AN12844

Power consumption and measurement of i.MX RT6xx DSP

Rev. 0 — May, 2020

Application Note

Contents

1 Introduction

This document focuses on i.MX RT6xx DSP current consumption under low power mode. It describes the functions that help a user to manage power usage and also includes measurement of current based on MIMXRT6xx EVK board.

2 i.MX RT6xx overview

The i.MX RT6xx is a dual-core microcontroller for embedded applications, featuring an Arm Cortex-M33 CPU combined with a Cadence Xtensa HiFi4 advanced Audio Digital Signal Processor CPU. It offers a rich set of peripherals and very low power consumption.

The Arm Cortex-M33 is a next generation core based on the Armv8-M architecture that offers system enhancements. The Cadence Xtensa HiFi4 Audio DSP engine is a highly optimized audio processor designed for efficient execution of audio and voice codecs and pre- and post-processing modules.

The i.MX RT6xx is designed to allow the Cortex-M33 to operate at frequencies of up to 300 MHz and the HiFi4 DSP to operate at frequencies of up to 600 MHz.

i.MX RT6xx has rich features such as clock generation unit and power control.

Clock generation unit:

- Crystal oscillator with an operating range of 1 MHz to 25 MHz.
- Internal 48 MHz or 60 MHz IRC oscillator. Trimmed to +/- 1% accuracy.
- Internal 16 MHz IRC oscillator. Trimmed to +/- 3% accuracy.
- Internal 1 MHz low-power oscillator with 10% accuracy. Serves as the watchdog oscillator and clock for the OS Event Timer and the Systick. Also available as the system clock.
- 32 kHz real-time clock (RTC) oscillator that can optionally be used as a system clock.
- Main System PLL.
- Audio PLL for the audio subsystem.
- 480 MHz USB PLL (internal to the USB PHY).
- Clock output function with divider that can reflect any of the internal clock sources.

Power control:

- Main power supply is 1.8 V +/- 5%.
- Analog supply is 1.71 V - 3.6 V.
- Triple VDDIO supply (can be shared or independent) is 1.71 V - 3.6 V.
- USB supply is 3.0 V - 3.6 V.
- Reduced power modes.
- RBB/FBB to provide additional control over power/performance trade-offs.

- Power-On Reset (POR)

3 System configuration

The i.MX RT6xx includes various power switches and clock switches to allow fine tuning power usage to match requirements at different performance levels and reduced power modes.

3.1 Clock source and system clock

The system control block facilitates the clock generation. Many clocking variations are possible. The Main PLL can be configured to use a number of clock inputs and produce output clocks up to the maximum chip frequency, and can be used to run most on-chip functions.

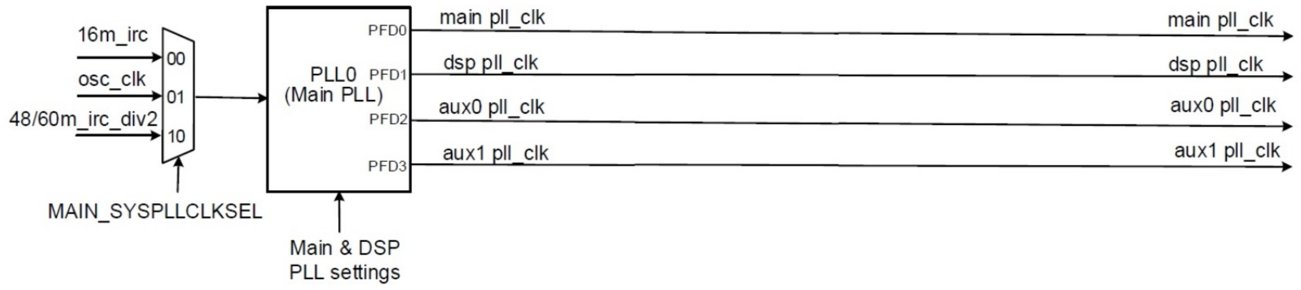


Figure 1. Main PLL

Figure 1 shows that the clock sources of the Main PLL can be 16m_irc, clk_in, and 48/60m_irc_div2. Also, the PFD settings can be used to alter the PLL VCO frequency before it is output from the PLL. Each PFD output may have a different setting. The PFD output frequency is given by:

PFD Output = 18/N x FVCO

where N = 12 to 35

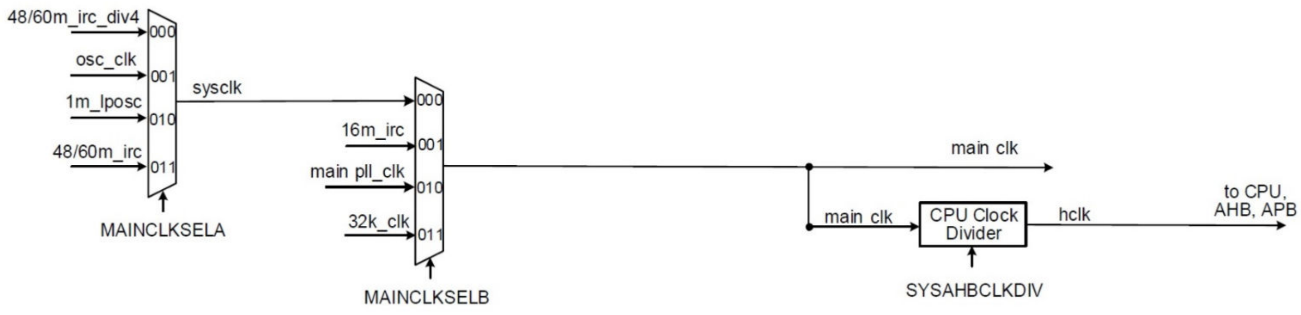


Figure 2. Main clock

Figure 2 shows that 48/60m_irc_div4, clk_in, 1m_lposc, and 48/60m_irc are Main clock select A sources, and 16m_irc, main_pll_clk (from Figure 1), and 32k_clk are Main clock select B sources. After main_clk is decided, the setting of CPU Clock Divider controls the divider for CPU clock. Similar to Figure 2, in Figure 3, DSP clock has its own clock source and DSP Clock Divider register divides down final clock to DSP CPU.

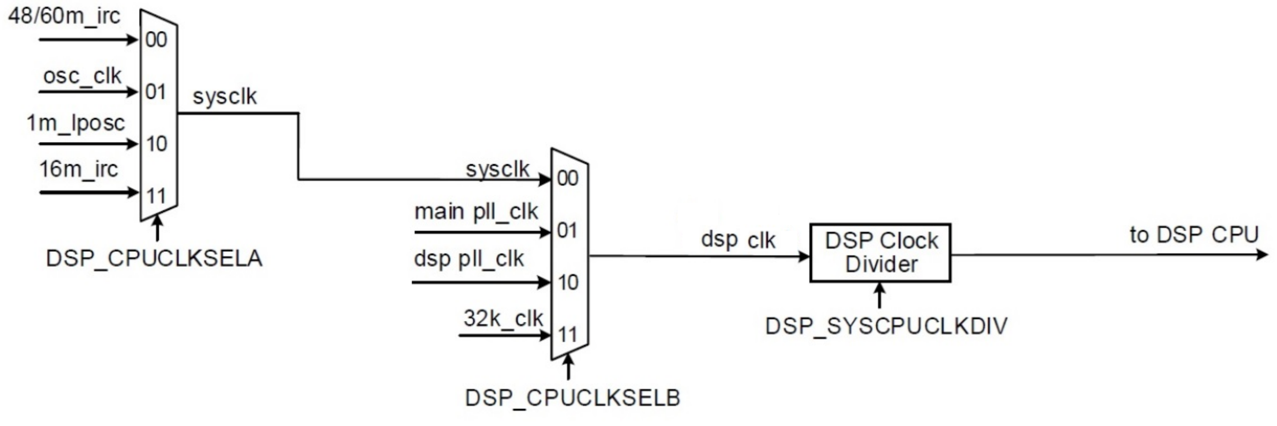


Figure 3. DSP clock

3.2 Registers impacting power consumption

There are 6 groups of system configuration function clock control registers group 0/1, reset control registers group 0/1, and other system registers group 0/1. Clock control register group has the clock control register CLKCTLx_PSCCTLx. This register enables the clocks to individual system and peripheral blocks. Reset control registers group has the Peripheral reset control register RSTCTLx_PRSTCTLx. This register allows software to reset specific peripherals. Other system registers group has the run configuration register SYSCTLx_PDRUNCFGx. This register controls the power to various blocks during normal operation. Configuring PDRUNCFG is typically accomplished using a power API that handles all the details of altering PDRUNCFG bits. For the details of these registers, see the i.MX RT6xx user manual.

3.3 Important pins and description

In this application note, external PMIC is used to supply core logic power. So, the pin of LDO_enable must be pull down.

Table 1. Important pins

Symbol	Description
LDO_enable	When 1, enables the on-chip regulator to power core logic through the VDDCORE pins. Tie low if an off-chip power management IC (PMIC) is used to supply power to core logic. This pin cannot be left floating. 100K external pull-up or 10K external pull-down is recommended
VDDCORE	Power supply for core logic. May be supplied from the internal LDO or externally by an off-chip power management IC (PMIC). An external filter capacitor is always required on these pins

4 Low power application design

This section describes simulation of low power applications. It allows selection of different frequencies such as 48 MHz, 24 MHz, 16 MHz, or 8 MHz for Arm Cortex-M33 and HiFi4 DSP. It turns off unnecessary blocks and lets Arm core enter sleep mode. Depending on the requirement, the user can select the block to be turned off. See the previous chapter for more information. Also, after DSP is initiated and it does some computations, it can execute a WAITI instruction to enter low power mode. So, the whole system can further reduce total power consumption.

4.1 Developing environment

- Install the IAR Workbench 8.40.1 or newer version on your laptop.
- If using lower version: Copy the content of the IAR i.MX RT600 patch into the

IAR Systems\Embedded Workbench 8.0\arm\config\.

This allows you to select the proper device in your IAR project.

- Run pre-installed terminal PC application (for example, Tera Term)

Use serial communication configuration:

• Baud: 115200

• Data size: 8-bit

• Stop bit: 1

• Parity: no

- Install Xtensa On Chip Debugger Daemon.
- Install Xtensa Development Environment 8.0.10.
- Install DSP Build Configuration rt600_sram_2019_1.

4.2 Function introduction

This section introduces the functions used in this measurement.

Table 2. CLOCK_AttachClk

Name	CLOCK_AttachClk
Prototype	void CLOCK_AttachClk(clock_attach_id_t connection);
Input parameter	Clock to be configured
Result	None
Description	Configure the clock selection multiplexors

Table 3. CLOCK_SetClkDiv

name	CLOCK_SetClkDiv
prototype	void CLOCK_SetClkDiv(clock_div_name_t div_name, uint32_t divider);
Input parameter	Clock divider name and value to be divided
Result	None
Description	Setup peripheral clock dividers

Table 4. CLOCK_InitSysPfd

name	CLOCK_InitSysPfd
prototype	void CLOCK_InitSysPfd(clock_pfd_t pfd, uint8_t divider);
Input parameter	Which PFD clock to enable and the PFD divider value
Result	None

Table continues on the next page...

Table 4. CLOCK_InitSysPfd (continued)

Description	Initialize the system PLL PFD
Note	PFD settings are kept between 12-35

Table 5. CLOCK_DisableClock

Name	CLOCK_DisableClock
Prototype	static inline void CLOCK_DisableClock(clock_ip_name_t clk);
Input parameter	Clock IP name
Result	None
Description	Disable the selected clock

Table 6. RESET_SetPeripheralReset

Name	RESET_SetPeripheralReset
Prototype	void RESET_SetPeripheralReset(reset_ip_name_t peripheral);
Input parameter	Assert reset to this peripheral
Result	None
Description	Asserts reset signal to the specified peripheral module

Table 7. POWER_EnablePD

Name	POWER_EnablePD
Prototype	static inline void POWER_EnablePD(pd_bit_t en);
Input parameter	peripheral for which to enable the PDRUNCFG bit
Result	None
Description	API to enable PDRUNCFG bit in the Sysctl0.
Note	Enabling the bit powers down the peripheral.

Table 8. POWER_ApplyPD

Name	POWER_ApplyPD
Prototype	void POWER_ApplyPD (void);
Input parameter	None

Table continues on the next page...

Table 8. POWER_ApplyPD (continued)

Result	None
Description	Apply updated PDRUNCFG bits in SYSCTL0.
Note	This is from Power API and available in the power library provided with the SDK

Table 9. BOARD_SetPmicVoltageForFreq

Name	BOARD_SetPmicVoltageForFreq
Prototype	void BOARD_SetPmicVoltageForFreq(uint32_t main_clk_freq, uint32_t dsp_main_clk_freq);
Input parameter	Main clock frequency and DSP main clock frequency
Result	None
Description	PMIC based on input frequency to provide different voltages

Table 10. POWER_EnterSleep

Name	POWER_EnterSleep
Prototype	void POWER_EnterSleep (void);
Input parameter	None
Result	None
Description	Configures and enters in Sleep low power mode
Note	This is from Power API and available in the power library provided with the SDK

Table 11. some_computation

name	some_computation
prototype	int some_computation(void);
Input parameter	None
Result	None
Description	Do some computations to simulate user application

Table 12. XT_WAITI

Name	XT_WAITI
Prototype	void XT_WAITI(int s);

Table continues on the next page...

Table 12. XT_WAITI (continued)

Input parameter	Timeout interval
Result	None
Description	Wait for interrupt

5 Measurement

5.1 Board overview and measuring point

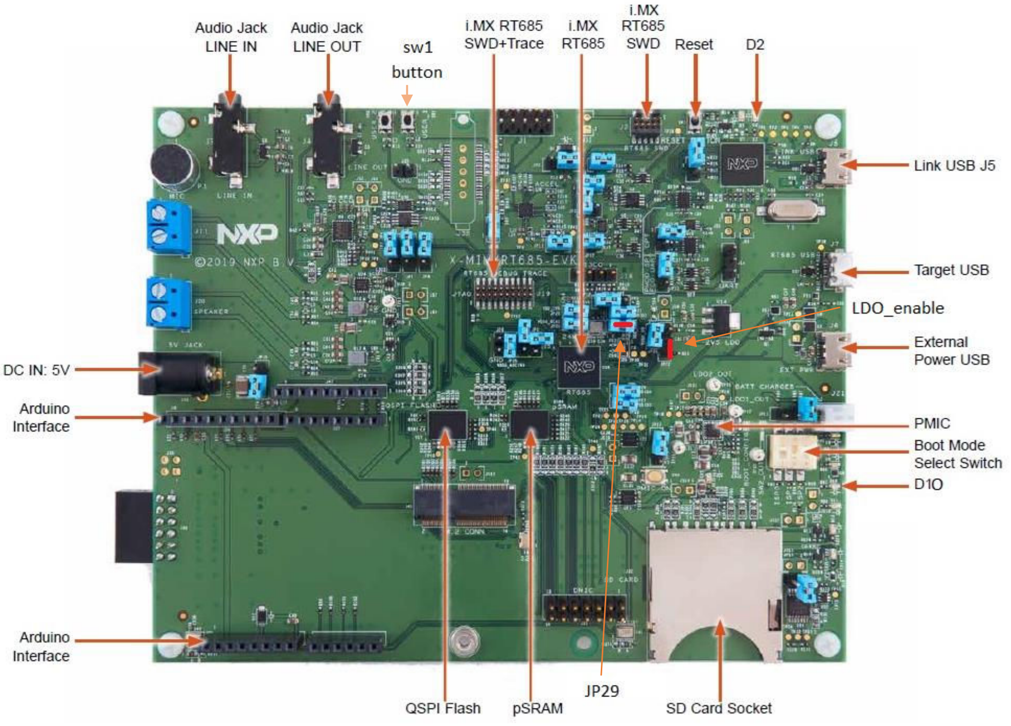


Figure 4. MIMXRT6xx EVK board overview

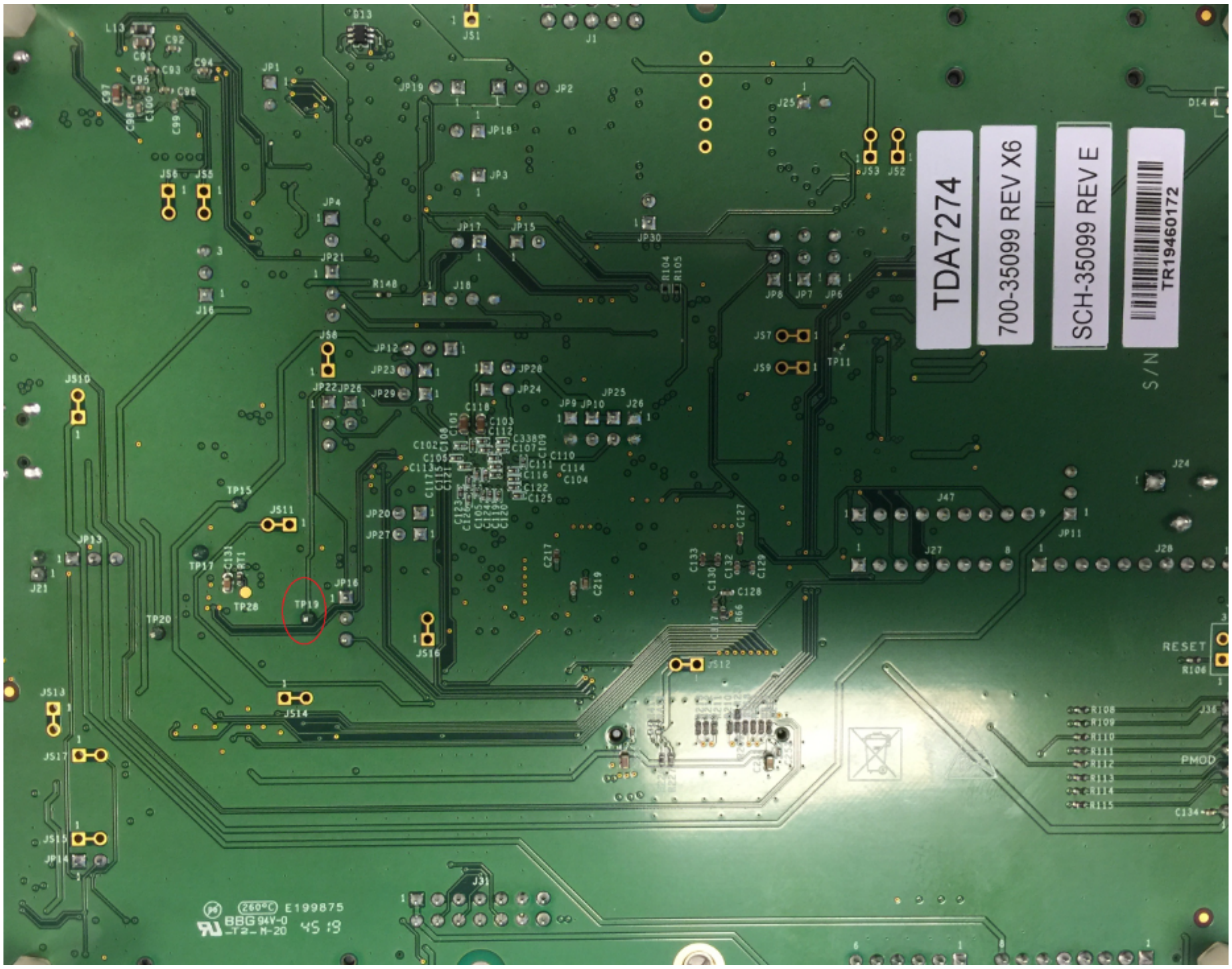


Figure 5. Reverse side of MIMXRT6xx EVK board

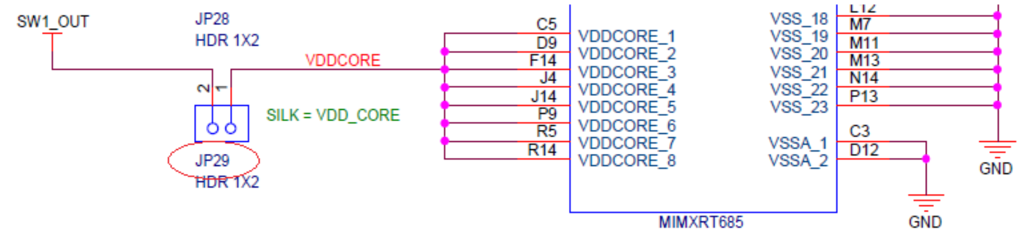


Figure 6. Measure point JP29

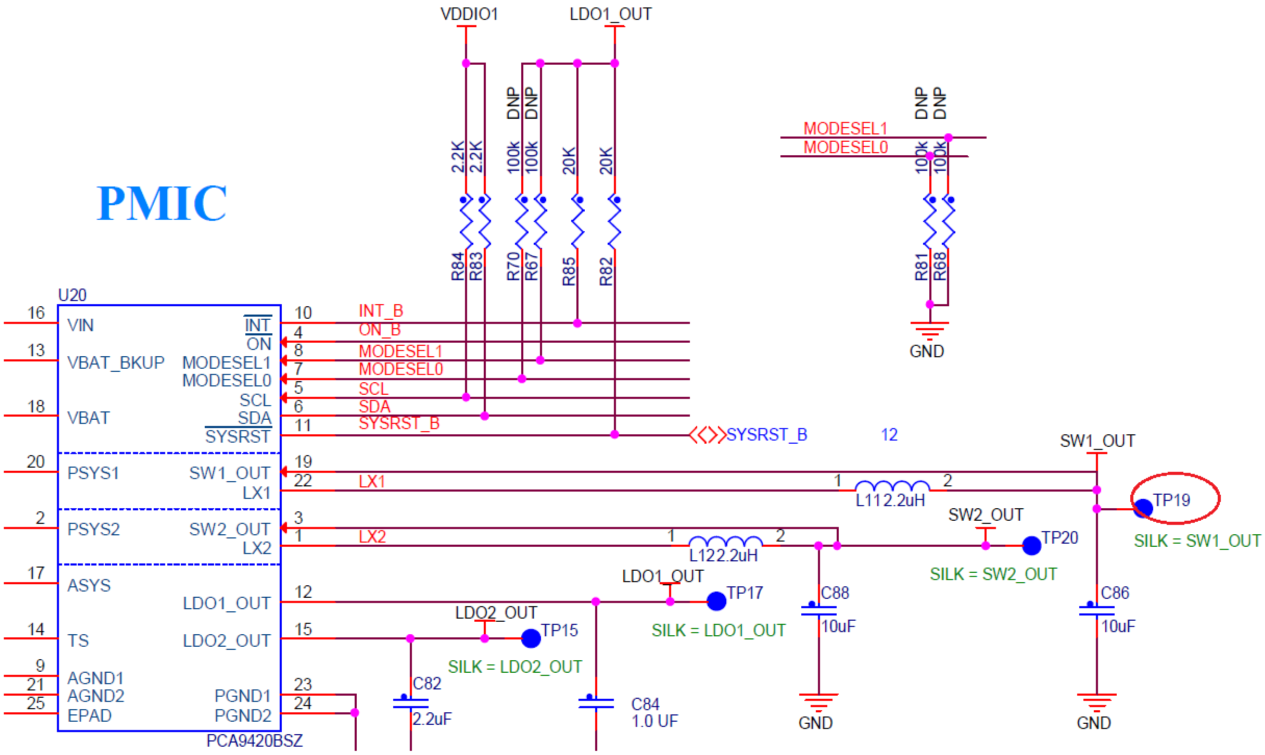


Figure 7. Measure point TP19

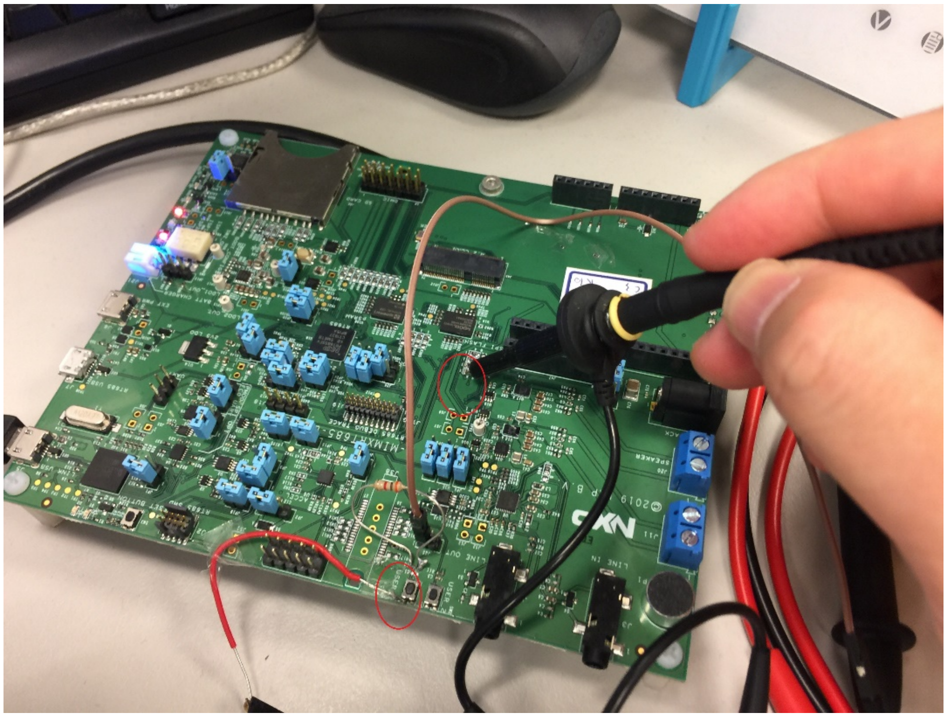


Figure 8. DSP wake up time measure point

Remember to install jumper top to JP22 position 2-3 to pull low LDO_ENABLE and use an off-chip PMIC PCA9420 to supply power to core logic. Measure current value on JP29 and voltage on TP19 for VDDCORE power consumption.

See Figure 8 for dsp wake up time measure point sw1 and R398 (red LED).

5.2 DSP wake up measurement

When DSP executes a WAITI instruction to enter low power mode, user can push the button (SW1) to wake it. After wake up, an interrupt callback function is called and it will turn on the red LED (D9) immediately. So we can measure the time interval between when the user button is pushed and the LED is turned on. In Figure 9 to Figure 12, the green line is the push button waveform, and the orange line is the LED waveform. These show the time interval for different kind of frequencies.

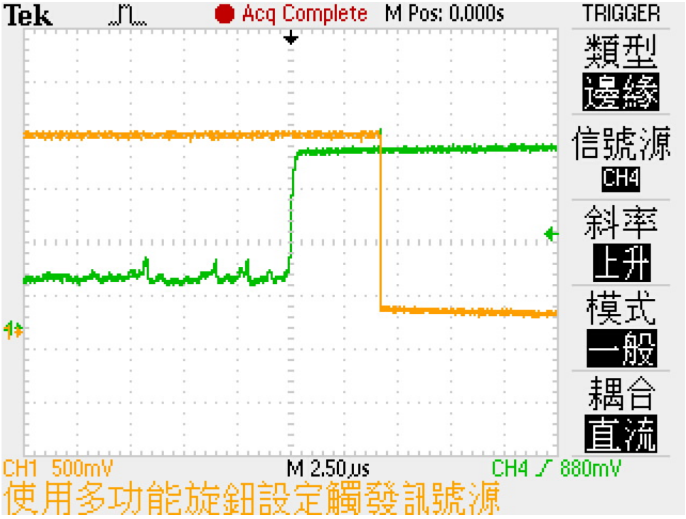


Figure 9. Arm 48 MHz, DSP 48 MHz

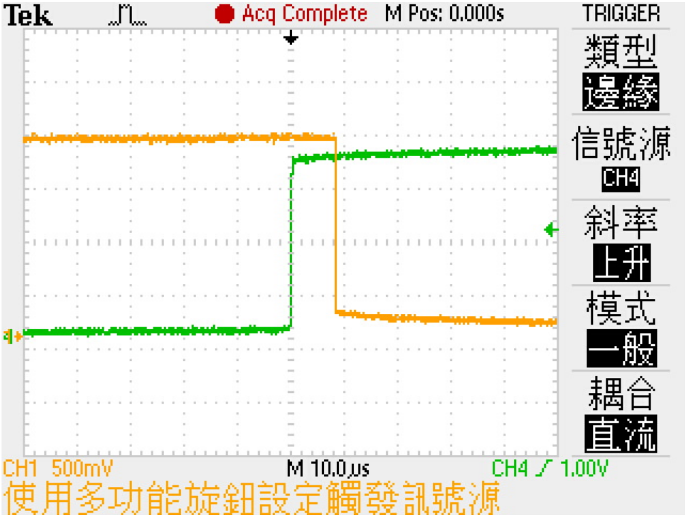
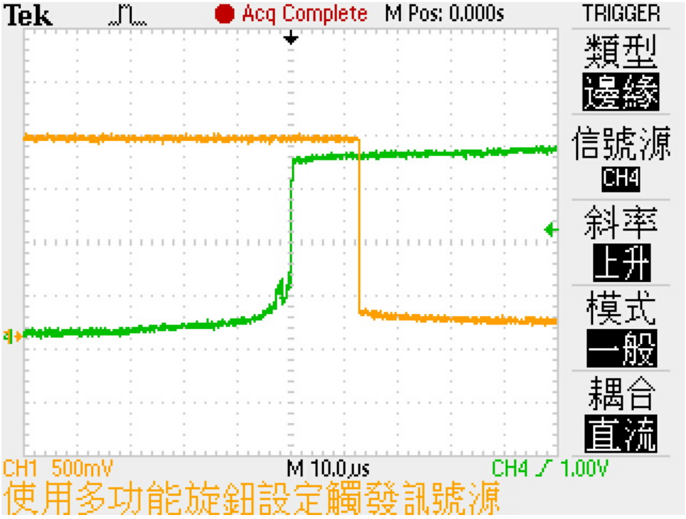


Figure 10. Arm 24 MHz, DSP 24 MHz

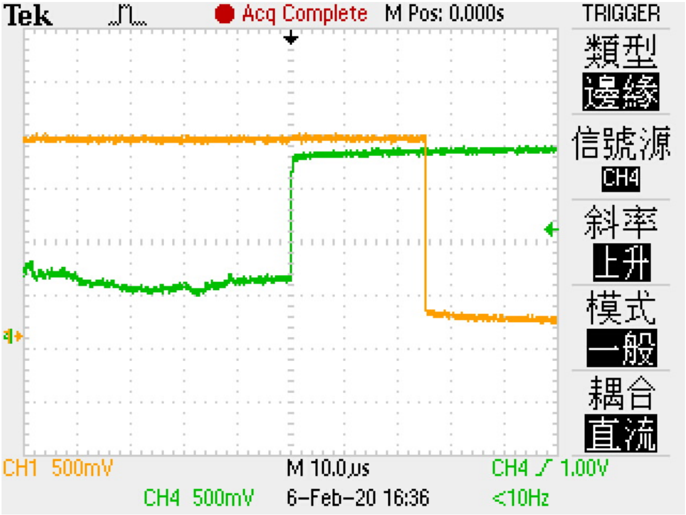


Acq Complete M Pos: 0.000s

CH1 500mV M 10.0μs CH4 ⟋ 1.00V

使用多功能旋鈕設定觸發訊號源

Figure 11. Arm 16 MHz, DSP 16 MHz



Acq Complete M Pos: 0.000s

CH1 500mV M 10.0μs CH4 ⟋ 1.00V

CH4 500mV 6-Feb-20 16:36 <10Hz

Figure 12. Arm 8 MHz, DSP 8 MHz

NOTE

It is recommended to remove button SW1 electric capacity (C3) and replace resistor (R10) to a smaller ohm capacity to ensure that the measured result is more accurate. See Figure 14 to know the comparison between before and after when electric capacity is removed and replaced with a smaller ohm resistor.

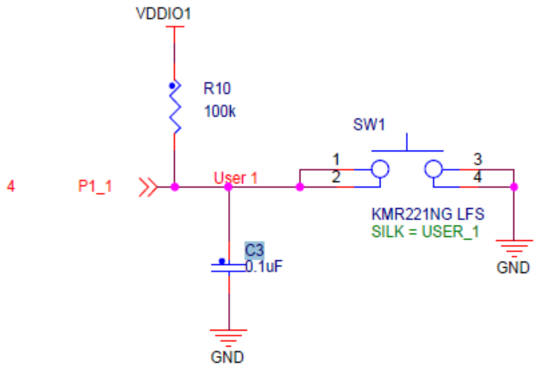


Figure 13. SW1 electric capacity C3 and resistor R10

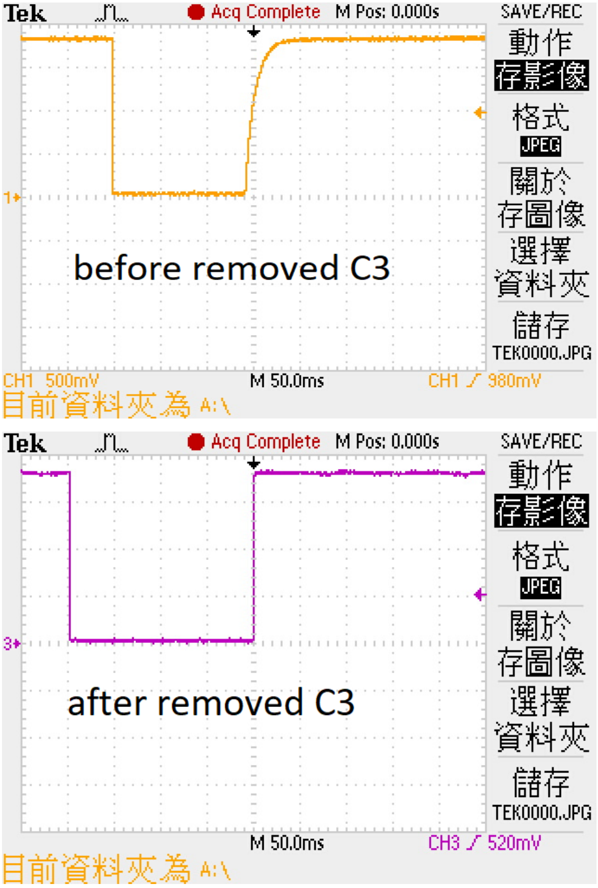


Figure 14. Push button wave

5.3 Measure result

Table 13. Result

Frequency	Vddcore (V)	Current (mA)	Power (mW)	Wake up time(µs)
ARM 48 MHz, DSP 48 MHz	0.7	6.85	4.795	4
ARM 24 MHz, DSP 24 MHz	0.7	6.07	4.249	8
ARM 16 MHz, DSP 16 MHz	0.7	5.86	4.102	11
ARM 8 MHz, DSP 8 MHz	0.7	5.71	3.997	23

6 Abbreviations and descriptions

Table 14. Abbreviations and descriptions

Name	Description
1m_lposc	Internal 1 MHz low power oscillator (LPOSC). Can be used as a low speed/low power system clock and/or to drive selected peripheral functions.
16m_irc	Internal 16 MHz oscillator (SFRO). May be used for as a clock source for the main and/or audio PLLs, main_clk, DSP clock, and many peripheral functions.
32k_clk	The output of the RTC oscillator, intended to be used with a 32.768 kHz crystal. The 32 kHz clock must be enabled in the RTCOSCCTRL register. May be used for as a clock source for main_clk, DSP clock, and selected peripheral functions.
48/60m_irc	FFRO internal oscillator with a default frequency of 48 MHz, user selectable as either 48 MHz or 60 MHz. May be used for as a clock source for main_clk, DSP clock, and selected peripheral.
dsp_main_clk	The clock used to derive the DSP CPU clock.
dsp_pll_clk	PFD1 output of the Main PLL, optionally divided by the DSP PLL clock divider. This clock can potentially be the base clock of dsp_main_clk.
main_clk	The clock used to derive hclk (which is used by the Cortex-M33, AHB bus, APB bus, and others) and used as the source clock for many other peripheral functions
main_pll_clk	PFD0 output of the Main PLL, optionally divided by the Main PLL clock divider. This clock can potentially be the base clock of main_clk, dsp_main_clk, and a number of peripheral functions.

7 References

1. i.MX RT6xx User manual
2. i.MX RT600 Rev B0 Data Sheet
3. MIMXRT6xx EVK board schematic (SPF-35099_E)
4. PCA9420 Data Sheet (Power management IC for low-power microcontroller applications)
5. AN12085-How to use i.MX RT Low-Power feature
6. AN12094-Power consumption and measurement

7. Getting Started with Xplorer for MIMXRT600

8 Revision history

Table 15. Revision history

Revision number	Date	Substantive changes
0	05/2020	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, 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, Vybrid, 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, Arm7TDMI, 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.

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

Date of release: May, 2020
Document identifier: AN12844