Shew Juan Kok

Senior Design & AI Engineer • ASIC + AI Systems

Reading, Berkshire, UK • +44 7387 729605 • kokshewjuan_job@outlook.com
linkedin.com/in/shewjuankok • juansync7.github.io

PROFESSIONAL SUMMARY

Senior Design & AI Engineer pairing deep ASIC/front-end expertise (SystemVerilog, SoC architecture, DFT, functional verification) with hands-on AI platform engineering. Currently architecting **AI Harness**, an internal AI knowledge and agentic platform built as a reusable base layer of APIs, data services, and orchestrated workflows. Built the business case and security review that brought Claude into production across the company — translating between engineering, IT, and leadership to take an AI tool from evaluation to adoption. Strong track record building production LLM systems that are observable, governed, and safe by construction.

WORK EXPERIENCE

Senior Design & AI Engineer, Aion Silicon — London, UK **Feb 2026 – Present**

- Architecting **AI Harness**, an internal AI platform serving as a reusable base layer (APIs, data services, durable workflows) on which new agentic capabilities can be stood up quickly and safely. Stack: LangGraph (agent orchestration), Temporal (durable workflows), LiteLLM (model routing), Langfuse (observability), polyglot persistence across PostgreSQL, Weaviate, and MinIO; guardrails via NeMo Guardrails and PII handling via Presidio.
- Designed the platform's multi-modal retrieval architecture: vector search (Weaviate), graph traversal over a Neo4j knowledge graph built from parsed SystemVerilog ASTs, hierarchical tree retrieval, and live search tools, routed by query intent rather than tuning a single retriever for every case.
- Established the evaluation and instrumentation discipline that keeps the platform measurable: retrieval and generation metrics (Hit Rate, Recall@k, MRR, NDCG@k, Faithfulness, Answer Relevance) feeding Langfuse-driven observability, turning model, chunking, and reranker choices into decisions rather than guesswork.
- Developing a detection-and-surfacing pipeline for chip-integration project monitoring: ambient agents flag schedule slippage, cross-functional file-mismatch errors, and workflow-correctness issues against accumulated past-run knowledge, raising signals to engineers early with audit trails so cross-team workflows stay aligned and compute resources are used efficiently.

Senior Design Engineer, Aion Silicon — London, UK **Aug 2025 – Feb 2026**

- Made the technical and business case for adopting **Claude** as a company-wide AI assistant alongside Microsoft Copilot: authored the IT security assessment, navigated engineering, IT/security, and leadership stakeholders, and secured approval — taking the tool from evaluation through to rollout.
- Re-engineered a RISC-V vector core to interface with an external VPU accelerator, analysing dispatch hazards, retirement-order constraints, and ISA-compliance edge cases, then implementing custom logic for instruction dispatch, hazard detection, and synchronised retirement to preserve in-order architectural state under decoupled execution.
- Built Python, TCL, and Bash automation across synthesis, lint, and DFT iteration loops to compress design-cycle turnaround on complex SoC projects.

ASIC Design Engineer, Aion Silicon — Reading, UK **Sep 2023 – Aug 2025**

- Led front-end design from architectural specification through RTL implementation across multiple IP and SoC projects in SystemVerilog, applying parameterisable design patterns and pipelined architectures, and verifying with constrained-random methodologies and SVA assertions to meet rigorous specification requirements.
- Identified inconsistent RTL coding practices across design teams as a source of integration friction; established a company-wide linting methodology using Synopsys VC Spyglass and Questa Static Formal, codified into automated checks adopted by all teams.
- Owned DFT implementation for a large-scale, ASIL-B compliant SoC, debugging ATPG coverage holes with Synopsys TestMax to push test coverage above 90% and meet automotive safety requirements.
- Performed early-stage design exploration with Synopsys RTL-Architect, running global placement and routing on RTL to surface PPA risk before back-end handoff and shift design-critical decisions left, and architected system-level integration of third-party and in-house IP for advanced SoC implementations.

PROJECTS

AI-Synapse — open-source skill-orchestration framework for AI coding agents github.com/JuanSync7/ai-synapse

- Built a composable framework intended as a company "brain" of reusable AI capabilities: skills, agents, protocols, and tools as auto-discovered artifacts invoked as slash commands across Claude Code, Codex, and Gemini for autonomous documentation, code generation, testing, and multi-agent pipelines.
- Designed a governed artifact lifecycle (brainstorm → create → improve → certify) with two-tier validation: deterministic pre-commit structural checks (taxonomy, registry, schema) plus LLM-based quality gatekeeping at PR time, and implemented the **cortex** CLI dispatcher enforcing metadata consistency across artifact types (Python and Shell, GitHub Actions CI).

code-doc-monitor — code→documentation drift monitor with auditable LLM remediation github.com/JuanSync7/code-doc-monitor

- Built an open-source tool that detects when documentation drifts from the code it describes, then routes each drift to a pluggable LLM backend to fix, invalidate, or escalate — recording the original drift and the proposed fix as a versioned, human-reviewable ReviewRecord (public JSON schema) so remediation stays auditable and a person keeps the review seat. Backends — offline mock, headless Claude Code CLI, Anthropic Messages API, and a LangGraph agent — are selected by config, never a code change.
- Implemented the remediation agent as a deterministic LangGraph state machine (select → compose → invoke → parse, with a bounded re-ask loop) whose prompt is assembled from separate Markdown artifacts loaded only when a node needs them, and whose runtime driver (Claude Code CLI / Anthropic API / local endpoint) is itself a config choice. Offline-by-default test suite, ruff/mypy-clean, ≥90% coverage across 322 tests, with a two-gate CI pipeline that exercises a live Claude backend on a schedule.

KGWeave — knowledge-graph subsystem for code and document retrieval github.com/JuanSync7/KGWeave

- Built a pipeline that safely converts SystemVerilog RTL into a queryable knowledge graph by parsing it into an abstract syntax tree (pyslang) rather than text-matching, capturing module hierarchy, instantiation, and signal connectivity as graph relationships with full provenance.
- Implemented entity extraction, community detection, and query expansion behind a stable, Pydantic-typed client API (admin, ingest, query), with its own eval harness and golden sets for retrieval quality.

EDUCATION

Master of Engineering (MEng), Electrical and Electronics Engineering, First-Class Honours **2023**

University of Southampton (UK, 2021–2023) and University of Southampton Malaysia (2017–2021).

SKILLS

Programming & Scripting: Python (primary), SystemVerilog (primary HDL), Bash, TCL, C, C++, SystemC (TLM modelling, HLS), Markdown.

AI / ML Engineering: Agentic systems and AI dev tooling (multi-agent pipelines, skill/tool orchestration); RAG and retrieval (vector, graph, hybrid) with LangGraph and Temporal; vector and graph stores (Weaviate, Neo4j); model serving (vLLM with PagedAttention, Ollama, LiteLLM); evaluation and observability (Langfuse, OpenTelemetry); guardrails and PII (NeMo Guardrails, Presidio); document processing (Docling).

Hardware Design & Verification: SystemVerilog RTL and constrained-random verification with SVA; pipelined architectures, FSM design, parameterisable IP, ECC/redundancy for safety-critical implementations; in-house and third-party IP integration; RISC-V vector architecture (VXU, VPU integration, dispatch/hazard handling); AMBA AXI/AHB, PCIe, DDR.

EDA Tools: DFT: Synopsys TestMax; Front-end: Synopsys Design Compiler, RTL-Architect, Platform Architect; Static Formal: Synopsys VC Spyglass, Questa Static Formal; Verification: Verdi, VCS, coverage and waveform analysis; Open Source: Verible, Slang, Verilator.

Infrastructure & Tooling: Docker, Podman, Nginx, Temporal, Git, Subversion (SVN); polyglot data stores (PostgreSQL, Weaviate, MinIO, Neo4j).

LANGUAGES

English (Native / Bilingual) • Mandarin Chinese (Native / Bilingual) • Malay (Native / Bilingual)

Additional: Requires UK skilled-worker visa sponsorship. Continuous learner with a focus on production AI systems.