AN14773

i.MX RT700 HiFi4 and HiFi1 Power Consumption Measurement and Data Sheet Power Numbers Replicate

Rev. 2.0 — 10 November 2025 Application note

Document information

Information	Content
Keywords	AN14773, i.MX RT700, HiFi 4, HiFi 1, power consumption, FFT, power performance
Abstract	This application note introduces how to reproduce the power consumption data of HiFi4 and HiFi1 in the data sheet for the MIMXRT700-EVK board.

1 Introduction

The i.MX RT700 includes both HiFi4 DSP and HiFi1 DSP:

- HiFi4 DSP, located next to the compute domain, functions as the DSP core for executing audio voice codecs, pre, and postprocessing modules. It also offloads machine learning functions for accelerated processing.
- HiFi1 DSP, located in the sense domain, primarily supports always-on applications, Bluetooth LE audio, and Sensor HiFi1.

This application note does not aim to introduce the status of each power domain in various power modes. Instead, it introduces how to reproduce the power consumption data of HiFi4 and HiFi1 in the data sheet for the MIMXRT700-EVK board.

By using the i.MX RT700 SDK, users quickly replicate the power numbers by following the steps mentioned in this application note. This approach helps users understand the power consumption performance of i.MX RT700 HiFi4 and HiFi1 DSPs.

2 How to replicate power numbers

To replicate the power consumption numbers using the i.MX RT700 SDK, follow the steps below:

1. Open the NXP SDK official website, mcuxpresso.nxp.com/en, download the latest i.MX RT700 SDK, and use the power-related examples to measure the power consumption data.
   Note: *As shown in Figure 1, this application note uses SDK version 25.06.00 as an example. If a newer version is available, use the latest SDK package.*

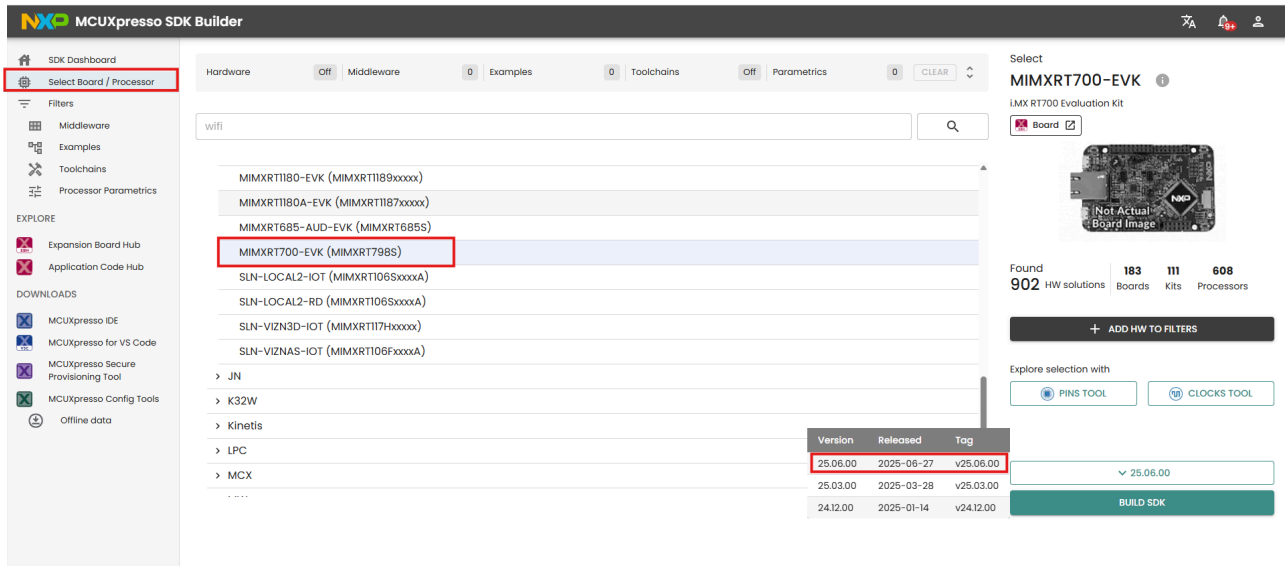

Figure 1. Downloading and selecting the i.MX RT700 SDK IN MCUXpresso SDK builder

2. As shown in Figure 2, locate the power mode demonstration projects in the SDK directory. The three relevant projects are:
   - power_mode_comp_only
   - power_mode_switch
   - power_mode_with_hifi
3. Understand the purpose of each project:
   - The **power_mode_comp_only** project targets users who use only CPU0. In this project, CPU1 enters Deep Sleep mode at the beginning.
   - The **power_mode_switch** project shows how CPU0 and CPU1 switch between different power modes.

- The **power_mode_with_hifi** project evaluates the power consumption of HiFi4 and HiFi1 while running fast Fourier transform (FFT). This application note describes the measurement steps in detail.
  Note: *Document AN14600 explains how to reproduce most low-power cases, excluding HiFi4 and HiFi1.*

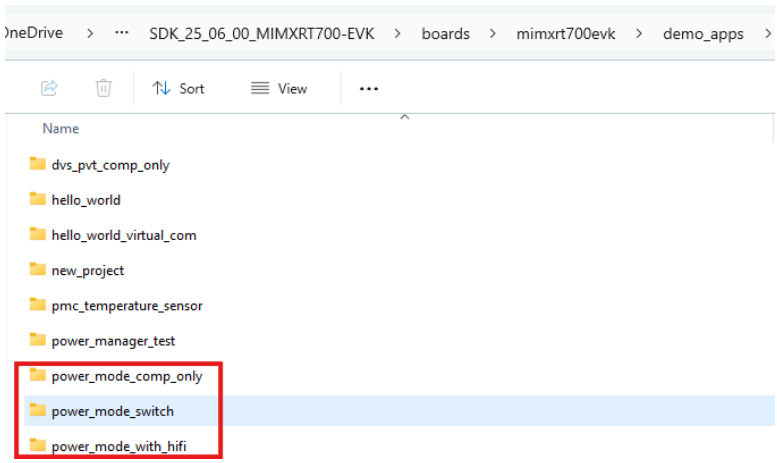

Figure 2. Power mode demonstration projects in the SDK directory

4. Refer to Table 1 that lists all the power cases from the data sheet and identifies the corresponding project in which each case is measured. Use IAR projects and the debug_target option to replicate the power numbers, as all power consumption data in this application note is based on IAR projects.

Table 1. i.MX RT700 power cases and corresponding projects

i.MX RT700 power cases in the data sheet			
Power demo example projects	**Case**	**HiFi4**	**HiFi1**
power_mode_ comp_only	CPU0 CM33 CoreMark, Sense Deep Sleep	HiFi4 DSP power down	HiFi1 DSP stalled
power_mode_ comp_only	CPU0 CM33 Sleep, Sense Deep Sleep	HiFi4 DSP power down	HiFi1 DSP stalled
power_mode_switch	Compute DSR, CPU1 CM33 CoreMark	HiFi4 DSP power down	HiFi1 DSP stalled
power_mode_switch	Compute DSR, CPU1 CM33 Sleep	HiFi4 DSP power down	HiFi1 DSP stalled
power_mode_with_ hifi	CPU0 CM33 Sleep, Sense Deep Sleep, HiFi4 DSP FFT, HiFi1 DSP stalled	HiFi4 Run FFT	HiFi1 DSP stalled
power_mode_with_ hifi	Compute DSR, CPU1 CM33 Sleep, HiFi4 DSP power down, HiFi1 DSP FFT	HiFi4 DSP power down	HiFi1 Run FFT
power_mode_ comp_only	Compute and Sense Dual Deep Sleep mode	HiFi4 DSP power down	HiFi1 DSP stalled
power_mode_ comp_only	Compute and Sense Dual Deep Sleep Async	HiFi4 DSP power down	HiFi1 DSP stalled
power_mode_switch	Full DSR mode (FDSR)	HiFi4 DSP power down	HiFi1 DSP stalled
power_mode_ comp_only	Deep Power Down mode (DPD)	HiFi4 DSP power down	HiFi1 DSP power down
power_mode_ comp_only	Full Deep Power Down mode (FDPD)	HiFi4 DSP power down	HiFi1 DSP power down

5. Before replicating the power consumption data, observe that the power numbers vary across different chips. The actual measurements depend on the specific chip in use. Refer to the data in the data sheet as standard.
   Note: *This application note provides measurement results for reference only.*

2.1 Case: CPU0 CM33 Sleep, Sense Deep Sleep, HiFi4 DSP FFT, HiFi1 DSP stalled

The following procedure outlines the step-by-step method to replicate the power consumption measurements for the i.MX RT700 and HiFi DSPs. It also describes how to validate these measurements using the provided SDK, hardware setup, and measurement tools.

Table 2. CPU0 CM33 Sleep, Sense Deep Sleep, HiFi4 FFT, HiFi1 stalled

Symbol	Description	Minimum	Type	Maximum	Unit	Condition
IVDD2	VDD2 supply current	-	87.52	-	mA	COMPUTE_MAIN_CLK = 1 MHz, DSP_CLK = 325 MHz, VDD2 = 1.1 V
IVDD2	VDD2 supply current	-	61.05	-	mA	COMPUTE_MAIN_CLK = 1 MHz, DSP_CLK = 250 MHz, VDD2 = 1.0 V
IVDD2	VDD2 supply current	-	41.87	-	mA	COMPUTE_MAIN_CLK = 1 MHz, DSP_CLK = 192 MHz, VDD2 = 0.9 V
IVDD2	VDD2 supply current	-	21.81	-	mA	COMPUTE_MAIN_CLK = 1 MHz, DSP_CLK = 110 MHz, VDD2 = 0.8 V
IVDD2	VDD2 supply current	-	8.27	-	mA	COMPUTE_MAIN_CLK = 1 MHz, DSP_CLK = 45 MHz, VDD2 = 0.7 V

1. Begin by selecting the two reference power consumption data points from Table 2, which shows IVDD2 values under different operating conditions:
   - DSP_CLK = 192 MHz, VDD2 = 0.9 V, IVDD2 = 41.87 mA
   - DSP_CLK = 325 MHz, VDD2 = 1.1 V, IVDD2 = 87.52 mA
2. Configure the hardware by installing JP1 and JP3 on the MIMXRT700 EVK board, and short JP2 between pins 2 and 3. Ensure that the i.MX RT700 is powered by the external power management integrated circuit (PMIC) PCA9422.
3. Open the Xplorer IDE and import the *naturedsp* demo example from the SDK package. The path to the example is: `boards\mimxrt700evk\dsp_examples\naturedsp\hifi4`. Replace main.c with main_dsp.c from the folder: `boards\mimxrt700evk\demo_apps\power_mode_with_hifi\cm33_core0\`, then select the 'Release' target.

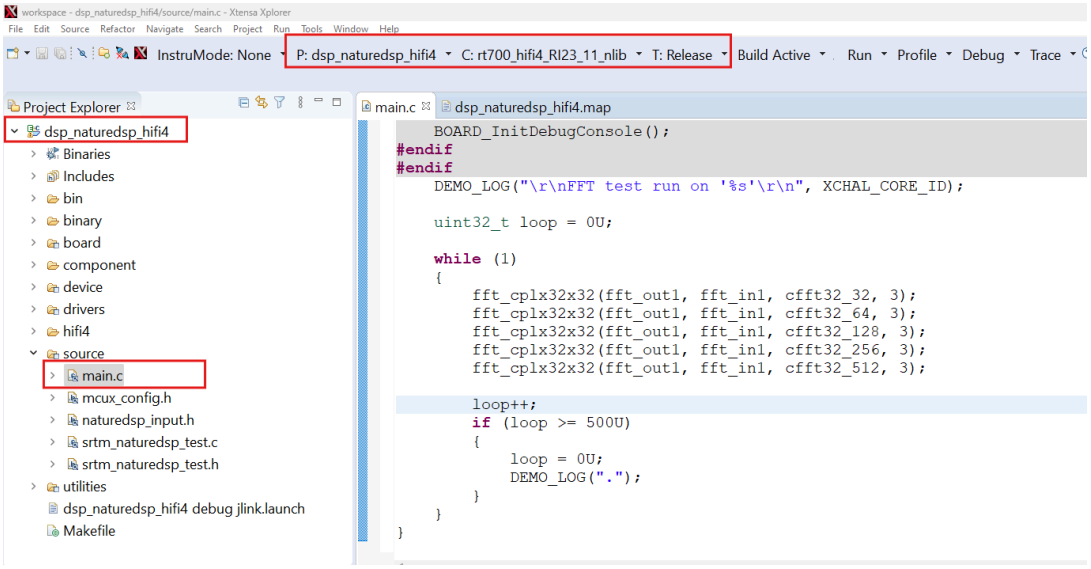

Figure 3. Path to the linker file in dsp_naturedsp_hifi4 project

Note:

- *Table 2 shows that the data and code reside in SRAM partition 14, so update the linker file of the `dsp_naturedsp_hifi4` project to match this configuration.*
- *Figure 4 shows the path of 'dsp_naturedsp_hifi4' linker file.*

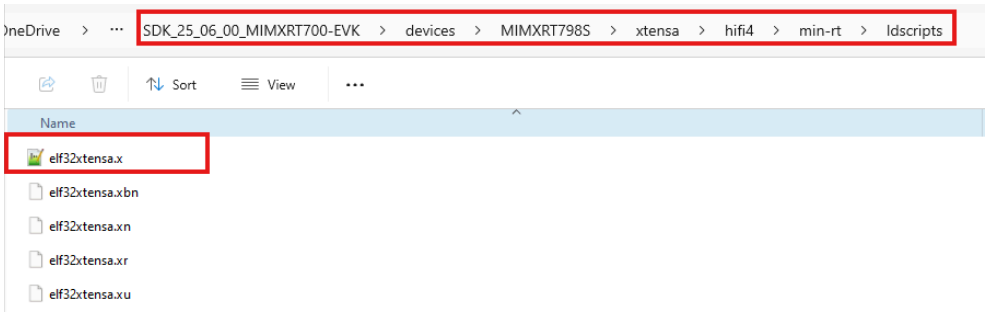

Figure 4. Modified linker file contents for SRAM mapping

Note:

- *To put both data and text into SRAM partition 14, refer to the attached file `elf32xtensa_sram_p14.x` and modify the contents of elf32xtensa.x.*
- *Figure 5 shows the difference between the default and modified versions.*

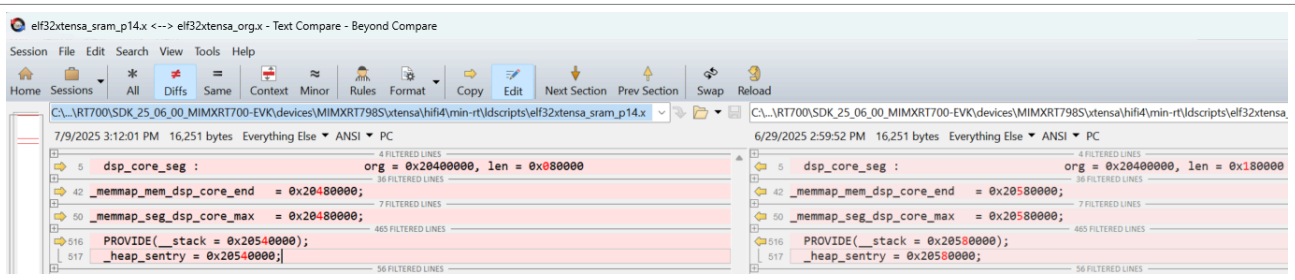

Figure 5. Comparison between default and modified linker files

4. After compiling with the provided linker file, four binaries are generated in the `dsp_naturedsp_hifi4` project folder, as shown in Figure 6. To ensure that both data and test sections are placed in SRAM partition 14, verify the memory allocation by checking the map file of the `dsp_naturedsp_hifi4` project in the Xplorer IDE.

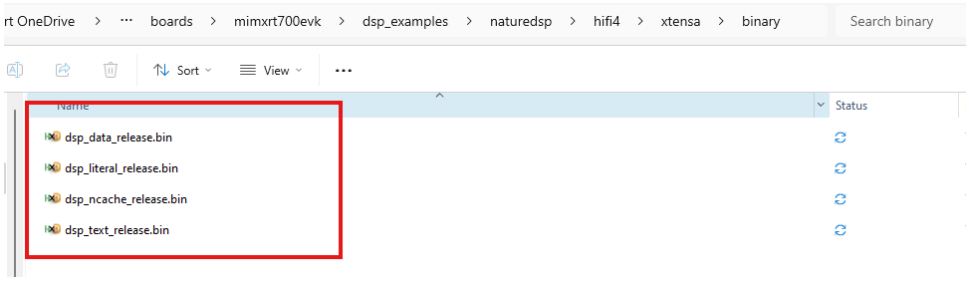

Figure 6. Binary output files after compilation

5. Copy the following DSP binary files into the `power_mode_with_hifi` project folder, as shown in Figure 7:
   - dsp_data_release.bin
   - dsp_literal_release.bin
   - dsp_ncache_release.bin
   - dsp_text_release.bin

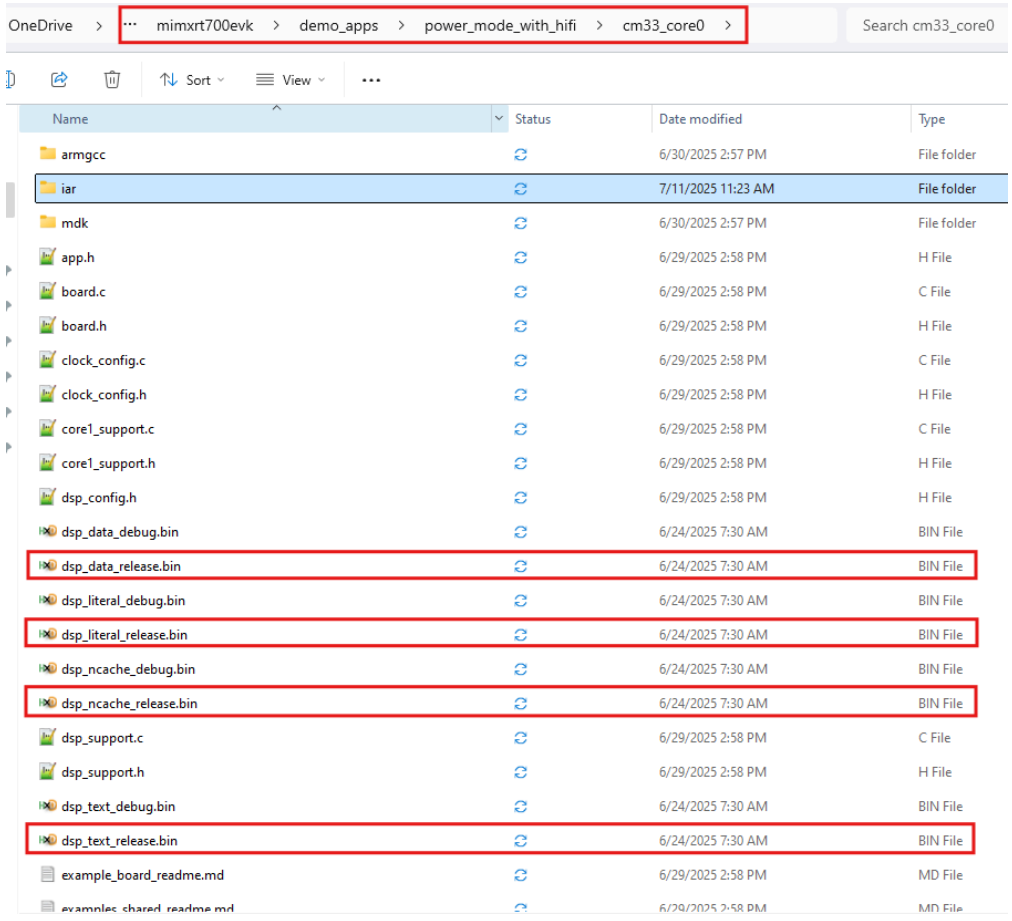

Figure 7. Copying DSP binary files

6. Open the `power_mode_with_hifi` project, then access the secondary project and first select the `debug_target`. Update the `DEMO_POWER_SUPPLY_OPTION` and `DEMO_POWER_ENABLE_DEBUG` settings as shown in Figure 8. After making these changes, compile the secondary project.
   Note: *Different SDK versions can have varying default configurations, but these configurations must be updated to match the same as in Figure 8.*

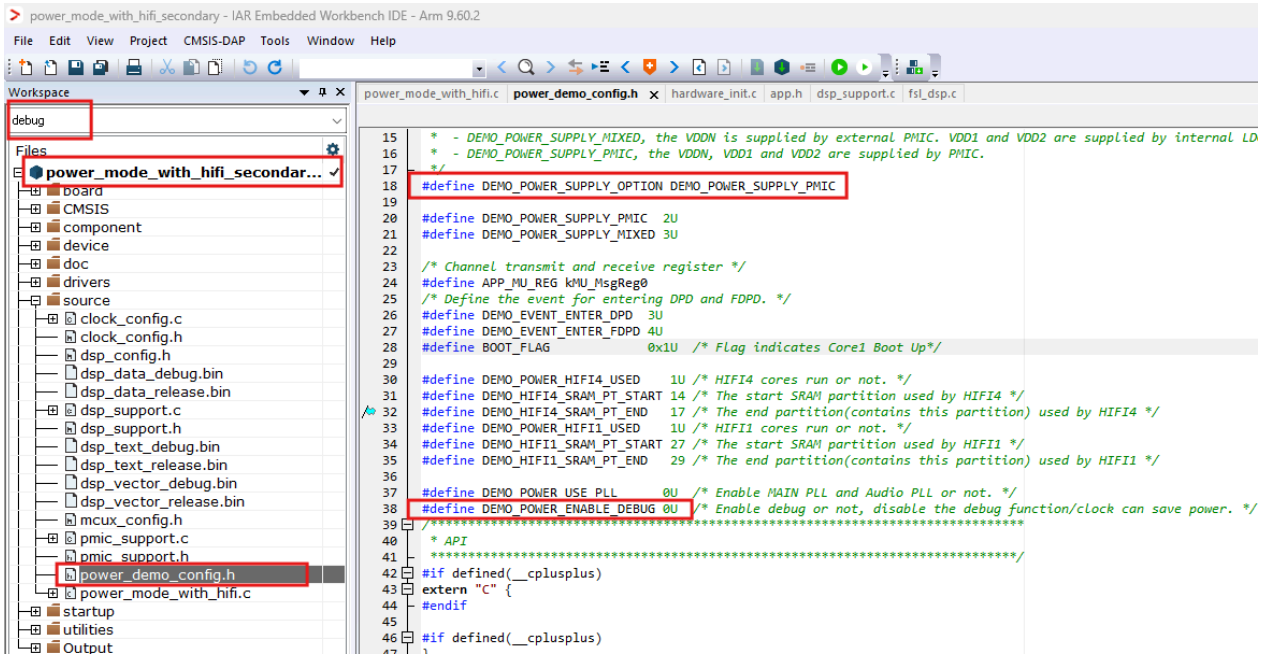

Figure 8. Setting PMIC and debug options in secondary project

7. Open the primary project and select the `debug_target`. Update the `DEMO_POWER_SUPPLY_OPTION`, `DEMO_POWER_ENABLE_DEBUG`, and `DEMO_HIFI4_SRAM_PT_END` settings as shown in Figure 9. Once the modifications are complete, compile the primary project.

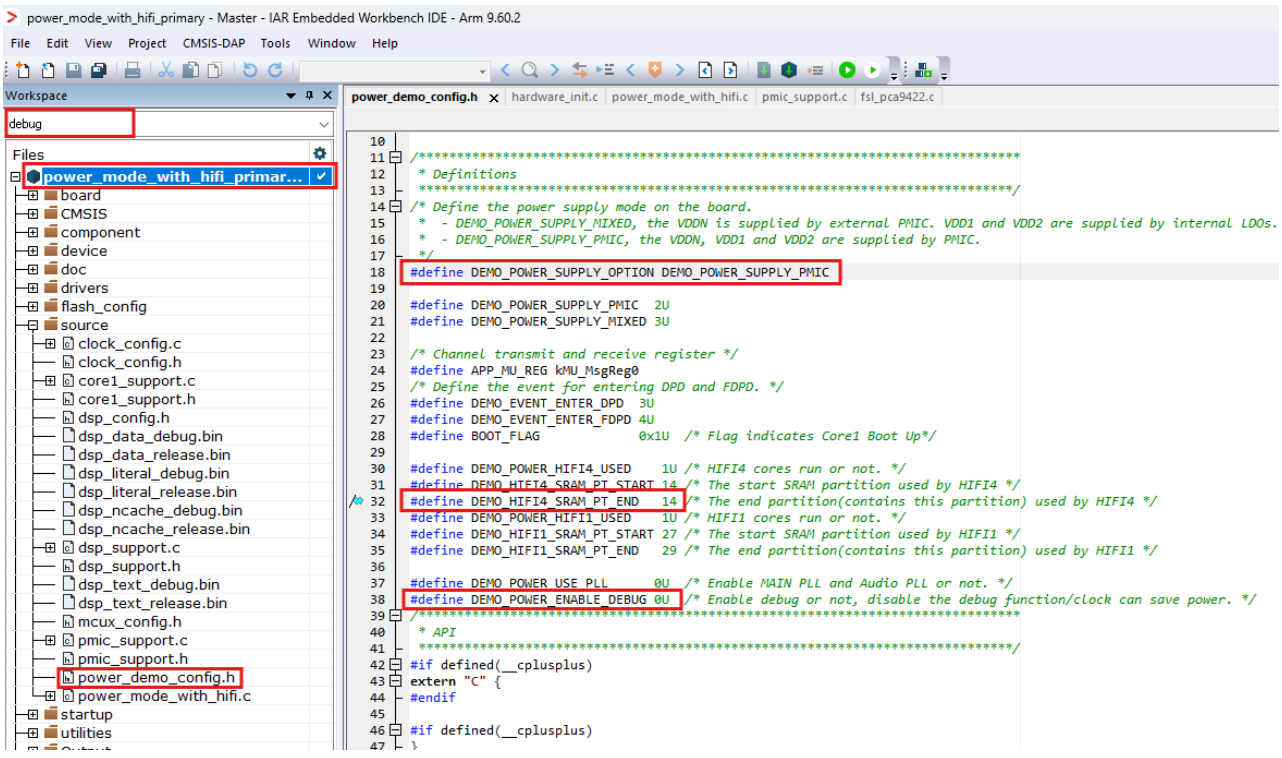

Figure 9. Modifying primary project for PMIC, debug, and SRAM settings

8. To generate the binary file `power_mode_with_hifi_primary.bin`, as shown in Figure 10, compile the primary project.

Note: *Ensure that the build completes successfully and the binary is located in the expected output directory.*

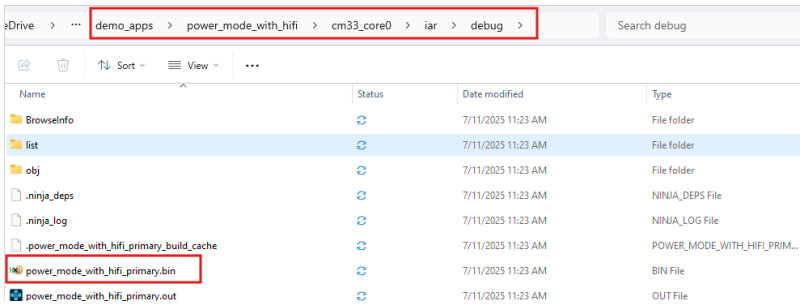

Figure 10. Generating application binary

9. Download the generated binary to the flash address 0x28000000. Figure 11 provides an example of using **JLINK** to download a binary to the address 0x28000000.
   Note: *JLINK is one of several tools that can be used for flashing binaries. Other supported methods include* `blhost`*, among others. Choose the method that best suits your development environment.*

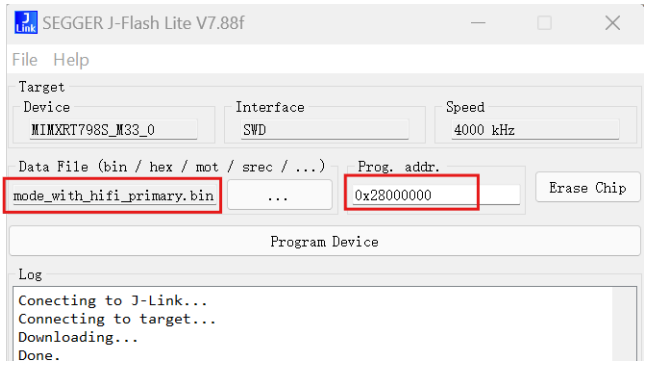

Figure 11. Flashing the binary image using JLINK

10. After the demo boots successfully, log from both CPU0 and CPU1 COM ports is displayed, as shown in Figure 12. The default setting is DSP_CLK = 192 MHz and VDD2 = 0.9 V.
    To initiate power-saving modes, input the characters '2S' or '2N' in the CPU1 COM port to set the Sense domain into Deep Sleep mode. Then, input the characters '1S' or '1T9' in the CPU0 COM port to transition CPU0 CM33 into Sleep mode. During this time, HiFi4 continues running an FFT operation.
    To observe the power consumption of VDD2, as illustrated in Figure 12, measure the JP1 current on the MIMXRT700 EVK board. For accurate power measurements, use tools such as Joulescope or any other reliable digital multimeter.
    Note: *The character inputs used to initiate power-saving modes have specific functions:*
    - *'2' selects Power Mode 2*
    - *'S' enables 'keypress to wake up' as the wake-up source*
    - *'N' indicates that no wake-up source is selected*

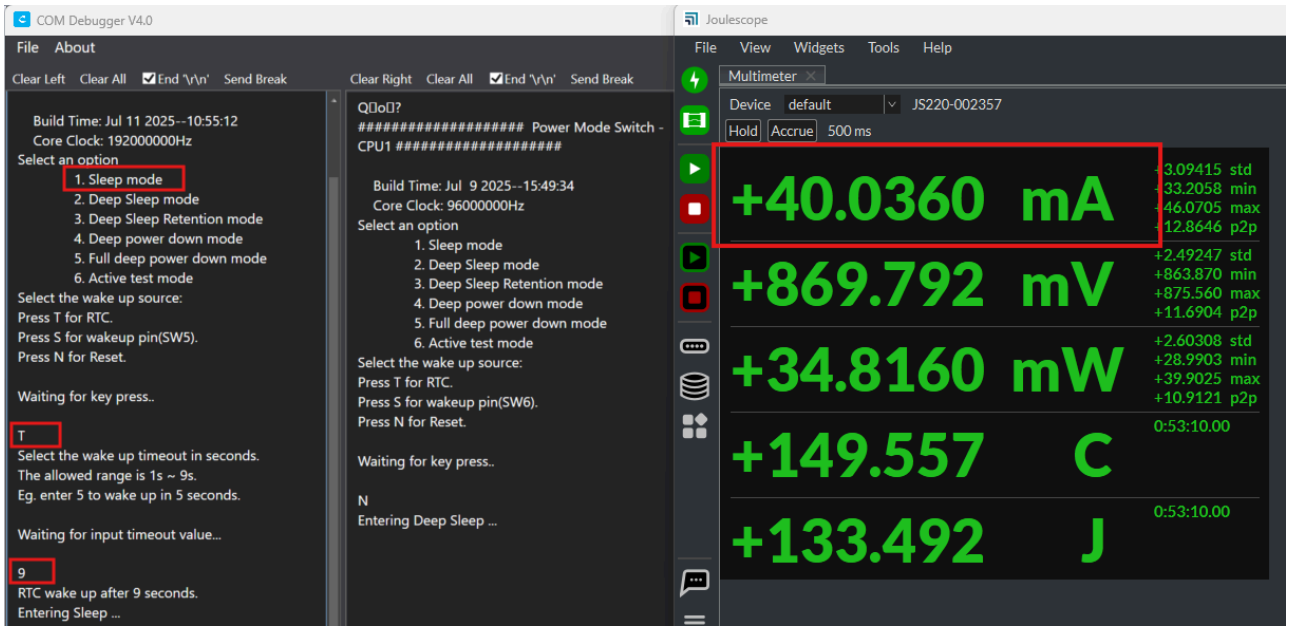

Figure 12. Demo output and power consumption observed on MIMXRT700 EVK

11. It is important to note that the internal impedance of the ammeter can introduce a voltage drop during current measurement. To obtain accurate current readings at the intended voltage point, voltage compensation is necessary. To ensure that the actual voltage at VDD2 remains close to 0.900 V, increase the VDD2 voltage based on the internal resistance of the ammeter. The resulting power consumption under these conditions is shown in Figure 13.

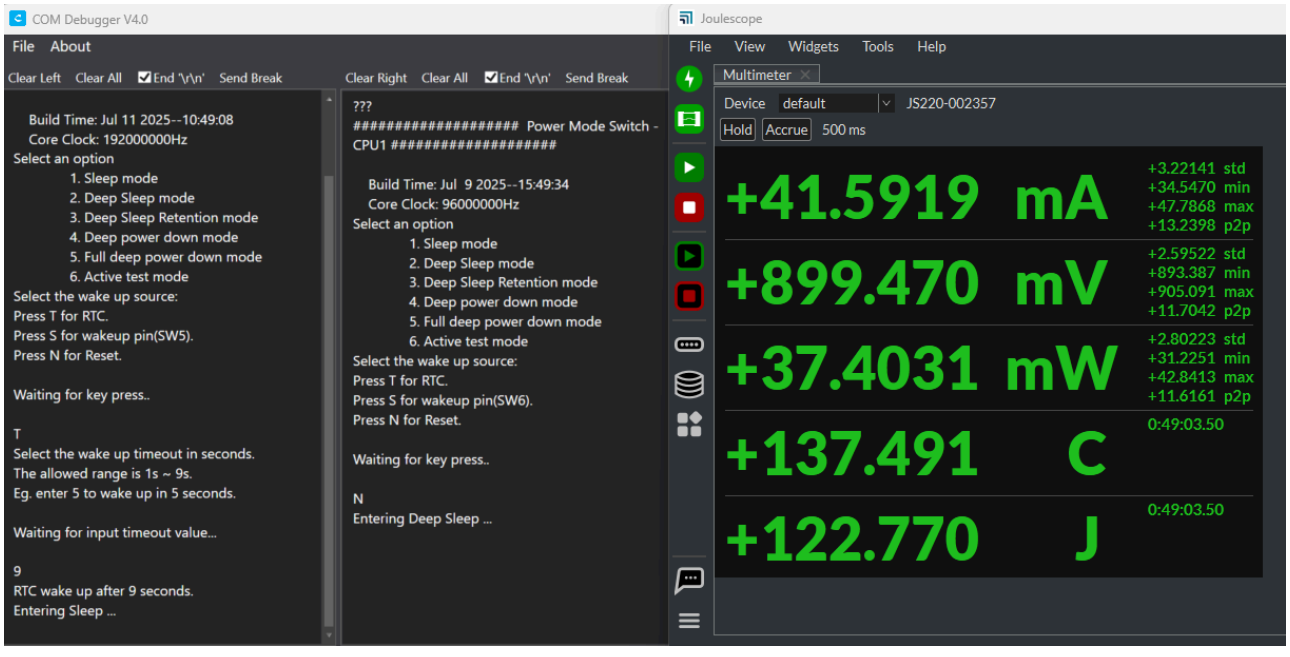

Figure 13. Power consumption of VDD2 after voltage compensation

12. Continue measuring power consumption across different voltage and frequency settings. Adjust the `DSP_CLK` frequency from 192 MHz to 325 MHz, and modify the VDD2 voltage from 0.9 V to 1.1 V. To implement these changes in the CPU0 project, add the following code snippet to the `BOARD_PowerConfigAfterCPU1Booted(void)` function to implement dynamic adjustment of both frequency and voltage, as shown in Figure 14.

```
const clock_fro_config_t g_fro0Config_BOARD_BootClockHSRUN =
    {
        .targetFreq = 325000000,                /* FRO0 TUNER output clock frequency: 325000000Hz */
        .range = 100,                           /* FRO0 range value: ± 100counts */
        .trim1DelayUs = 5,                      /* FRO0 Trim1 delay: 5us */
        .trim2DelayUs = 150,                    /* FRO0 Trim2 delay: 150us */
        .refDiv = 1,                            /* The FRO reference divider is 1 */
        .enableInt = 0,                         /* The FRO interrupts are disabled */
        .coarseTrimEn = false,                  /* The coarse value autotrimming is disabled */
    };

void BOARD_PowerConfigAfterCPU1Booted(void)
{
    uint32_t miniVolts[4] = {0U};
    miniVolts[0] = 630000U; /* For DeepSleep. */

    POWER_ConfigRegulatorSetpointsForFreq(kRegulator_Vdd2LDO, freqs, miniVolts, 0U, 4U);

    /*!< Set SystemCoreClock variable. */
    SystemCoreClock = 325000000U;

    g_runVolt = POWER_CalcVoltLevel(kRegulator_Vdd2LDO, SystemCoreClock, 0U); /* Calculate the voltage per frequency. */

#if defined(DEMO_POWER_SUPPLY_OPTION) && (DEMO_POWER_SUPPLY_OPTION == DEMO_POWER_SUPPLY_MIXED)
    /* VDDN use external PMIC supply, VDD1&VDD2 use internal LDO. */
    POWER_SetVddnSupplySrc(kVddSrc_PMIC);
    POWER_SetVdd1SupplySrc(kVddSrc_PMC);
    POWER_SetVdd2SupplySrc(kVddSrc_PMC);

    POWER_SelectRunSetpoint(kRegulator_Vdd2LDO, 2U);
#if defined(DEMO_POWER_HIFI4_USED) && (DEMO_POWER_HIFI4_USED != 0U)
    POWER_SelectSleepSetpoint(kRegulator_Vdd2LDO, 2U); /* HIFI4 can keep running in deep sleep mode. */
#else
    POWER_SelectSleepSetpoint(kRegulator_Vdd2LDO, 0U);
#endif                                  /* DEMO_POWER_HIFI4_USED */
    POWER_SelectRunSetpoint(kRegulator_Vdd1LDO, 0U);
    POWER_SelectSleepSetpoint(kRegulator_Vdd1LDO, 0U);
    POWER_ApplyPD();
#elif defined(DEMO_POWER_SUPPLY_OPTION) && (DEMO_POWER_SUPPLY_OPTION == DEMO_POWER_SUPPLY_PMIC)
    POWER_SelectRunSetpoint(kRegulator_Vdd2LDO, 0U);
    POWER_SelectSleepSetpoint(kRegulator_Vdd2LDO, 0U);
    POWER_ApplyPD();

    BOARD_SetPmicVdd2Voltage(g_runVolt);

    DEMO_DeinitDebugConsole();
    POWER_DisablePD(kPDRUNCFG_PD_FRO1);
    CLOCK_EnableFroClkOutput(FRO1, kCLOCK_FroDiv1OutEn);
    CLOCK_AttachClk(kFRO1_DIV1_to_COMPUTE_BASE);
    CLOCK_AttachClk(kCOMPUTE_BASE_to_COMPUTE_MAIN);
    BOARD_BootClockRUN_InitClockModule(kClockModule_XTAL_OSC); /* Enable SOSC used for FRO trim. */
    CLOCK_EnableFroClkFreqCloseLoop(FRO0, &g_fro0Config_BOARD_BootClockHSRUN, kCLOCK_FroDiv1OutEn | kCLOCK_FroDiv3OutEn | kCLOCK_FroDiv6OutEn);
    POWER_EnablePD(kPDRUNCFG_PD_SYSXTAL);
    CLOCK_AttachClk(kFRO0_DIV1_to_COMPUTE_MAIN);
    CLOCK_AttachClk(kLPOSC_to_COMPUTE_BASE);
    CLOCK_DisableFro(FRO1);
    DEMO_InitDebugConsole();
#endif
```

Figure 14. Power measurement results before voltage compensation at 1.1 V

13. Repeat steps List item, List item, and List item to continue measuring power consumption. The updated power reading for **VDD2** under the new voltage and frequency settings is shown in Figure 15.

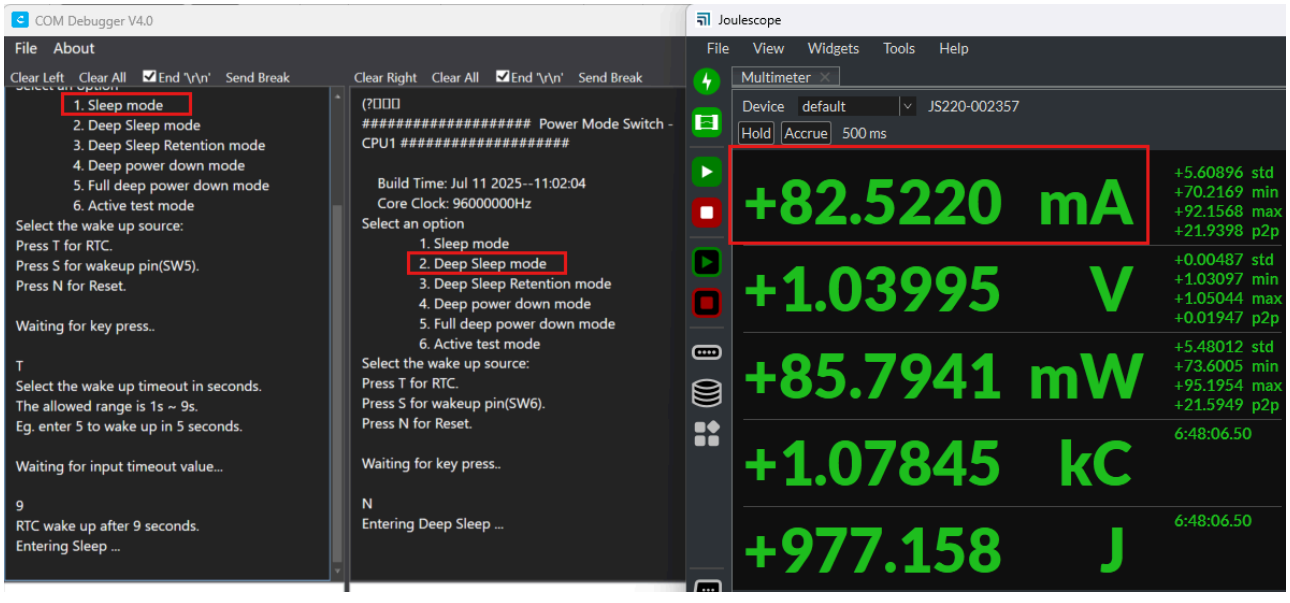

Figure 15. Final power measurement after voltage compensation at 1.1 V

14. Repeat step List item to perform voltage compensation. The updated power consumption of **VDD2** appears as shown in Figure 16. Power consumption data obtained from actual measurements can differ from the typical values listed in the data sheet. This difference primarily results from factors, such as measurement

accuracy and ambient temperature. Also, for DSP_CLK frequencies of 110 MHz and 45 MHz, it is necessary to change the DSP clock divider to ensure that the FRO0 frequency remains above 150 MHz.

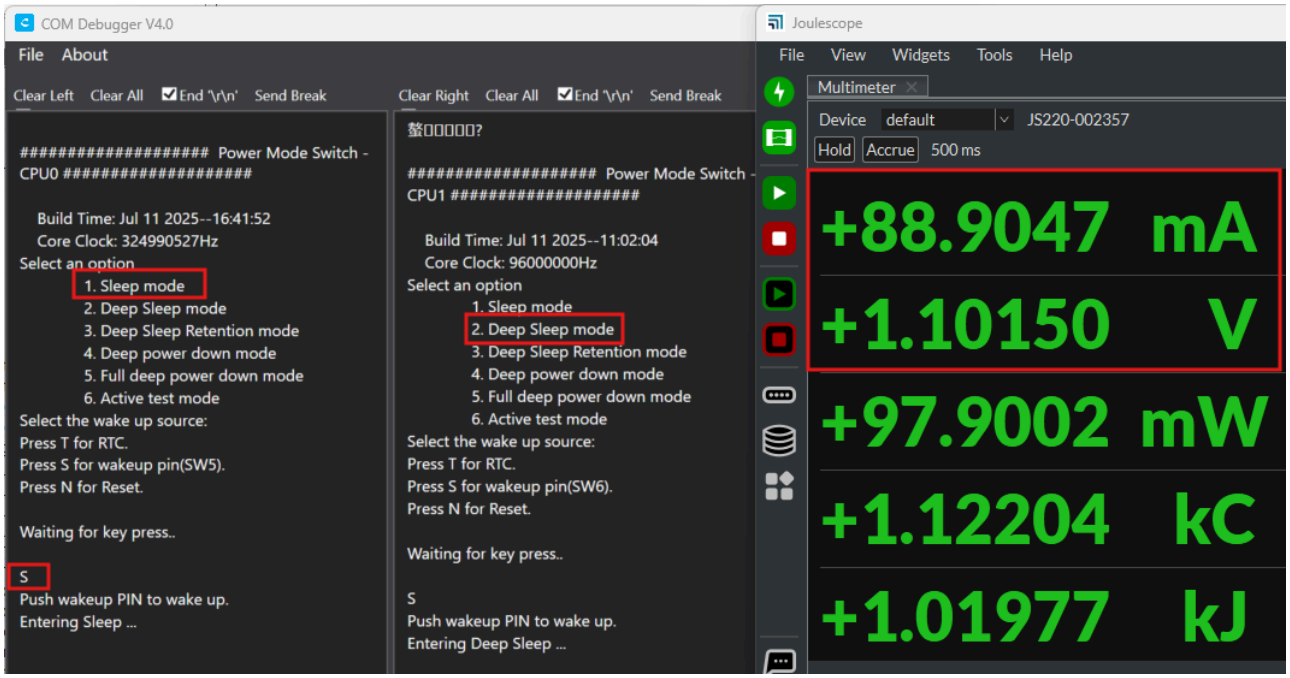

Figure 16. Final measurements after compensation

2.2 Case: Compute DSR, CPU1 CM33 Sleep, HiFi4 DSP power down, and HiFi1 DSP FFT

The following procedure outlines the step-by-step method to replicate the power consumption measurements and configurations for the Compute DSR, CPU1 CM33 Sleep, HiFi4 DSP power down, and HiFi1 DSP FFT use case on the MIMXRT700 EVK board.

Table 3. Compute DSR, CPU1 CM33 Sleep, HiFi4 DSP power down, and HiFi1 DSP FFT

Symbol	Description	Minimum	Type	Maximum	Unit	Condition
IVDD1	VDD1 supply current	-	32.03	-	mA	SENSE_MAIN_CLK = 1 MHz, SENSE_DSP_CLK = 250 MHz, VDD1 = 1.1 V
IVDD1	VDD1 supply current	-	23.46	-	mA	SENSE_MAIN_CLK = 1 MHz, SENSE_DSP_CLK = 205 MHz, VDD1 = 1.0 V
IVDD1	VDD1 supply current	-	16.29	-	mA	SENSE_MAIN_CLK = 1 MHz, SENSE_DSP_CLK = 160 MHz, VDD1 = 0.9 V
IVDD1	VDD1 supply current	-	9.26	-	mA	SENSE_MAIN_CLK = 1 MHz, SENSE_DSP_CLK = 100 MHz, VDD1 = 0.8 V
IVDD1	VDD1 supply current	-	3.87	-	mA	SENSE_MAIN_CLK = 1 MHz, SENSE_DSP_CLK = 45 MHz, VDD1 = 0.7 V

1. Begin by selecting the following power consumption data as an example:
   - SENSE_DSP_CLK = 160 MHz, VDD1 = 0.9 V, IVDD1 =16.29 mA

2. Configure the hardware by installing JP1 and JP3 on the MIMXRT700 EVK board, and short JP2 between pins 2 and 3. Ensure that the i.MX RT700 is powered by the external PMIC PCA9422.
3. Open the Xplorer IDE and import the *naturedsp* demo example from the SDK package. The path to the example is: `boards\mimxrt700evk\dsp_examples\naturedsp\hifi4`. Replace main.c with main_dsp.c from the folder: `boards\mimxrt700evk\demo_apps\power_mode_with_hifi\cm33_core0\`, then select the 'Release' target.

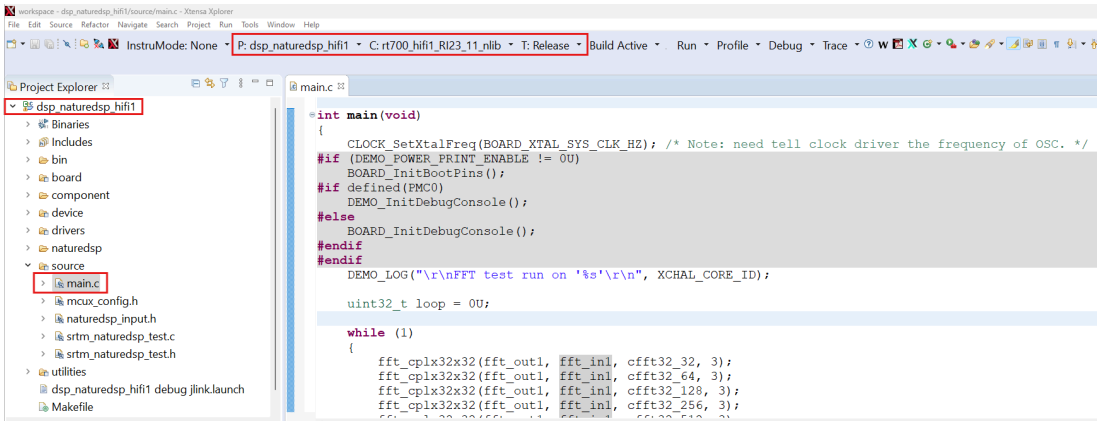

Figure 17. Path to the linker file in dsp_naturedsp_hifi4 project

Note:

- Table 3 shows that the data and code reside in SRAM partition 27, so update the linker file of the `dsp_naturedsp_hifi1` project to match this configuration.
- Figure 18 displays the path to the `dsp_naturedsp_hifi1` linker file.

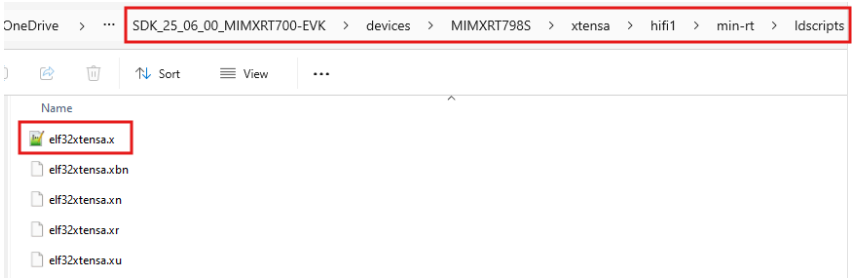

Figure 18. Modified linker file contents for SRAM mapping

Note:

- *To put both data and text into SRAM partition 27, refer to the attached file `elf32xtensa_sram_p27.x` and modify the contents of elf32xtensa.x.*
- *Figure 19 shows the difference between the default and modified versions.*

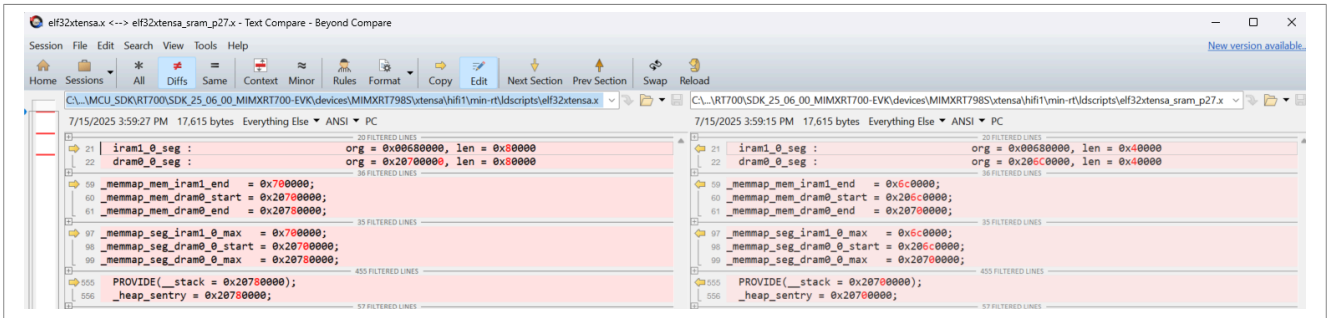

Figure 19. Comparison between default and modified linker files

4. After compiling with the provided linker file, three binaries are generated in the `dsp_naturedsp_hifi1` project folder, as shown in Figure 20. To ensure that both data and test sections are placed in SRAM

partition 27, verify the memory allocation by checking the map file of the `dsp_naturedsp_hifi1` project in the Xplorer IDE.

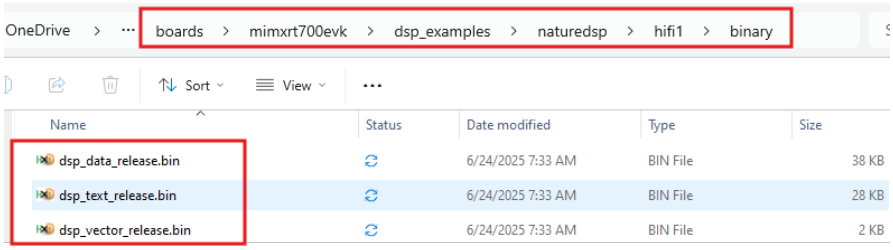

Figure 20. Binary output files after compilation

5. Copy the following DSP binary files into the `power_mode_with_hifi` project folder, as shown in Figure 21:
   - dsp_data_release.bin
   - dsp_text_release.bin
   - dsp_vector_release.bin

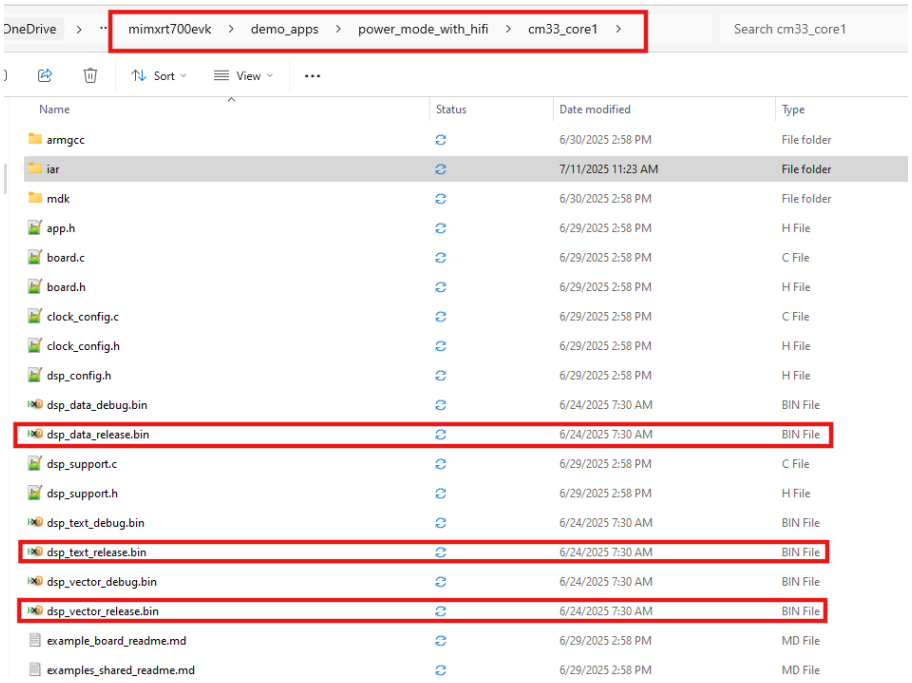

Figure 21. Copying DSP binary files

6. Open the `power_mode_with_hifi` project, then access the secondary project and first select the `debug_target`. Update the `DEMO_POWER_SUPPLY_OPTION` and `DEMO_POWER_ENABLE_DEBUG` settings as shown in Figure 22. After making these changes, compile the secondary project.
   Note: *Different SDK versions can have varying default configurations, but these configurations must be updated to match the same as in Figure 22.*

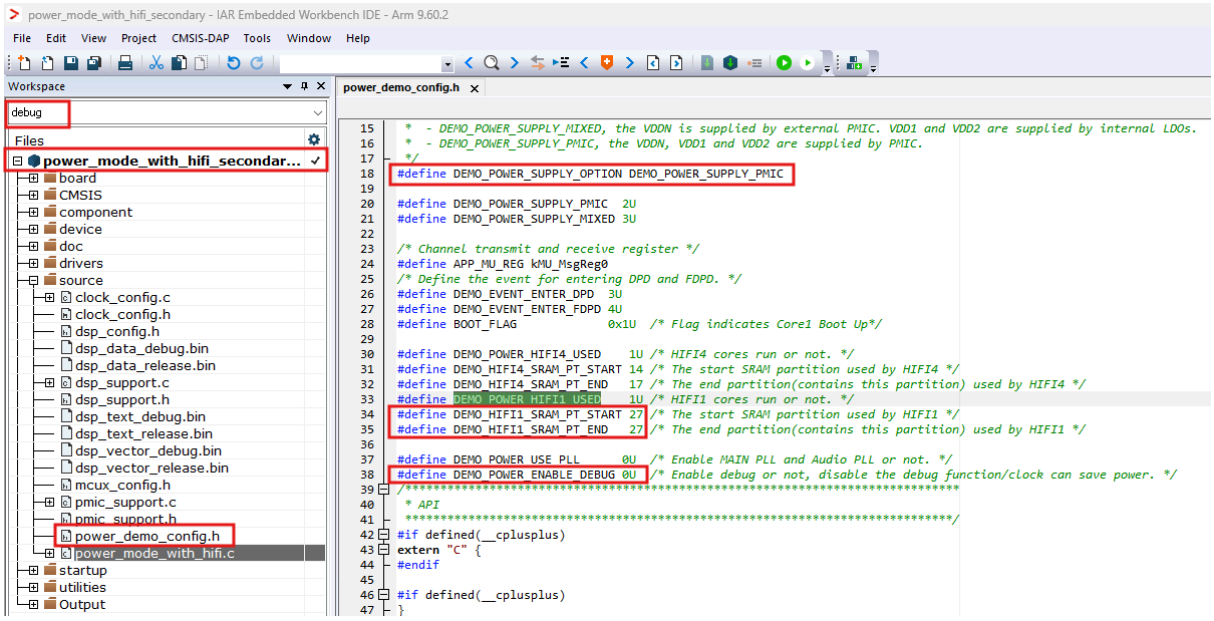

Figure 22. Setting PMIC and debug options in secondary project

7. Modify the SENSE_DSP_CLK frequency from 96 MHz to 160 MHz, as shown in Figure 23. Also, for SENSE_DSP_CLK frequencies of 100 MHz and 45 MHz, it is necessary to change the DSP clock divider to ensure that the FRO2 above 150 MHz.

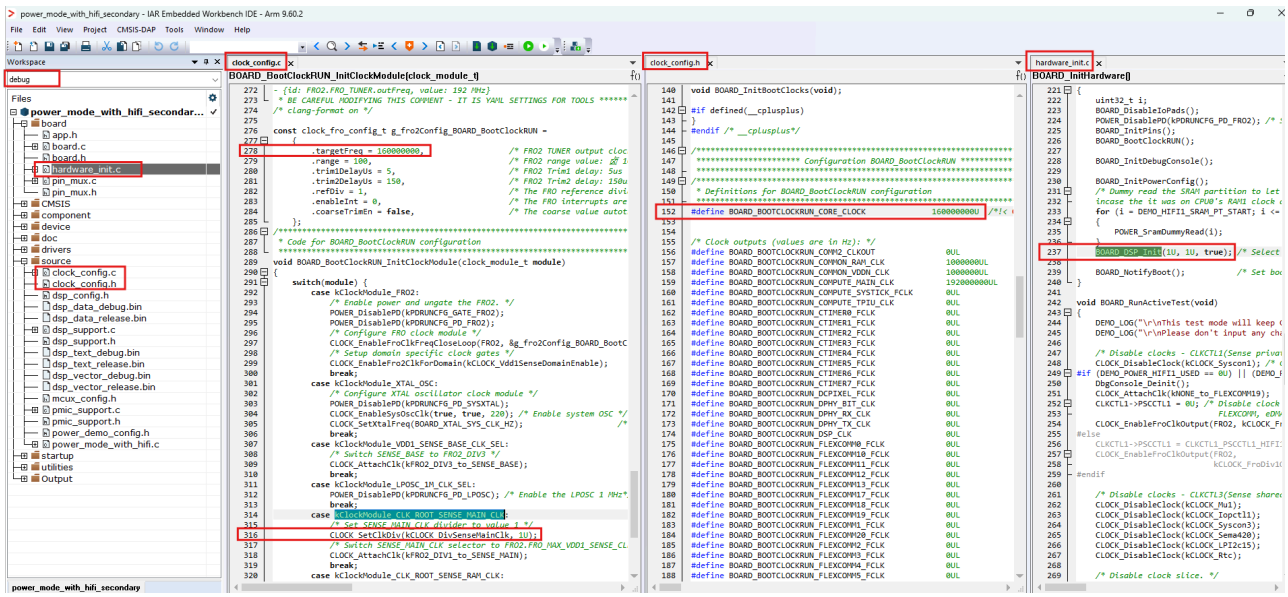

Figure 23. Changing SENSE_DSP_CLK frequency from 96 MHz to 160 MHz

8. Open the primary project and select the `debug_target`. Update the `DEMO_POWER_SUPPLY_OPTION`, `DEMO_POWER_ENABLE_DEBUG`, and `DEMO_HIFI4_SRAM_PT_END` settings as shown in Figure 24. Once the modifications are complete, compile the primary project.

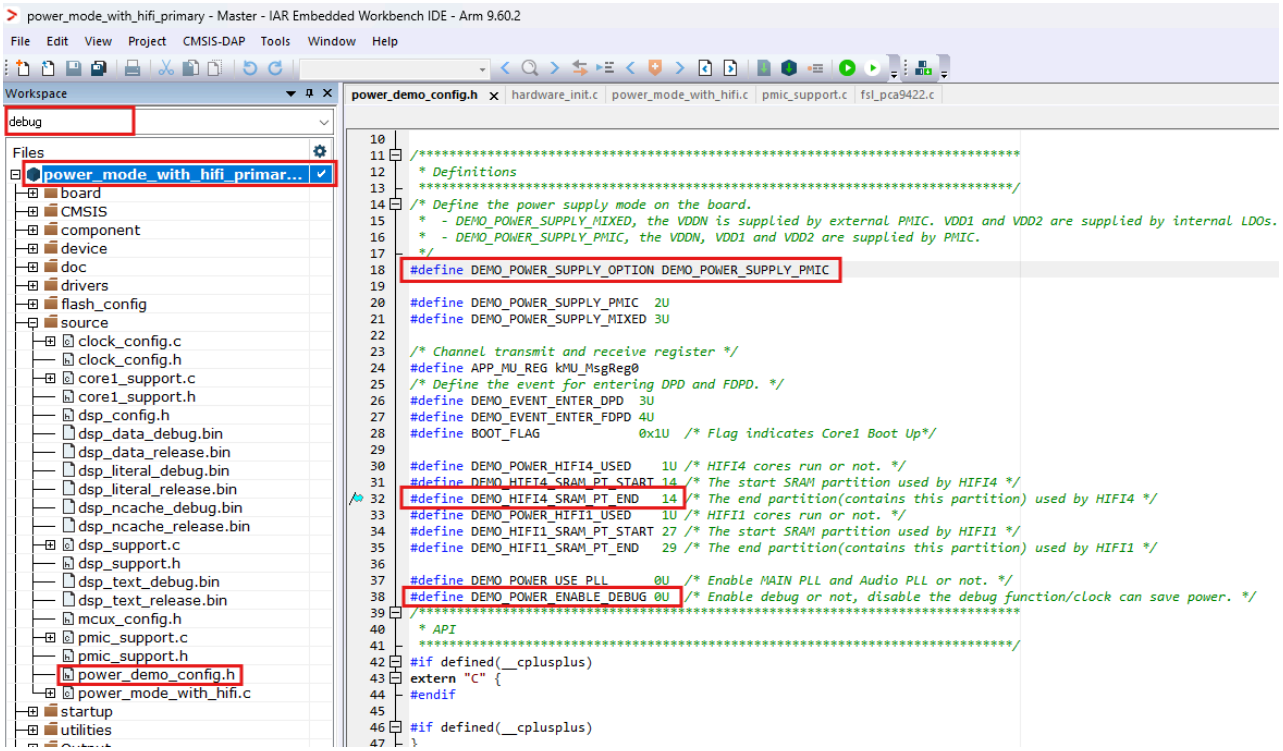

Figure 24. Modifying primary project for PMIC, debug, and SRAM settings

9. To generate the binary file `power_mode_with_hifi_primary.bin`, as shown in Figure 25, compile the primary project.
   Note: *Ensure that the build completes successfully and the binary is located in the expected output directory.*

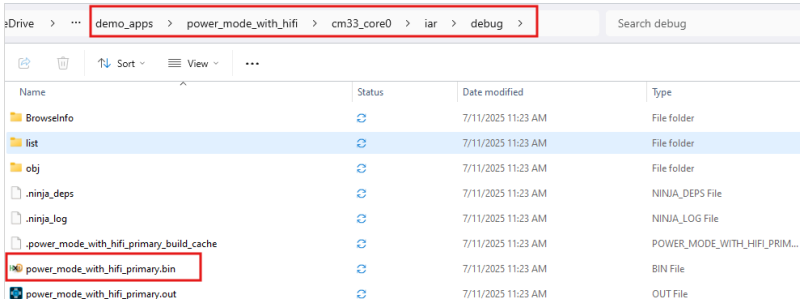

Figure 25. Generating application binary

10. Download the generated binary to the flash address 0x28000000. Figure 26 provides an example of using **JLINK** to download a binary to the address 0x28000000.
    Note: *JLINK is one of several tools that can be used for flashing binaries. Other supported methods include `blhost`, among others. Choose the method that best suits your development environment.*

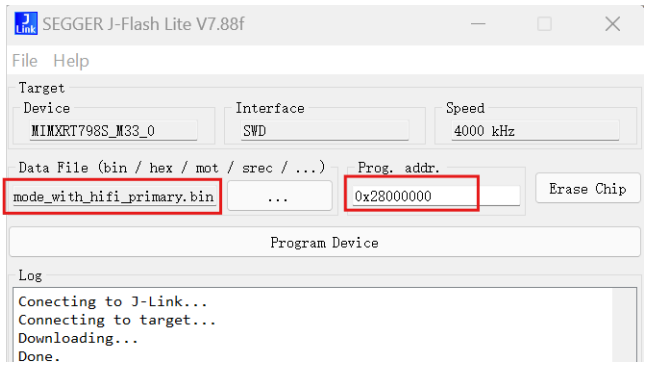

Figure 26. Flashing the binary image using JLINK

11. After the demo boots successfully, the log from both CPU0 and CPU1 COM ports appears, as shown in Figure 27. The default configuration sets SENSE_DSP_CLK to 160 MHz and VDD1 to 0.9 V.
    To initiate power-saving modes, input the characters 3 N in the CPU0 COM port to set the Compute domain into DSR mode. Then, input 1S or 1T9 in the CPU1 COM port to transition CPU1 CM33 into Sleep mode, while HiFi1 continues running an FFT operation.
    To observe the power consumption of VDD1, as illustrated in Figure 27, measure the JP3 current on the MIMXRT700 EVK board. For accurate power measurements, use tools such as Joulescope or any other reliable digital multimeter.

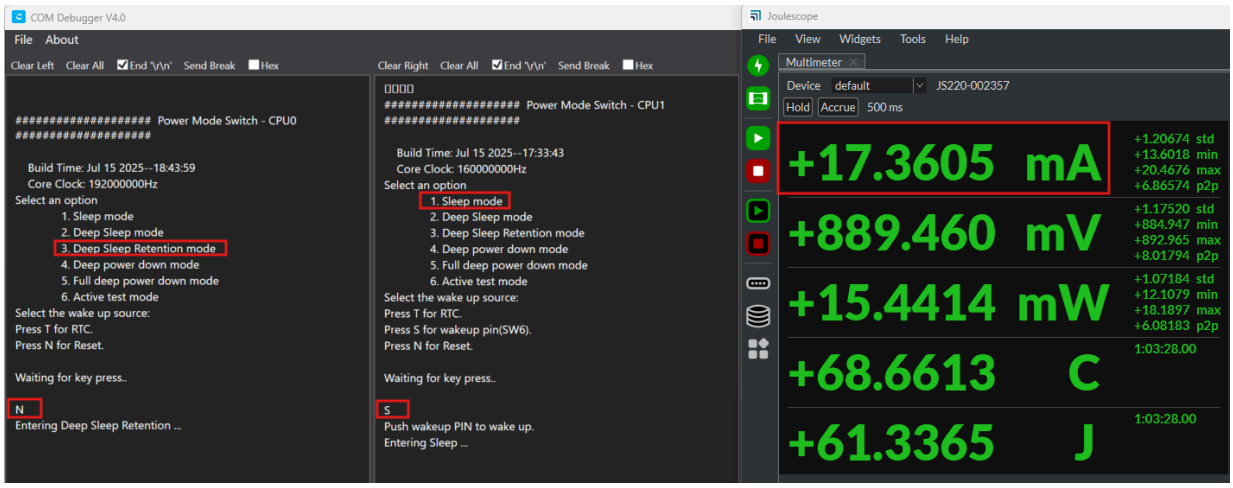

Figure 27. Demo output and power consumption observed on MIMXRT700 EVK

3 Acronyms and abbreviations

Table 4. Acronyms

Acronym	Expanded Form
CM33	Cortex-M33
DPD	Deep Power Down
DSP	digital signal processor
DSR	Deep Sleep Retention
FFT	fast Fourier transform
FDPD	Full Deep Power Down
FDSR	Full Deep Sleep Retention

Table 4. Acronyms...*continued*

Acronym	Expanded Form
FRO	free running oscillator
JP	jumper pin
PMIC	power management-integrated circuit
VDD	voltage drain drain

4 References

Refer to the below URLs for more documents that provide other information on the i.MX RT700 devices:

- *i.MX RT700 Crossover Microcontroller Data Sheet* (document IMXRT700EC)
- See Design Resources tab for the i.MX RT700 board design files

5 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2025 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6 Revision history

Table 5 summarizes the revisions to this document.

Table 5. Revision history

Document ID	Release date	Description
AN14773 v.2.0	10 November 2025	Initial public release
AN14773 v.1.0	28 August 2025	Initial internal release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. 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 B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

i.MX RT700 HiFi4 and HiFi1 Power Consumption Measurement and Data Sheet Power Numbers Replicate

Contents

1 Introduction 2
2 How to replicate power numbers 2
2.1 Case: CPU0 CM33 Sleep, Sense Deep Sleep, HiFi4 DSP FFT, HiFi1 DSP stalled 4
2.2 Case: Compute DSR, CPU1 CM33 Sleep, HiFi4 DSP power down, and HiFi1 DSP FFT 11
3 Acronyms and abbreviations 16
4 References 17
5 Note about the source code in the document 17
6 Revision history 17
Legal information 18