Fujitsu Compiler Test Suite: New Test Suite for Fortran/C/C++

Takahiro Kawashima (@kawashima-fj)

2025 AsiaLLVM Developers' Meeting

June 2025

Agenda

FUJITSU

- **Fujitsu Compiler Test Suite**: Test suite released by Fujitsu
  - What is it?
  - Characteristics
  - How the community can utilize it
- **Flang CI**: Post-merge CI operated by Fujitsu and Linaro
  - What is it?
  - How the community can utilize it
- **Future plan**

FUJITSU

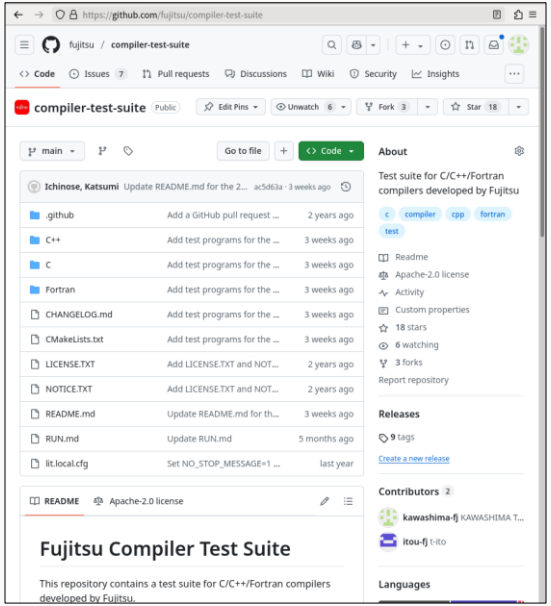
Fujitsu Compiler Test Suite

Test suite released by Fujitsu

What is Fujitsu Compiler Test Suite

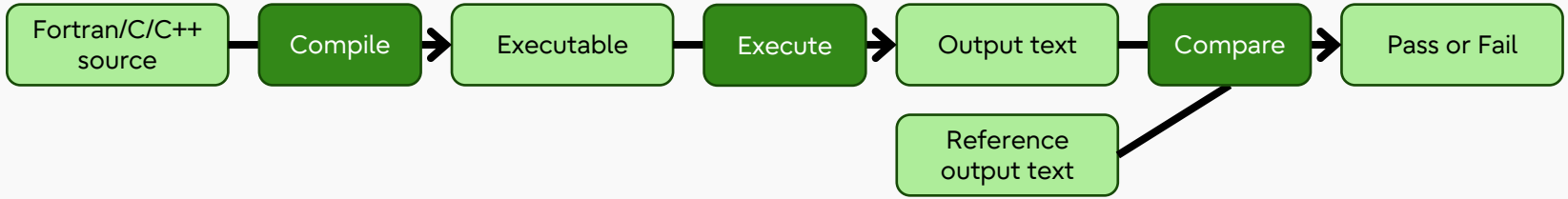
FUJITSU

- Test suite for Fortran/C/C++ compilers
- Released on GitHub at the end of 2023 by Fujitsu
  - https://github.com/fujitsu/compiler-test-suite
- Mainly intended to enhance test of Flang
- Accumulated over many years for testing proprietary Fujitsu compilers
- Use the LLVM test-suite infrastructure (CMake, lit, fpcmp, ...)



Characteristics of the Fujitsu Compiler Test Suite (1/4) FUJITSU

- Whole program test
  - Programs are compiled and executed, and the outputs are verified
    - Like “SingleSource” tests in the LLVM test-suite
  - Compile-only tests also exist



- Single-source tests only (currently)
  - Multiple-source tests will be added in the future
  - Typically, 10–100 lines

Characteristics of the Fujitsu Compiler Test Suite (2/4) FUJITSU

- Tests for Fortran, C, and C++, with OpenMP

Language	Fortran	C	C++
# of tests	~64,000	~26,000	~5,000
% of OpenMP tests	7%	6%	6%

(at version 2025-06)

  - No tests for OpenMP offloading feature
  - No tests for OpenACC, CUDA, or other programming model extensions other than OpenMP
  - (FYI: the LLVM test-suite contains ~8,500 Fortran tests, most of which were imported from the GFortran test suite)
- Many code patterns used in the HPC domain
  - Floating-point arithmetic, loops, math function calls, ...
- Not categorized in directories

Characteristics of the Fujitsu Compiler Test Suite (3/4) FUJITSU

- Various Fortran tests
  - No tests for coarray (yet)
    - Because Flang does not support coarray yet
    - Fujitsu has internal tests, which can be released if the community is interested
  - Almost all features of Fortran 2008 and earlier (except coarray)
    - Only basic tests for features introduced in Fortran 2008; Combination tests are insufficient
  - Some features introduced in Fortran 2018
    - Assumed type, assumed rank, etc.
  - Include extensions that Flang and the Fujitsu Fortran Compiler have in common
    - e.g., specifying type size with “*”, such as “REAL*8”

Fortran Standards	77–95	2003	2008	2018
% of tests	51%	32%	16%	≤ 1%

(at version 2025-06)

Characteristics of the Fujitsu Compiler Test Suite (4/4) FUJITSU

- Check correctness (not performance)
  - Language standard conformance
  - Optimization correctness
  - Code generation correctness
  - Fortran and OpenMP runtime libraries' behavioral correctness
- No dependencies on external libraries (MPI or math libraries)
- Include Linux/AArch64-specific tests
  - Most other tests can be run on machines other than Linux/AArch64
  - No architecture-specific tests other than AArch64
  - Tested only on Linux/AArch64
  - Disabling Linux/AArch64-specific tests on other platforms is not yet implemented

How to Use the Fujitsu Compiler Test Suite

FUJITSU

- Use it as an add-on to the LLVM test-suite
  - If you can run “SingleSource” tests in the LLVM test-suite, it’s easy
  - Details in https://github.com/fujitsu/compiler-test-suite/blob/main/RUN.md

Command example

```
git clone https://github.com/llvm/llvm-test-suite.git
git clone https://github.com/fujitsu/compiler-test-suite.git
    llvm-test-suite/Fujitsu
mkdir llvm-test-suite-build
cd llvm-test-suite-build

cmake -G Ninja
    -D CMAKE_C_COMPILER=<path to llvm build>/bin/clang
    -D CMAKE_Fortran_COMPILER=<path to llvm build>/bin/flang
    -D TEST_SUITE_SUBDIRS=Fujitsu
    -D TEST_SUITE_FORTRAN=ON
    -C ../llvm-test-suite/cmake/caches/O0.cmake
    ../llvm-test-suite
ninja -k 0
lit -o results.json .
```

(**Green/bold** lines are required)

Directory structure

```
llvm-test-suite
 |- SingleSource
 |- MultiSource
 |- Fortran
 |- External
 |- CMakeLists.txt
 |- ...
 |- Fujitsu    ← Added
     |- Fortran
     |- C
     |- C++
     |- CMakeLists.txt
     |- lit.local.cfg
     |- ...
```

Fujitsu Compiler Test Suite for the LLVM Community

FUJITSU

- You can use the Fujitsu Compiler Test Suite as a supplement to the LLVM test-suite, especially for:
  - Flang frontend/runtime
  - OpenMP runtime
  - Loop optimization
- Some (many?) Flang developers already use it

FUJITSU

Flang CI

Post-merge CI operated by Fujitsu and Linaro

What is Flang CI

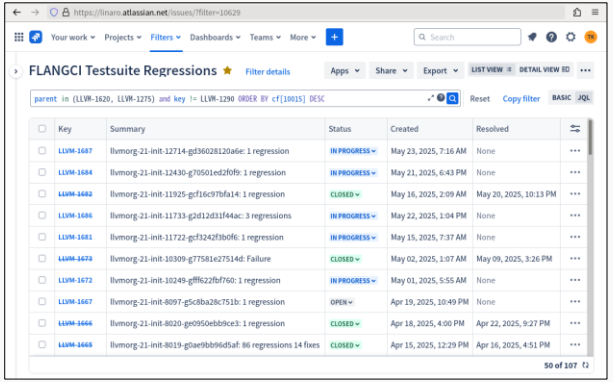
FUJITSU

- Post-merge CI
  - Detect regressions introduced in the LLVM `main` branch
  - Run six times a day
  - Identify the regression cause commit by automatic bisecting search
- Community-driven CI operated by Fujitsu and Linaro
- Use the Fujitsu Compiler Test Suite
  - Despite the name, all Fortran, C, and C++ tests are run
- Run on Ubuntu 24.04 / AArch64 machines (AWS Graviton3)
- Operation started on July 2024
  - Email notification started on August 2024

Flang CI for the LLVM Community

FUJITSU

- Email is sent to the author/committer of the regression cause commit
  - How to reproduce is explained in the email
  - If you cannot reproduce it, please reply to the email
    - You may not be able to reproduce on an architecture different from the CI machine
  - There may be false detections by floating-point arithmetic precision errors
- Unresolved regressions are tracked at our site:
  - https://linaro.atlassian.net/issues/?filter=10629



Result of Flang CI

FUJITSU

- 53 regressions were detected in the past 10 months
  - Include regressions also detected by the LLVM Project's official CI (Buildbot)
- All regressions except a recent one have been fixed
  - Thanks for your contribution!

Failures detected by Flang CI
(as of 2025-06-06)

Cause		Open	Closed	Total
Bug	New regression	1	**52**	**53**
	Latent bug	3	3	6
Not Bug	Revert commit	0	3	3
	Test problem	5	16	21
	Precision error	11	4	15
Other		0	5	5
Uninvestigated		6	0	6
Total		26	83	109

Components where regressions were detected
(as of 2025-06-06)

Component		Number
Flang frontend	Preprocessor	3
	Fortran	7
	OpenMP	10
Flang runtime		3
Clang frontend		5
LLVM middle-end	Optimization	18
LLVM backend	AArch64	4
	Other	1
Other		2
Total		53

Future Plan

Future Plan

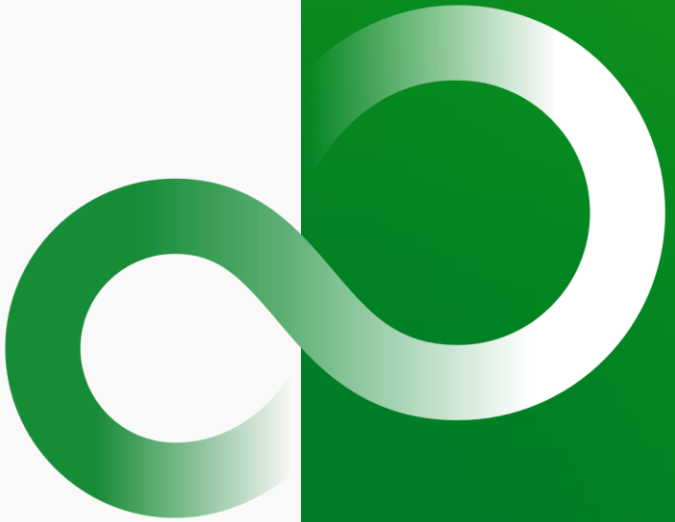
FUJITSU

- Improve the reliability of the test suite and the CI
  - Fix issues in the Fujitsu Compiler Test Suite (non-conforming, implementation-defined, ...)
    - Now working. Gradually resolving.
  - Reduce false detections by precision errors of floating-point arithmetic in Flang CI
    - Now working. Hopefully completed this year (2025).
- Enhance the test suite
  - Increase the number of tests by making Fujitsu's remaining internal tests runnable with test runner of the LLVM test-suite
    - Now working. Gradually increasing. Hopefully completed this year (2025).
- **Promote wider adoption**
  - Integrate the Fujitsu Compiler Test Suite into the LLVM test-suite
    - After improving the test suite reliability. Require consensus with the community.

Acknowledge

- This presentation is based on results obtained from a project, JPNP21029, subsidized by the New Energy and Industrial Technology Development Organization (NEDO).

Thank you



FUJITSU

Backup Slides

Fujitsu Compiler Test Suite: Know-how

FUJITSU

- Use “`-k 0`” flag for Ninja to keep going even if one build fails
  - There are tests that cannot be built by Flang yet
- Don’t forget to set `LD_LIBRARY_PATH` for `libomp.so`
- Check known issues if you have trouble
  - GitHub issues labeled with “known issue”
    - https://github.com/fujitsu/compiler-test-suite/issues?q=is%3Aissue%20state%3Aopen%20label%3A%22known%20issue%22

Fujitsu Compiler Test Suite: Issues

FUJITSU

- Known issues
  - This test suite includes tests that cannot be compiled or executed correctly by any existing version of Flang. (#22)
  - When tests are run in parallel, some test executions may fail randomly. (#2)
  - There may be test programs which have implementation-defined/ unspecified/undefined behaviors. (#3)
  - Compiling with fast-math flag options (`-ffast-math`, `-Ofast`, etc.) may result in verification errors in some or many test programs. (#4)
  - This test suite does not support platforms other than Linux/AArch64 currently. (#5)
- See https://github.com/fujitsu/compiler-test-suite/wiki/KnownIssues for details
- If you find another issue, please let us know in a GitHub issue.

Flang CI: Configurations

FUJITSU

- Four configurations
  1. Optimization with fast-math, AArch64 SVE VLA (`main-aarch64-Ofast-sve_vla`)
  2. Optimization with fast-math, AArch64 SVE VLS, LTO (`main-aarch64-Ofast-sve_vls-lto-lld`)
  3. Optimization without the fast-math flag (`main-aarch64-O3-neoverse_v1-sve_vla-mpipeliner-stack_arrays`)
     - If FAIL with 1 and 2 but PASS with 3: Likely a bug related to fast-math or a false detection due to precision errors
  4. No optimization (`main-aarch64-O0-debug`)
     - If FAIL with 1, 2, and 3 but PASS with 4: Likely a bug in optimizations

```
1: -O3 -ffast-math -march=armv8.4-a+sve -msve-vector-bits=scalable
   -mllvm -scalable-vectorization=preferred
   -mllvm -treat-scalable-fixed-error-as-warning=false -DNDEBUG
2: -O3 -ffast-math -march=armv8.4-a+sve -msve-vector-bits=256
   -mllvm -treat-scalable-fixed-error-as-warning=false -flto -fuse-ld=lld -DNDEBUG
3: -O3 -mcpu=neoverse-v1 -msve-vector-bits=scalable -mllvm -scalable-vectorization=preferred
   -mllvm -treat-scalable-fixed-error-as-warning=false -mllvm -aarch64-enable-pipeliner
   -mllvm -pipeliner-mve-cg -DNDEBUG (-fstack-arrays (Fortran only))
4: -O0 -g
```

Flang CI: How to Reproduce a Regression (1/3)

FUJITSU

1. The author/committer of the regression cause commit receive an email.
2. Identify the cause commit and the failed tests written in the email.
3. Check the failed tests in https://github.com/fujitsu/compiler-test-suite.
   - The file path of a test differs slightly from the test name. e.g., `Fujitsu/Fortran/0123/Fujitsu-Fortran-0123_0045.test` → `Fortran/0123/0123_0045.*`

```
Dear contributor,

Our automatic CI has detected problems related to your patch(es). Please
find some details below.

In tcwg_flang_test/main-aarch64-Ofast-sve_vls-lto-lld, after:
  | commit llvmorg-21-init-12345-g0123456789ab
  | Author: My Name <someone@example.com>
  | Date:   Tue June 10 12:34:56 2025 +0900
  |
  |     [flang] Add foobar feature (#123456)

Produces 2 regressions:
  |
  | regressions.sum:
  | Running test-suite:Fujitsu/Fortran/0123 ...
  | FAIL: test-suite :: Fujitsu/Fortran/0123/Fujitsu-Fortran-0123_0045.test
  | FAIL: test-suite :: Fujitsu/Fortran/0123/Fujitsu-Fortran-0123_0067.test
  | # "FAIL" means : the execution of the compiled binary failed / output
of the binary differs from the expected one

Used configuration :
 * Toolchain : cmake -G Ninja ../llvm/llvm "-DLLVM_ENABLE_PROJECTS=clang;ll
d;flang;openmp;clang-tools-extra" -DCMAKE_BUILD_TYPE=Release -DLLVM_ENABLE_
ASSERTIONS=True -DCMAKE_INSTALL_PREFIX=../llvm-install "-DLLVM_TARGETS_TO_B
UILD=AArch64" -DCLANG_DEFAULT_LINKER=lld
 * Testsuite : export LD_LIBRARY_PATH=$¥WORKSPACE/llvm-install/lib/aarch64-
unknown-linux-gnu$¥{LD_LIBRARY_PATH:+:$¥LD_LIBRARY_PATH}
 cmake -GNinja -DCMAKE_C_COMPILER="$¥WORKSPACE/llvm-install/bin/clang" -DCM
AKE_CXX_COMPILER="$¥WORKSPACE/llvm-install/bin/clang++" -DCMAKE_Fortran_COM
PILER="$¥WORKSPACE/llvm-install/bin/flang-new" -DCMAKE_BUILD_TYPE=Release -
DCMAKE_C_FLAGS= -DCMAKE_CXX_FLAGS= -DCMAKE_Fortran_FLAGS= -DCMAKE_C_FLAGS_R
ELEASE="-O3 -ffast-math -march=armv8.4-a+sve -msve-vector-bits=256 -mllvm -
treat-scalable-fixed-error-as-warning=false -flto -fuse-ld=lld -DNDEBUG" -D
CMAKE_CXX_FLAGS_RELEASE="-O3 -ffast-math -march=armv8.4-a+sve -msve-vector-
bits=256 -mllvm -treat-scalable-fixed-error-as-warning=false -flto -fuse-ld
=lld -DNDEBUG" -DCMAKE_Fortran_FLAGS_RELEASE="-O3 -ffast-math -march=armv8.
4-a+sve -msve-vector-bits=256 -mllvm -treat-scalable-fixed-error-as-warning
=false -flto -fuse-ld=lld -DNDEBUG" -DTEST_SUITE_FORTRAN=ON -DTEST_SUITE_SU
BDIRS=Fujitsu "$¥WORKSPACE/test/test-suite"

(snip)
```

Cause commit

Failed tests

Flang CI: How to Reproduce a Regression (2/3)

FUJITSU

4. Try running the failed tests manually using the flags written in the email.
   - You may need to remove architecture-specific flags (e.g., `-march=... -mcpu=... -msve-vector-bits=...`).
5. Build the whole test suite if you cannot reproduce the regression in 4.
   - Some tests require additional flags set by the test suite (e.g., `-fopenmp`).
   - The link for instruction to reproduce the build is written in the email.
   - You may need to remove architecture-specific flags.

```
Dear contributor,

Our automatic CI has detected problems related to your patch(es). Please
find some details below.

(snip)                                          Flags

Used configuration :
 * Toolchain : cmake -G Ninja ../llvm/llvm "-DLLVM_ENABLE_PROJECTS=clang;ll
d;flang;openmp;clang-tools-extra" -DCMAKE_BUILD_TYPE=Release -DLLVM_ENABLE_
ASSERTIONS=True -DCMAKE_INSTALL_PREFIX=../llvm-install "-DLLVM_TARGETS_TO_B
UILD=AArch64" -DCLANG_DEFAULT_LINKER=lld
 * Testsuite : export LD_LIBRARY_PATH=$¥WORKSPACE/llvm-install/lib/aarch64-
unknown-linux-gnu$¥{LD_LIBRARY_PATH:+:$¥LD_LIBRARY_PATH}
 cmake -GNinja -DCMAKE_C_COMPILER="$¥WORKSPACE/llvm-install/bin/clang" -DCM
AKE_CXX_COMPILER="$¥WORKSPACE/llvm-install/bin/clang++" -DCMAKE_Fortran_COM
PILER="$¥WORKSPACE/llvm-install/bin/flang-new" -DCMAKE_BUILD_TYPE=Release -
DCMAKE_C_FLAGS= -DCMAKE_CXX_FLAGS= -DCMAKE_Fortran_FLAGS= -DCMAKE_C_FLAGS_R
ELEASE="-O3 -ffast-math -march=armv8.4-a+sve -msve-vector-bits=256 -mllvm -
treat-scalable-fixed-error-as-warning=false -flto -fuse-ld=lld -DNDEBUG" -D
CMAKE_CXX_FLAGS_RELEASE="-O3 -ffast-math -march=armv8.4-a+sve -msve-vector-
bits=256 -mllvm -treat-scalable-fixed-error-as-warning=false -flto -fuse-ld
=lld -DNDEBUG" -DCMAKE_Fortran_FLAGS_RELEASE="-O3 -ffast-math -march=armv8.
4-a+sve -msve-vector-bits=256 -mllvm -treat-scalable-fixed-error-as-warning
=false -flto -fuse-ld=lld -DNDEBUG" -DTEST_SUITE_FORTRAN=ON -DTEST_SUITE_SU
BDIRS=Fujitsu "$¥WORKSPACE/test/test-suite"

We track this bug report under https://linaro.atlassian.net/browse/LLVM-
1234. Please let us know if you have a fix.

(snip)                                          Link to instruction

Instruction to reproduce the build : https://gitlab.com/Linaro/tcwg/ci/inte
resting-commits/-/raw/master/llvm/sha1/a0d699a8e686cba99690cf28463d14526c5b
fbc8/tcwg_flang_test/main-aarch64-Ofast-sve_vls-lto-lld/reproduction_instru
ctions.txt

(snip)
```

Flang CI: How to Reproduce a Regression (3/3)

FUJITSU

6. Check which configurations succeed and which fail.
   - See the bug report on our site.
   - See “Flang CI: Configurations” a few pages back.
   - Even if you cannot reproduce the regression, this information may help you investigate.
7. Reply to the email if you cannot reproduce the regression.

```
Dear contributor,

Our automatic CI has detected problems related to your patch(es). Please
find some details below.

(snip)

Used configuration :
 * Toolchain : cmake -G Ninja ../llvm/llvm "-DLLVM_ENABLE_PROJECTS=clang;lld;flang;openmp;clang-tools-extra" -DCMAKE_BUILD_TYPE=Release -DLLVM_ENABLE_ASSERTIONS=True -DCMAKE_INSTALL_PREFIX=../llvm-install "-DLLVM_TARGETS_TO_BUILD=AArch64" -DCLANG_DEFAULT_LINKER=lld
 * Testsuite : export LD_LIBRARY_PATH=$¥WORKSPACE/llvm-install/lib/aarch64-unknown-linux-gnu$¥{LD_LIBRARY_PATH:+:$¥LD_LIBRARY_PATH}
 cmake -GNinja -DCMAKE_C_COMPILER="$¥WORKSPACE/llvm-install/bin/clang" -DCMAKE_CXX_COMPILER="$¥WORKSPACE/llvm-install/bin/clang++" -DCMAKE_Fortran_COMPILER="$¥WORKSPACE/llvm-install/bin/flang-new" -DCMAKE_BUILD_TYPE=Release -DCMAKE_C_FLAGS= -DCMAKE_CXX_FLAGS= -DCMAKE_Fortran_FLAGS= -DCMAKE_C_FLAGS_RELEASE="-O3 -ffast-math -march=armv8.4-a+sve -msve-vector-bits=256 -mllvm -treat-scalable-fixed-error-as-warning=false -flto -fuse-ld=lld -DNDEBUG" -DCMAKE_CXX_FLAGS_RELEASE="-O3 -ffast-math -march=armv8.4-a+sve -msve-vector-bits=256 -mllvm -treat-scalable-fixed-error-as-warning=false -flto -fuse-ld=lld -DNDEBUG" -DCMAKE_Fortran_FLAGS_RELEASE="-O3 -ffast-math -march=armv8.4-a+sve -msve-vector-bits=256 -mllvm -treat-scalable-fixed-error-as-warning=false -flto -fuse-ld=lld -DNDEBUG" -DTEST_SUITE_FORTRAN=ON -DTEST_SUITE_SUBDIRS=Fujitsu "$¥WORKSPACE/test/test-suite"

We track this bug report under https://linaro.atlassian.net/browse/LLVM-
1234. Please let us know if you have a fix.

(snip)
```

Link to the bug report

```
Instruction to reproduce the build : https://gitlab.com/Linaro/tcwg/ci/interesting-commits/-/raw/master/llvm/sha1/a0d699a8e686cba99690cf28463d14526c5bfbc8/tcwg_flang_test/main-aarch64-Ofast-sve_vls-lto-lld/reproduction_instructions.txt

(snip)
```

Resource

FUJITSU

- Linaro Connect 2025 “LIS25-232 Quality Assurance of Flang Using the Fujitsu Compiler Test Suite”
  - https://resources.linaro.org/en/resource/cQZfP7Z82euYwM57mYn1Z3

Thank you