 이 점을 개선하기 위해 실무 프로젝트에서는 일정 내 빠른 실행을 목표로 60% 완성 → 피드백 반영 → 개선 반복의 방식으로 균형을 잡고 있습니다.

가온칩스의 인재상 중 하나가 ‘소통이 잘 되는 사람’인 만큼, 저는 이러한 경험과 태도를 바탕으로 OJT 및 프로젝트 수행 과정에서 유연하고 지속적으로 성장하는 엔지니어로 발전하겠습니다.

활동 내역

반도체 기초 이론

- 반도체 Device Physics: 반도체 물리 기초 및 소자 동작 원리

- 디지털 설계 원론: 디지털 회로 설계 기초 이론 및 설계 방법론

RTL 설계 및 검증

- SystemVerilog 설계
 - Design: RTL 설계 문법 및 구조
 - Verification: 검증 방법론 및 테스트벤치 작성
 - Assertion: SVA(SystemVerilog Assertions)를 활용한 속성 검증
- ARM-based Modern SoC Architecture: ARM 아키텍처 기반 SoC 구조 이해
- Coverage-driven Verification: 커버리지 기반 검증 방법론 및 목표 커버리지 달성을 위한 도구 활용

검증 도구 활용

- Synopsys VCS: RTL 시뮬레이션 및 검증
- Synopsys Verdi: 디버깅 및 파형 분석 도구 활용

저전력 설계

- Low Power Design Methodology
 - 클럭 게이팅, 전원 게이팅 등 저전력 설계 기법
 - 전력 분석 및 최적화 방법론

논리 합성 (Logic Synthesis)

- Synopsys Design Compiler (DC)
 - 논리 합성 수행
 - 제약 조건(Constraints) 작성
 - 리소스 및 타이밍 최적화
 - 실무 환경에서의 합성 플로우 구축

타이밍 분석 (Static Timing Analysis)

- Synopsys PrimeTime (STA Sign-off)
 - 정적 타이밍 분석 수행
 - Setup/Hold Time 검증
 - 타이밍 클로저 달성
 - Sign-off 기준 준수 방법

테스트 설계

- DFT Technology
 - 스캔 체인(Scan Chain) 설계
 - BIST(Built-In Self-Test) 설계
 - Power-aware DFT 방법론
 - 테스트 패턴 생성 및 검증