1 Introduction

The i.MX RT1170 crossover processor is setting speed records at 1 GHz. This ground-breaking family combines superior computing power and multiple media capabilities with more usability and real-time functionality. The dual-core i.MX RT1170 runs on the Arm® Cortex®-M7 core at 1 GHz and Arm Cortex-M4 at 400 MHz, while providing best-in-class security. The i.MX RT1170 MCU offers support over a wide temperature range and is qualified for consumer, industrial, and automotive markets.

The RT1170 series is divided into two parts: single-core and dual-core. For details on the dual-core part and the most common settings, see AN13104. This application note outline steps to enter the STBY mode on a single core RT1170 and simulate a single-core state on RT1170 EVK.

NOTE

The hardware is MIMXRT1170 EVK RevC1 (referred as EVK) and the software is based on SDK 2.9.0 with IAR IDE. The SDK demo has different projects for single-core and dual-core. The SDK demo also supports flash target and ram target.

2 Overview

#unique_3 shows that the basic principle of entering STBY is that both CPUs enter low-power mode and issue STBY requirements. The first principle is the CPU entering a low-power mode. The low-power mode here means CPU’s status: . It can be WAIT, STOP, or SUSPEND mode. Each CPU can be in any state except run mode.

Such as:

• CM7 WAIT, CM4 STOP
• CM7 STOP, CM4 STOP
• CM7 SUSPENDED, CM4 WAIT
• CM7 SUSPEND, CM4 SUSPEND

The second principle is that both CPUs have sent the STBY request. If any CPU did not send it, the system cannot enter the STBY mode.
For single core silicon, CM4 cannot send this request signal. Therefore, some commands are a must to force the system enter the STBY mode.

For a single core silicon, the basic principle is CM7 enter a low-power mode and sent the stby_request signal. The low-power mode here can be WAIT, STOP, or SUSPEND mode.

The basic flow is:

**3 Steps to enter the STBY mode on single-core RT1170**

To enter the STBY mode, both the cores send the stby_request. However, in a single-core silicon, such as RT1172 and RT1176, CM4 cannot send it. Therefore, the Force_STBY function is used. The Force_STBY function forces CM4 requesting the STBY mode. Once the M7 enters the low-power mode and sends an stby_request, the whole chip enters the STBY mode.

```c
GPC_STBY_CTRL->STBY_MISC |= GPC_STBY_CTRL_STBY_MISC_FORCE_CPU1_STBY_MASK;
```

**4 Steps to simulate single-core state on RT1170 EVK**

EVK is based on RT1176 silicon, which is a dual-core silicon. Customer can simulate it as a single-core state after some settings.

**NOTE**

Simulation as a single-core state after some settings is only for early evaluation.

The basic method is CM4 entering suspend mode and sending the STBY request.

1. After system boot, CM7 configures some registers.
2. Write PGMC CM4 CPC register as CPU mode with power-off at SUSPEND.
3. Write GPC1 IRQ mask registers as 0xFFFFFFFF so that no interrupt can wake it up.
4. Write GPC1 NON IRQ mask register as 0x3 to avoid a pending interrupt from debugger stop entering low-power mode.
5. Write GPC1 control register to request SUSPEND mode and SBTY mode.
6. CM7 releases CM4 by SRC_SCR register. The silicon is now in the dual-core state.
7. CM4 runs one instruction: assert WFI to trigger GPC1 low-power sequence to let itself from entering SUSPEND mode and send the STBY request.

8. When CM4 is in SUSPEND mode, CM7 writes some registers to lock CM4 by software.

9. Write GPC1 AUTHEN register to lock CM1 register access.

10. Write CM4 CCGR slice in CCM to gate off CM4 clock.

11. Write CM4 CCGR AUTHEN register ‘allow_list’ and ‘lock’ fields meaning no CPU can access CM4 CCGR register again. Thus CM7 software cannot turn on CM4 clock again.

12. Write PGMC CM4 CPC AUTHEN register to lock any access so that software cannot turn on CM4 power again.

The attachment of this application note is provided with detailed code for reference. The basic method is to let CM4 enter suspend mode and send STBY request. Based on this, mask all the wake-up sources in GPC CM1 and then lock the access rights of these registers: any CPU cannot access and modify. CM4 then enters the suspend mode and there is no wake-up source to wake it up and any CPU does not have the access right to change these settings.

The following API is able to complete above configuration.

```c
void Powerdown_CM4(void);
```

For details, see the attachment with this application note.

## 5 Differences between a single-core silicon and simulated single-core on RT1176

There are some differences between a single-core silicon and a simulated single-core on RT1176. The first one is RDC. The single-core silicon cannot use RDC to assign a peripheral to a domain. If CM7 wants to access the RDC, a fault occurs. However, a simulated single-core on RT1176 can use RDC. For single-core silicon, all the peripherals are assigned to M7 domain. As a result, only 1 stop request from M7 domain is required during the handshake and only need to check the stop_ack from M7 domain. For details on the handshake, see AN13104, Chapter 4.5.

<table>
<thead>
<tr>
<th></th>
<th>Single Core</th>
<th>Simulated single core on RT1176</th>
</tr>
</thead>
<tbody>
<tr>
<td>STBY Method</td>
<td>Force_STBY</td>
<td>CM4 enters Suspend STBY</td>
</tr>
<tr>
<td>RDC</td>
<td>Not support</td>
<td>Support</td>
</tr>
<tr>
<td>Handshake</td>
<td>Only CM7 sends stop_req and checks the stop_ack</td>
<td>See AN13104, Chapter 4.5</td>
</tr>
</tbody>
</table>

## 6 Steps to create a single-core project on SDK

The MIMXRT1170 EVK supports single-core program executions. This section lists the steps to set up a single-core project by using different IDEs (IAR, Keil, and Arm GCC).

### 6.1 Run a demo using IAR

The following steps illustrate the power_mode_switch demo from SDK 2.9.0.

1. The location of the desired demo is:

   ```
   <install_dir>/boards/evkmimxrt1170/demo_apps/power_mode_switch/bm/core0/iar/power_mode_switch_bm_core0.eww
   ```

2. Import the demo into the IAR IDE.
Steps to create a single-core project on SDK

1. Enter STBY mode on a single-core RT1170, Rev. 0, 02 February 2021.

Figure 3. Open the demo in IAR

3. Debug the default build target as shown in Figure 4.

Figure 4. Build target

4. Select Project > Edit Configuration.

Figure 5. Edit configurations

5. Add a build target.
a. Click the New button.
b. Name it as single_core.
c. Click OK.

6. Enter the project option and configure the following settings.

a. Add the macro ‘SINGLE_CORE_M7’ in the Preprocessor of C/C++ Compiler category.
b. Delete all the information in the linker input as shown in Figure 9

c. In the Debugger category, disable the multicore.
7. Debug and run the code.

8. On successful debug, the terminal window prompts the following message.

![Terminal output](image)

**Figure 11. Terminal output**

### 6.2 Run a demo using Keil® MDK/μVision

1. Open the `power_mode_switch` project by using Keil. The project location is:

```
<install_dir>\boards\evkmimxrt1170\demo_apps\power_mode_switch\bm\core0\mdk\power_mode_switch_bm_core0.uvmpw
```
2. Change the target to flexsip_nor_debug.

3. Open the options for the target.

4. In C/C++ section, add 'SINGLE_CORE_M7' to the Define field.
5. Open the options for the file fsl_incbin.S under utilities.

6. Deselect the **Include in Target Build** checkbox.
7. Debug and run the project.

![Terminal output](image)

Figure 18. Terminal output

### 6.3 Run a demo using Arm® GCC

This section shows how to run a demo in a single-core mode by using GCC IDE.

**NOTE**
Ensure that CMake and Segger J-Link are installed.

1. Open the `cmakelist.txt` in the project directory.
2. Delete or comments the following three parts of the code need.

a. The code that include core1.

```cpp
set(CMAKE_MODULE_PATH
    ${ProjDirPath}/../.../.../.../devices/MIMXRT1176/drivers
    ${ProjDirPath}/../.../.../.../devices/MIMXRT1176/utilities/incbin
    ${ProjDirPath}/../.../.../.../devices/MIMXRT1176
    ${ProjDirPath}/../.../.../.../CMSIS/Include
    ${ProjDirPath}/../.../.../.../components/uart)
```

Figure 20. Delete or comment line 54 through line 60

b. The path that include `incbin`.

```cpp
target_include_directories(
    ${NCKDS_SDK_PROJECT_NAME} PRIVATE
    ${ProjDirPath}/../.../common
)
```

Figure 21. Delete or comment line 64

c. The include for `incbin`.
3. Open the file flags.cmake.

4. Add macro ‘SINGLE_CORE_M7’ to the target build.
5. Save changes, double-click on `build_debug.bat` to build the target.

6. If successful, there are two new folders appear in the project. The folders are: CmakeFiles, and debug.

7. Open the J-Link GDB server application and connect to the device.
8. Open the GCC ARM Embedded tool chain command window.

9. To launch the window, from the Windows operating system select, **Start menu > GNU Tools ARM Embedded <version> > GCC Command Prompt**.

10. Run the `arm-none-eabi-gdb.exe <application_name>.elf`. The .elf file appears in the debug folder. For this example, it is `arm-none-eabi-gdb.exe power_mode_switch_bm_core0.elf`. 
11. Run the following commands:
   a. Target remote localhost: 2331
   b. Monitor reset
   c. Monitor halt
   d. Load

12. Run the command `monitor go` to start the demo and the result appears on the terminal window.
Figure 30. Terminal output

This is single core M7.
CPU wakeup source 0x0...

*******************************************************************************
Power Mode Switch Demo for iMXRT1176
*******************************************************************************

System previous setpoint is 0
System current setpoint is 0
M7 previous CPU mode is RUN
M7 current CPU mode is RUN
M7 CLK is 696 MHz

Please select the desired operation:
Press  A  to demonstrate typical set point transition.
Press  B  to demonstrate cpu mode switch in setpoint 0.
Waiting for select...
Information in this document is provided solely to enable system and software implementers to use NXP products. There
are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the
information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor
does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any
and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided
in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over
time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical
experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to
standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this
document, including without limitation specifications and product descriptions, at any time and without notice. This
document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities.
Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce
the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other
open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no
liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.
Customer shall select products with security features that best meet rules, regulations, and standards of the intended
application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all
legal, regulatory, and security related requirements concerning its products, regardless of any information or support that
may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com)
that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP,
HITAG, UCODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX,
MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTFLX, SMARTMX, STARPLUG,
TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AttiVec, CodeWarrior, ColdFire, ColdFire+, the Energy
Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ
Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Hybrid, Airfast, BeeKit, BeeStack, CoreNet,
Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are
trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7,
ArmTDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle,
Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2,
ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile are trademarks or registered trademarks of Arm Limited (or its
subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights,
designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The
Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and
service marks licensed by Power.org.

© NXP B.V. 2021. All rights reserved.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 02 February 2021
Document identifier: AN13116