```
0.list  {

        0       difference signal
        1       variable 0
        2       constant 0
        3       store 1
        4       store 2
        5       stored value
        6       signal storage component
        7       iterate 5 of 8 at 9 by 10
        8       <value variable-name>
        9       <value variable-location>
        10      <integer value>
        11      decrement 5 of 8 at 9 by 10
        12      map 5 to 13 xor 14
        13      <single character-string>
        14      <multi character-string>
        15      save 5 as 16 xor 20 with 17 18 xor 19
        16      <in-directory filename>
        17      extension
        18      <extension string variable>
        19      <extension integer variable>
        20      <subdirectory filename> at node layer 21
        21      <positive integer variable>
        22      xor as 23
        23      exclusively or
        24      concatenate 8 to 10 as 16 xor 20 with 17 18 xor 20
        25      (<x>) where x is taken as user-input or 26
        26      system generated input
        27      <directory name>
        28      <subdirectory name>
        29      high 1
        30      high 2
        31      low 1
        32      low 2
        33      medium 1
        34      medium 2
        35      high quantum 1
        36      low quantum 1
        37      medium quantum 1
        38      high quantum 2
        39      low quantum 2
        40      medium quantum 2

}

1.list  {

        0       DEFINE STATUS_FLAGS {
                0       FLAG_ZERO
                1       FLAG_NEGATIVE
                2       FLAG_EQUAL
                3       FLAG_GREATER
```

```
                4       FLAG_LESS
        }

        1       INSTR COMPARE <operand1> <operand2>
                0       PARAM <operand1> : TYPE { 0.5 | 0.8 AT 0.9 | <literal_value> }
                1       PARAM <operand2> : TYPE { 0.5 | 0.8 AT 0.9 | <literal_value> }
                2       ACTION : Calculates difference; SET/CLEAR flags (1.0.0-1.0.4) accordingly.

        2       INSTR BRANCH_IF <condition> <target>
                0       PARAM <condition> : TYPE { FLAG_ZERO | NOT FLAG_ZERO | FLAG_NEGATIVE | NOT FLAG_NEGATIVE | FLAG_EQUAL | NOT FLAG_EQUAL | FLAG_GREATER |
NOT FLAG_GREATER | FLAG_LESS | NOT FLAG_LESS }
                1       PARAM <target> : TYPE <instruction_address> | <label>

        3       INSTR JUMP <target>
                0       PARAM <target> : TYPE <instruction_address> | <label>

        4       INSTR LOAD <address> <variable_name>
                0       PARAM <address> : TYPE { 0.9 | <address_literal> }
                1       PARAM <variable_name> : TYPE { 0.8 }

        5       INSTR STORE_VAL <value> <address>
                0       PARAM <value> : TYPE { 0.5 | <literal_value> }
                1       PARAM <address> : TYPE { 0.9 | <address_literal> }

        6       DEFINE <label> : <instruction_address>
        7       TYPE <instruction_address> : <integer value>
        8       TYPE <address_literal> : <integer value>
        9       TYPE <literal_value> : { 0.10 | 0.13 | 0.14 }
        10      INSTR HALT

        11      ASSERT TURING_COMPLETE {
                0       REQUIRES : { Memory RW, Basic Ops, Conditional Branching, Iteration, Unconditional Branching }

                1       PROVIDED_BY {
                        0       Memory RW : { 0.3, 0.4, 0.5, 0.6, 0.9, 0.15, 1.4, 1.5 }
                        1       Basic Ops : { 0.0, 0.12, 0.22, 0.24, 1.1 }
                        2       Conditional Branching : { 1.0, 1.1, 1.2 }
                        3       Iteration : { 0.7, 0.11 }
                        4       Unconditional Branching : { 1.3 }
                }

                2       STATUS : Achieved by combining 0.list with primitives in 1.list (1.0-1.10).
        }

}

2.list  {

        0       DEFINE REQUIREMENT <req_id> {
                0       DESCRIPTION : <multi character-string>
                1       TYPE : { FUNCTIONAL | NON_FUNCTIONAL | PERFORMANCE | SECURITY | SAFETY }
                2       SOURCE : <multi character-string>
                3       PRIORITY : { 0.29 | 0.30 | 0.33 | 0.34 | 0.31 | 0.32 }
```

```
        4       STATUS : { DEFINED | IMPLEMENTED | VERIFIED | FAILED }
        5       CRITERIA <criteria_id> {
                0       DESCRIPTION : <multi character-string>
                1       METRIC : <multi character-string>
                2       TARGET_VALUE : <literal_value>
        }
}

1       DEFINE USE_CASE <uc_id> {
        0       DESCRIPTION : <multi character-string>
        1       ACTORS : LIST_OF <multi character-string>
        2       PRECONDITIONS : LIST_OF <condition_description>
        3       STEPS {
                0       <step_description>
                }
        4       POSTCONDITIONS : LIST_OF <condition_description>
        5       RELATES_TO_REQ : LIST_OF <req_id>
}

2       DEFINE MODULE <module_id> {
        0       NAME : <multi character-string>
        1       DESCRIPTION : <multi character-string>
        2       IMPLEMENTS_REQ : LIST_OF <req_id>
        3       SUBMODULES : LIST_OF <module_id>
        4       INTERFACES : LIST_OF <interface_id>
        5       SOURCE_REF : <subdirectory filename> AT 0.27
        6       VERSION : <version_string>
}

3       DEFINE INTERFACE <interface_id> FOR MODULE <module_id> {
        0       NAME : <multi character-string>
        1       DIRECTION : { INPUT | OUTPUT | BIDIRECTIONAL | MONITOR }
        2       TYPE : { HARDWARE | SOFTWARE_API | BUS | NETWORK }
        3       PROTOCOL : <protocol_name>
        4       SIGNALS : LIST_OF <signal_id>
}

4       DEFINE SIGNAL <signal_id> {
        0       NAME : <multi character-string>
        1       DATA_TYPE : { 0.10 | 0.13 | 0.14 | BOOLEAN | FIXED_POINT(<bits>,<fractional_bits>) | <custom_struct_id> }
        2       WIDTH_BITS : <integer value>
        3       CLOCK_DOMAIN : <clock_name>
}

5       DEFINE CONNECTION <conn_id> {
        0       SOURCE : <module_id>.<interface_id>.<signal_id>
        1       DESTINATION : <module_id>.<interface_id>.<signal_id>
        2       TYPE : { DIRECT | BUS | BUFFERED | QUEUED }
        3       ASSERT TYPE_MATCH(2.5.0, 2.5.1)
}

6       DEFINE TEST_PLAN <plan_id> {
        0       SCOPE : <multi character-string>
```

```
        1       TARGET_MODULES : LIST_OF <module_id>
        2       TARGET_REQS : LIST_OF <req_id>
        3       STRATEGY : { UNIT | INTEGRATION | SYSTEM | REGRESSION | PERFORMANCE }
        4       TEST_CASES : LIST_OF <test_case_id>
}

7       DEFINE TEST_CASE <test_case_id> {
        0       DESCRIPTION : <multi character-string>
        1       VERIFIES_REQ : LIST_OF <req_id>
        2       EXECUTION_ENV : <environment_description>

        3       SETUP {
                0       <instruction using 0.list, 1.list, 2.list>
        }

        4       STIMULUS {
                0       <instruction using 0.list, 1.list, 2.list>
        }

        5       EXPECTATION <expect_id> {
                0       CHECK_AT : { END_OF_TEST | LABEL <label> | TIME <time_value> }

                1       ASSERT {
                        0       1.1 COMPARE <source1> <source2>
                        1       EXPECT 1.0.2 IS SET
                }
        }

        6       TEARDOWN {
                0       <instruction using 0.list, 1.list, 2.list>
        }

}

8       INSTR RUN_TEST_CASE <test_case_id>
        0       ACTION : Execute 2.7.3 (SETUP)
        1       ACTION : Execute 2.7.4 (STIMULUS)
        2       ACTION : Monitor state/outputs and evaluate 2.7.5 (EXPECTATION)
        3       ACTION : Execute 2.7.6 (TEARDOWN)
        4       OUTPUT : { PASS | FAIL(<expect_id>) | ERROR(<step>, <message>) }
        5       LOG_RESULT TO <file_or_storage>

9       DEFINE BUILD_CONFIG <build_id> {
        0       TARGET_PLATFORM : <platform_description>
        1       TOOLCHAIN : <toolchain_description>
        2       MODULES_INCLUDED : LIST_OF <module_id>
        3       BUILD_OPTIONS : <multi character-string>
        4       OUTPUT_ARTIFACT : <in-directory filename>
}

10      INSTR BUILD <build_id>
        0       ACTION : Simulate compilation/synthesis using config 2.9
```

```
        1       OUTPUT : <build_log> AND 2.9.4 artifact reference

11      DEFINE DEPLOYMENT <deploy_id> {
        0       BUILD_ARTIFACT_REF : 2.9.4
        1       TARGET_ENV : <environment_description>

        2       DEPLOY_SCRIPT {
                0       <instruction using 0.list, 1.list, 2.list>
        }

}

12      INSTR DEPLOY <deploy_id>
        0       ACTION : Execute 2.11.2 script
        1       OUTPUT : { SUCCESS | FAIL(<step>, <message>) }

13      DEFINE ISSUE <issue_id> {
        0       REPORTED_BY : <multi character-string>
        1       DATE_REPORTED : <integer value>
        2       AFFECTS_MODULE : LIST_OF <module_id>
        3       AFFECTS_REQ : LIST_OF <req_id>
        4       DESCRIPTION : <multi character-string>
        5       SEVERITY : { CRITICAL | HIGH | MEDIUM | LOW }
        6       STATUS : { OPEN | IN_PROGRESS | RESOLVED | CLOSED }
}

14      DEFINE PATCH <patch_id> {
                0       RESOLVES_ISSUE : LIST_OF <issue_id>
                1       MODIFIES_MODULE : LIST_OF <module_id>
                2       DESCRIPTION : <multi character-string>
                3       PATCH_DATA_REF : <subdirectory filename> AT 0.27
        }

        15      INSTR SIMULATE MODULE <module_id> WITH STIMULUS <stimulus_sequence>
                0       PARAM <stimulus_sequence> : LIST_OF { WRITE <signal> <value> AT TIME <t> | ... }
                1       OUTPUT : EXECUTION_TRACE | WAVEFORM_DATA(<signal_list>)

        16      INSTR ANALYZE TIMING FOR MODULE <module_id>
                0       OUTPUT : { TIMING_MET | SLACK_REPORT(<path>, <value>) | VIOLATION(<path>, <value>) }

        17      INSTR ANALYZE POWER FOR MODULE <module_id>
                0       OUTPUT : POWER_ESTIMATE(<static_mW>, <dynamic_mW>)

}
```