Fast Model SPKG Workflow

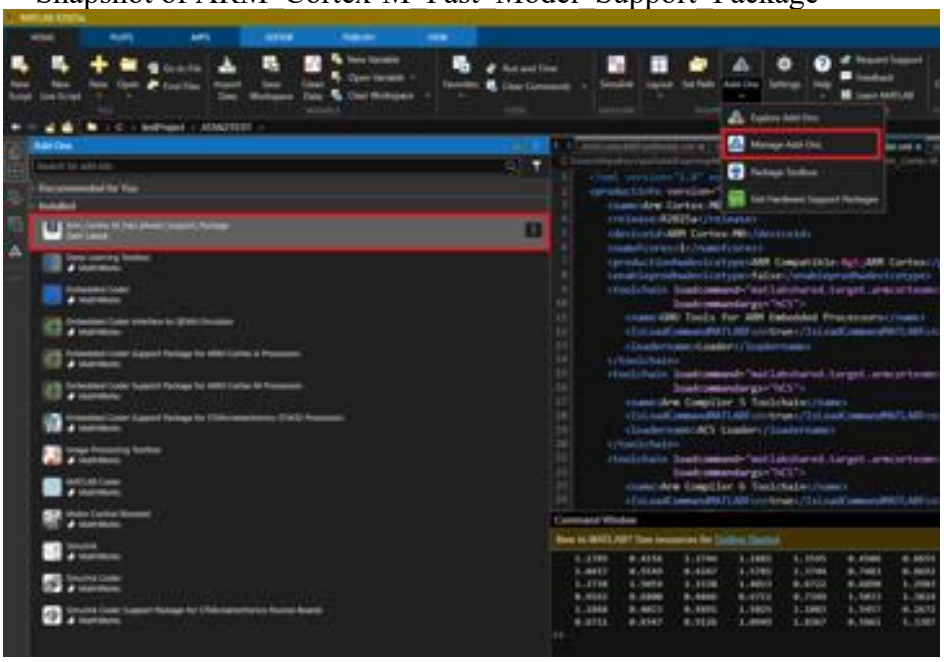
This documentation mentions how to run PIL using fast Model Support Package and benchmark in MATLAB 25a using ARM Development Studio latest version.

1) Installation of Fast Model SPKG in 25a

1) Installation of Fast Model SPKG in 25a
For further details, see [URL].
While a direct link to this specific section cannot be generated within this context, you can easily reference it by copying the section header, "1) Installation of Fast Model SPKG in 25a," and using your document's search function (Ctrl+F or Cmd+F) to quickly locate this part. For detailed sharing, consider bookmarking or highlighting this section within your documentation platform or adding an internal anchor, if supported.

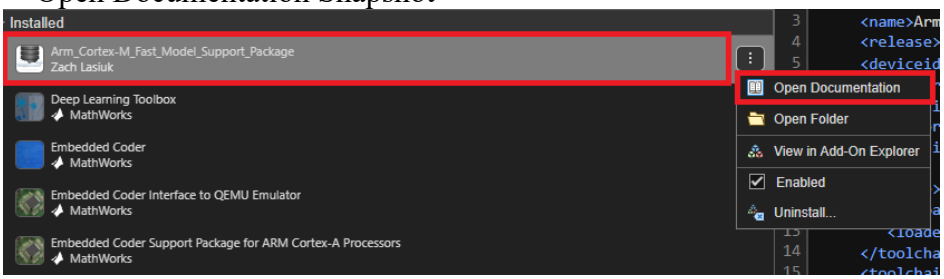
1. Install Fast Model SPKG from Arm_Cortex-M_Fast_Model_Support_Package - File Exchange - MATLAB Central . This is compatible with 2018b, to make it work for 25a, few work arounds were needed.
2. Once Installed, In Add-Ons → Manage Add-Ons, newly installed SPKG is shown.

Snapshot of ARM_Cortex-M_Fast_Model_Support_Package



3. Documentation for this newly installed SPKG can be obtained as shown in snapshot below -

Open Documentation Snapshot



4. Follow the below steps to make SPKG work in 25a.

Resolution for SPKG incompatibility

Issue - SPKG won't show up and we won't see any Fast Model related hardware in Simulink.

```
sl_refresh_customizations
Warning: An error occurred while evaluating the rtwTargetInfo file "C:\Users\[redacted]\AppData\Roaming\MathWorks\MATLAB
Add-Ons\Toolboxes\Arm_Cortex-M_Fast_Model_Support_Package\ArmCortexM0FastModel\rtwTargetInfo.m": 'm0' has been created in R2018b and is not compatible
with this release of MATLAB. Please recreate the target in R2025a before attempting to register or use the target.
> In coder.targetreg.internal/TargetRegistry/registerTargetInfosOnPath
In coder.targetreg.internal.TargetRegistry.initialize
In coder.targetreg.internal.TargetRegistry.getWithoutDataLoad
In RTW.TargetRegistry.getInstance
In registerTargetInfo (line 14)
In customizationSimulinkCoderCore (line 16)
In slCustomizer/refresh
In slCustomizer.callRefresh
In slCustomizer>@()slCustomizer.callRefresh()
In slCustomizer.staticRefresh
In sl_refresh_customizations
Warning: An error occurred while evaluating the rtwTargetInfo file "C:\Users\[redacted]\AppData\Roaming\MathWorks\MATLAB
Add-Ons\Toolboxes\Arm_Cortex-M_Fast_Model_Support_Package\ArmCortexM0pFastModel\rtwTargetInfo.m": 'm0plus' has been created in R2018b and is not
compatible with this release of MATLAB. Please recreate the target in R2025a before attempting to register or use the target.
> In coder.targetreg.internal/TargetRegistry/registerTargetInfosOnPath
```

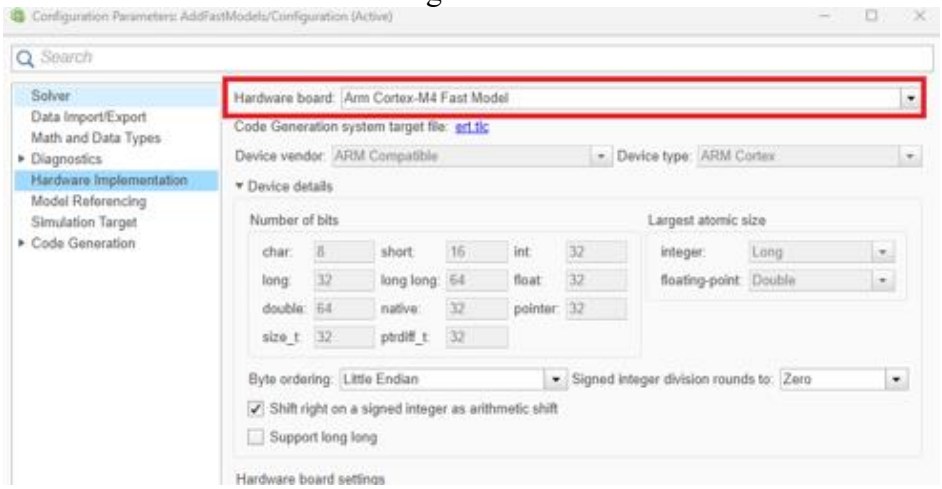
Resolution -

- Open Folder (`Manage Add-Ons --> ARM_Cortex-M_Fast_Model_Support_Package -> Open Folder` section in manage add-ons) of this newly installed SPKG
- Search for "R2018b" in the folder. Replace all instances of R2018b with R2025a in all xml files.

- After this run `sl_refresh_customizations` , that will resolve the issue and in hardware section we can see fast Models SPKG working

5. Open Simulink model. Got to ‘Hardware Implementation’ setting. Under hardware board, any Arm Cortex-M Fast Model can be selected now.

M4 fast Model hardware configure screenshot

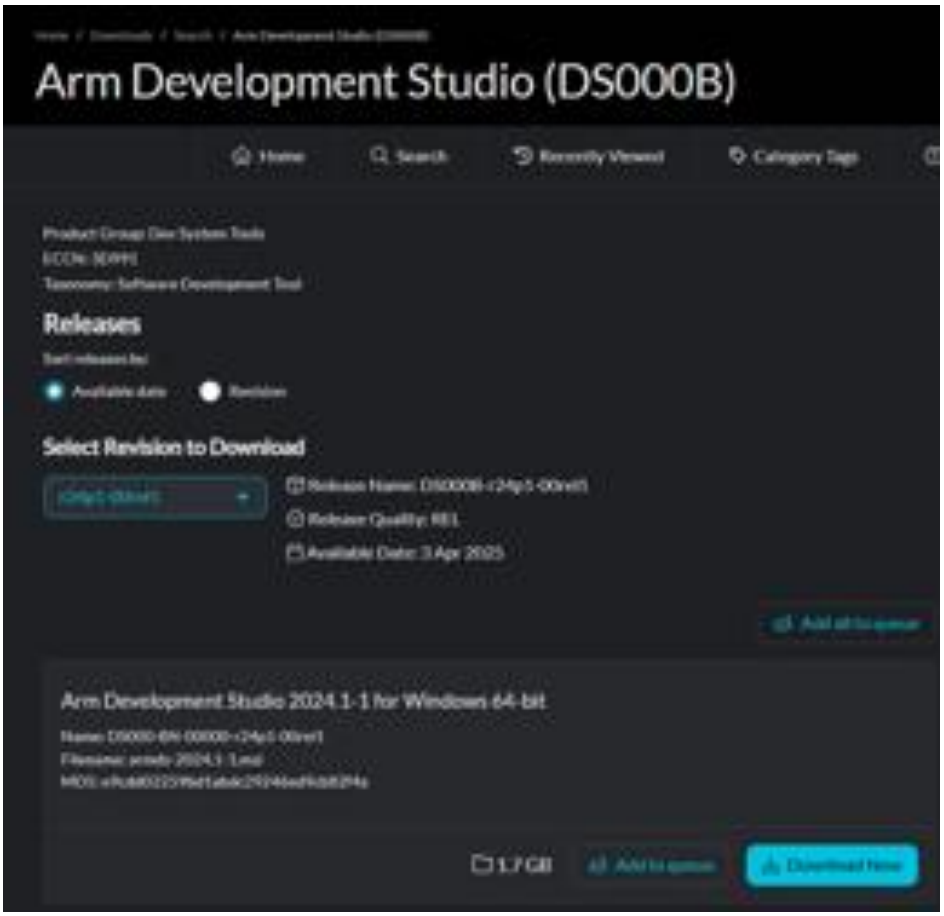


---

2) ARM Development Studio installation

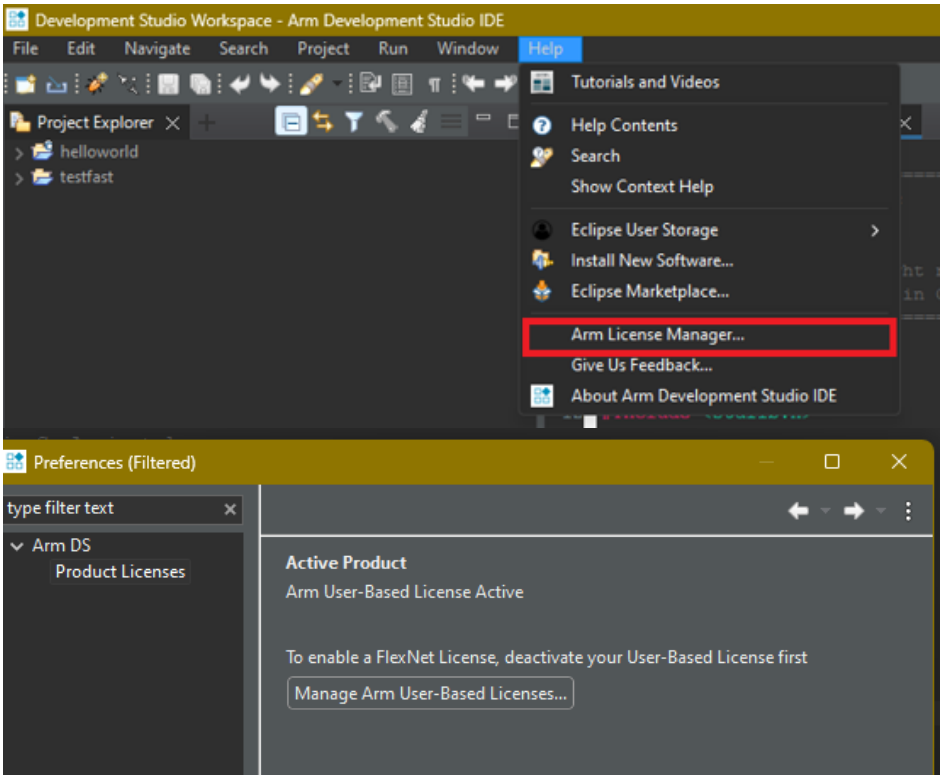
1. Install ARM Development Studio. Fast Model support package documentation is pointing DS-5, but this workflow is tested with latest ARM development studio release (2024.1-1 for Windows 64-bit). This can be downloaded from this link.

ARM Development Studio Snapshot



2. Install and open ARM development studio. Open Arm DS IDE. User needs to have valid ARM development studio License. Our team had a license which we used for license configuration.

˅ Configure License



- This license can also be checked in online ARM portal through this [link]

---

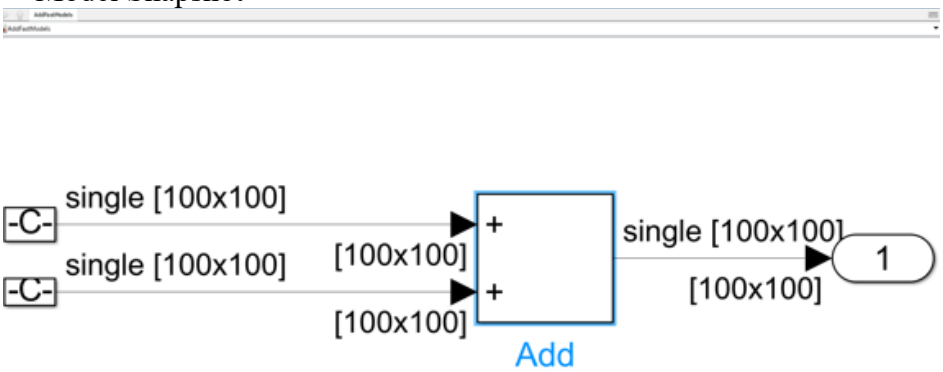
3) Setup In MATLAB

1. Set the relevant environment variables in MATLAB. Documentation for SPKG points to DS-5 paths, But paths are modified to use ARM Development Studio.

```
setenv('ARM_PRODUCT_PATH',"C:\Program Files\Arm\Development Studio 2024.1-1\sw\mappings")
setenv("ARM_TOOL_VARIANT", "ult")
```

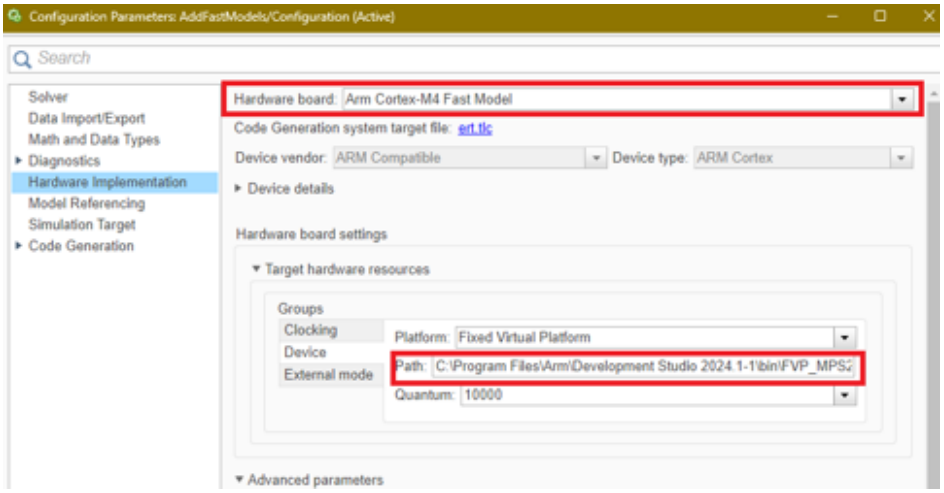
2. Create a Model, here simple addition Model is created.

Model Snapshot



3. Open Model settings. Set the hardware to M4 fast Model. Under Target hardware resources → Device, Update the Path to `"C:\Program Files\Arm\Development Studio 2024.1-1\bin\FVP_MPS2_Cortex-M4.exe"` (FVP path needs to be updated based on hardware chosen)

M4 fast Model configuration



4. We faced compilation errors during build, So we updated Include paths in `Code Generation >> Build configuration >> C Compiler` as shown below :

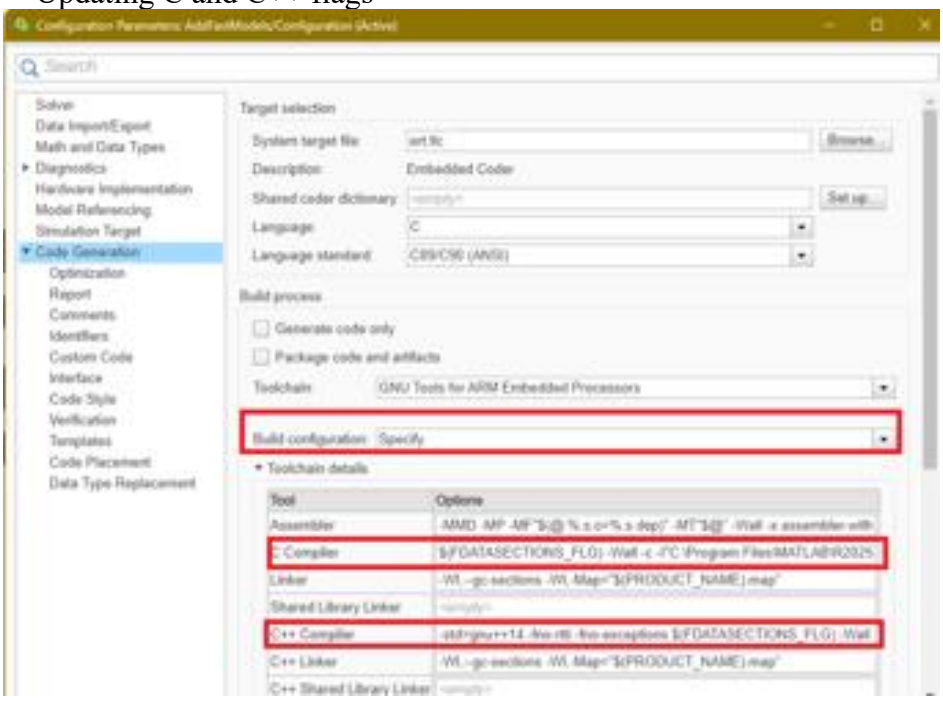
Resolution for compilation errors.

Issue - Once we tried to build , we faced Compilation error

Resolution -

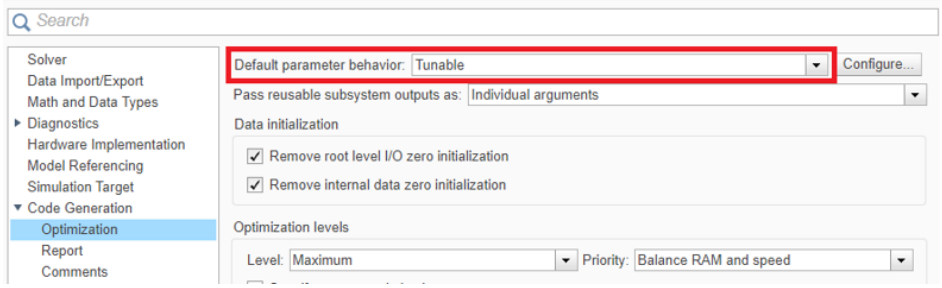
- Append `-I"C:\Program Files\MATLAB\R2025a\toolbox\coder\rtiostream\src"` in C and C++ compiler flags as shown in below snapshot.

Updating C and C++ flags



5. Set the default parameter behavior to tunable as shown in snapshot below.

Default parameter behavior setting



4) Simulink Model PIL Execution

Run PIL simulation using below script

```
% Fast Model PIL workflow
```


```
rng(0); in1 = rand(10,10,'single'); in2 = in1;
ModelName = 'AddFastModels'; % Model Name needs to be updated when other
model is used.
open_system(ModelName);
load_system(ModelName);
set_param(ModelName,'SimulationMode','processor-in-the-loop (pil)');
outFastModel = sim(ModelName);
```

Output Log for Fast Model PIL is shown below -

Output Log

```
### # Output for PIL - outFastModel = sim(ModelName)
### Skipped unpacking from Simulink cache file "AddFastModels.slxc" because
the relevant build artifacts on disk are up to date.
```

```
### Searching for referenced models in model 'AddFastModels'.
### Total of 1 models to build.
### Starting build procedure for: AddFastModels
### Generating code and artifacts to 'Model specific' folder structure
### Generating code into build folder:
C:\testProject\ATAN2TEST\AddFastModels_ert_rtw
### Generated code for 'AddFastModels' is up to date because no structural,
parameter or code replacement library changes were found.
### Using toolchain: GNU Tools for ARM Embedded Processors
### 'C:\testProject\ATAN2TEST\AddFastModels_ert_rtw\AddFastModels.mk' is up
to date.
### Building 'AddFastModels': "C:\PROGRA~1\MATLAB\R2025a\bin\win64\gmake"  -f
AddFastModels.mk buildobj

C:\testProject\ATAN2TEST\AddFastModels_ert_rtw>cd .

C:\testProject\ATAN2TEST\AddFastModels_ert_rtw>if "buildobj" == ""
("C:\PROGRA~1\MATLAB\R2025a\bin\win64\gmake"  -f AddFastModels.mk all )  else
("C:\PROGRA~1\MATLAB\R2025a\bin\win64\gmake"  -f AddFastModels.mk buildobj )
"### Successfully generated all binary outputs."

C:\testProject\ATAN2TEST\AddFastModels_ert_rtw>exit /B 0
### Successful completion of build procedure for: AddFastModels

Build Summary

Top model targets:

Model          Build Reason                                  Status          Build
Duration
================================================================================
=========
AddFastModels  Compilation artifacts were out of date.  Code compiled.  0h 0m
5.0574s

1 of 1 models built (0 models already up to date)
Build duration: 0h 0m 5.8001s
### Connectivity configuration for component "AddFastModels": m4 ###
### Preparing to start PIL simulation ...
### Using toolchain: GNU Tools for ARM Embedded Processors
### 'C:\testProject\ATAN2TEST\AddFastModels_ert_rtw\pil\AddFastModels.mk' is
up to date.
### Building 'AddFastModels': "C:\PROGRA~1\MATLAB\R2025a\bin\win64\gmake"  -f
AddFastModels.mk all

C:\testProject\ATAN2TEST\AddFastModels_ert_rtw\pil>cd .

C:\testProject\ATAN2TEST\AddFastModels_ert_rtw\pil>if "all" == ""
("C:\PROGRA~1\MATLAB\R2025a\bin\win64\gmake"  -f AddFastModels.mk all )  else
("C:\PROGRA~1\MATLAB\R2025a\bin\win64\gmake"  -f AddFastModels.mk all )
"### Invoking postbuild tool "Binary Converter" ..."
"C:/ProgramData/MATLAB/SupportPackages/R2025a/3P.instrset/gnuarm-
armcortex.instrset/win/bin/arm-none-eabi-objcopy" -O binary
./AddFastModels.elf ./AddFastModels.bin
"### Done invoking postbuild tool."
"### Invoking postbuild tool "Hex Converter" ..."
```

```
"C:/ProgramData/MATLAB/SupportPackages/R2025a/3P.instrset/gnuarm-
armcortex.instrset/win/bin/arm-none-eabi-objcopy" -O ihex ./AddFastModels.elf
./AddFastModels.hex
"### Done invoking postbuild tool."
"### Invoking postbuild tool "Executable Size" ..."
"C:/ProgramData/MATLAB/SupportPackages/R2025a/3P.instrset/gnuarm-
armcortex.instrset/win/bin/arm-none-eabi-size" ./AddFastModels.elf
   text    data     bss     dec     hex filename
  11299   80104   40560  131963   2037b ./AddFastModels.elf
"### Done invoking postbuild tool."
"### Successfully generated all binary outputs."

C:\testProject\ATAN2TEST\AddFastModels_ert_rtw\pil>exit /B 0
### Updating code generation report with PIL files ...
Warning: Model Web view in code generation report requires Simulink Report
Generator, which is not installed.
> In Simulink.report.ReportInfo/emitWebview
In rtw.report.ReportInfo/emitHTML>loc_emitHTML_V2
In rtw.report.ReportInfo/emitHTML
In rtw.report.ReportInfo/generate>locGenerate
In rtw.report.ReportInfo/generate
In rtw.report.generate
In coder.connectivity/SimulinkInterface/updateReport (line 695)
In rtw.pil/SILPILInterface/buildApplication (line 1191)
In rtw.pil.ModelBlockPIL.XILBuildHook (line 366)
In slprivate
In pil_target_sim
In sl_feval
### Starting application: 'AddFastModels_ert_rtw\pil\AddFastModels.elf'
### Downloading
application...C:\testProject\ATAN2TEST\AddFastModels_ert_rtw\pil\AddFastModel
s.elf
Simulation complete. Application will now be terminated safely.

outFastModel =

  Simulink.SimulationOutput:

                  tout: [51x1 double]
                  yout: [1x1 Simulink.SimulationData.Dataset]

     SimulationMetadata: [1x1 Simulink.SimulationMetadata]
           ErrorMessage: [0x0 char]
```

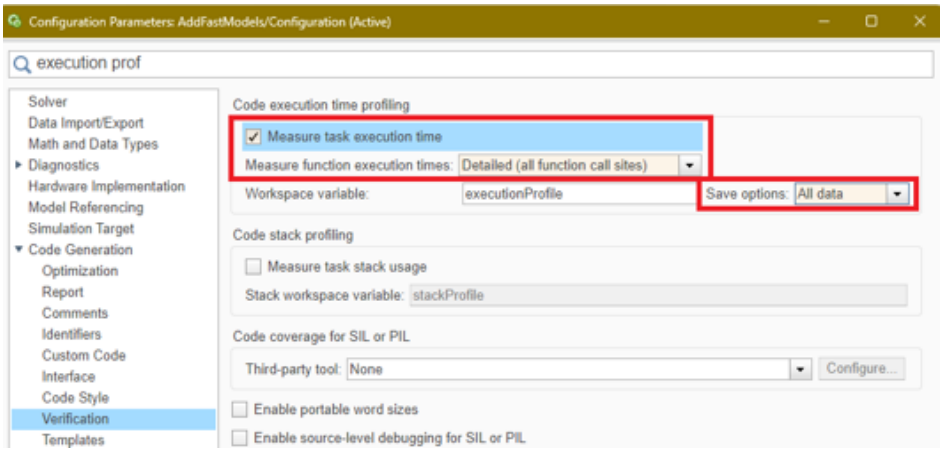
---

5) Benchmarking

1. Set profiling to true.

Profiling set to true snapshot

Set `Measure task execution time --> True`, `Measure function execution times --> Detailed`, Workspace `save options → All Data`



2. Generate`ARM Cortex-M4 MPS2` (M4 QEMU), `STM32 Nucleo F401RE` and `ARM Cortex-M4 Fast Model` execution times.

Script to generate execution timings

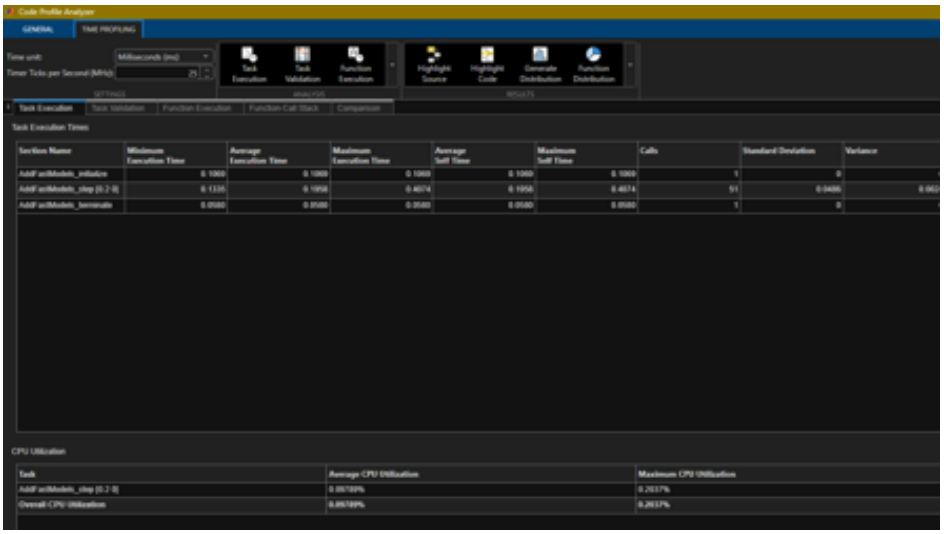
```
%% Compare timings.
in1 = rand(10,10,'single'); in2 = in1;
ModelName = 'AddFastModels';
open_system(ModelName);
load_system(ModelName);
%%
% M4 QEMU
set_param(ModelName,"HardwareBoard",'ARM Cortex-M4 (MPS2)');
set_param(ModelName,'SimulationMode','processor-in-the-loop (pil)');
outM4QEMU = sim(ModelName);

%%
% M4 Fast Model FVP
set_param(ModelName,"HardwareBoard",'Arm Cortex-M4 Fast Model');
% Set device to "C:\Program Files\Arm\Development Studio2024.1-
1\bin\FVP_MPS2_Cortex-M4.exe"
set_param(ModelName,'SimulationMode','processor-in-the-loop (pil)');
outM4FM = sim(ModelName);

%%
% M4 Hardware
set_param(ModelName,"HardwareBoard",'STM32 Nucleo F401RE');
set_param(ModelName,'SimulationMode','processor-in-the-loop (pil)');
outM4Hardware = sim(ModelName);
```

3. Visualize benchmark numbers on QEMU, M4 Fast Model and Hardware.

QEMU Benchmarking

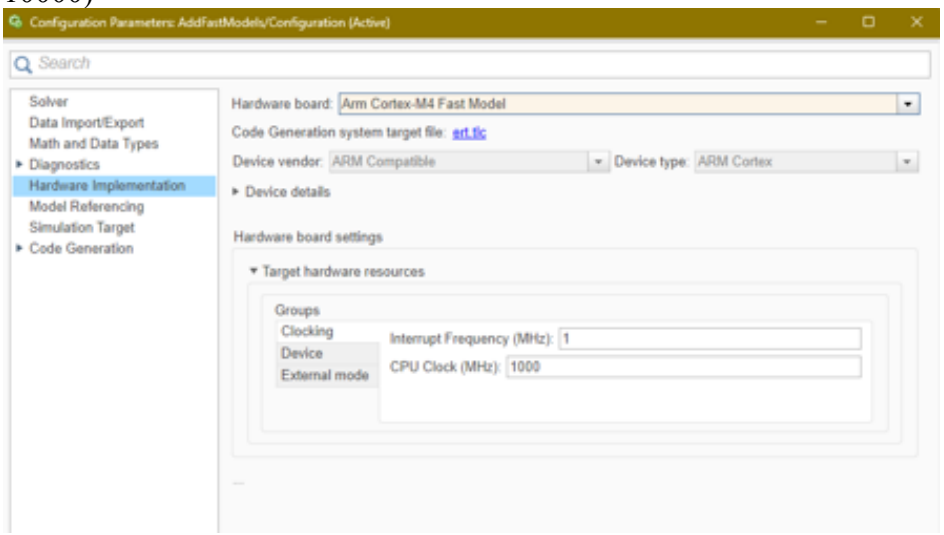
Timings of M4QEMU - `coder.profile.show(outM4QEMU.executionProfile)`

Clock - 25MHz



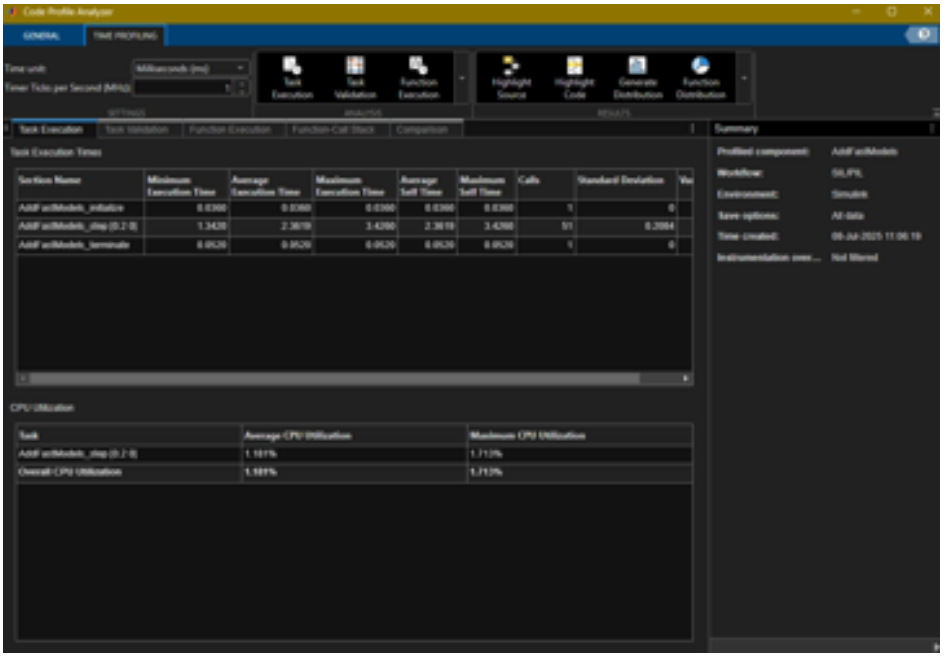
˅ Fast Model Benchmarking

˅ Default M4 fast Model Values (Interrupt Frequency - 1, CPU Clock - 1000 MHz, Quantum - 10000)



Benchmarking is done with default values - (Interrupt Frequency - 1, CPU Clock - 1000 MHz, Quantum - 10000).

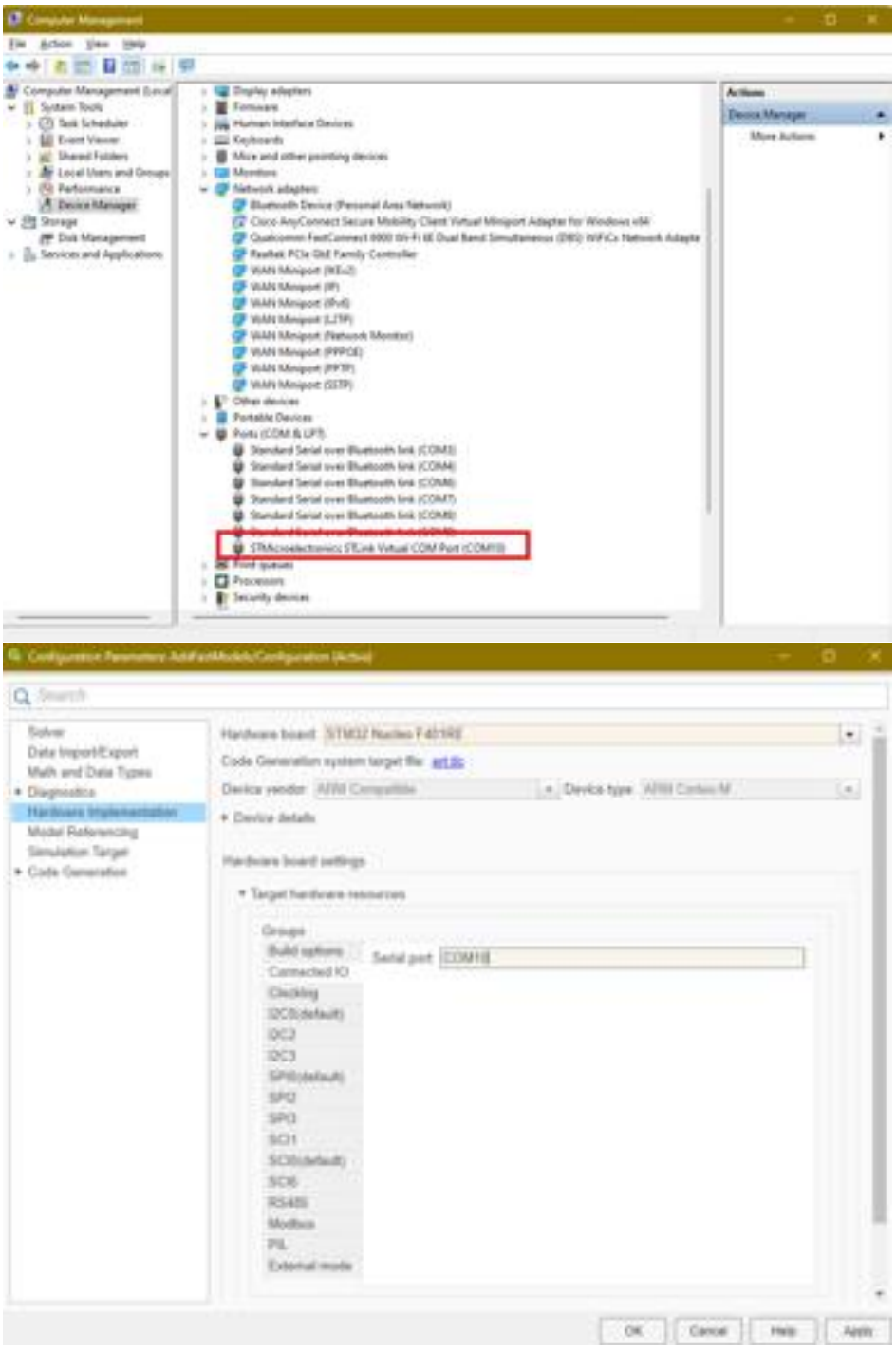
Timings of M4 Fast Model - `coder.profile.show(outM4FM.executionProfile)`, Timer Ticks Per Second(MHz) is different from that of QEMU.

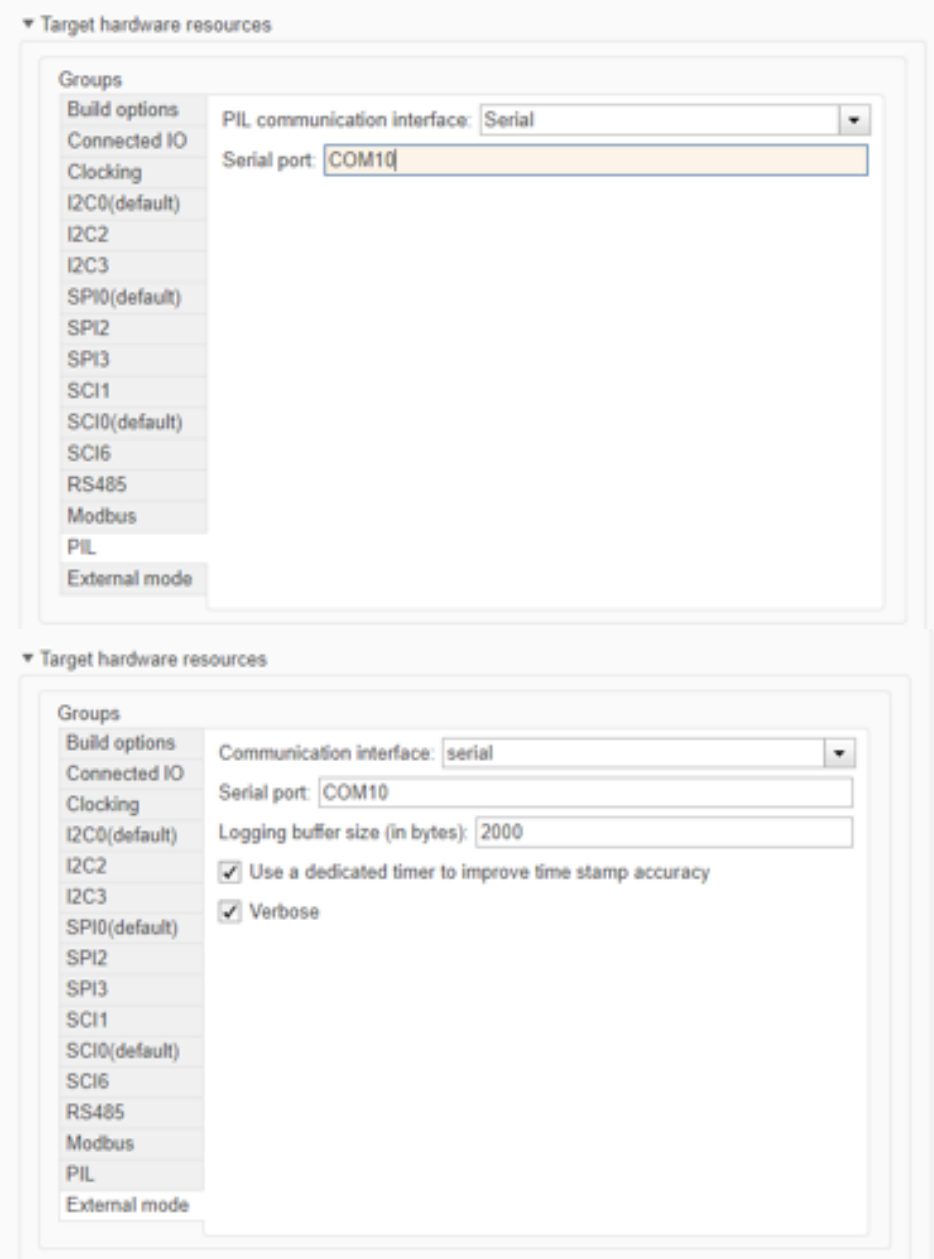


Hardware Benchmarking

- Set COM Port based on hardware COM port-

Configure COM Port


Computer Management
Device Manager
Ports (COM & LPT)
STMicroelectronics STLink Virtual COM Port (COM10)
Hardware board
Code Generation system target file
Device vendor
Device type
Hardware board settings
Target hardware resources
Serial port
COM10
OK
Cancel
Help
Apply



- Timings of M4 Fast Model - `coder.profile.show(outM4Hardware.executionProfile)`, Clock is 84MHz.

-