Updates of TF-M build system restructure
Build split
Review of proposal

+ Separate builds of BL, TF-M SPE and NSPE
  - Build each module as a CMake external project under a virtual root project
  - Simulate actual NS + TF-M integration scenarios
  - Simplify build configuration, such as multi-core settings and FP compiler flags

```
if(BL)
  ExternalProject_Add(BL
    ....
  )
endif()

ExternalProject_Add(TF-M-SPE
  ....
)

if(NS)
  ExternalProject_Add(TF-M-NSPE
    ....
  )
endif()
```
Implementation

Major restructure

- Almost identical to those in proposal
  • A virtual root project + independent external projects
    ◦ A root configuration step for fundamental configs
    ◦ Collect command line arguments in root project and pass them to external projects
  • Secure/non-secure/bootloader specific configs
    ◦ And share common config
  • Flags `S_BUILD/NS_BUILD/BL_BUILD` indicate current build
    ◦ Also used to wrap platform builds
    ◦ Alternatively separate platform `CMakeLists.txt` into `platform_s/ns/bl.cmake`
  • Define root source/build directory variables
    `TFM_SOURCE/BINARY_DIR`, to replace `CMAKE_SOURCE/BINARY_DIR`
    ◦ `CMAKE_SOURCE/BINARY_DIR` vary in each external project

```
if(BL_BUILD)
  target_sources(platform_bl2 ...)
  ...
endif()
if(S_BUILD)
  target_sources(platform_s ...)
  ...
endif()
if(NS_BUILD)
  target_sources(platform_ns ...)
  ...
endif()
```

Virtual project

```
Root project config

TFM_SOURCE_DIR/TFM_BINARY_DIR

Y

BL2?

N

BL config

BL build

SPE config

TF-M build

NS?

N

NSPE config

NSPE build

Backward compatible install
```

target platform's `CMakeLists.txt`
Implementation

New features since proposal

+ Integrate Kconfig-based configuration
  - Kconfig generates a common project config header file which can be easily included by all external projects
  - Still require a post config step to collect configs not supported yet (by Kconfig)
Implementation

New features since proposal

- Integrate Kconfig-based configuration
  - Simplify the overall configuration after Kconfig can support more configs
  - By skipping duplicated configuration steps in multiple external projects
Implementation

New features since proposal

• Download 3rd party libraries only once
  • Avoid duplicated downloads of shared 3rd party libraries used by multiple projects.
  • Paths of 3rd party libraries are maintained in a temporary CMake file under build folder
  • When a 3rd party library is requested for the first time, its path is written into the temporary CMake file
  • All external projects include this temporary CMake file to fetch 3rd party library which have been downloaded.
• Implemented in fetch_remote_library(). Main routines are unaware of this change.
Progress and status

 Patch status
  - Patch set under review
    - TF-M changes
    - tf-m-tests changes
  - OpenCI nightly test passed
    - With some tweaks on static check

 Plan
  - Any review comment is welcome!
  - Documentation update
  - It is expected that major implementations can be merged in 2023Q2
A step forward

Build NS from NS
Start NS/tests build from tf-m-tests

**Current build sequence**

- All types of builds are triggered from TF-M
  - tf-m-tests is integrated as a component
  - TF-M and tf-m-tests are tightly coupled

**Proposed build sequence**

- Invert the dependency
  - TF-M alone builds secure images only
  - Build NS/Regression/PSA Arch tests from tf-m-tests
What’s the benefit?

+ Simplify TF-M development and integration
  - Minimize complexities of various NS scenarios in TF-M, such as diverse NS app source directories.
  - Integrate TF-M with another NS implementation simply via `add_subdirectory()`
  - TF-M only focus on its own 3rd party dependencies. Tf-m-tests takes over the management of regression tests and PSA Arch tests.

```
simplified TF-M development and integration
```

Build TF-M secure images alone. Secure binaries are released for downstreaming integration

Build TF-M together with NS RTOS. Minimize the effort to configure TF-M to fit NS specific implementation. For example, `NS_EVALUATION_APP_PATH` is unnecessary any more.
Implementation
Straightforward based on build split

- **Tf-m-tests**
  - Download *latest* trusted-firmware-m
    - Still rely on trusted-firmware-m cmake utils and common config
  - **Build S via add_subdirectory()**
  - Build NS as an external project

- **Trusted-firmware-m**
  - Remove NS build
  - Support to accept parent scope configuration
    - Build command configs

- **OpenCI**
  - Trigger regression tests from tf-m-tests
  - tf-m-tests patches can be verified via openCI
Side effects

- Tweak the build scripts
  - Build regression tests/PSA Arch tests from tf-m-tests folder
  - Otherwise, TF-M build fails with test flags set
Side effects

TF-M and tf-m-tests commit synchronization

- tf-m-tests always downloads *latest* TF-M code base in this proposal
  - It is the responsibility of tf-m-tests to track and verify TF-M latest master branch
  - TF-M is updated frequently. Extremely inconvenient to track and update TF-M commit ID in tf-m-tests.
  - It enables tf-m-tests patches to be verified by per-patch CI
    - If those patches only affect tf-m-tests

- However
  - If users work on a specific TF-M/tf-m-tests commit, it becomes difficult to figure out the corresponding peer commit.
Side effects

+ TF-M and tf-m-tests commit synchronization
  • Insert a trailer in *tf-m-tests* commit message to “hook” a TF-M change ID
    + Tf-m-tests git commit-msg automatically inserts a TF-M-Change-Id-hook trailer when committing
    + The trailer value is set to TBD as a dummy one by default
    + Developers shall manually fill in the corresponding TF-M Change ID
    + The Change ID keeps the same while the commit ID varies during development of a patch

commit db3b7cdefb8b08f2fcfd3ff9941f5bbe1084c759c
Author: David Hu <david.hu@arm.com>
Date:   Wed Mar 8 10:22:10 2023 +0800

This is a commit title
This is commit message detail.
Signed-off-by: David Hu <david.hu@arm.com>
Change-Id: Iadb8b30d524cdffedd3e75bda25f92a5b7e4b81db
TF-M-Change-Id-hook: TBD

commit db3b7cdefb8b08f2fcfd3ff9941f5bbe1084c759c
Author: David Hu <david.hu@arm.com>
Date:   Wed Mar 8 10:22:10 2023 +0800

This is a commit title
This is commit message detail.
Signed-off-by: David Hu <david.hu@arm.com>
Change-Id: Iadb8b30d524cdffedd3e75bda25f92a5b7e4b81db
TF-M-Change-Id-hook: Iefdc09834a723ce1f78fd67ace1d83

```
git commit -s ...
git commit --amend
```
Side effects

TF-M and tf-m-tests commit synchronization

• Insert a trailer in \textit{tf-m-tests} commit message to “hook” a TF-M change ID

Hook \textit{tf-m-tests} change to the coupled TF-M commit

Hook \textit{tf-m-tests} change to the “current” latest TF-M commit.
If other commits are merged in TF-M during development, unnecessary to update the hook.
But eventually CI tests with the latest TF-M shall be passed.
Side effects

+ TF-M and tf-m-tests commit synchronization
  - Users work on a specific *tf-m-tests* commit
    - Locate the corresponding TF-M commit according to the TF-M-Change-Id-hook
    - Use a local copy of TF-M repo with the corresponding commit ID
  
  - Users work on a specific *TF-M* commit
    - TF-M provides a tool to locate the corresponding tf-m-tests commit
    - Use a local copy of tf-m-tests repo with the commit ID found out by the tool
Status

🔗 A PoC for preview
  • TF-M patch
  • Tf-m-tests patch

🔗 Ongoing tasks
  • OpenCI
    ✚ Tweak tests build
    ✚ Enable per-patch CI check against tf-m-tests trailer
  
  • tf-m-tests
    ✚ Enable the trailer in commit-msg hook
  
  • TF-M
    ✚ Implement the tool for locating tf-m-tests commit
Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
ありがとう
Asante
Merci
감사합니다
धन्यवाद
 شكرا
ধন্যবাদ
תודה