NXP Semiconductors
Application Note

Document Number: AN12051
Rev. 0, 11/2017

Migration Guide from i.MX6UL to i.MXRT

1. Introduction

This document describes the key points of the embedded system migration from the i.MX6UL application processor to i.MXRT crossover processor. It describes the hardware board design considerations, the software, and tools.

i.MX6UL application processor and i.MXRT crossover processor are designed for different markets, so we won't compare their advantages and disadvantages in this document. The audiences of this document are the users:

- who have developed embedded system with i.MX6UL and decided to migrate the project to i.MXRT;
- who have i.MX6UL experience and wish to start project with i.MXRT.

This document only describes the software migration based on the NXP SDK release with FreeRTOS support, Linux BSP not involved.

Contents

2. SoC comparison

Table 1 compares the features of i.MX6UL and i.MXRT1050.

Table 1. Feature comparison

Feature	i.MX6UL	i.MXRT1050
CPU core	Arm Cortex-A7 528 MHz NEON + MMU	Arm Cortex-M7 600 MHz FPU + MPU
Cache	32 K/32 K L1 I/D-Cache 128 K L2 Cache	32 K/32 K L1 I/D-Cache
Internal RAM	128 KB OCRAM	512 KB FlexRAM (OCRAM+I/DTCM)
Power Management	LDO	Integrated DCDC/LDO
Serial Flash I/F	Dual-ch DDR QuadSPI XIP supported	Dual-ch DDR QuadSPI/OctalSPI/HyperBus XIP supported
DRAM I/F	LPDDR2/DDR3/DDR3L 16-bit, 400 MHz	SDRAM 8/16-bit, 166 MHz
External Memory	8/16-bits Parallel NOR Flash 8-bits MLC/SLC NAND Flash (HW ECC)	8/16-bits Parallel NOR Flash 8/16-bits SLC NAND Flash (SW ECC)
eMMC/SD I/F	eMMC 4.5/SD 3.0 x 2	eMMC 4.5/SD 3.0 x 2
USB	USB OTG HS w/PHY x 2	USB OTG HS w/PHY x 2
Ethernet	10/100 with IEEE1588 x 2	10/100 with IEEE1588 x 1
Graphics	CSC, Resize, Combine, Rotate, BitBlit	CSC, Resize, Combine, Rotate, BitBlit
Display	24-bit RGB up to WXGA	24-bit RGB up to WVGA
Camera Sensor	24-bit parallel	24-bit parallel
Audio	ESAI, SAI/I2S x 3, SPDIF Tx/Rx, ASRC, MQS	SAI/I2S x 2, Multi-channel I2S x 1, SPDIF Tx/Rx, MQS
ADC/ACMP	12-bits ADC x 2	12-bits ADC x 2, ACMP x 4
Timers and PWM	GPT x 2, EPIT x 2, PWM x 4, WDOG x 3	GPT x 2, PIT x 4, QTimer x 4, FlexPWM x 4, QuadDecoder x 4, WDOG x 3
Connectivity	UART x 8, SPI x 4, I2C x 4, FlexCAN x 2	LPUART x 8, LPSPI x 4, LPI2C x 4, FlexIO x 2, FlexCAN x 2

2.1. Core

As the CPU core is different, i.MXRT is a high-performance microcontroller and i.MX6UL is a low-power application processor, as shown in *Table 2*. The software needs to be updated for the migration from i.MX6UL to i.MXRT.

Table 2. Core comparison

Features	SOCs		Comments
	i.MXRT1050	**i.MX6UL**	
Arm Core	Cortex-M7 (600 MHz, Arm7-M)	Cortex-A7 (528 MHz, Arm7-A)	Both cores are power efficient Arm core. M7 is the high-performance core among the Cortex-M series, and A7 is the low-power core among the Cortex-A series.
FPU/DSP	VFPv5 8/16-bit SIMD	VFPv4 NEON (SIMDv2)	NEON can support advanced SIMD.
Memory Management	MPU	MMU	With MMU virtual memory map support, i.MX6UL can run Rich Operating Systems, like Linux.

Table 2. Core comparison

Features	SOCs		Comments
Cache	L1 32 KB/32 KB I/D-Cache	L1 32 KB/32 KB I/D-Cache	Identical
Interrupt Controller	NVIC	GIC	NVIC has limited interrupt sources support, but with low interrupt latency. GIC supports multi-cores and many interrupt sources by sharing the interrupt lines. GIC takes more clock cycles to jump to IRQ handler than the NVIC.
Security	N/A	TrustZone	TrustZone technology provides system-wide hardware isolation for trusted software.

2.2. System modules

2.2.1. Clocks

i.MXRT reuses most clock tree of i.MX6UL, with seven dedicated PLLs. i.MXRT uses the AHB clock root for the Arm core, which can be generated from PLL1 and PLL2's PFD.

Table 3. Clock module comparison

Clocks	i.MXRT1050	i.MX6UL	Comments
PLLs	7 x PLLs	7 x PLLs	Identical. Arm/System/USB1/Audio /Video/ENET/USB2 PLLs
Arm core clock	ahb_clk (generated from PLL1 or PLL2 PFD)	arm_clk (generated from PLL1)	i.MXRT provides more flexible clock sources for Arm core
Clock generation	24 MHz/32 KHz external crystal	24 MHz/32 KHz external crystal	Identical

2.2.2. Power

i.MXRT reuses the power architecture and low power modes of i.MX6UL, but with a new DCDC analog IP. The power supply to the Arm core, which requires voltage range from 0.9 V to 1.3 V, can be generated by internal DCDC from 3.3 V source. This means users can use one 3.3 V power source for all the power domains except USB (5 V). The power supply and sequence design on the PCB board is much easier than i.MX6UL. For details, please refer to *3.1*.

2.2.3. FlexRAM

i.MXRT brings a new FlexRAM IP, which divides the on-chip SRAM into several 32 KB memory banks. Each bank can be configured as Tightly-Coupled Memory (TCM) or OCRAM. When the banks are configured as TCM, the CPU core can access the banks directly without waiting and bypassing the L1 cache. When the banks are configured as OCRAM, the CPU core can access the banks by L1 cache, same as OCRAM on i.MX6UL.

2.2.4. DMA

i.MXRT uses Enhanced DMA (EDMA) instead of Smart DMA (SDMA). The EDMA supports the following features:

1. 32 channels
2. fixed-priority and round-robin channel arbitration
3. programmable support for scatter/gather DMA processing
4. channel activation by software initiation, channel-to-channel link, and hardware request

With the DMAMUX support, 128 hardware trigger sources can be routed to the EDMA's 32 channels, and a periodic trigger mode for DMA channel is supported by PIT timer trigger source. The EDMA does not use the embedded script, like SDMA, to transfer, and the data flow is controlled in the Transfer-Control Descriptor (TCD) which is filled by CPU. It's flexible to use EDMA with a well-prepared TCD.

2.2.5. XBAR

i.MXRT integrates a new on-chip cross trigger network, as shown in *Figure 1*.

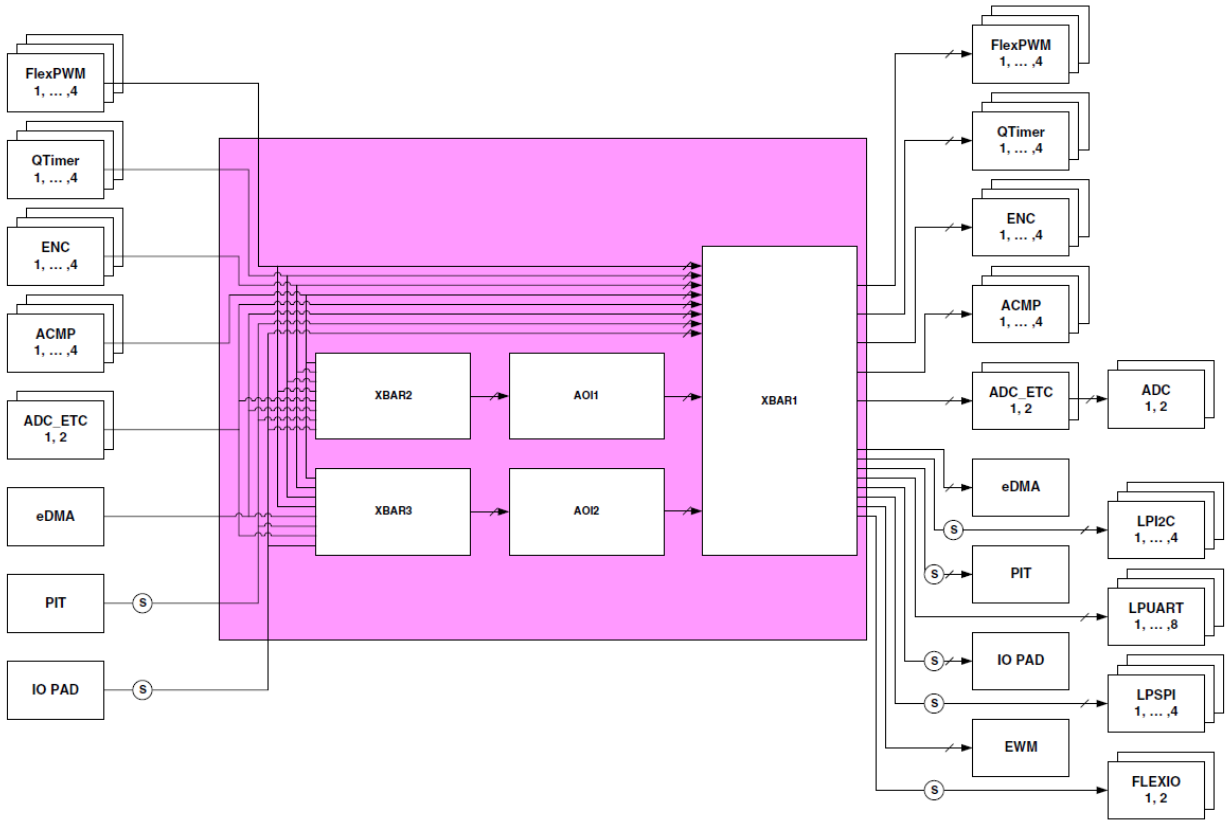

Figure 1. Cross trigger network

This Cross-Trigger Network consists of three Inter-Peripheral Crossbar Switches (XBARs) and two And-Or-Invert (AOIs). XBAR allows any input (typically from external GPIO or internal module

outputs) connected to any output under users' control. The model allows user configuration of data paths between internal modules and between internal modules and GPIO. AOI supports the signal generation by inverting the signals from AOI inputs. The typical use case of the cross-trigger networking is the motor control.

2.3. External Memory

2.3.1. FlexSPI

FlexSPI is designed with a flexible sequence engine (LUT table) to support various vendor devices:

- Serial NOR Flash or other device with similar SPI protocol as Serial NOR Flash
- Serial NAND Flash
- HyperBus device (HyperFlash/HyperRAM)
- FPGA device

It supports the following features:

- Flash eXcute-In-Place (XIP)
- Flash access mode
  - Single/Dual/Quad/Octal access mode
  - Single Data Rate (SDR) and Double Data Rate (DDR) mode (it's called STR/DTR in some of the Flash device spec).
  - Individual/Parallel mode (Parallel mode means Read/Program Access to two serial flash devices in parallel)
- Sampling clock mode
  - Internal dummy read strobe loopbacked internally
  - Internal dummy read strobe loopbacked from pad (DQS pin)
  - Flash provided read strobe (DQS pin)
- Memory mapped access by AHB bus and SW trigger access by IP Bus

FlexSPI can support more Flash, RAM devices than the QuadSPI IP on the i.MX6UL, even the fastest HyperFlash and HyperRAM. The maximum bandwidth is: 166 MHz x 2 (DDR) x 8 (Octal) = 332 MB/s.

2.3.2. SEMC

SEMC is a brand-new IP and a multi-standard memory controller optimized for both high-performance and low pin-count. It can support multiple external memories in the same application with shared address and data pins. It supports:

- SDRAM interface
  - Supporting both 8-bit and 16-bit modes, up to 166 MHz
  - Up to 512 Mb per Chip Select (CS) and up to 4x CS
- NOR Flash and SRAM interface

  - Supporting both 8-bit and 16-bit modes
  - Async mode
  - Address and Data Multiplexing (ADM)
  - Up to 128 Mb per Chip Select (CS)
  - Supporting NOR Flash program via IPBus configuration
- 8080 display interface
  - Supporting 8/16/24-bit modes
  - Up to 100 MHz
- NAND Flash
  - 8/16-bit NAND SLC FLASH with SW ECC
  - Async mode only
  - Only supporting devices capable of cache read/write and page-based operation
- Dynamic I/O sharing which enables multiple external memory device in parallel
- Multi-device access pattern scheduler

For i.MX6UL, MMDC supports DDR, EMI supports PNOR/PSRAM, and GPMI supports NAND. However, i.MXRT uses one SEMC IP to support SDRAM, PNOR, PSRAM and NAND Flash with shared pins, which reduces the SoC, BOM cost and simplifies the hardware board design and layout.

2.4. Analog

2.4.1. ADC

i.MXRT reuses the ADC module of i.MX6UL, with a new ADC_ETC module added. The ADC_ETC module enables multiple users to share the ADC modules in a Time-Division-Multiplexing (TDM) mode. The external triggers can be from Cross BAR(XBAR) and TSC. The ADC_ETC has two channels, each supporting one TSC and four external triggers from XBAR to one ADC module. The ADC_ETC can support interrupt mode and DMA mode.

2.4.2. ACMP

The Analog Comparator (ACMP) module reuses that of Kinetis products. It provides a circuit for comparing two analog input voltages. The comparator circuit is designed to operate across the full range of the supply voltage, known as rail-to-rail operation. It supports programmable hysteresis control, selectable interrupt on rising-edge, falling-edge, or both rising or falling edges of the comparator output. The comparator filter and sampling features can be combined to multiple working modes, like Continuous, Sampled/(Non)Filtered mode, Windowed mode, etc. It also supports the DMA transfer**.**

2.5. Timers

i.MXRT includes three new timer IPs, especially for the motor control use cases besides the GPT, PIT and WDOG.

2.5.1. FlexPWM

FlexPWM module contains PWM submodules, each controlling a single half-bridge power stage. FlexPWN provides the fault channel supportand can generate various switching patterns, including highly sophisticated waveforms. The main features of the FlexPWM IP module are:

- Each FlexPWM module is configured with four channels, each supporting A, B, and X PWM output. All channels are configured for standard edge placement and have their input capture function enabled.
- All four FlexPWM can work in the synchronous mode.
- All four sub-modules are with A, B and X outputs.
- No sub-module uses nano-edge placement, but supporting accumulative fractional clock calculation.
- All four sub-modules support input capture with FIFO_DEPTH = 1 (without FIFO) and CYC_WIDTH = 4 (4-bit wide).
- Four fault inputs.

2.5.2. Quad timer

The Quad Timer provides four timer channels with a variety of controls affecting both individual and multi-channel features, including:

- up/down count
- cascading of counters
- programmable modulo
- count once/repeated
- counter preload
- compare registers with preload
- shared use of input signals
- prescaler controls
- independent capture/compare
- fault input control
- programmable input filters
- multi-channel synchronization

The input of Quad Timer can be sourced from PADs or XBAR1. It’s used as a precise timer or a Quadrature decoder (supporting to PHASEA/B and INDEX).

2.5.3. Quadrature decoder

The enhanced quadrature decoder module provides the interfaces to position/speed sensors in the industrial motor control applications. It has five input signals: PHASEA, PHASEB, INDEX, TRIGGER, and HOME. This module is used to decode shaft position, revolution count and speed. The key features of the Quadrature Decoder module are:

- including logic to decode quadrature signals
- 32-bit position counter capable of modulo counting
- 16-bit position difference register

2.6. Connectivity

2.6.1. LPUART/LPSPI/LPI2C

i.MXRT updates the UART/SPI/I2C IP to the Low Power version. They support to wake up the SoC in the STOP low power mode (suspend), which means SoC can go into the STOP mode to save power consumption when waiting for the data from external peripheral connected by UART, SPI or I^2C.

For i.MX6UL, the I2C module supports the Fast mode. For i.MXRT, the LPI^2C module supports the Standard, Fast, Fast+ (1 MHz), Ultra-Fast (5 MHz) and HS slave mode.

2.6.2. FlexIO

The FlexIO is a highly configurable module which provides a wide range of functionalities, including:

- Emulation of a variety of serial/parallel communication protocols
- Flexible 16-bit timers supporting a variety of trigger, reset, enable and disable conditions
- Programmable logic blocks allowing the implementation of digital logic functions on-chip and configurable interaction of internal and external modules
- Programmable state machine for offloading basic system control functions from CPU

FlexIO supports a wide range of protocols including, but not limited to, UART, I^2C, SPI, I^2S, Camera IF, Motorola 68 K/Intel 8080 bus and PWM/Waveform generation. Users can use FlexIO to emulate those hardware signals if there aren't more IP resources to be used, which is a very flexible and extendable solution.

3. Hardware design migration

This chapter covers some important things, like power supply, DRAM layout and boot devices on the hardware design migration from i.MX6UL to i.MXRT.

3.1. Power supply

Since the i.MXRT has integrated the DCDC inside the SoC, the power supply to the i.MXRT is much simpler than i.MX6UL and the hardware design is easier as well. Users can use one single 3.3 V power source for most of the power, even for VDD_SNVS_IN (in case users do not want to use SNVS power mode). The on-chip DCDC can convert the 3.3 V input into the low voltage (0.9 V to 1.25 V, depending on the power mode) which Arm core needs. *Figure 2* shows the power trees of both i.MX6UL and i.MXRT.

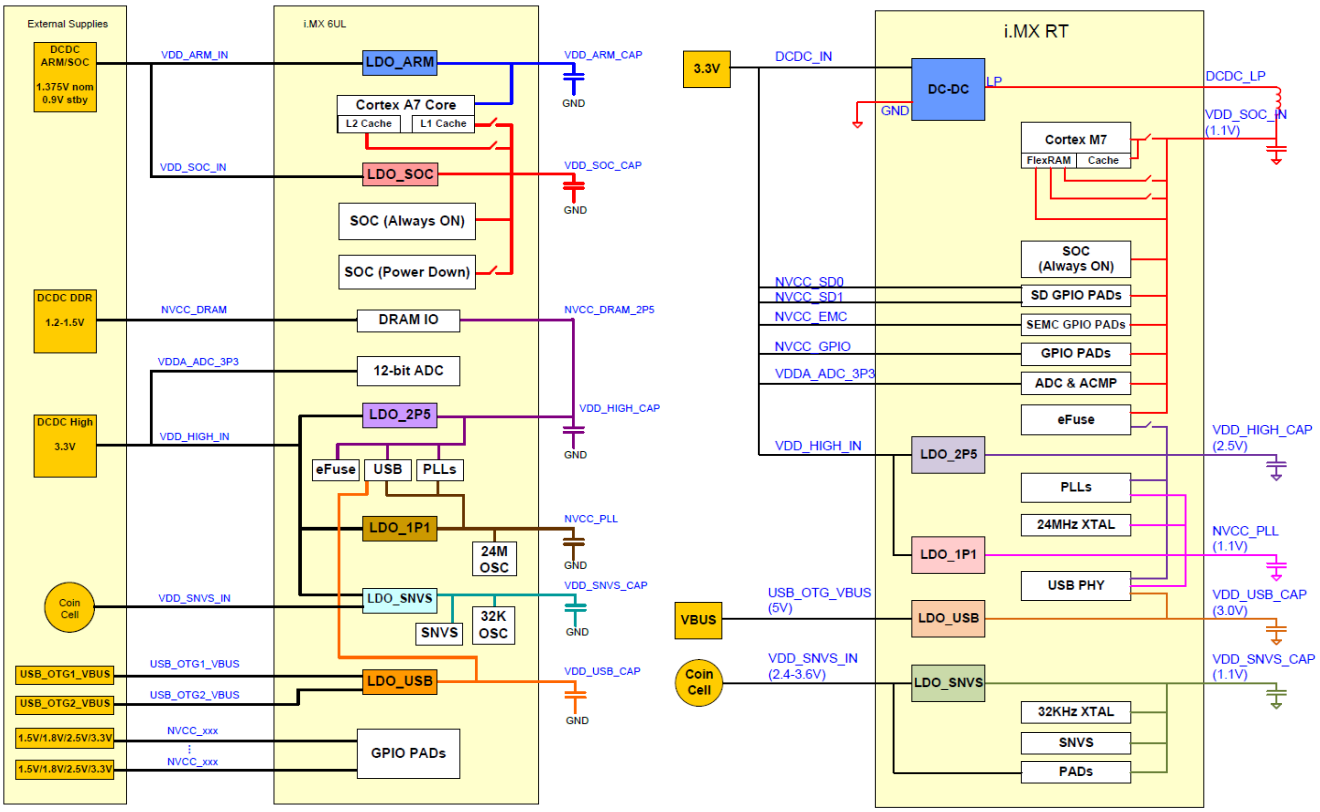

Figure 2. Power supply of i.MX6UL vs i.MXRT

3.2. SMEC for external memory

i.MXRT introduces a new IP, SEMC, to external memory interfaces. SEMC is a multi-standard memory controller optimized for both high-performance and low pin-count. SEMC supports multiple external memories (SDRAM, Raw NAND, PNOR, PSRAM and 8080 display) in the same application with shared address and data pins. *Table 4* shows the SEMC pin mux usage for different external memories and 8080 display interfaces.

Table 4. SEMC PinMux

I/O name	SDRAM	NAND	NOR	PSRAM	8080
SEMC_DA[0]	D0	D0	D0/A0	D0/A0	D0
SEMC_DA[1]	D1	D1	D1/A1	D1/A1	D1
SEMC_DA[2]	D2	D2	D2/A2	D2/A2	D2
SEMC_DA[3]	D3	D3	D3/A3	D3/A3	D3
SEMC_DA[4]	D4	D4	D4/A4	D4/A4	D4
SEMC_DA[5]	D5	D5	D5/A5	D5/A5	D5
SEMC_DA[6]	D6	D6	D6/A6	D6/A6	D6
SEMC_DA[7]	D7	D7	D7/A7	D7/A7	D7
SEMC_DA[8]	D8	D8	D8/A8	D8/A8	D8
SEMC_DA[9]	D9	D9	D9/A9	D9/A9	D9
SEMC_DA[10]	D10	D10	D10/A10	D10/A10	D10
SEMC_DA[11]	D11	D11	D11/A11	D11/A11	D11
SEMC_DA[12]	D12	D12	D12/A12	D12/A12	D12
SEMC_DA[13]	D13	D13	D13/A13	D13/A13	D13

Table 4. SEMC PinMux

I/O name	SDRAM	NAND	NOR	PSRAM	8080
SEMC_DA[14]	D14	D14	D14/A14	D14/A14	D14
SEMC_DA[15]	D15	D15	D15/A15	D15/A15	D15
SEMC_DM[0]	DQM0 (DQ Mask low byte)	-	-	LB# (Write Data Mask low byte)	-
SEMC_DM[1]	DQM1 (DQ Mask high byte)	-	-	UB# (Write Data Mask high byte)	-
SEMC_ADDR[0]	A0	-	A16	A16	-
SEMC_ADDR[1]	A1	-	A17	A17	-
SEMC_ADDR[2]	A2	-	A18	A18	-
SEMC_ADDR[3]	A3	-	A19	A19	-
SEMC_ADDR[4]	A4	-	A20	A20	-
SEMC_ADDR[5]	A5	-	A21	A21	-
SEMC_ADDR[6]	A6	-	A22	A22	-
SEMC_ADDR[7]	A7	-	A23	A23	-
SEMC_ADDR[8]	A8	CS4	CS5	CS6	CS7
SEMC_ADDR[9]	A9	CLE	-	CRE	-
SEMC_ADDR[10]	A10	-	-	-	-
SEMC_ADDR[11]	A11	WE#	WE#	WE#	WRx
SEMC_ADDR[12]	A12	RE#	OE#	OE#	RDx
SEMC_BA[0]	BA0	BA0			-
SEMC_BA[1]	BA1	ALE	ADV#	ADV#	DCx
SEMC_CAS	CAS#	-	-	-	-
SEMC_RAS	RAS#	-	-	-	-
SEMC_CLK	CLK	-	-	-	-
SEMC_CKE	CKE	-	-	-	-
SEMC_WE	WE#	-	-	-	-
SEMC_CS0	CS0	-	-	-	-
SEMC_CSx[0]	CS1/2/3	CS4	A24/CS5	A24/CS6	CS7
SEMC_CSx[1]	CS1/2/3	CS4	A25/CS5	A25/CS6	CS7
SEMC_CSx[2]	CS1/2/3	CS4	A26/CS5	A26/CS6	CS7
SEMC_CSx[3]	CS1/2/3	CS4	A27/CS5	A27/CS6	CS7
SEMC_RDY	CS1/2/3	R/B#	A27/C55	A27/C56	CS7
SEMC_DQS	DQS (Dummy read strobe loopback)				

3.3. HyperFlash support

i.MXRT supports the following boot devices:

- Serial NOR Flash (Quad/Octa/Hyper Flash) via FlexSPI
- Serial NAND Flash via FlexSPI
- Parallel NOR Flash via SEMC
- RAW NAND Flash via SEMC
- SD/MMC
- SPI NOR/EEPROM via LPSPI

HyperFlash is a brand new boot device supported by i.MXRT. With high performance, up to 166 MHz DTR and octa data lines, HyperFlash supports data throughput up to 332 MB/s. IHyperFlash is good for code XIP and data storage.

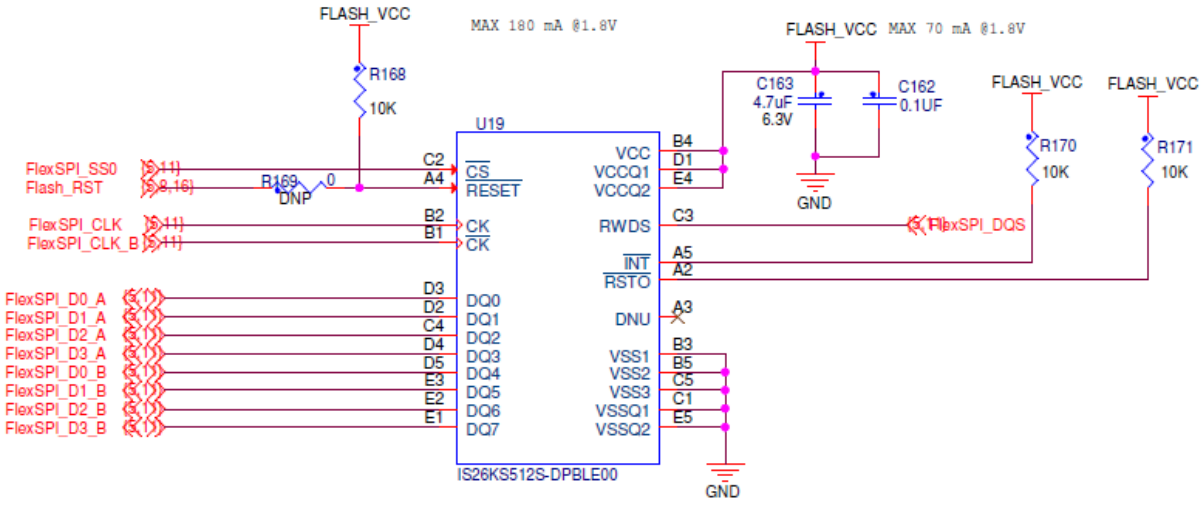

Figure 3. HyperFlash connection

To connect HyperFlash to i.MXRT, the FlexSPI A/B groups are used together to work in the combination mode, as shown in *Figure 3*. FlexSPI A group provides the CLK, SS, DQS and D0-D3, and FlexSPI B group provides the CLK_B (different clocks guarantee the stability of the high frequency) and D4-D7. The power supply to the HyperFlash is 1.8 V.

For the hardware connection of other boot devices like Parallel NOR and Raw NAND, please refer to *3.2*.

3.4. SDRAM layout

i.MXRT supports SDRAM only and i.MX6UL supports DDR, so the DRAM layout is much easier in i.MXRT.

The SDRAM interface, running up to 166 MHz, is one of the critical interfaces for chip routing. It must have the controlled impedance for the 50 Ω single-end trace. Ideally, route all the signals at the same length as MIMXRT1050-EVK board. Please refer to the MIMXRT1050-EVK board layout for routing all signals at the same length (±50 mil). SDRAM routing needs to be divided into three groups: data, address, and control. On the MIMXRT1050-EVK, the layout divides all SDRAM signals into two groups:

- all data lines and DM[x]
- all address lines and control lines

Due to 4-layer board design, the two groups of routing are referred to GND plane for impedance control. One group is at the top layer while the other is at the bottom layer.

For detailed layout recommendation, please check the Hardware Development Guide of i.MXRT.

4. Software migration

NXP SDK supports i.MXRT and i.MX6UL with different middleware and features, as shown in *Table 5*:

Table 5. Middlewear/features comparison

Middleware/Features	i.MXRT1050	i.MX6UL
DSP	CMSIS-DSP library	N/A
GUI	emWin	N/A
Networking	LwIP	LwIP
File System	FatFS	FatFS
USB	Device/Host/TypeC-PD	Device/Host
Security	mbedTLS/WolfSSL	mbedTLS
SDHC	SD/MMC	SD/MMC
WiFi	QCA4002	N/A
JPEG Enc/Dec	libjpeg	N/A

4.1. Core system

4.1.1. Using the DSP inside M7

By taking advantage of the DSP in M7 core, i.MXRT SDK provides the CMSIS-DSP library to support the following functions:

- Basic, Fast and Complex math functions
- Filters
- Matrix functions
- Transforms
- Motor control functions
- Statistical functions
- Support functions
- Interpolation functions

i.MX6UL does not provide any DSP (NEON) software library.

To use this CMSIS-DSP library, please refer to the example in the SDK package: <SDK>\ CMSIS\DSP_Lib\Examples\ARM\ and the reference API can be found here:

http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html

4.1.2. Using the L1 cache

Since the i.MXRT uses the M7 core with Memory Protection Unit (MPU), the L1 cache usage has some differences from the i.MX6UL A7 core with MMU. The cache operation API is same in the SDK, like enable/disable/flush/clean. But the configurations of memory type, access permission and cache policy for different memory region are different. To use L1 cache on i.MX6UL, the MMU must be turned on with translation/page table (minimum 1 M section size, and 4 KB page size) initialized for the virtual to physically memory map. For i.MXRT, the MPU is used to directly configure the memory type and cache policy based on the physical memory region. Please refer to the AN “Using the i.MXRT L1 Cache” for more details.

4.1.3. Using EDMA

The usage of the EDMA is a little different than SDMA in the SDK. Before using EDMA, users must configure the DMAMUX to route the different peripheral DMA requests (up to 128 requests) to 32 channels, or use always-on channel for memory-to-memory transfer, or use periodic trigger transfer. For more details, please refer to the EDMA driver examples in the SDK.

4.2. Memory system

4.2.1. Using FlexRAM

The FlexRAM includes an internal SRAM, which is up to 512 KB on i.MXRT1050. It supports ITCM, DTCM and OCRAM with configurable size, which shares the 512 KB SRAM. This configuration is performed by the IOMUXC_GPR_GPR14 registers in the IOMUX module or fixed in the eFuse. The TCM has high-performance with zero wait access, but it cannot be accessed when the CPU core is powered off in low power mode. The OCRAM can be accessed by other peripherals like eDMA, even when the CPU core is powered off. The critical application code/data which requires low latency and high-performance is recommended to store in the TCM, especially the vector table and interrupt handler. Please refer to the AN “Using the i.MXRT FlexRAM” for more details.

4.2.2. Using SDRAM

The way of using the SDRAM in i.MXRT is almost same as the i.MX6UL. The typical two methods are described below:

1. Use JLink script or .mac file in IAR to initialize the SEMC for external SDRAM when doing application development and debug.
5. Configure SEMC in the application where pieces of code are first loaded to TCM/OCRAM. After SDRAM is accessible, copy code/data or allocate memory from the SDRAM to use SDRAM.

NOTE

When using the 2nd method, users should make sure that the toolchains, like IAR, will not perform the zero initialization or data copy on SDRAM before SEMC initialization. This is configurable in the linker file.

4.3. Timers

4.3.1. Using FlexPWM

For i.MX6UL, the FlexPWM provides more complex and flexible features than the PWM modules. The i.MXRT SDK provides more APIs and functions for users as below:

- Fault control with level, clear and recover mode
- Reload logic for new prescaler, period and pulse width
- Multiply channel PWM output setup, duty cycle update and LDOK set
- Input capture setup
- Output trigger enablement for SoC intern modules inter-connection

For more details, please refer to the SDK driver examples.

4.3.2. Using Quad Timer

To use the QTimer as a PWM output or a timer is almost same as the GPT, with similar API in the SDK. The basic program flow is:

1. initialize QTimer IP
2. output PWM or Input Capture configuration
3. enable interrupt
4. start the timer

Since QTimer supports many capture input modes, like Edge-Count, Gated-Count, Quadrature-Count (w/ or w/o index input) Signed-Count modes etc., the users must configure the required channel, primary source, secondary source and count mode correctly according to the use cases. Please refer to the “SDK drivers” and “Functional Modes” sections described in the Quad Timer chapter of Reference Manual for more details.

4.4. Analog

4.4.1. Using ADC ETC

i.MXRT reuses the ADC modules of the i.MX6UL, with a new module called ADC External Trigger Control (ETC), added. With ETC, i.MXRT provides a hardware trigger function as well as the software trigger function in i.MX6UL. To use the hardware trigger, a trigger source from other modules or pins should be configured in XBAR, routing to one or several (up to four in i.MXRT1050) ADC ETC trigger

inputs. Then users configure the trigger(s) with mode, chain length, delay and priority parameters. Per the chain length, users configure the mode (Back-to-Back or interval), which ADC channel uses, completes interrupt enable or not, etc. to set up each chain. The ADC conversion result is stored in the trigger result register. For more details, please refer to the SDK ADC ETC driver examples.

4.5. Connectivity

4.5.1. LPUART/LPSPI/LPI2C

The SDK drivers for these three IP modules provide almost the same API as the IPs in the i.MX6UL, except for some feature changes. Users should be able to port the application code by using these drivers from i.MX6UL to i.MXRT conveniently.

4.5.2. FlexIO

The typical use case of the FlexIO module is to emulate the UART, SPI, I2C and I2S protocols. i.MXRT SDK provides those protocols with high-level drivers implemented for users. Usually, users should not configure the FlexIO IP directly, but use the high-level drivers. For more details, please refer to the FlexIO driver examples in the SDK.

5. Tools migration

i.MXRT delivers more tools than the i.MX6UL for the SDK and RTOS ecosystem. Users can re-use the tools for i.MX6UL or migrate to the new tools for easy use or lower cost.

Table 6. Tools comparison

Tools	i.MXRT1050	i.MX6UL
IDE	• IAR Embedded Workbench 8.11 • MDK 5.23 • MCUXpresso	• IAR Embedded Workbench 7.80
Debugger	• On-board DAP-LINK (EVK) • JLINK Plus (or above version)	• JLINK Plus (or above version)
Flash downloader	• FLashloader (work with ROM) • MFGTool • DAP-LINK MSD • IDE integrated flashloader	• MFGTool • IDE integrated flashloader

6. References

- i.MX 6UL Application Processor Reference Manual
- i.MX RT1050 Processor Reference Manual
- Kinetis SDK v.2.2 API Reference Manual (In the SDK release package)

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN12051
11/2017