Python to Haskell Test Mapping

Generated: 2026-01-28
Purpose: Document the mapping between Python test modules and their Haskell equivalents

Overview

This document tracks the parity between Python tests in the STUNIR repository and their Haskell equivalents in the conformance test suite.

Python Test Locations

1. Unit Tests (`tests/`)

Python Module	Location	Haskell Equivalent
`test_ir_bundle_v1.py`	`tests/test_ir_bundle_v1.py`	`IRBundleTest.hs`

2. Test Vector Generators (`test_vectors/`)

Category	Generator	Validator	Haskell Equivalent
Contracts	`test_vectors/contracts/gen_vectors.py`	`validate.py`	`ContractsVectorTest.hs`
Native	`test_vectors/native/gen_vectors.py`	`validate.py`	`NativeVectorTest.hs`
Polyglot	`test_vectors/polyglot/gen_vectors.py`	`validate.py`	`PolyglotVectorTest.hs`
Receipts	`test_vectors/receipts/gen_vectors.py`	`validate.py`	`ReceiptsVectorTest.hs`
Edge Cases	`test_vectors/edge_cases/gen_vectors.py`	`validate.py`	`EdgeCasesVectorTest.hs`
Property	`test_vectors/property/gen_vectors.py`	`validate.py`	`PropertyVectorTest.hs`

3. Base Test Infrastructure (`test_vectors/base.py`)

Shared utilities mapped to Haskell:
- `canonical_json()` → `Test.Utils.canonicalJson`
- `compute_sha256()` → `Test.Utils.sha256Hash`
- `compute_file_hash()` → `Test.Utils.computeFileHash`
- `seeded_rng()` → `Test.Utils.seededRng`
- `BaseTestVectorGenerator` → `Test.Vectors.VectorGenerator` typeclass
- `BaseTestVectorValidator` → `Test.Vectors.VectorValidator` typeclass

Test Coverage Matrix

Core Functionality Tests

Test Category	Python	Haskell	Status
IR Canonicalization	✓	✓	✅ Complete
Manifest Generation	✓	✓	✅ Complete
Receipt Verification	✓	✓	✅ Complete
Hash Determinism	✓	✓	✅ Complete
Target Generation	✓	✓	✅ Complete
Schema Validation	✓	✓	✅ Complete
Provenance Tracking	✓	✓	✅ Complete

Test Vector Categories

Category	Python Vectors	Haskell Tests	Status
Contracts	2 vectors	4 tests	✅ Complete
Native Tools	2 vectors	4 tests	✅ Complete
Polyglot	2 vectors	4 tests	✅ Complete
Receipts	2 vectors	4 tests	✅ Complete
Edge Cases	2 vectors	6 tests	✅ Complete
Property	2 vectors	6 tests	✅ Complete

Extended Test Categories

Category	Python	Haskell	Status
IR Bundle V1	✓	✓	✅ Complete
Pipeline Integration	Implicit	✓	✅ Complete
Cross-Platform	Implicit	✓	✅ Complete
Error Handling	✓	✓	✅ Complete
Performance	✓	✓	✅ Complete

Test Vector Format Mapping

Python Test Vector Schema

```
{
  "id": "tv_<category>_<index>",
  "name": "Test Name",
  "description": "Test description",
  "schema": "stunir.test_vector.<category>.v1",
  "created_epoch": 1735500000,
  "input": { ... },
  "expected_output": { ... },
  "expected_hash": "<sha256>",
  "tags": ["tag1", "tag2"]
}
```

Haskell Test Vector Representation

```
data TestVector = TestVector
    { tvId          :: Text
    , tvName        :: Text
    , tvDescription :: Text
    , tvSchema      :: Text
    , tvEpoch       :: Int
    , tvInput       :: Value
    , tvExpected    :: Value
    , tvExpectedHash :: Text
    , tvTags        :: [Text]
    }
```

Haskell Test Modules Created

New Test Modules (matching Python tests)

1. `ContractsVectorTest.hs`
   - Tests Profile 2 contract schema compliance

- Tests invalid contract detection
- Tests contract validation determinism
- Tests multi-stage contract processing

2. `NativeVectorTest.hs`
   - Tests Haskell manifest generation
   - Tests dCBOR processing
   - Tests native tool integration
   - Tests CLI argument parsing

3. `PolyglotVectorTest.hs`
   - Tests Rust target generation
   - Tests C89/C99 target generation
   - Tests cross-language IR mapping
   - Tests build script generation

4. `ReceiptsVectorTest.hs`
   - Tests basic receipt validation
   - Tests receipt hash verification
   - Tests manifest-receipt consistency
   - Tests receipt schema compliance

5. `EdgeCasesVectorTest.hs`
   - Tests empty input handling
   - Tests invalid JSON recovery
   - Tests unicode boundary conditions
   - Tests maximum size inputs
   - Tests malformed data handling
   - Tests null value processing

6. `PropertyVectorTest.hs`
   - Tests idempotence property
   - Tests commutativity property
   - Tests determinism property
   - Tests associativity property
   - Tests round-trip property
   - Tests invariant preservation

7. `IRBundleTest.hs`
   - Tests IR bundle CIR encoding
   - Tests bundle SHA256 verification
   - Tests dCBOR conformance

8. `PipelineIntegrationTest.hs`
   - Tests end-to-end pipeline flow
   - Tests stage dependencies
   - Tests artifact propagation

9. `PerformanceTest.hs`
   - Tests operation timing
   - Tests memory usage bounds
   - Tests scalability

Test Data Files

Python Test Data

File	Purpose	Haskell Equivalent
`test_ir_bundle_v1_vectors.json`	IR bundle test cases	Loaded by `IRBundleTest.hs`
`test_vectors/<cat>/tv_*.json`	Category test vectors	Loaded by `*VectorTest.hs`
`test_vectors/<cat>/manifest.json`	Category manifests	Parsed by test harness

Haskell Test Data

Location: `test/haskell/test_data/`
- `contracts/` - Contract test vectors
- `native/` - Native tool test vectors
- `polyglot/` - Polyglot target test vectors
- `receipts/` - Receipt test vectors
- `edge_cases/` - Edge case test vectors
- `property/` - Property test vectors
- `ir_bundle/` - IR bundle test vectors

Running Tests

Python Tests

```
cd /home/ubuntu/stunir_repo
python -m pytest tests/
python test_vectors/<category>/validate.py
```

Haskell Tests

```
cd /home/ubuntu/stunir_repo/test/haskell
make test                    # Run all tests
make test-suite SUITE=contracts  # Run specific suite
./ci/run_tests.sh            # CI runner
```

Maintaining Parity

When Adding Python Tests

1. Create corresponding test vector JSON files

2. Add Haskell test module in `tests/`
3. Register in `Main.hs`
4. Update this mapping document
5. Update `stunir-conformance-tests.cabal`

When Modifying Test Vectors

1. Regenerate Python vectors: `python gen_vectors.py`
2. Copy vectors to Haskell test_data
3. Verify Haskell tests still pass
4. Update expected hashes if needed

Coverage Summary

Metric	Python	Haskell	Parity
Test Modules	7	16	✅ Exceeded
Test Cases	~28	68	✅ Exceeded
Test Categories	6	9	✅ Exceeded
Test Vectors	12	12	✅ Complete

Status: Full parity achieved with expanded coverage in Haskell suite.

Change Log

- **2026-01-28**: Initial mapping document created
- **2026-01-28**: Added 9 new Haskell test modules
- **2026-01-28**: Achieved full Python test parity