NXP Semiconductors
Application Note

Document Number: AN12090
Rev. 0, 11/2017

Using Multi-Channel Feature of SAI

1. Introduction

The i.MX RT series are the industry's first crossover processors provided by NXP. They offer high-performance processing optimized for the lowest power consumption and best real-time response.

The Arm® Cortex®-M7 core delivers 3020 CoreMark/1284 DMIPS @ 600 MHz, which is a strong guarantee for high-performance applications. Another advantage of the Cortex-M7 core is that it contains three SAIs and SAI1 has four transmit (TX) data lines which support the 7.1 surround sound data transmission.

This document introduces the technology of the system, including SAI and FLAC. It provides guidance on how to use the multi-channel SAI feature to develop applications running in a correct high-performance way.

The software used as the example in this document is based on the i.MX RT1050 SDK release. The development environment is the IAR Embedded Workbench® 8.11. The hardware used to verify the example is the MIMXRT1050-EVB board.

Contents

2. SAI and FLAC overview

This chapter introduces the SAI module in the i.MX RT1050 and FLAC pure software decoding. Some other modules that bring advantages for the high-performance use case are also mentioned.

2.1. SAI

The I^2S module provides a Synchronous Audio Interface (SAI) that supports full-duplex serial interfaces with frame synchronization, such as I^2S, AC97, TDM, and codec/DSP interfaces.

2.1.1. Features

NOTE

Some of the features are not supported across all SAI instances. Only SAI1 supports four TX and RX data lines.

- Transmitter with independent bit clock and frame sync that supports four data lines.
- Receiver with independent bit clock and frame sync that supports four data lines.
- Each data line supports a maximum frame size of 32 words.
- Word size from 8 bits to 32 bits.
- Word size configured separately for the first word and the remaining words in the frame.
- Asynchronous 32 × 32-bit FIFO for each transmit and receive data line.
- Supports graceful restart after an FIFO error.
- Supports automatic restart after an FIFO error without a software intervention.
- Supports packing of 8-bit and 16-bit data into each 32-bit FIFO word.
- Supports combining multiple-data-line FIFOs into a single-data-line FIFO.

2.1.2. Block diagram

Figure 1 shows the module clocks.

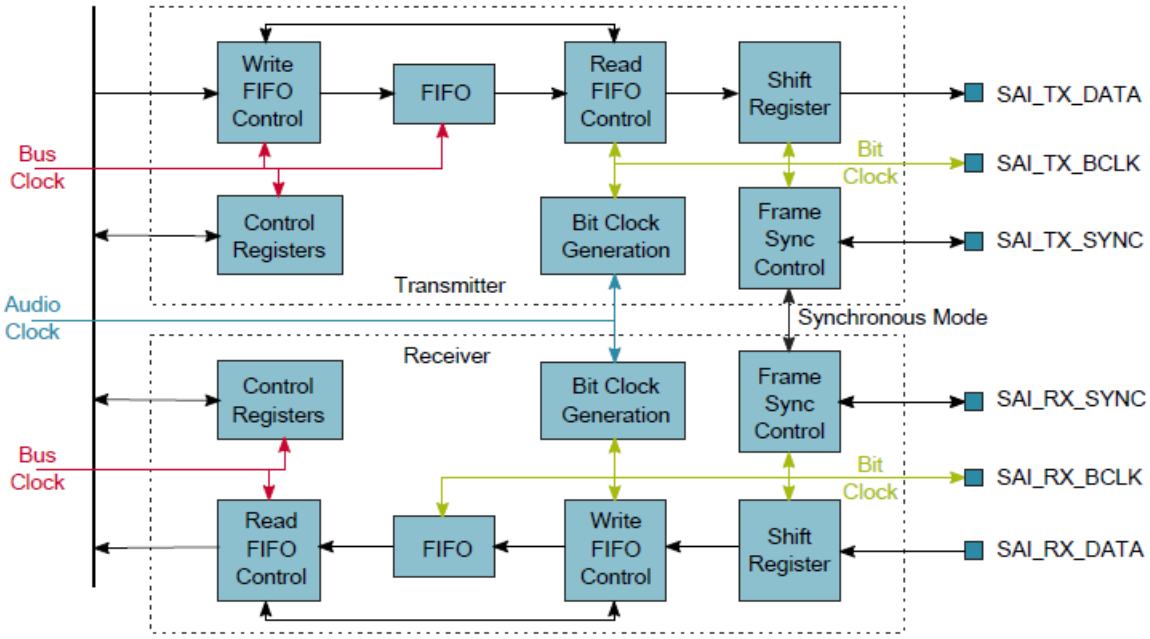

Figure 1. I²S/SAI block diagram

2.1.3. Clocking

The audio master clock is used to generate the bit clock when the receiver (or transmitter) is configured for an internally-generated bit clock. The transmitter and receiver can independently select between the bus clock and up to three audio master clocks to generate the bit clock.

The SAI transmitter and receiver support asynchronous free-running bit clocks that can be generated internally from an audio master clock or supplied externally. There is also the option for a synchronous bit clock and frame sync operation between the receiver and the transmitter or between multiple SAI peripherals.

2.1.4. Data FIFO

Each transmit and receive channel includes a 32 × 32-bit FIFO. The FIFO data is accessed using the SAI transmit/receive data registers. This use case has 16 transmit FIFOs.

The FIFO warning flag is set according to the number of entries in the FIFO. The transmit warning flag is set when the number of entries in any of the enabled transmit FIFOs is empty and cleared when the number of entries in each enabled transmit FIFO is not empty. The FIFO warning flag can generate an interrupt or a DMA request.

2.2. FLAC decoder

The Free Lossless Audio Codec (FLAC) is an audio format similar to MP3 but lossless, which means that the audio in the FLAC is compressed without any loss in quality. FLAC is free, which means that it is available at no cost. It also means that the specification of the format is fully open to the public to be used for any purpose (the FLAC project reserves the right to set the FLAC specification and certify the compliance) and neither the FLAC format nor any of the implemented encoding/decoding methods are covered by any known patent. It also means that the whole source code is available under the open-source license and its core library is BSD. It is the first truly open and free lossless audio format.

The channel assignment of the eight channels is: front left, front right, front center, LFE, back left, back right, side left, and side right. The FLAC decoding on the i.MX RT1050 is done in a purely software way.

2.3. Other modules

The block diagram in Figure 2 provides a view of the chip's major functional components and core complexes. The modules in the red rectangles bring advantages for designing a high-performance or system-integrated use case.

The Cortex-M7 core delivers 3020 CoreMark/1284 DMIPS @ 600 MHz, which is powerful enough for high-performance applications.

The Smart External Memory Controller (SEMC) is a multi-standard memory controller optimized for both high performance and low pin count. It supports multiple external memories in the same application with the shared address and data pins. The supported interface includes the SDRAM, NOR flash, SRAM, and NAND flash, as well as the 8080 display interface. The SDRAM interface supports both 8-bit and 16-bit modes. It also supports up to 512 Mb per each Chip Select (CS) and up to four chip selects.

This chip has two Ultra Secured Digital Host Controller (uSDHC) modules for the SD/eMMC interface. It provides the interface between the host system and the SD/SDIO/MMC cards.

The LPI2C is a low-power Inter-Integrated Circuit (I^2C) module that supports an efficient interface to an I^2C bus as a master and/or as a slave.

The Watchdog Timer (WDOG) protects against system failures by providing a method to escape from unexpected events or programming errors.

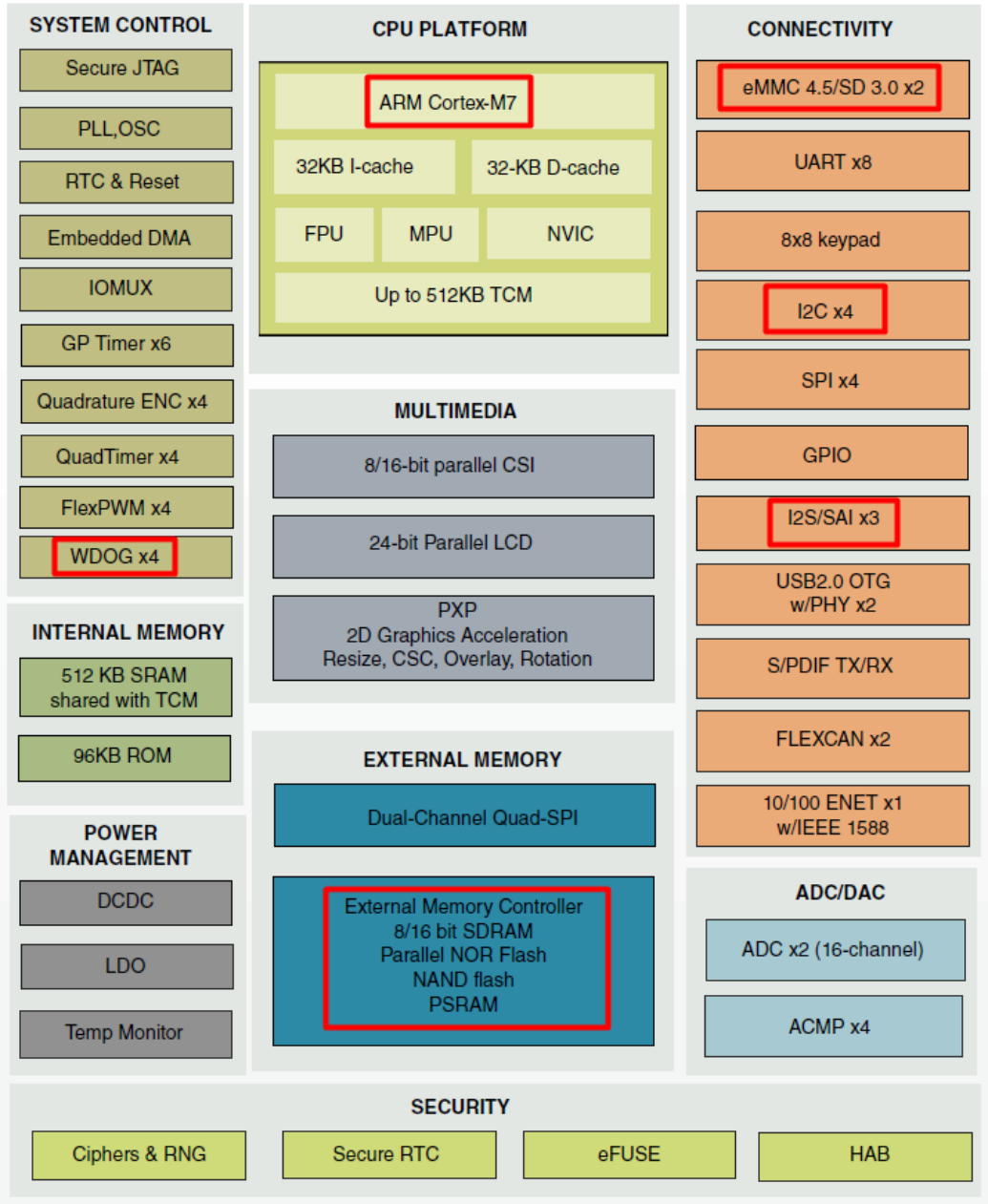

Figure 2. Simplified block diagram

3. Implementation of SAI 7.1 multi-channel feature

This chapter describes the design points of the 7.1 surround sound use case. The multi-channel SAI configuration and the audio stream process are most important.

3.1. 7.1 surround sound use case system

Figure 3 shows the 7.1 surround sound use case system architecture. The core decodes a FLAC file from a micro SD card and transfers the data to SAI1, which has four transmit data lines (one line contains two

channels of data). The CS42448 codec then gets four data lines and clocks to process the data. Finally, the codec transmits eight channels of data to the speakers using DACs.

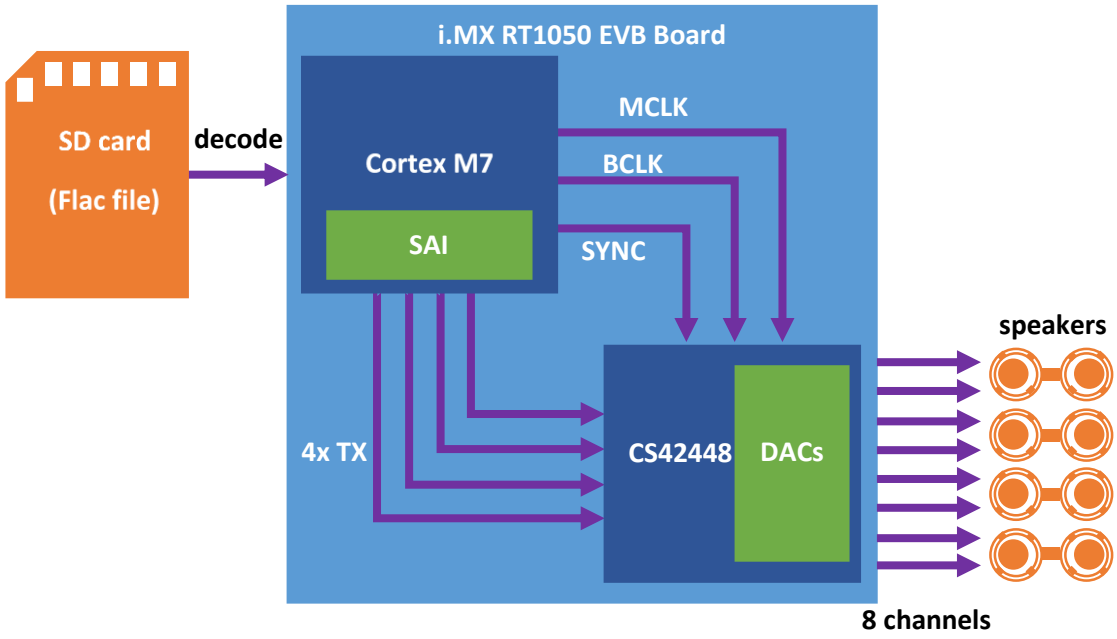

Figure 3. 7.1 surround sound use case system architecture

Figure 4 shows the use case software system flow. The FLAC file decoding is done in a purely software way on the i.MX RT1050. There are these main steps:

1. Board, pin, and clock initialization, such as USDHC and SAI.
2. Watchdog initialization to prevent program errors.
3. Initialization and configuration of the CS42448 codec by LPI2C.
4. Creation and initialization of a new FLAC stream decoder.
5. Detection of the FLAC file's information to configure the SAI TX.
6. Audio playback during decoding.
7. When one audio file finishes, the next file starts (go to step 4). If all files on the SD card are finished, the use case plays the audio from the beginning.

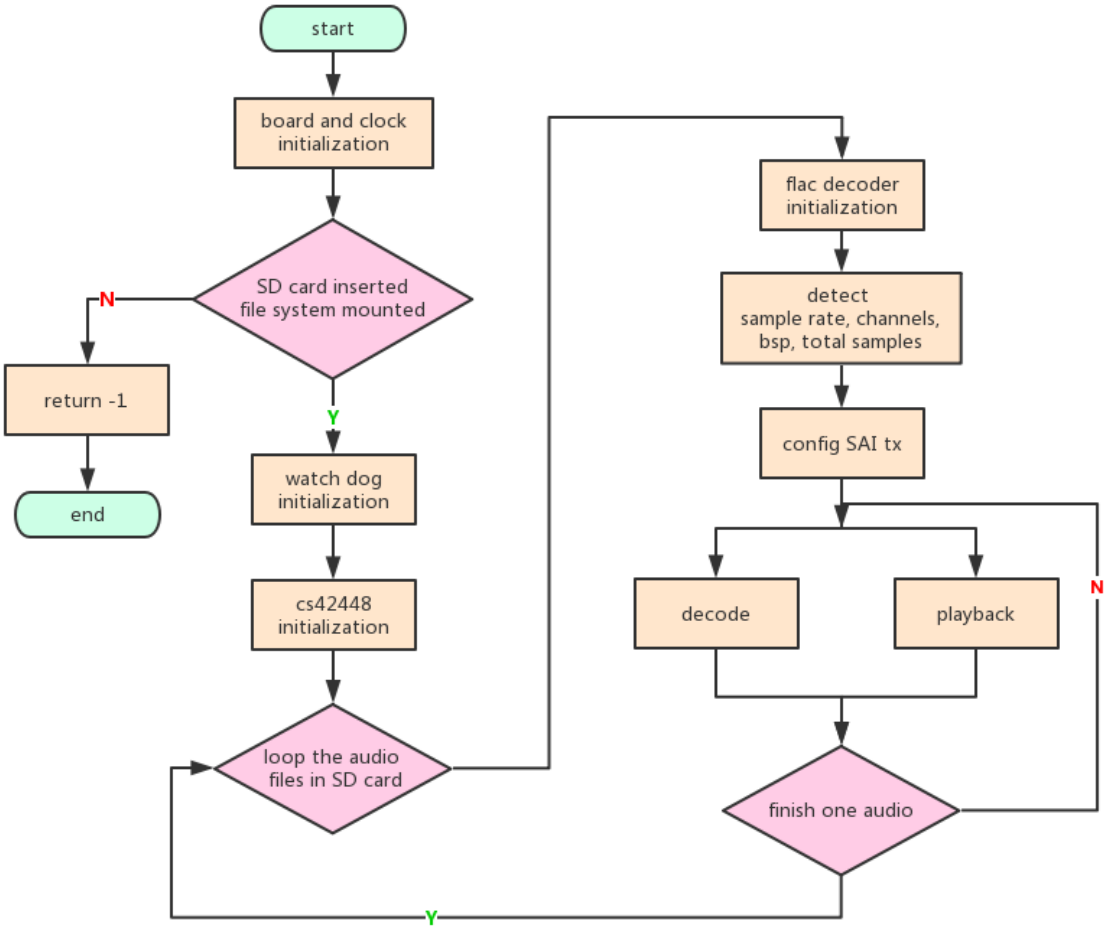

Figure 4. Use case software system flow

3.2. System initialization

The system initialization initializes the pins and clocks.

Set the USDHC1 source to PLL2 PFD0 352 MHz and set the SAI1 source to PLL4 (Audio PLL) 786.43 MHz. The total divider is 1 for the USDHC1 and 60 for the SAI1.

The pins that must be set are listed in Table 1.

Table 1. Pins to set

LPUART	LPI2C	USDHC	SAI
LPUART2_TX LPUART2_RX	LPI2C4_SCL LPI2C4_SDA	USDHC1_CMD USDHC1_CLK USDHC1_DATA0 USDHC1_DATA1 USDHC1_DATA2 USDHC1_DATA3 USDHC1_VSELECT GPIO2_IO28 GPIO1_IO05	SAI1_MCLK SAI1_TX_BCLK SAI1_TX_SYNC SAI1_TX_DATA00 SAI1_TX_DATA01 SAI1_TX_DATA02 SAI1_TX_DATA03

3.3. CS42448 codec configuration

The CS42448 codec provides six multi-bit analog-to-digital converters and eight multi-bit digital-to-analog delta-sigma converters. The codec can operate with either differential or single-ended inputs and outputs.

The codec has a recommended power-up sequence. The core can communicate with the codec using LPI2C and the initialization and configuration of the codec can be applied using LPI2C.

DAC Digital Interface Format should be set as *Left Justified* (same as the SAI configuration). The default *DAC Functional Mode* register is set to auto-detect the sample rates, so you don't have to set the master speed mode (the i.MX RT works in the master mode and the CS42448 is a slave).

The first byte sent to the CS42448 after a start condition consists of a 7-bit chip address field and an R/W bit (high for a read, low for a write). The upper five bits of the 7-bit address field are fixed to 10010. To communicate with the CS42448, the chip address field (which is the first byte sent to the CS42448) should match 10010, followed by the settings of AD1 and AD0. The eighth bit of the address is the R/W bit. If the operation is a write, the next byte is the Memory Address Pointer (MAP) which selects the register to read or write.

The I^2C control port timing is shown in Figure 5.

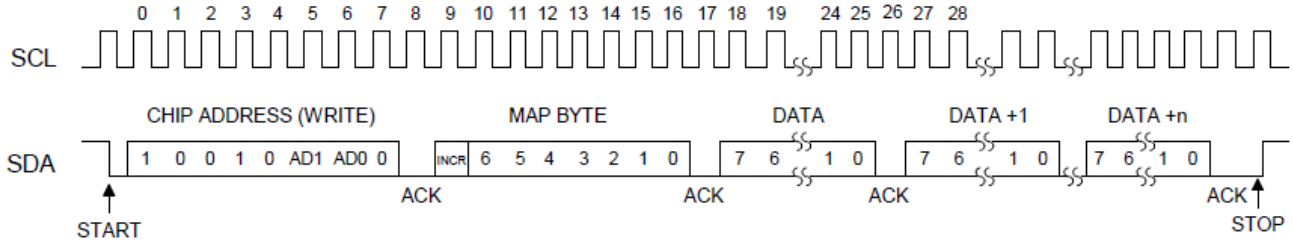

Figure 5. Control port timing

3.4. Multi-channel SAI configuration

In this use case, the transmitter is configured for asynchronous operation and the transmitter uses its own bit clock and frame sync.

The frequency of the sample rate clock (SAI_TX_SYNC) depends on the audio stream sample rate. Most of FLAC 7.1 audio files have 24-bit sampling. Therefore, the bit width is configured to 24 bits.

The code in Example 1 shows a write of audio stream data to the SAI Transmit Data (TDR0~3) registers. Each register corresponds to one SAI TX data line. Writing to these registers when the transmit FIFO is not full pushes the data written into the transmit data FIFO. The writes to these registers when the transmit FIFO is full are ignored. The operation of the audio playback can be written in the *SAI_TxIRQHandler()* function.

The Transmit Channel Enable (TCE) register is set to generate the FIFO request and warning flags at the end of each frame for the transmit operation.

The steps and code section to enable the multi-channel SAI1 are as follows:

1. Get the default configuration (master mode, master clock from the system clock, bit clock using the master clock divider, left-justified format, asynchronous mode, and master clock output enables).
2. Initialize the SAI1 TX peripheral.
3. Configure the SAI1 TX audio format (*mclk* and *bclk* if needed, sample rate, bit width, stereo, data channel, watermark).
4. Enable the SAI1 interrupt.
5. Enable the SAI1 TX interrupt requests for the FIFO warning.
6. Enable the SAI1 TX.
7. Enable the 0~3 channel FIFO.

Example 1.

```
/* Get default configuration */
SAI_TxGetDefaultConfig(&config);
/* Initializes the SAI1 TX peripheral */
SAI_TxInit(DEMO_SAI, &config);

/* Configure the audio format */
format.bitWidth = DEMO_SAI_BITWIDTH;
format.channel = 0U;
format.sampleRate_Hz = sample_rate;
format.masterClockHz = DEMO_SAI_CLK_FREQ;
mclkSourceClockHz = DEMO_SAI_CLK_FREQ;
SAI_TxSetFormat(DEMO_SAI, &format, mclkSourceClockHz, format.masterClockHz);

/*  Enable interrupt and SAI TX*/
EnableIRQ(DEMO_SAI_IRQ);
SAI_TxEnableInterrupts(DEMO_SAI, kSAI_FIFOWarningInterruptEnable |
kSAI_FIFOErrorInterruptEnable);
SAI_TxEnable(DEMO_SAI, true);

/* Enable the corresponding data channel FIFO for transmit operation */
DEMO_SAI->TCR3 &= ~I2S_TCR3_TCE_MASK;
DEMO_SAI->TCR3 |= I2S_TCR3_TCE(0xF);
```

3.5. Audio stream process

Most 7.1 surround sound audio files are 24-bit and the CS42448 codec supports up to 24 bits, so the high 8 bits are useless. The decoding speed is higher than the playback sample rate (most files have 48 kHz). The decoding operation waits for the SAI transmit to finish.

Two buffers are allocated for the “ping-pong” operation to play audio files while decoding. When decoding the audio stream data to buffer0, the data in buffer1 are being sent to the SAI FIFO (playback). The operation is shown in Figure 6. In each block period, the left buffer is buffer0 and the right buffer is buffer1.

The FLAC driver decodes the audio stream block by block (one block contains 4608 frame samples). One audio file contains many blocks so the use case decodes and plays continuously.

When one audio file finishes, the system selects the next file. When all files on the micro SD card have finished playing, the use case loops the files from the beginning.

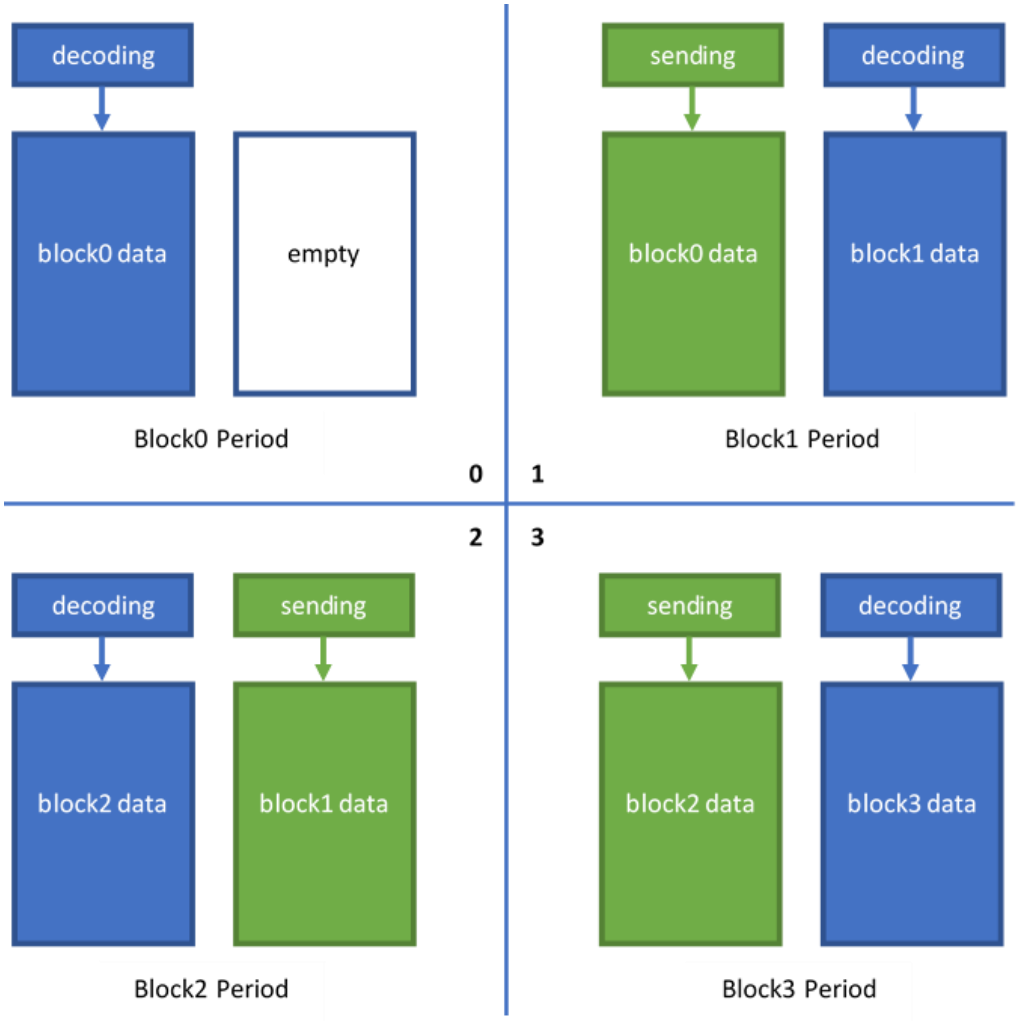

Figure 6. “Ping-pong” operation of playback and decoding

4. Conclusion

To use the SAI multi-channel feature correctly and efficiently, there are several tips:

- Be careful with the system initialization, especially with system clocks and pins.
- The codec's power-up sequence should be set well and the configuration must be aligned with the SAI's configuration.
- The SAI's clocks, TX interrupt, audio format, and TX FIFO read/write are the most important things when using the multi-channel feature. The transmit warning flag is set when the number of entries in any of the enabled transmit FIFOs is empty.
- The audio stream process logic must be designed carefully. Many global variables are used in different functions to prevent a conflict between the decoding and sending of data (playback).

NOTE

Because the Cortex-M7 core can be used in DSP applications for both fixed-point and floating-point operations, the DSP extension is optimized for fixed-point applications. Floating-point operations are accelerated using the optional floating-point unit.

5. Reference

- i.MX RT1050 Processor Reference Manual (Revision: C)
- i.MX RT1050 Validation Board Schematics (Revision: A)
- Kinetis SDK v.2.2 API Reference Manual (In the SDK release package)
- CS42448 Data Sheet
- The DSP capabilities of ARM Cortex-M4 and Cortex-M7 Processors

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN12090
Rev. 0
11/2017