1. Overview

1.1 Overview

This manual explains the Audio device driver in R-Car H3/M3/M3N/E3/D3 Linux.

1.2 Function

This module controls the PCM I/F that is provided by ALSA, and transmits/receives the data to/from the Audio Codec LSI (AK4613) on the R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board.

1.2.1 Connected Device

In the R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board, three SSIs (ten channels) on R-Car H3/M3/M3N is connected to the following peripheral device. And connected to CS2000-CP as Clock Divider. CS2000-CP, AK4613VQ and ADV7482W are connected by I2C.

Table 1-1 SSI Connected device (R-Car H3/M3/M3N)

SSI channel	Peripheral device
SSI0 (output: playback)	CODEC: AK4613VQ
SSI1 (input: capture)	
SSI4 (input)	ADV7482W
SSI2, SSI3, SSI5 - SSI9	None

In the R-CarD3 System Evaluation Board, two SSIs (ten channels) on R-Car D3 is connected to the following peripheral device. And connected to CS2000-CP as Clock Divider. And connected to CS2000-CP as Clock Divider. CS2000-CP and AK4613VQ are connected by I2C.

Table 1-2 SSI Connected device (R-Car D3)

SSI channel	Peripheral device
SSI3 (output: playback)	CODEC: AK4613VQ
SSI4 (input: capture)	

Table 1-3 I2C Connected device

Peripheral device	I2C channel	I2C slave address

CS2000-CP	[R-Car H3/M3/M3N] channel 2 [R-Car D3] channel 0	0x9F for read, 0x9E for write.

AK4613VQ	[R-Car H3/M3/M3N] channel 2 [R-Car D3] channel 0	0x21 for read, 0x20 for write.
[R-Car H3/M3/M3N] ADV7482W [R-Car D3] -	channel 4	0xE1 for read, 0xE0 for write.

In the R-Car E3 System Evaluation Board, four SSIs (ten channels) on R-Car E3 is connected to the following peripheral device. And connected to CS2000-CP as Clock Divider. CS2000-CP, AK4613VQ, ADV7482W and ADV7511W are connected by I2C.

Table 1-4 SSI Connected device (R-Car E3)

SSI channel	Peripheral device
SSI0 (output: playback)	CODEC: AK4613VQ
SSI1 (input: capture)	
SSI3 (input)	ADV7482W
SSI6 (output)	ADV7511W
SSI2, SSI4, SSI5, SSI7 - SSI9	None

Table 1-5 I2C Connected device (R-Car E3)

Peripheral device	I2C channel	I2C slave address
CS2000-CP	channel 3	0x9F for read, 0x9E for write.
AK4613VQ	channel 3	0x21 for read, 0x20 for write.
ADV7482W	channel 0	0xE1 for read, 0xE0 for write.
ADV7511W	channel 0	0x73 for read, 0x72 for write

1.2.2 Clock of connected device

The following figure shows the clock of connected device.

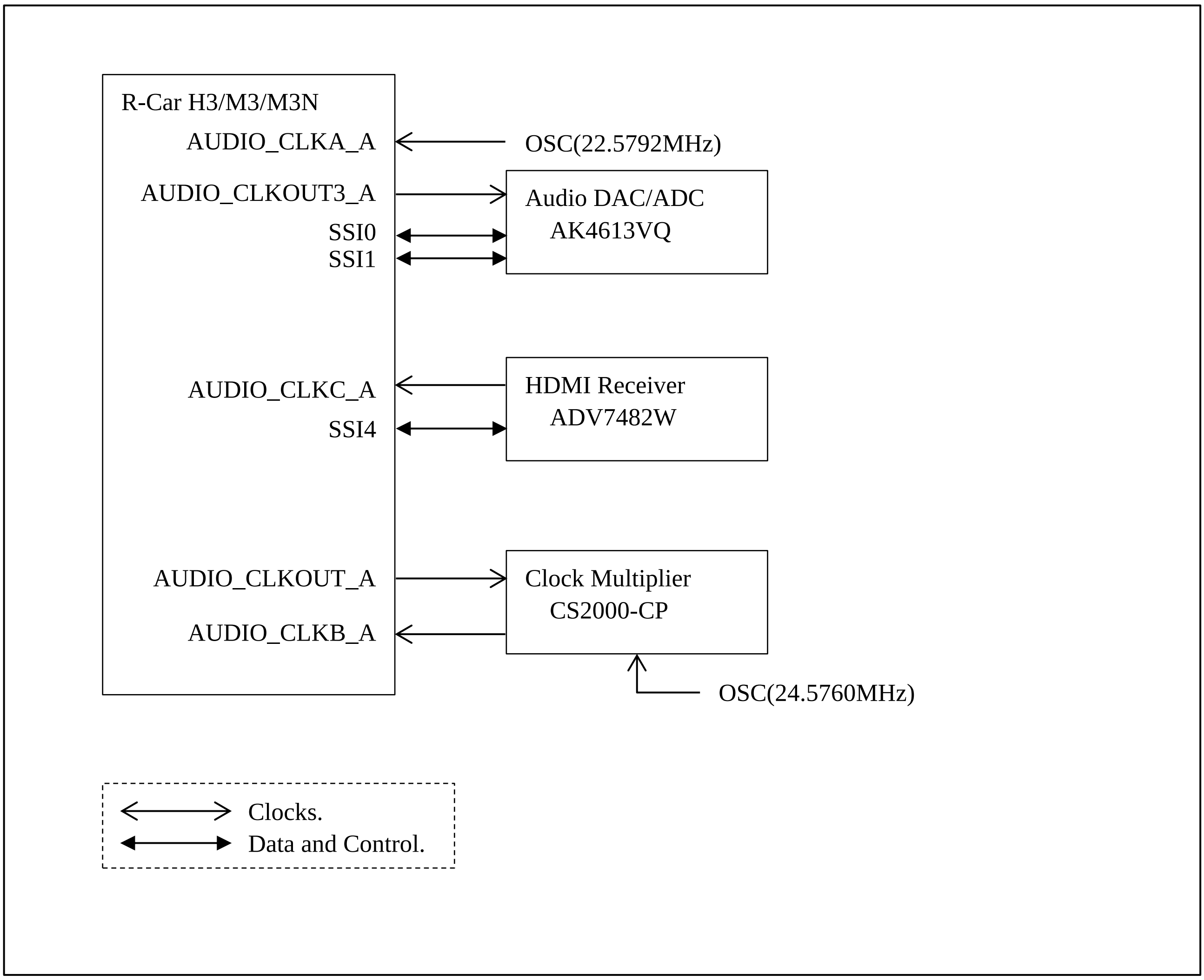


Figure 1-1 Clock of connected device (R-Car H3/M3/M3N)

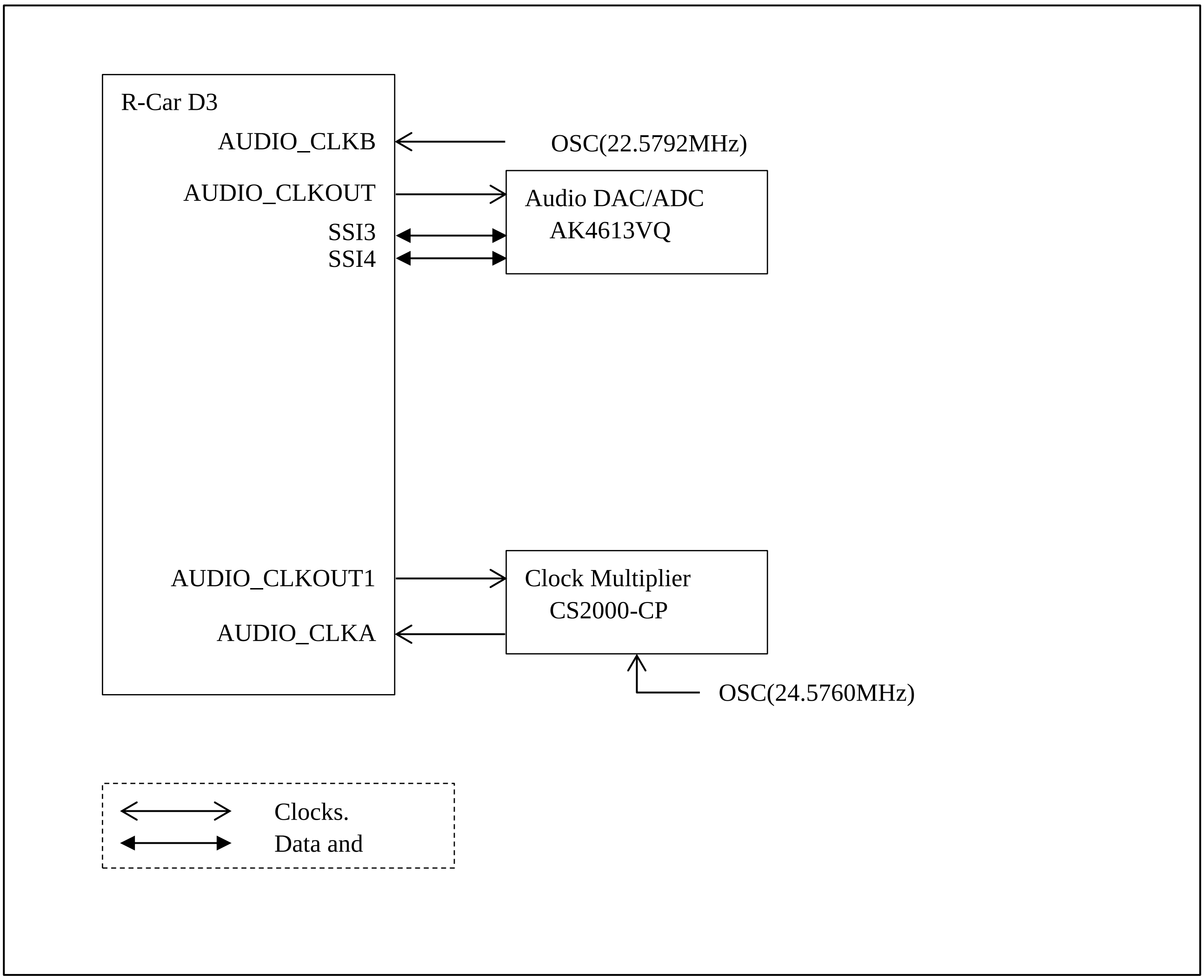


Figure 1.2 Clock of connected device (R-Car D3)

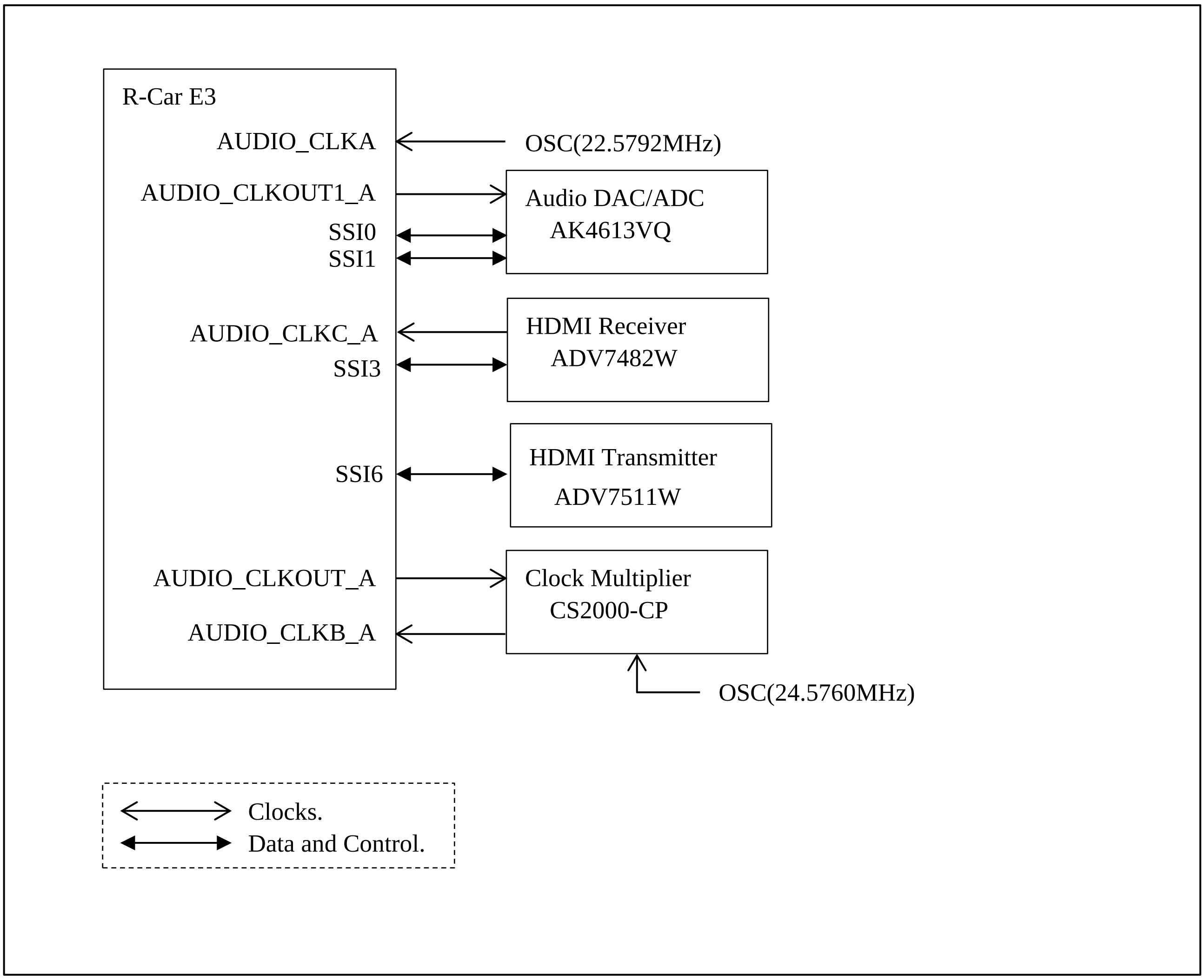


Figure 1-3 Clock of connected device (R-Car E3)

1.2.3 PCM

Support for this module's PCM data depends on the codec support status. At the R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board, 16-bit data is converted to 24-bit data by ALSA library and processed.
As an example of conversion by the ALSA library, the case of specifying the “plughw” option and S16_LE format is applicable. At the R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board, the following command will convert 16-bit data to 24-bit data by the ALSA library.

```
# cat /dev/zero | aplay -D plughw:0,0 -d 30 -f S16_LE -r 48000
```

Also, the supported PCM rate is limited by the clock range that can be supplied. At the R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board, data of 8000 to 24000 Hz and 88200 to 192000 Hz are resampled by ALSA library. Please refer to Table 1-7.

Table 1-6 PCM function

Data format	S16_LE: Little Endian signed 16 bits*1. S24_LE: Little Endian signed 24 bits.
Sampling rate	8000Hz, 11025Hz, 12000Hz, 16000Hz, 22050Hz, 24000Hz, 32000Hz, 44100Hz, 48000Hz, 88200Hz, 96000Hz, 176400Hz, 192000Hz
Audio clock	128fs, 256fs, 384fs, 512fs
Serial data format	I^{2}S (2 channel)
Number of Channels	Monaural*2 / Stereo 2ch / TDM 2,6,8ch.

Notes)
*1: This module supports the 16-bit little endian signed data format, and depending on the target board, it can operate by converting it to 24-bit with the ALSA library.
*2: The monaural output format is converted to 2ch by ALSA library.

Table 1-7 PCM re-sampling rate by ALSA

Sampling rate	Device’s Output	Device’s Input
8000Hz	32000Hz	32000Hz
11025Hz	44100Hz	44100Hz
12000Hz	48000Hz	48000Hz
16000Hz	32000Hz	32000Hz
22050Hz	44100Hz	44100Hz
24000Hz	48000Hz	48000Hz
32000Hz	32000Hz	32000Hz
44100Hz	44100Hz	44100Hz
48000Hz	48000Hz	48000Hz
88200Hz	48000Hz	48000Hz
96000Hz	48000Hz	48000Hz
176400Hz	48000Hz	48000Hz
192000Hz	48000Hz	48000Hz

1.2.4 Audio Codec

This module supports the following function of Audio Codec LSI (AK4613). Audio Codec LSI's default mode is 'slave mode' on R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board.

Table 1-8 AK4613 function

Data format	24bit, left justified (MSB first)	
Sampling rate	32000 - 48000Hz, 64000 - 96000Hz, *1 128000 - 192000Hz *1	
Channel	Output	2 (Codec IC has a 12 channel)
	Input	2 (Codec IC has a 4 channel)
Volume	DAC	
Playback source	Support: LOUT1 / ROUT1	
	Not support: (not connected at R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board) LOUT2 / ROUT2 / LOUT3 / ROUT3 / LOUT4 / ROUT4 / LOUT5 / ROUT5 / LOUT6 / ROUT6	
Capture source	Support: LIN1 / RIN1	
	Not support: (not connected at R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board) LIN2 / RIN2	

Notes) *1: Not support at R-CarH3-SiP/M3-SiP/M3N-SiP/D3/E3 System Evaluation Board. The maximum clock that can be supplied to the AK4613 is up to 12.288 MHz, so the corresponding maximum rate is up to sampling rate 48 kHz. Audio driver supports up to sampling rate 192kHz.

1.2.5 Routing

This module supplies the function of setting the routing for playback/capture path with using Control interface. The routing that this module supports is below. Refer to 4.3 in detail.

Table 1-9 Connected device

Operation	Support route
Playback	Memory -> SSIn
	Memory -> SCU(SRCm) -> SSIn
	Memory -> SCU(SRCm -> DVCl) -> SSIn
	Memory -> SCU(SRCm -> CTUk -> MIXj -> DVCl) -> SSIn
Capture	SSIn -> Memory
	SSIn -> SCU(SRCm) -> Memory
	SSIn -> SCU(SRCm -> DVCl) -> Memory

Notes) [R-Car H3/M3/M3N/E3] SSIn: n=0 to 9, SRCm: m=0 to 9, DVCl: l=0, 1, CTUk: k=0, 1, MIXj: j=0, 1.
[R-Car D3] SSIn: n=3 or 4, SRCm: m=5 or 6, DVCl: l=0, 1, CTUk: k=0, 1, MIXj: j=0, 1.

1.2.6 Sampling Rate Conversion

This module supports the sampling rate conversion function using the SRC. To use it, please set with 'device tree file', or the control interface. If both of which are set, the control interface is given priority.
Initial setting does not change the sampling rate.
For more information, please refer to 4.4.

1.2.7 Rate Continuous

By this setting, ALSA supports all sampling rate. But this driver only supports specific sampling rate, because this feature is disabled. If you would like to use other sampling rate, please refer to 4.5.

1.2.8 Mixing

Mixing two to four sources into one. Ratio is dynamically changeable.

1.2.9 Channel transfer unit

This function provides the channel count conversion. For example, it can convert “5.1ch” to “2ch”.

1.2.10 TDM format

R-Car Series, 3rd Generation supports TDM format (six SSI modules of ten SSI modules can be used for this function). Audio driver supports only TDM extend mode. In the TDM mode, the audio driver does not support the rate converting function

Table 1-10 TDM mode support status

TDM mode	Audio driver support status	
	Output	**Input**
TDM format “Basic Configuration”	not support	not support
TDM-16ch mode	not support	not support
TDM extend mode	support*	support*
TDM split mode	not support	not support
TDM ex-split mode	not support	not support

* Audio driver supports TDM, but R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board does not support this function

1.2.11 Ramp

This module supports Ramp function at MIX, and DVC. Ramp function is a function to gradually change to the specified volume.

1.3 Reference

1.3.1 Standards

The following table shows the standard that this module corresponds.

Table 1-11 Standard

Number	Issue	Title	Edition	Date
-	-	ALSA Sound ver.1.0.29	-	-

1.3.2 Related Documents

The following table shows the document related to this module.

Table 1-12 Related document

Number	Issue	Title	Edition	Data
-	Renesas Electronics	R-Car Series, 3rd Generation User's Manual: Hardware	Rev.2.20	Jun. 30, 2020
-	Renesas Electronics	R-CarH3-SiP System Evaluation Board Salvator-X Hardware Manual RTP0RC7795SIPB0011S	Rev.1.09	May. 11, 2017
-	Renesas Electronics	R-CarM3-SiP System Evaluation Board Salvator-X Hardware Manual RTP0RC7796SIPB0011S	Rev.0.04	Oct. 3, 2016
-	Renesas Electronics	R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS Hardware Manual	Rev.2.04	Jul. 17, 2018
-	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu Hardware Manual RTP0RC77990SEB0010S	Rev.0.03	Apr. 11, 2018
-	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu-4D (E3 board 4xDRAM) Hardware Manual	Rev.1.01	Jul. 19, 2018
	Renesas Electronics	R-CarD3 System Evaluation Board Draak Hardware Manual Hardware Manual RTP0RC77995SEB0010S	Rev.1.20	Jul. 25, 2017

Table 1-4 Related document

Number	Issue	Title	Edition	Date
MS1052-J-05	ASAHI KASEI	AK4613 4/12-Channel Audio CODEC	05	2015.6.11

1.4 Restrictions

There is no reference document on standards.

1.5 Notice

- The sampling rate uses the same setting in input/output. When playback/capture executes at the same time, the sampling rate should be specified to the same value.

- The combination of this audio driver and AK 4613 has the following notice.

At the time of initial playback, the volume setting is not reflected, and the loud sound is output. It also occurs at the time of the first playback that suspended and resumed. Also at the first capture or resumed, it will be recorded with loud sounds.

The volume setting of AK4613 codec must be done in LRCLK input state. However, since the current ALSA framework sets the codec before SSI start (LRCLK output), the initial volume setting will not be reflected.

As an ALSA framework, SSI and codec are made independently. When setting codec, it is not supported to control SSI, which is another module, to output LRCLK because it requires special remodeling to the framework.

Also, once the DAC/ADC turns off, it will be in the initial state, so you will need to set it again.

<Workaround (a)>

The following is example for avoiding the problem at playback.

1. Prepare the silent sound wav file as "silence.wav".
2. Play "silence.wav" for a short time.

   # aplay -d 1 silence.wav

After that, execute playback of the target wav file.

The following is example for avoiding the problem at capture.

1. Capture "dummy.wav" for a short time.

   # arecord -d 1 -f cd dummy.wav

After that, execute capture of the target wav file.

<Workaround (b)>

1. Change framework. It modifies “sound/soc/soc-pcm.c”. This changes the control order of clock supply.

```
static int soc_pcm_trigger(struct snd_pcm_substream *substream, int cmd)
{
        struct snd_soc_pcm_runtime *rtd = substream->private_data;
        struct snd_soc_component *component;
        struct snd_soc_rtdcom_list *rtdcom;
        struct snd_soc_dai *cpu_dai = rtd->cpu_dai;
        struct snd_soc_dai *codec_dai;
        int i, ret;

        /* Add */
        ret = snd_soc_dai_trigger(cpu_dai, substream, cmd);
        if (ret < 0)
                return ret;

        for_each_rtd_codec_dai(rtd, i, codec_dai) {
                ret = snd_soc_dai_trigger(codec_dai, substream, cmd);
                if (ret < 0)
                        return ret;
        }

        for_each_rtdcom(rtd, rtdcom) {
                component = rtdcom->component;

                ret = snd_soc_component_trigger(component, substream, cmd);
                if (ret < 0)
                        return ret;
        }

        /* Remove
        ret = snd_soc_dai_trigger(cpu_dai, substream, cmd);
        if (ret < 0)
                return ret;
        */

        if (rtd->dai_link->ops->trigger) {
                ret = rtd->dai_link->ops->trigger(substream, cmd);
                if (ret < 0)
                        return ret;
        }

        return 0;
}
```

2. Add start delay. It modifies “sound/soc/sh/rcar/dma.c”. This change waits for audio input/output start until AK 4613's volume transition time is completed.

```
/* Add */
static unsigned int start_delay = 230;
module_param(start_delay, uint, 0644);
MODULE_PARM_DESC(start_delay, "PCM stream start delay time (msecs)");

struct rsnd_dmaen {
        struct dma_chan              *chan;
        dma_cookie_t                cookie;
        unsigned int                dma_len;
        /* Add */
        struct delayed_work           work;
};

. . . .

static int rsnd_dmaen_cleanup(struct rsnd_mod *mod,
                              struct rsnd_dai_stream *io,
                              struct rsnd_priv *priv)
{
        struct rsnd_dma *dma = rsnd_mod_to_dma(mod);
        struct rsnd_dmaen *dmaen = rsnd_dma_to_dmaen(dma);

        /* Add */
        cancel_delayed_work_sync(&dmaen->work);

        /*
         * DMAEngine release uses mutex lock.
         * Thus, it shouldn't be called under spinlock.
         * Let's call it under nolock_start
         */
        if (dmaen->chan)
                dma_release_channel(dmaen->chan);

        dmaen->chan = NULL;

        return 0;
}

/* Add */
static void rsnd_dma_work(struct work_struct *work)
{
        struct rsnd_dmaen *dmaen = container_of(work, struct rsnd_dmaen,
work.work);

        dma_async_issue_pending(dmaen->chan);
}

static int rsnd_dmaen_prepare(struct rsnd_mod *mod,
                        struct rsnd_dai_stream *io,
                        struct rsnd_priv *priv)
{
        struct rsnd_dma *dma = rsnd_mod_to_dma(mod);
        struct rsnd_dmaen *dmaen = rsnd_dma_to_dmaen(dma);
        struct device *dev = rsnd_priv_to_dev(priv);

        /* maybe suspended */
        if (dmaen->chan)
```

```
                return 0;

        /*
         * DMAEngine request uses mutex lock.
         * Thus, it shouldn't be called under spinlock.
         * Let's call it under prepare
         */
        dmaen->chan = rsnd_dmaen_request_channel(io,
                                          dma->mod_from,
                                          dma->mod_to);
        if (IS_ERR_OR_NULL(dmaen->chan)) {
                dmaen->chan = NULL;
                dev_err(dev, "can't get dma channel¥n");
                return -EIO;
        }

        /* Add */
        INIT_DELAYED_WORK(&dmaen->work, rsnd_dma_work);

        return 0;
}

static int rsnd_dmaen_start(struct rsnd_mod *mod,
                       struct rsnd_dai_stream *io,
                       struct rsnd_priv *priv)
{
. . . .
        desc->callback          = rsnd_dmaen_complete;
        desc->callback_param     = rsnd_mod_get(dma);

        dmaen->dma_len          = snd_pcm_lib_buffer_bytes(substream);

        dmaen->cookie = dmaengine_submit(desc);
        if (dmaen->cookie < 0) {
                dev_err(dev, "dmaengine_submit() fail¥n");
                return -EIO;
        }

        /* Remove
        dma_async_issue_pending(dmaen->chan);
        */
        /* Add */
        schedule_delayed_work(&dmaen->work, msecs_to_jiffies(start_delay));

        return 0;
}
```

2. Terminology

The following table shows the terminology related to this module.

Table 2-1 Terminology

Terms	Explanation
ADG	Audio clock generator
ALSA	Advanced Linux Sound Architecture The term on ALSA is provided by the ALSA site. http://www.alsa-project.org/
ASoC	ALSA for SoC
CTU	Channel transfer unit
DAI	Digital Audio Interfaces
DMAC	Direct Memory Access Controller
DVC	Digital volume and mute function
I2C	Inter-Integrated Circuit
MIX	Mixing unit
PCM	Pulse Code Modulation
SCU	Sampling rate converter unit SCU is R-Car H3/M3/M3N/E3/D3 unit, includes SRC/CTU/MIX/DVC.
SRC	Sampling rate conversion
SSIU	Serial sound interface unit SSIU is R-Car H3/M3/M3N/E3/D3 unit, provides the function of SSI (Serial sound interface).
TDM	Time Division Multiplexing.

3. Operating Environment

3.1 Hardware Environment

The following table lists the hardware needed to use this module.

Table 3-1 Hardware Environment

Name	Version	Manufacture
R-CarH3-SiP System Evaluation Board Salvator-X	-	Renesas Electronics
R-CarM3-SiP System Evaluation Board Salvator-X	-	Renesas Electronics
R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS	-	Renesas Electronics
R-CarE3 System Evaluation Board Ebisu	-	Renesas Electronics
R-CarE3 System Evaluation Board Ebisu-4D	-	Renesas Electronics
R-CarD3 System Evaluation Board Draak	-	Renesas Electronics

3.2 Module Configuration

The following figure shows the configuration of this module. Audio driver controls Audio-DMAC-pp at direct for peripheral to peripheral transfer.

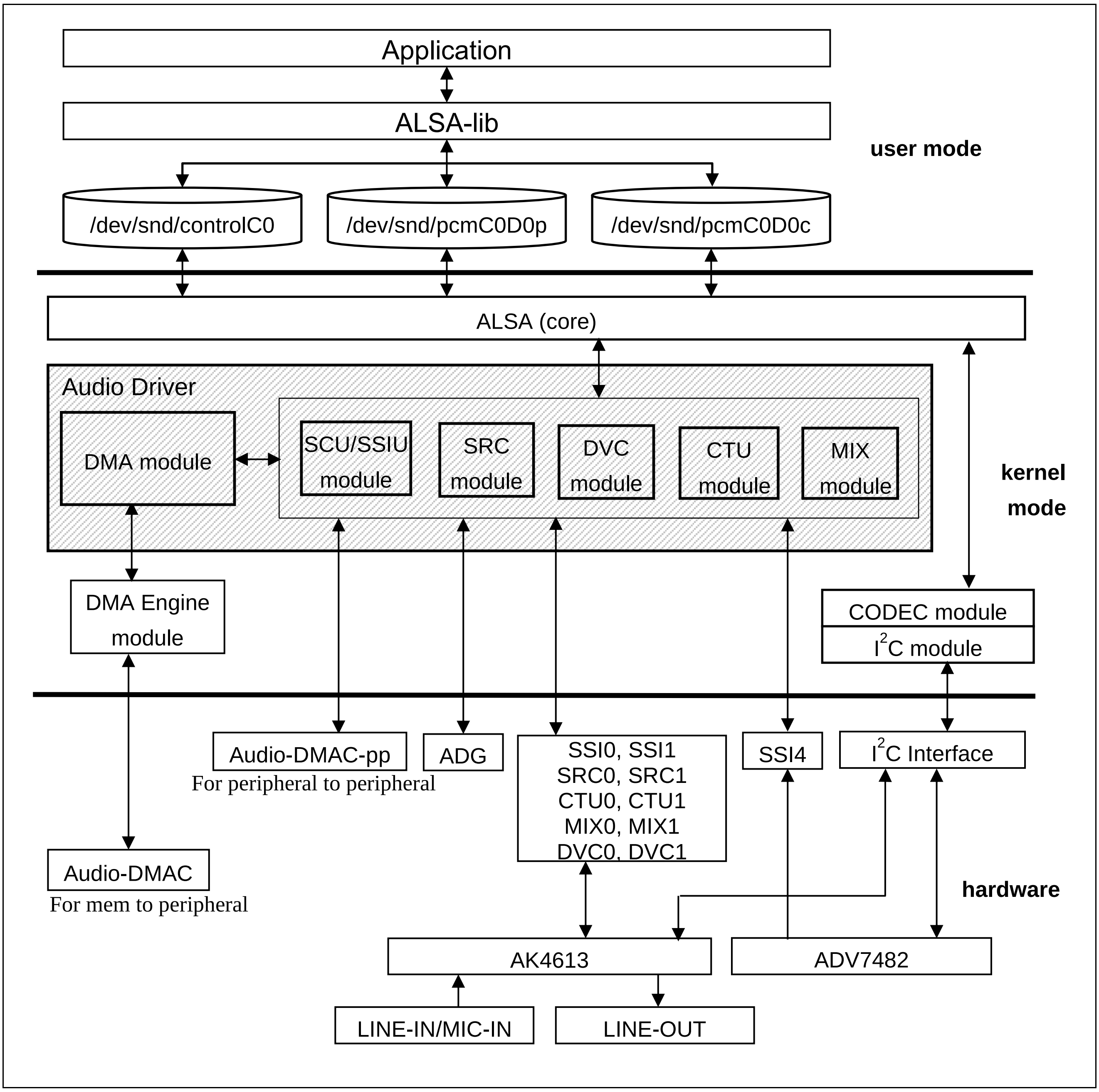


Figure 3-1 Audio Driver configuration (R-Car H3/M3/M3N)

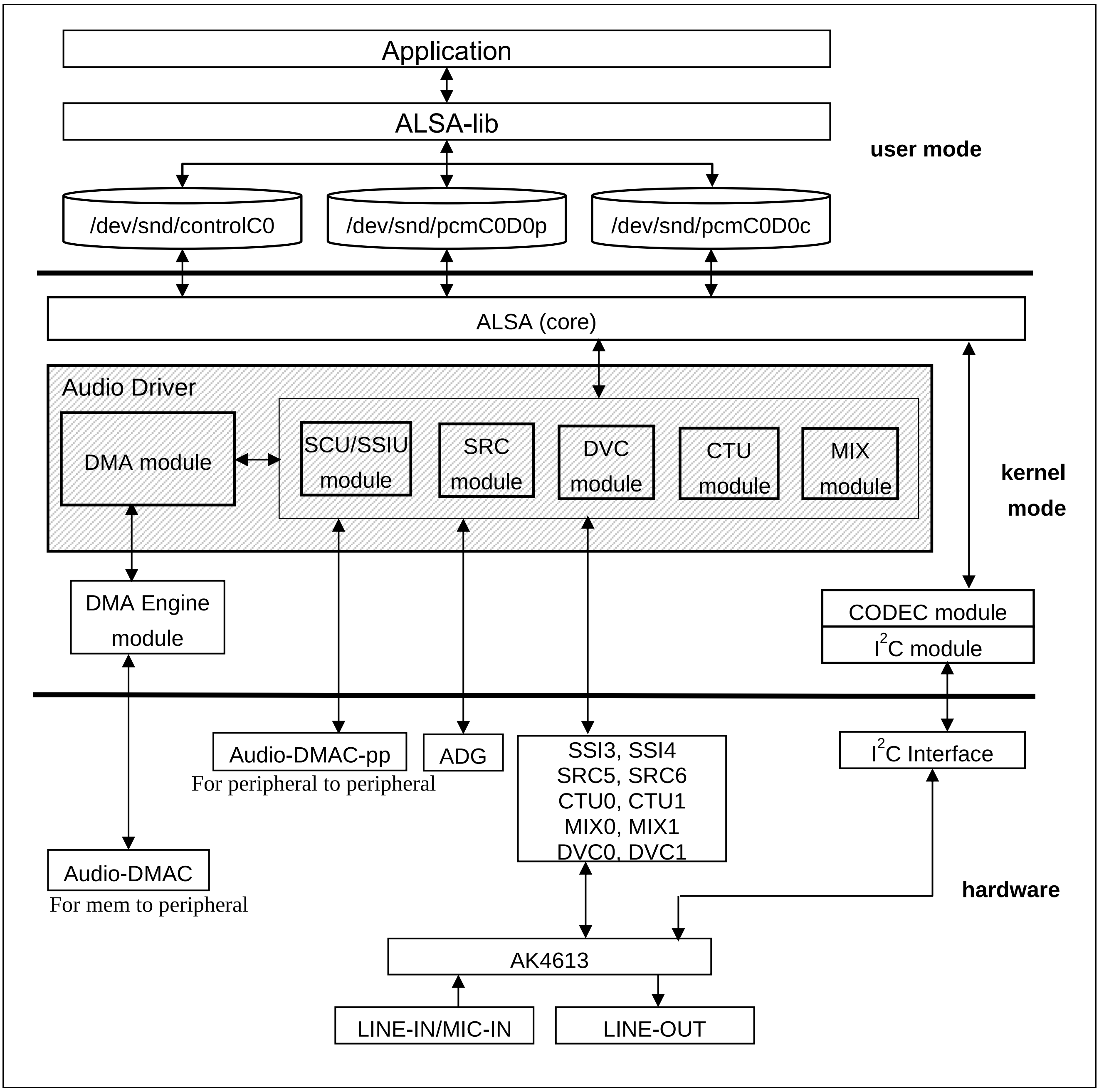


Figure 3.2 Audio Driver configuration(R-Car D3)

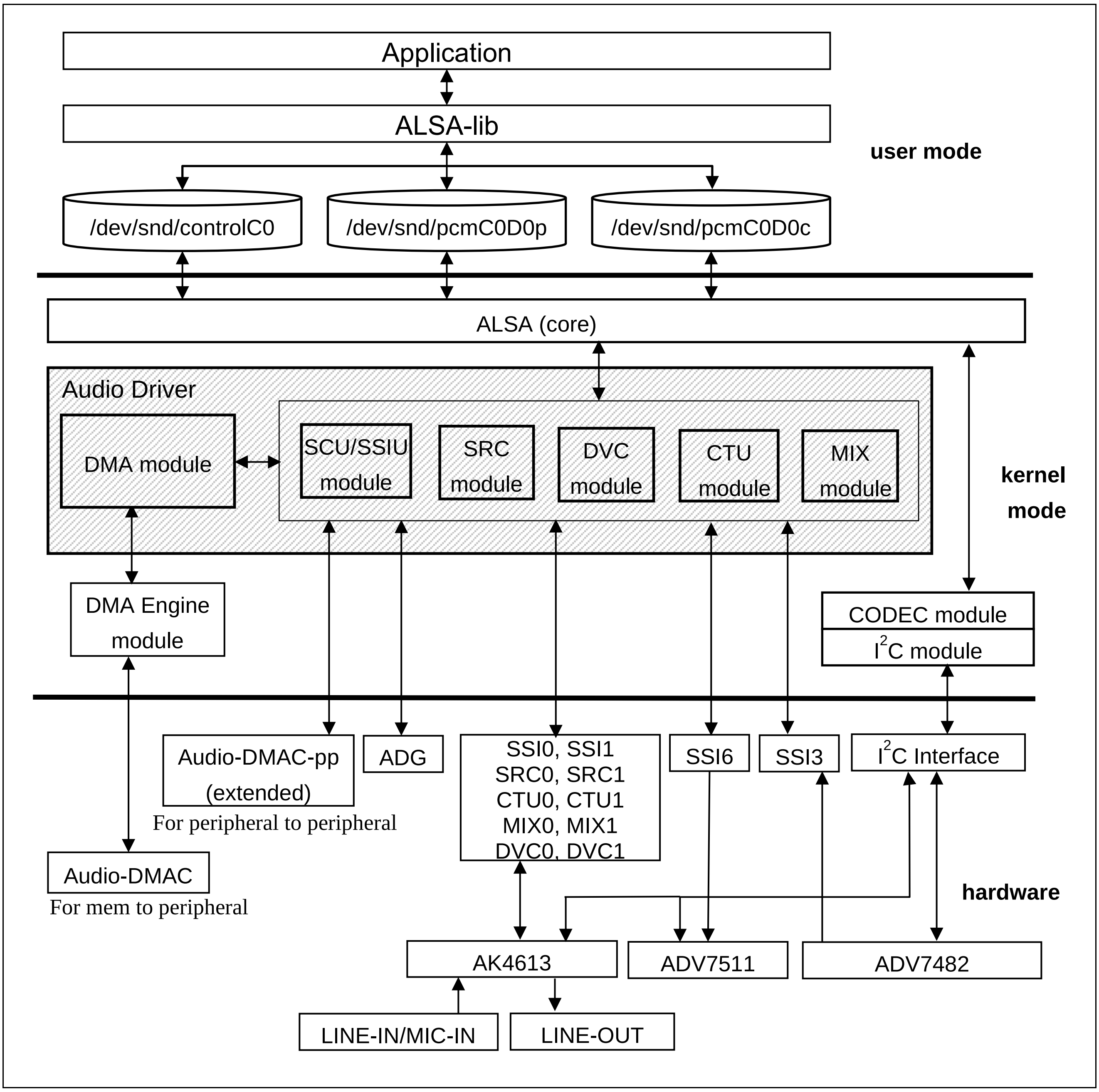


Figure 3-3 Audio Driver configuration (R-Car E3)

3.3 State Transition Diagram

There is no state transition diagram for this module.

4. External Interface

This module is based on ALSA sound. This manual describes only a peculiar function.

4.1 Device

A device is expressed as follows by the ALSA interface.

Table 4-1 ALSA Device Interface

ALSA interface	Device node
Information Interface	/proc/asound
Control Interface	/dev/snd/controlCX
PCM Interface	/dev/snd/pcmCXDX
Timer Interface	/dev/snd/timer

String of device node format "X" indicates a numeric character.

4.2 Device Node

The following table shows the device node of this module. This case is BSP standard settings.

Table 4-2 Device node

Device node	Major number	Minor number
/dev/snd/controlC0	116	0
/dev/snd/pcmC0D0c	116	24
/dev/snd/pcmC0D0p	116	16
/dev/snd/timer	116	33

4.3 Setting route

This module supplies the function of statically setting the routing for playback/capture path.

They are defined by “rcar_sound,dai” in device tree. Please refer to Table 4-3.

Table 4-3 Device tree files

Target CPU	Target board	Device tree files
R-Car H3	Salvator-X/XS	arch/arm64/boot/dts/renesas/r8a77950-salvator-x.dts, arch/arm64/boot/dts/renesas/r8a77951-salvator-x.dts, arch/arm64/boot/dts/renesas/r8a779m1-salvator-x.dts, arch/arm64/boot/dts/renesas/r8a77951-salvator-xs.dts, arch/arm64/boot/dts/renesas/r8a779m1-salvator-xs.dts. each include below: arch/arm64/boot/dts/renesas/r8a77950.dtsi, arch/arm64/boot/dts/renesas/r8a77951.dtsi, arch/arm64/boot/dts/renesas/r8a779m1.dtsi, arch/arm64/boot/dts/renesas/salvator-common.dtsi.
R-Car M3	Salvator-X/XS	arch/arm64/boot/dts/renesas/r8a77960-salvator-x.dts, arch/arm64/boot/dts/renesas/r8a77960-salvator-xs.dts, arch/arm64/boot/dts/renesas/r8a77961-salvator-xs.dts, arch/arm64/boot/dts/renesas/r8a779m3-salvator-xs.dts. each include below: arch/arm64/boot/dts/renesas/r8a77960.dtsi, arch/arm64/boot/dts/renesas/r8a77961.dtsi, arch/arm64/boot/dts/renesas/r8a779m3.dtsi, arch/arm64/boot/dts/renesas/salvator-common.dtsi.
R-Car M3N	Salvator-X/XS	arch/arm64/boot/dts/renesas/r8a77965-salvator-x.dts, arch/arm64/boot/dts/renesas/r8a779m5-salvator-x.dts, arch/arm64/boot/dts/renesas/r8a77965-salvator-xs.dts, arch/arm64/boot/dts/renesas/r8a779m5-salvator-xs.dts. each include below: arch/arm64/boot/dts/renesas/r8a77965.dtsi, arch/arm64/boot/dts/renesas/r8a779m5.dtsi, arch/arm64/boot/dts/renesas/salvator-common.dtsi.
R-Car E3	Ebisu	arch/arm64/boot/dts/renesas/r8a77990-ebisu.dts, arch/arm64/boot/dts/renesas/r8a77990-ebisu-4d.dts, arch/arm64/boot/dts/renesas/r8a77990-es10-ebisu.dts, arch/arm64/boot/dts/renesas/r8a77990-es10-ebisu-4d.dts. include below: arch/arm64/boot/dts/renesas/r8a77990.dtsi, arch/arm64/boot/dts/renesas/r8a77990-es10.dtsi.
R-Car D3	Draak	arch/arm64/boot/dts/renesas/r8a77995-draak.dts include below: arch/arm64/boot/dts/renesas/r8a77995.dtsi.

4.3.1 Data transmission paths

Data transmission paths in the Audio module are shown in Figure 4-1

The setting use below:

&src0	&ctu00	&mix0	&dvc0	&ssi0
&src1	&ctu01	&mix1	&dvc1	&ssi1
&src2	&ctu02			&ssi2
&src3	&ctu03			&ssi3
&src4				&ssi4
&src5	&ctu10			&ssi5
&src6	&ctu11			&ssi6
&src7	&ctu12			&ssi7
&src8	&ctu13			&ssi8
&src9				&ssi9

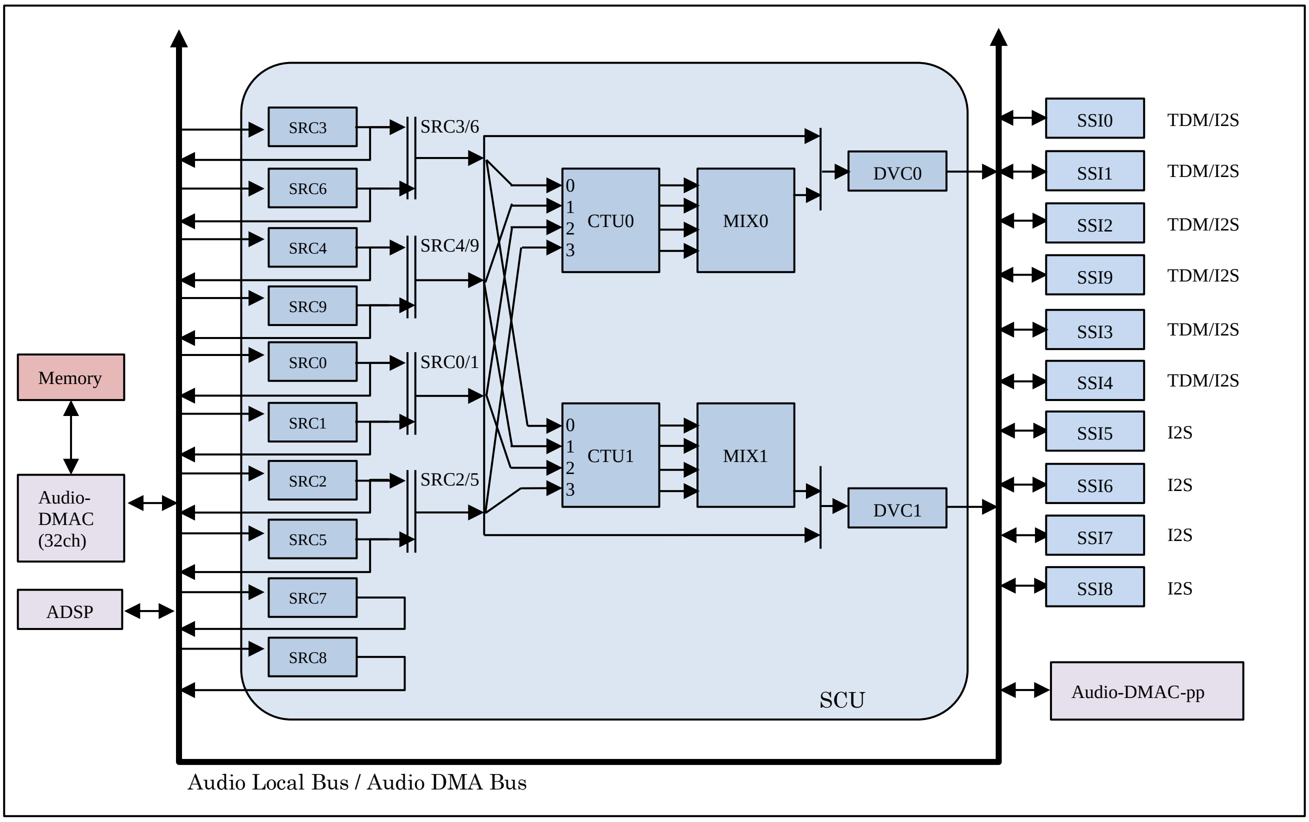


Figure 4-1 Data transmission paths (R-Car H3/M3/M3N)

Data transmission paths in the Audio module are shown in Figure 4.2(R-Car D3).

The setting use below:

&src5	&ctu00	&mix0	&dvc0	&ssi3
&src6	&ctu03	&mix1	&dvc1	&ssi4
	&ctu10			
	&ctu13			

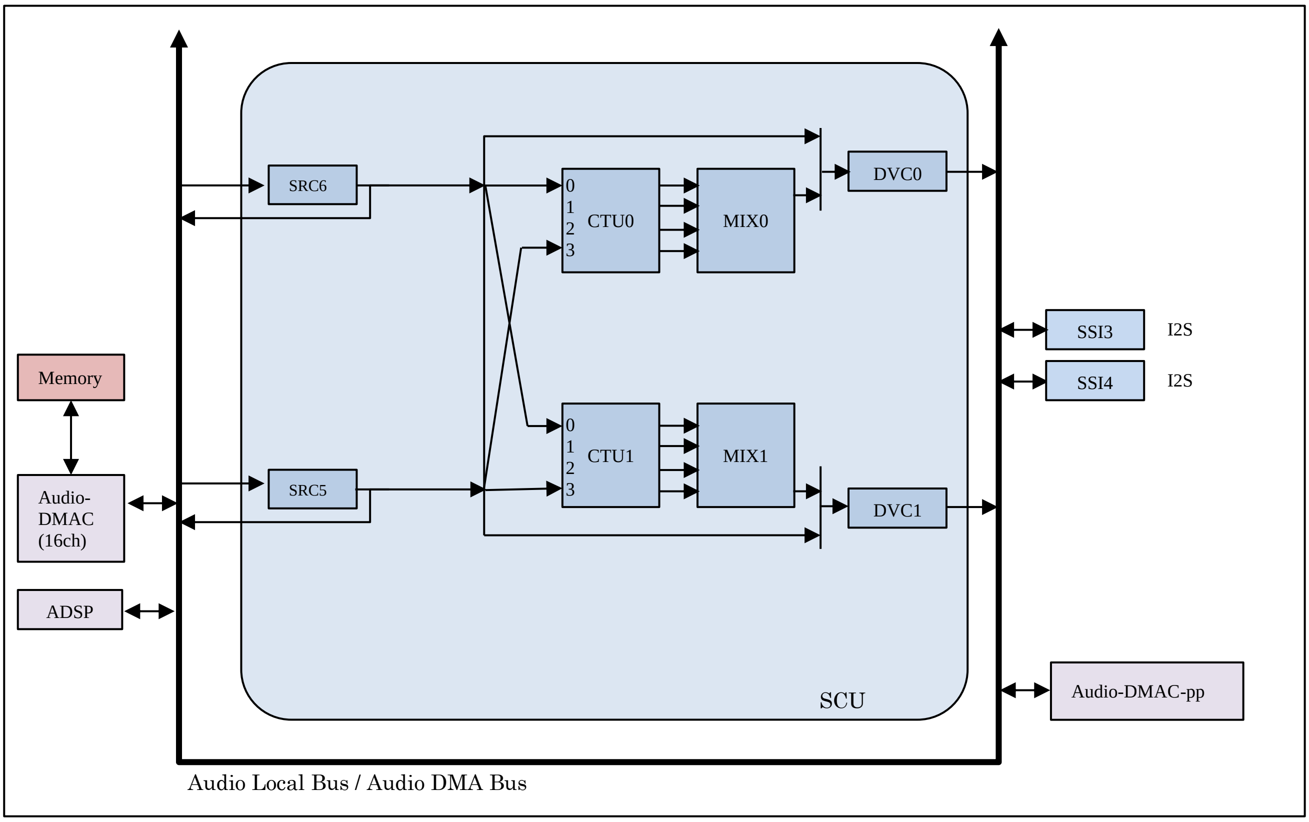


Figure 4.2. Data transmission paths(R-Car D3)

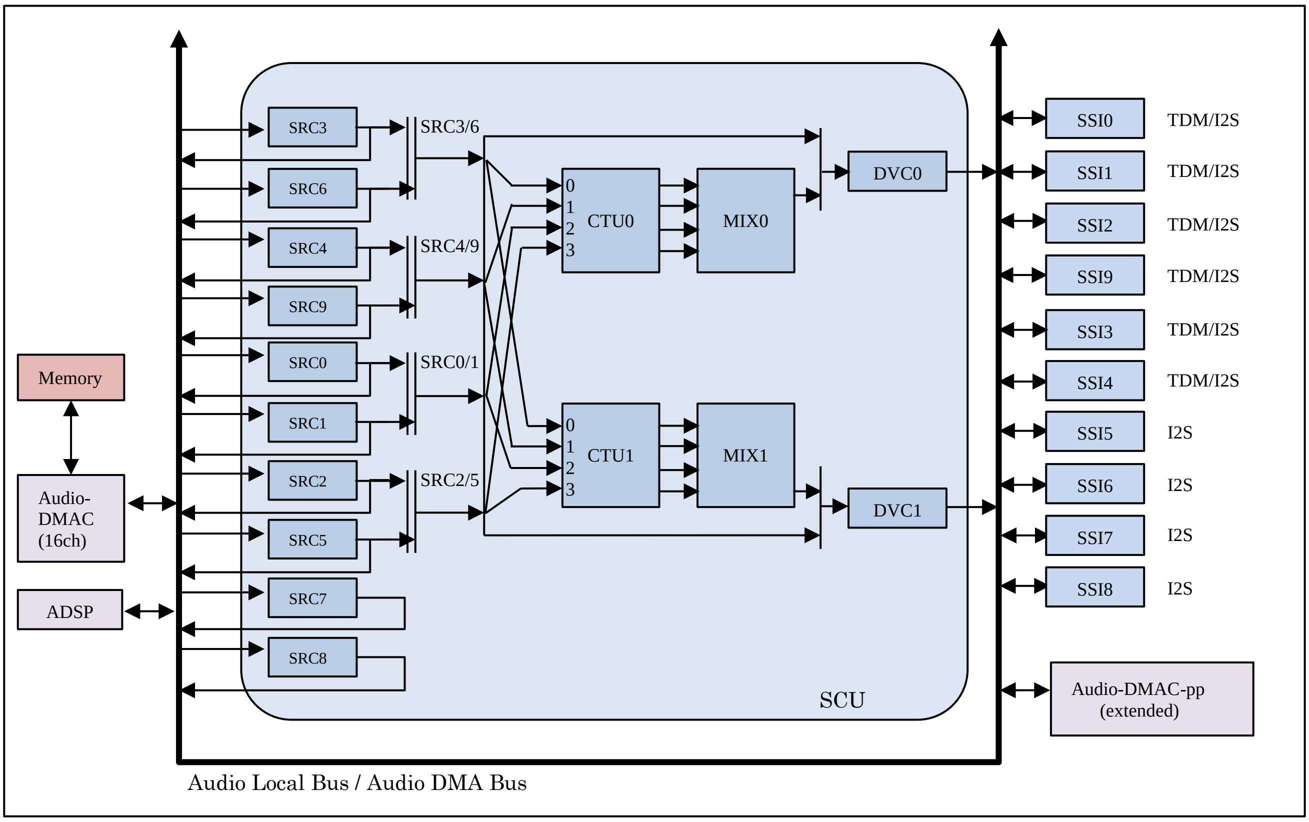


Figure 4-3 Data transmission paths (R-Car E3)

4.3.2 Setting route for playback

[R-Car H3/M3/M3N/E3] (1) - (4), [R-Car D3] (5) - (8)

(1) **Setting case of “Memory -> SSI0 -> CODEC”**

Route path shows the case of "Memory-> SSI0-> CODEC".

Transfer settings of audio DMAC is set to transfer from the memory to the “SSI00” at the driver. Audio DMAC-pp is no use at this case.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

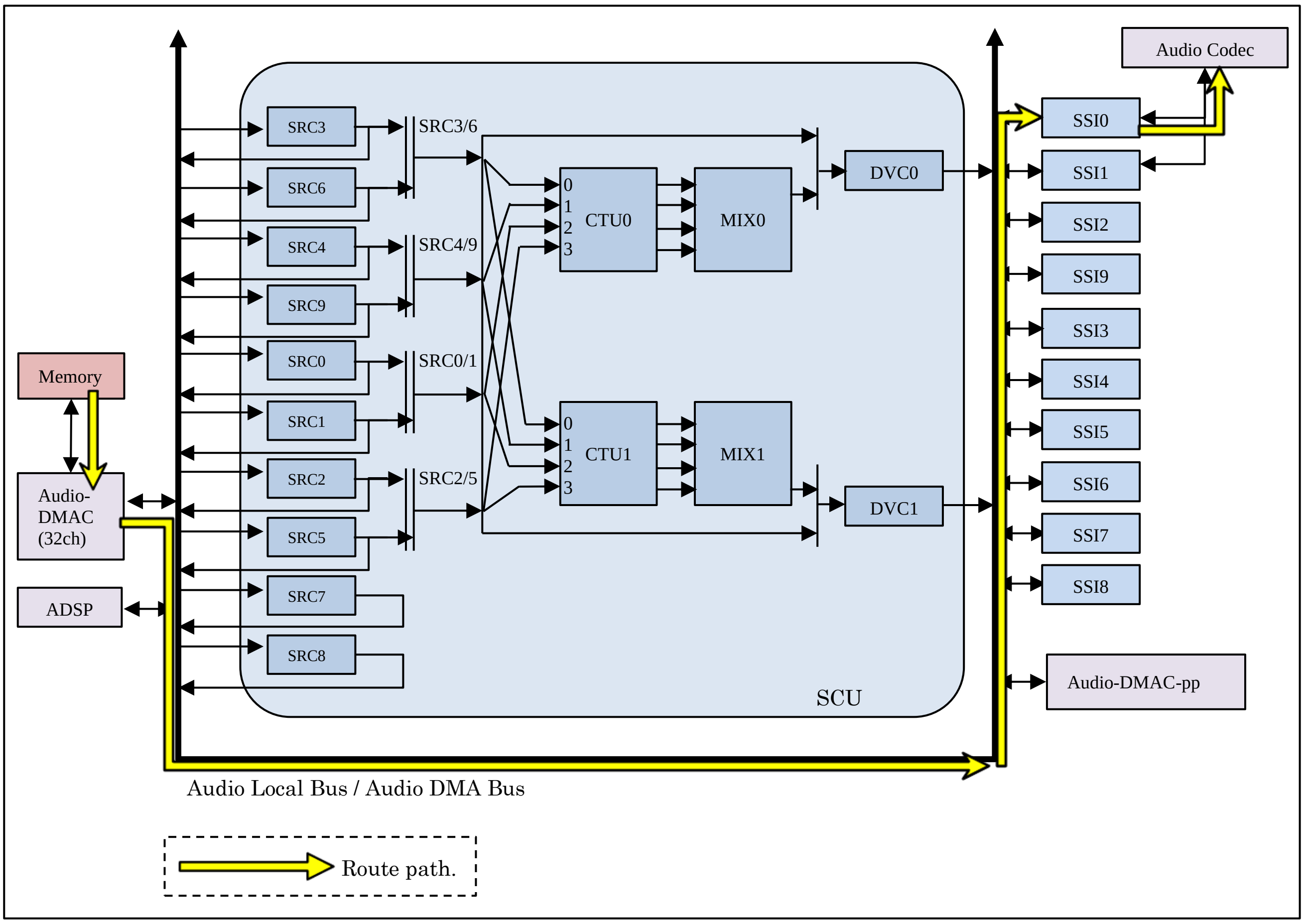


Figure 4-4 Memory->SSI0->CODEC data path (R-Car H3/M3/M3N)

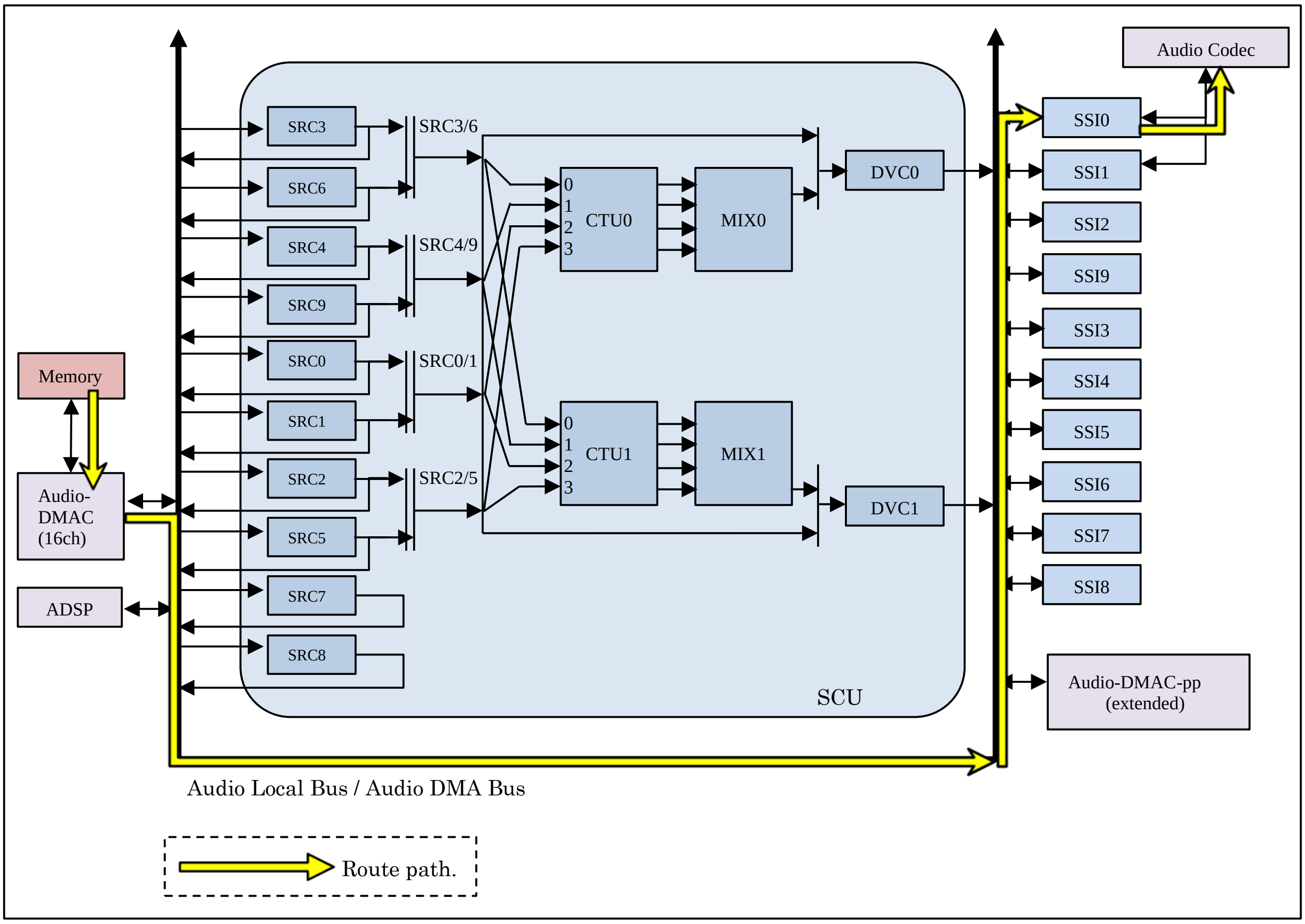


Figure 4-5 Memory->SSI0->CODEC data path (R-Car E3)

This route case's description example is shown below.

```
sound_card: sound {
        compatible = "audio-graph-card";

        label = "rcar-sound";

        dais = <&rsnd_port0>;
};


&rcar_sound {

       ...

        ports {
                rsnd_port0: port@0 {
                        rsnd_endpoint0: endpoint {
                                remote-endpoint = <&ak4613_endpoint>;

                                dai-format = "left_j";
                                bitclock-master = <&rsnd_endpoint0>;
                                frame-master = <&rsnd_endpoint0>;

                                playback = <&ssi0>;
                                capture  = <&ssi1>;
                        };
                };
        };
};
```

Figure 4-6 setting for Memory->SSI0->CODEC

(2) Setting case of “Memory -> SRC0 -> SSI0 -> CODEC

Route path shows the case of "Memory-> SRC0 -> SSI0-> CODEC".

Transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI0” at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SCU_SRCO0” to the “SSI00” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

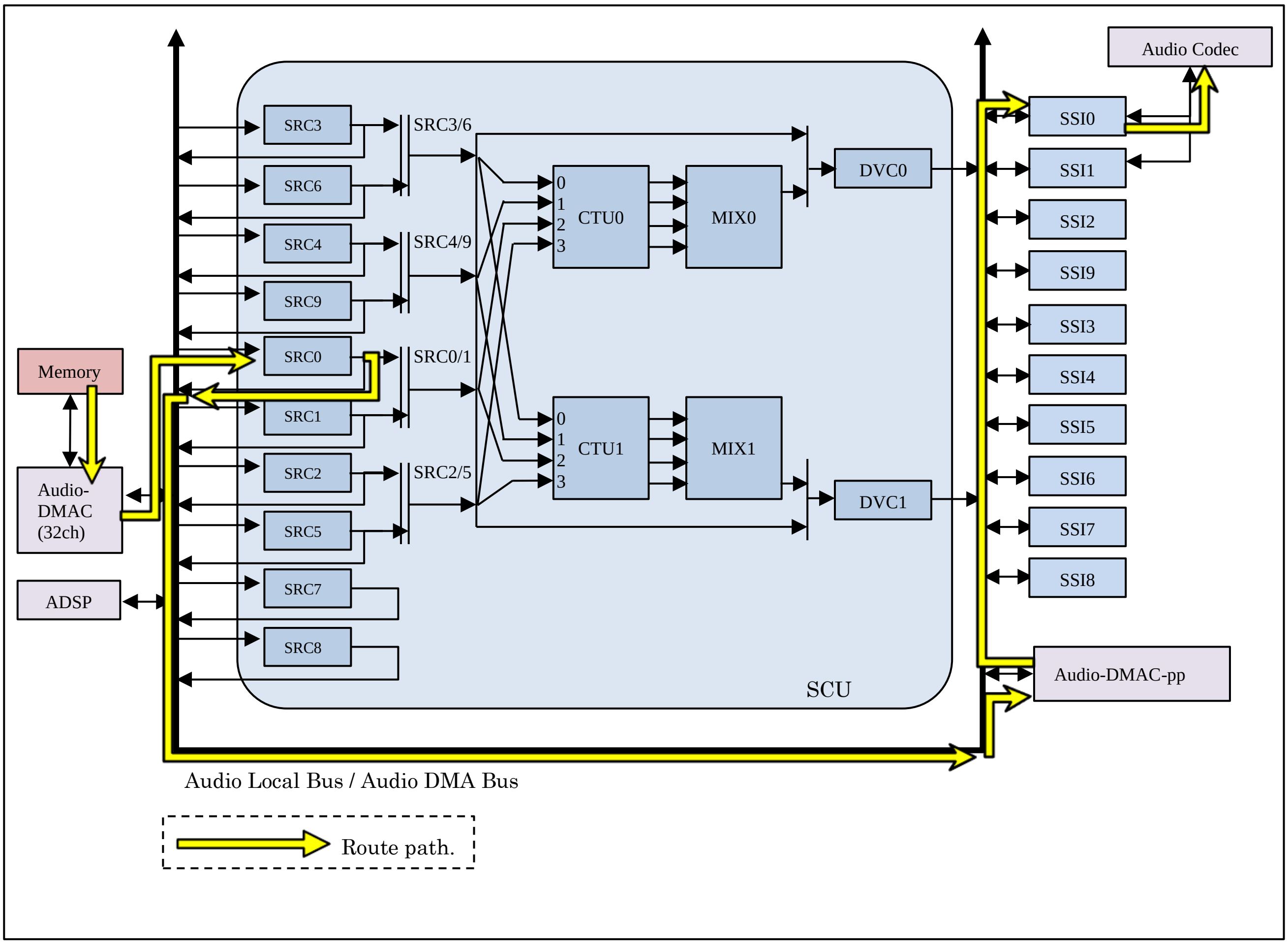


Figure 4-7 Memory->SRC0->SSI0->CODEC data path (R-Car H3/M3/M3N)

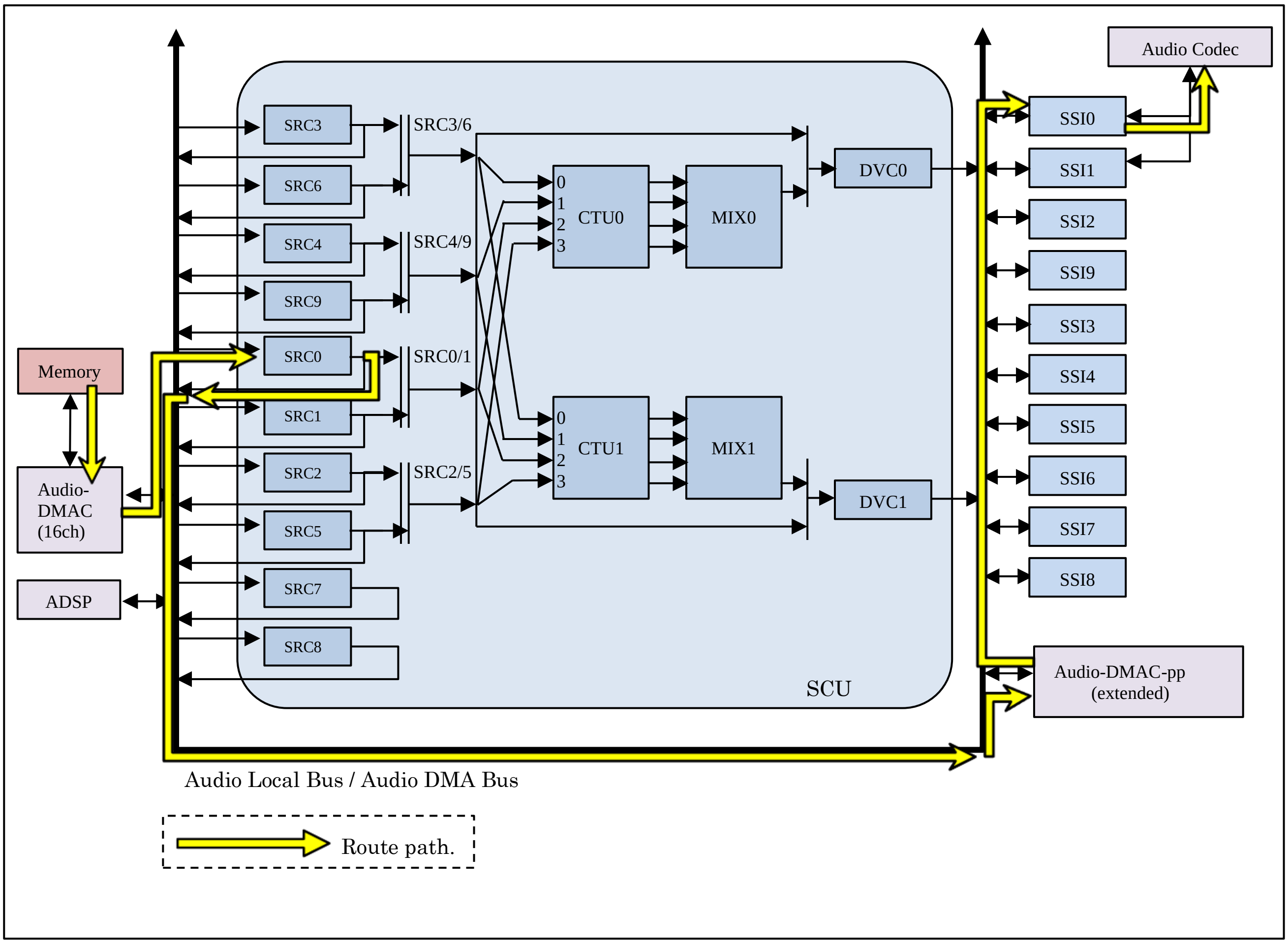


Figure 4-8 Memory->SRC0->SSI0->CODEC data path (R-Car E3)

This route case’s description example is shown below.

```
sound_card: sound {
       compatible = "audio-graph-card";

       label = "rcar-sound";

       dais = <&rsnd_port0>;
};

&rcar_sound {

      ...
       ports {
              rsnd_port0: port@0 {
                     rsnd_endpoint0: endpoint {
                            remote-endpoint = <&ak4613_endpoint>;

                            dai-format = "left_j";
                            bitclock-master = <&rsnd_endpoint0>;
                            frame-master = <&rsnd_endpoint0>;

                            playback = <&ssi0 &src0>;
                            capture  = <&ssi1 &src1>;
                     };
              };
       };
};
```

Figure 4-9 setting for Memory->SRC0->SSI0->CODEC

(3) Setting case of “Memory -> SRC0 -> DVC0 -> SSI0 -> CODEC”

Route path shows the case of "Memory-> SRC0 -> DVC0 -> SSI0 -> CODEC".

Transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI0” at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SCU_CMD0” to the “SSI00” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

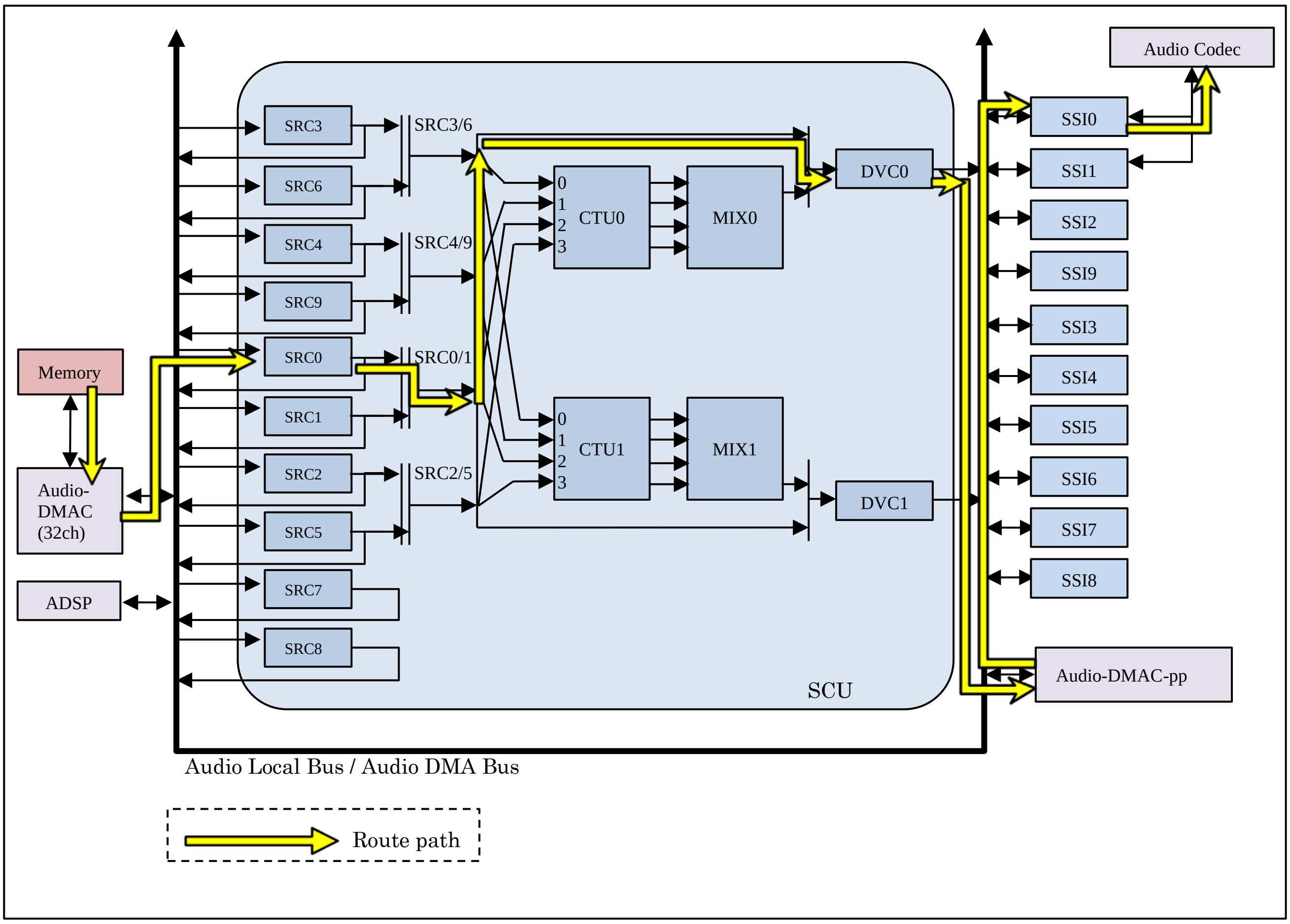


Figure 4-10 Memory->SRC0->DVC0->SSI0->CODEC data path (R-Car H3/M3/M3N)

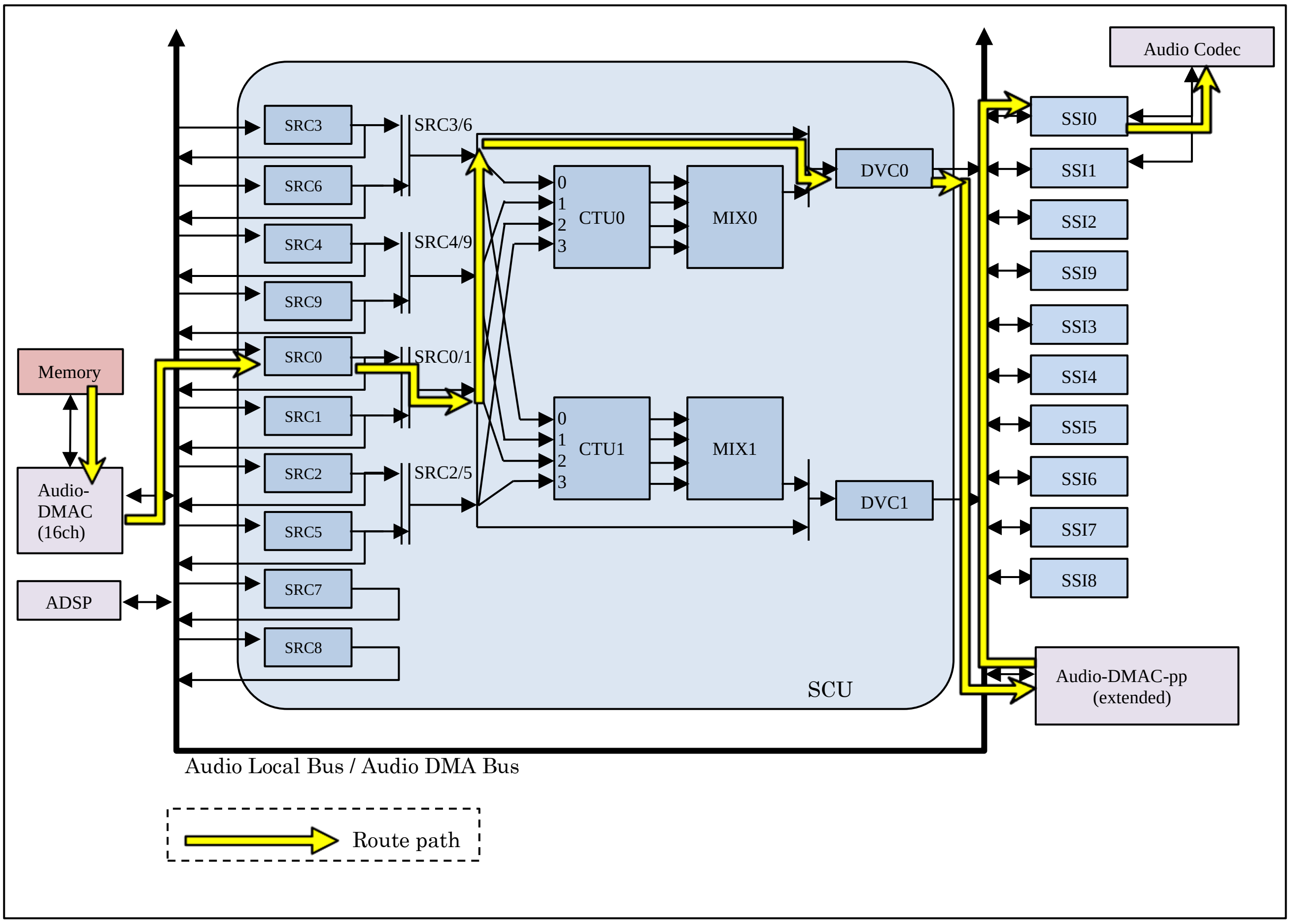


Figure 4-11 Memory->SRC0->DVC0->SSI0->CODEC data path (R-Car E3)

This route case’s description example is shown below.

```
sound_card: sound {
        compatible = "audio-graph-card";

        label = "rcar-sound";

        dais = <&rsnd_port0>;
};

&rcar_sound {

      ...

        ports {
                rsnd_port0: port@0 {
                        rsnd_endpoint0: endpoint {
                                remote-endpoint = <&ak4613_endpoint>;

                                dai-format = "left_j";
                                bitclock-master = <&rsnd_endpoint0>;
                                frame-master = <&rsnd_endpoint0>;

                                playback = <&ssi0 &src0 &dvc0>;
                                capture  = <&ssi1 &src1 &dvc1>;
                        };
                };
        };
};
```

Figure 4-12 setting for Memory->SRC0->DVC0->SSI0->CODEC

(4) Setting case of “Memory -> SRC1 -> CTU02 -> MIX0 -> DVC0 -> SSI0 -> CODEC” and “Memory -> SRC2 -> CTU03 -> MIX0 -> DVC0 -> SSI0 -> CODEC”

Route path shows the case of " Memory -> SRC1-> CTU02->MIX0->DVC0 -> SSI0 -> CODEC “ and “Memory -> SRC2-> CTU03->MIX0->DVC0 -> SSI0 -> CODEC“. This route path setting is mixing the two audio.

Transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI1” at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SCU_CMD0” to the “SSI00” at the driver.

Similarly, another transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI2” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

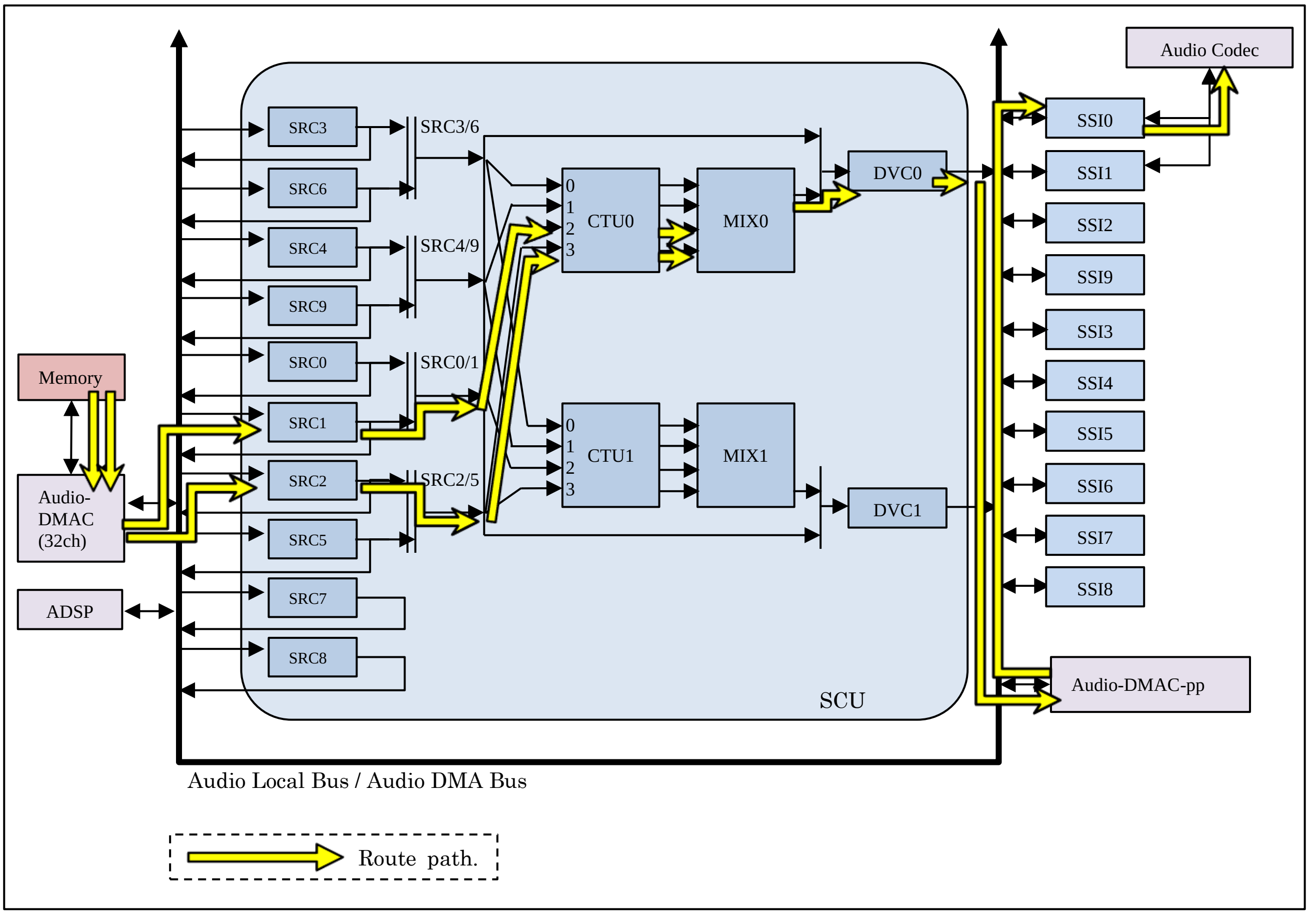


Figure 4-13 Memory->SRC1/2->CTU02/03->MIX0->DVC0->SSI0->CODEC data path (R-Car H3/M3/M3N)

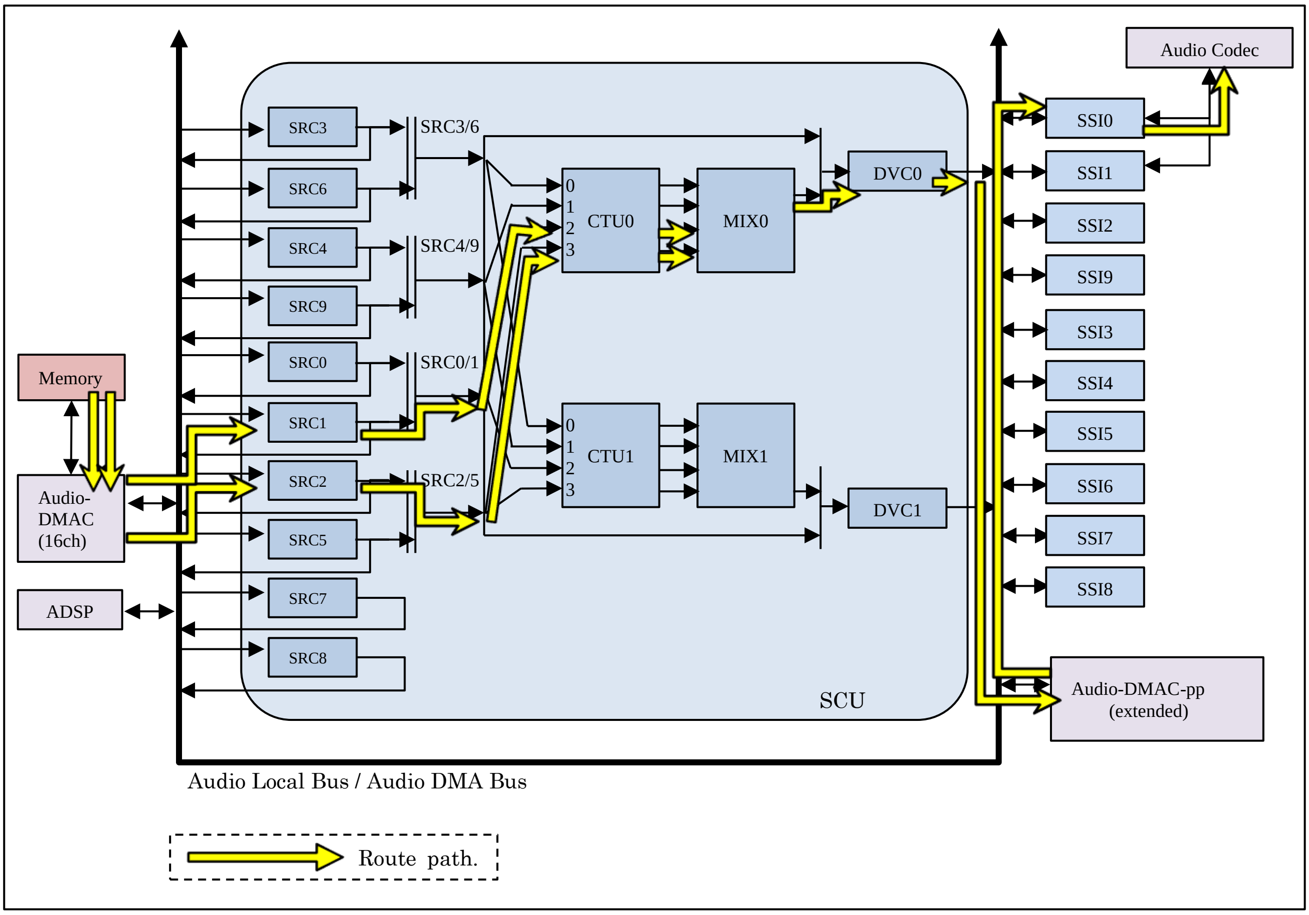


Figure 4-14 Memory->SRC1/2->CTU02/03->MIX0->DVC0->SSI0->CODEC data path (R-Car E3)

[In case of HDMI x 2] This route case's description example is shown below.

```
/delete-node/ &sound_card;
/delete-node/ &ak4613_endpoint;
/delete-node/ &rsnd_endpoint0;
/delete-node/ &rsnd_endpoint1;
/delete-node/ &rsnd_endpoint2;
/delete-node/ &rsnd_port0;
/delete-node/ &rsnd_port1;
/delete-node/ &rsnd_port2;
/ {
      sound {
           compatible = "audio-graph-scu-card";

           routing =  "ak4613 Playback", "DAI0 Playback",
                      "ak4613 Playback", "DAI1 Playback";

           dais =     <&rsnd_port00 /* ak4613 (MIX-0) */
                 &rsnd_port01 /* ak4613 (MIX-1) */
                 &rsnd_port1  /* HDMI0 */
                 &rsnd_port2  /* HDMI1 */
           >;
      };
};

&ak4613 {
      port {
           prefix = "ak4613";
           ak4613_ep1: endpoint@0 {
                 remote-endpoint = <&rsnd_endpoint00>;
           };
           ak4613_ep2: endpoint@1 {
                 remote-endpoint = <&rsnd_endpoint01>;
           };
      };
};

&dw_hdmi0_snd_in {
      remote-endpoint = <&rsnd_endpoint1>;
};

&dw_hdmi1_snd_in {
      remote-endpoint = <&rsnd_endpoint2>;
};

&rcar_sound {
       ...

       ports {
             rsnd_port00: port@0 {
                    reg = <0>;
                    rsnd_endpoint00: endpoint {
                           convert-rate = <48000>;
                           remote-endpoint = <&ak4613ep1>;
                           dai-format = "left_j";
                           bitclock-master = <&rsnd_endpoint00>;
                           frame-master    = <&rsnd_endpoint00>;

                           playback = <&src1 &ctu02 &mix0 &dvc0 &ssi0>;
                           capture  = <&ssi1 &src1 &dvc1>;
                    };
             };
```

```
            rsnd_port01: port@1 {
                  reg = <1>;
                  rsnd_endpoint01: endpoint {
                        convert-rate = <48000>;
                        remote-endpoint = <&ak4613_ep2>;
                        dai-format = "left_j";
                        bitclock-master = <&rsnd_endpoint01>;
                        frame-master    = <&rsnd_endpoint01>;
                        playback = <&src2 &ctu03 &mix0 &dvc0 &ssi0>;
                  };
            };
            rsnd_port1: port@2 {             reg = <2>;
                  rsnd_endpoint1: endpoint {
                        remote-endpoint = <&dw_hdmi0_snd_in>;

                        dai-format = "i2s";
                        bitclock-master = <&rsnd_endpoint1>;
                        frame-master = <&rsnd_endpoint1>;

                        playback = <&ssi2>;
                  };
            };
            rsnd_port2: port@3 {
                  reg = <3>;
                  rsnd_endpoint2: endpoint {
                        remote-endpoint = <&dw_hdmi1_snd_in>;

                        dai-format = "i2s";
                        bitclock-master = <&rsnd_endpoint2>;
                        frame-master = <&rsnd_endpoint2>;

                        playback = <&ssi3>;
                  };
            };
      };
};
```

Figure 4-15 setting for Memory->SRC1/2->CTU02/03->MIX0->DVC0->SSI0->CODEC

[In case of HDMI x 1] This route case’s description example is shown below.

```
/delete-node/ &sound_card;
/delete-node/ &ak4613_endpoint;
/delete-node/ &rsnd_endpoint0;
/delete-node/ &rsnd_endpoint1;
/delete-node/ &rsnd_port0;
/delete-node/ &rsnd_port1;
/ {
      sound {
           compatible = "audio-graph-scu-card";

           routing =  "ak4613 Playback", "DAI0 Playback",
                      "ak4613 Playback", "DAI1 Playback";

           dais =     <&rsnd_port00 /* ak4613 (MIX-0) */
                 &rsnd_port01 /* ak4613 (MIX-1) */
                 &rsnd_port1  /* HDMI0 */
           >;
      };
};

&ak4613 {
      port {
           prefix = "ak4613";
           ak4613_ep1: endpoint@0 {
                 remote-endpoint = <&rsnd_endpoint00>;
           };
           ak4613_ep2: endpoint@1 {
                 remote-endpoint = <&rsnd_endpoint01>;
           };
      };
};

&dw_hdmi0_snd_in {
      remote-endpoint = <&rsnd_endpoint1>;
};

&rcar_sound {
       ...

       ports {
             rsnd_port00: port@0 {
                    reg = <0>;
                    rsnd_endpoint00: endpoint {
                           convert-rate = <48000>;
                           remote-endpoint = <&ak4613ep1>;
                           dai-format = "left_j";
                           bitclock-master = <&rsnd_endpoint00>;
                           frame-master    = <&rsnd_endpoint00>;

                           playback = <&src1 &ctu02 &mix0 &dvc0 &ssi0>;
                           capture  = <&ssi1 &src1 &dvc1>;
                    };
             };
```

```
                rsnd_port1: port@2 {
                        reg = <2>;
                        rsnd_endpoint1: endpoint {
                                remote-endpoint = <&dw_hdmi0_snd_in>;

                                dai-format = "i2s";
                                bitclock-master = <&rsnd_endpoint1>;
                                frame-master = <&rsnd_endpoint1>;

                                playback = <&ssi2>;
                         };
                };
        };
};
```

Figure 4-16 setting for Memory->SRC1/2->CTU02/03->MIX0->DVC0->SSI0->CODEC

(5) Setting case of “Memory -> SRC5 -> SSI3 -> CODEC”

Route path shows the case of "Memory-> SRC5 -> SSI3-> CODEC".

Transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI5” at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SCU_SRCO5” to the “SSI30” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

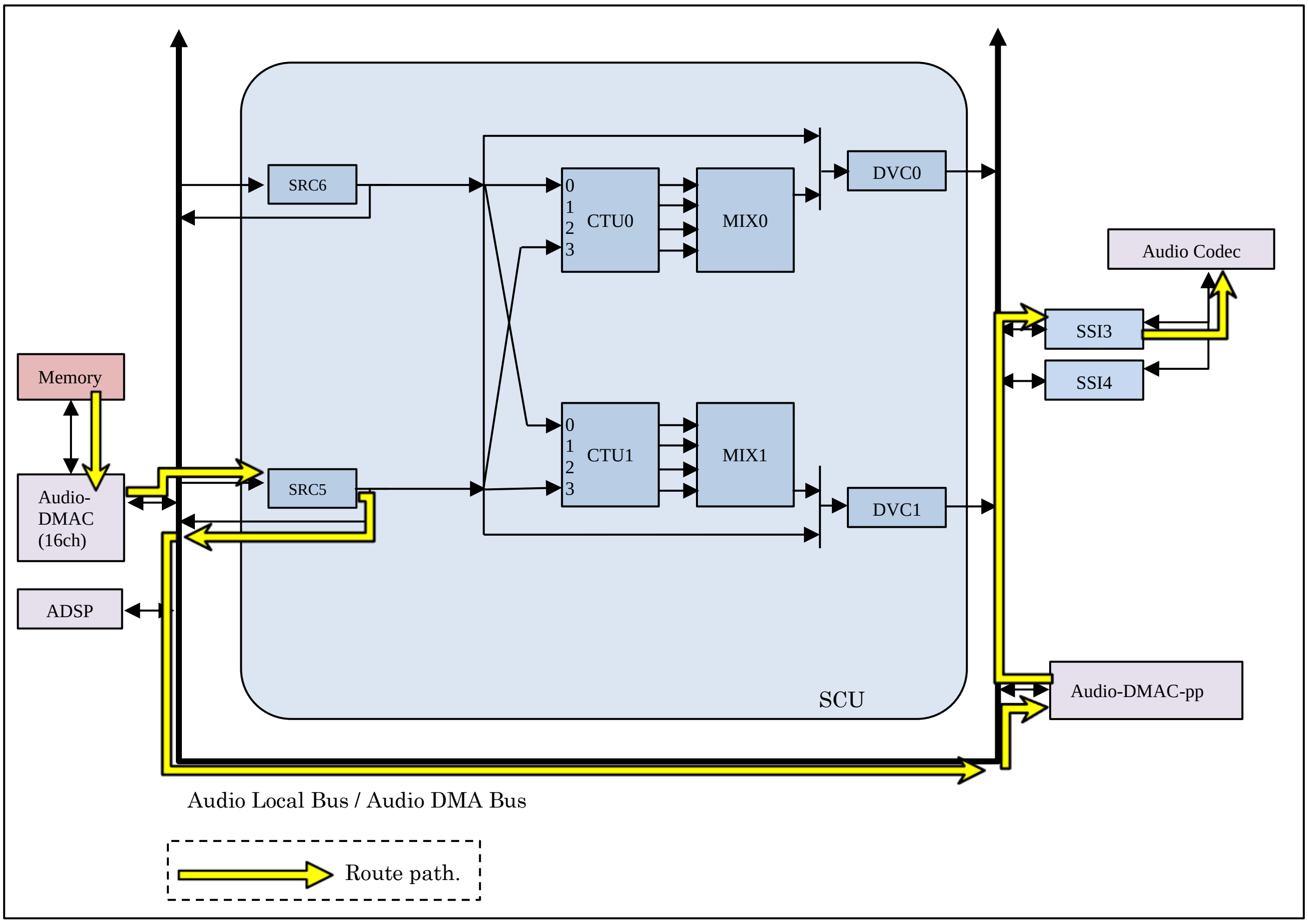


Figure 4.17 Memory->SRC5->SSI3->CODEC data path

This route case's description example is shown below.

```
&rcar_sound {

      ...


      rsnd_port0: port {
            rsnd_for_ak4613: endpoint {
                  playback = <&ssi3>, <&src5>;
                  ...
            };
      };
};
```

Figure 4.18 setting for Memory->SRC5->SSI3->CODEC

(6) Setting case of “Memory -> SRC5 -> DVC0 -> SSI3 -> CODEC”

Route path shows the case of "Memory-> SRC5 -> DVC0 -> SSI3 -> CODEC".

Transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI5” at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SCU_CMD0” to the “SSI30” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

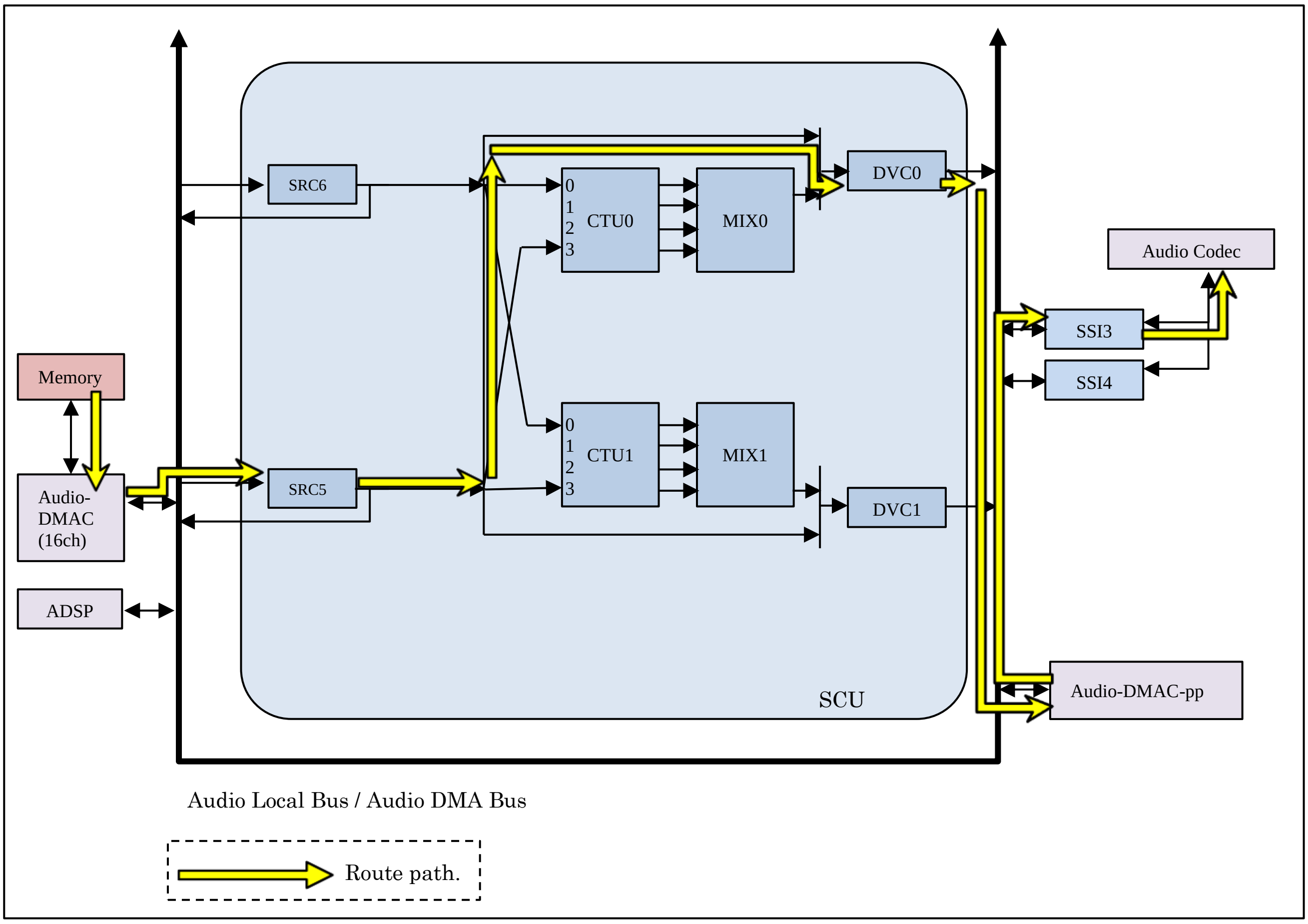


Figure 4.19 Memory->SRC5->DVC0->SSI3->CODEC data path

This route case’s description example is shown below.

```
&rcar_sound {

      ...


      rsnd_port0: port {
            rsnd_for_ak4613: endpoint {
                  playback = <&ssi3>, <&src5>, <&dvc0>;
                  ...
            };
      };
};
```

Figure 4.20 setting for Memory->SRC5->DVC0->SSI3->CODEC

(7) Setting case of “Memory -> SRC5 -> CTU00 -> MIX0 -> DVC0 -> SSI3 -> CODEC” and “Memory -> SRC6 -> CTU03 -> MIX0 -> DVC0 -> SSI3 -> CODEC”

Setting case of “Memory -> SRC5 -> CTU00 -> MIX0 -> DVC0 -> SSI3 -> CODEC” and “Memory -> SRC6 -> CTU03 -> MIX0 -> DVC0 -> SSI3 -> CODEC”

Route path shows the case of " Memory -> SRC5-> CTU00->MIX0->DVC0 -> SSI3 -> CODEC “ and “Memory -> SRC6-> CTU03->MIX0->DVC0 -> SSI3 -> CODEC“. This route path setting is mixing the two audios.

Transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI5” at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SCU_CMD0” to the “SSI30” at the driver.

Similarly, another transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI6” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

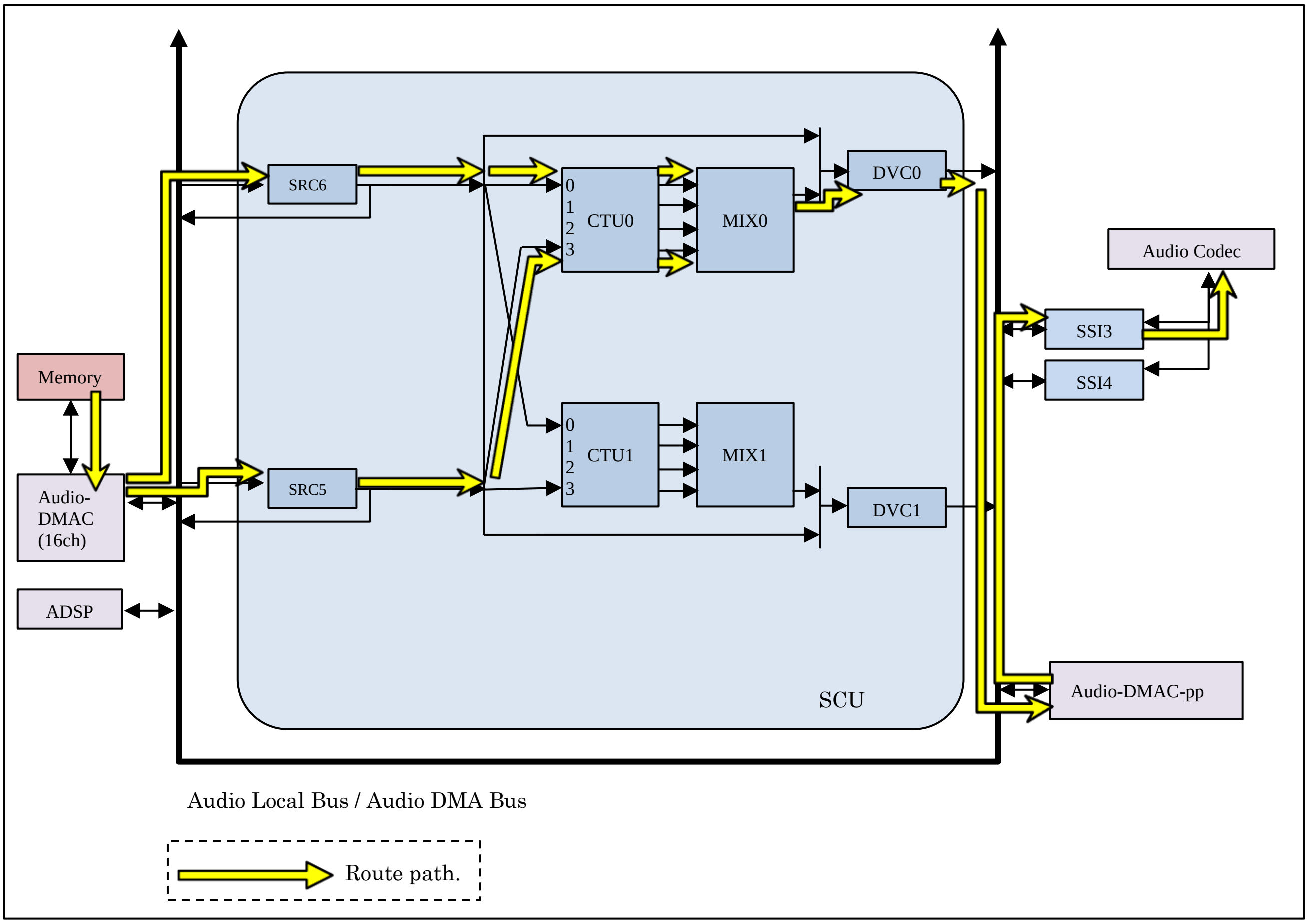


Figure 4.21 Memory->SRC6/5->CTU00/03->MIX0->DVC0->SSI3->CODEC data path

This route case’s description example is shown below.

```
&rcar_sound {

      ...

      rsnd_port0: port@0 {
            rsnd_endpoint0: endpoint {
                  playback = <&src6>, <&ctu00>, <&mix0>, <&dvc0>, <&ssi3>;
            };
      };
      rsnd_port1: port@1 {
            rsnd_endpoint1: endpoint {
                  playback = <&src5>, <&ctu03>, <&mix0>, <&dvc0>, <&ssi3>;
            };
      };
};
```

Figure 4.22 setting for Memory->SRC6/5->CTU00/03->MIX0->DVC0->SSI3->CODEC

This route case’s description example is shown below.

```
&rcar_sound {

	...


	rsnd_port0: port {
		rsnd_for_ak4613: endpoint {
			playback = <&ssi0>, <&src0>, <&dvc0>;
			...
		};
	};
};
```

Figure 4.23 setting for Memory->SRC0->DVC0->SSI0->CODEC

(8) Setting case of “Memory -> SRC1 -> CTU02 -> MIX0 -> DVC0 -> SSI0 -> CODEC” and “Memory -> SRC2 -> CTU03 -> MIX0 -> DVC0 -> SSI0 -> CODEC”

Route path shows the case of " Memory -> SRC1-> CTU02->MIX0->DVC0 -> SSI0 -> CODEC “ and “Memory -> SRC2-> CTU03->MIX0->DVC0 -> SSI0 -> CODEC“. This route path setting is mixing the two audios.

Transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI1” at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SCU_CMD0” to the “SSI00” at the driver.

Similarly, another transfer settings of audio DMAC is set to transfer from the memory to the “SCU_SRCI2” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

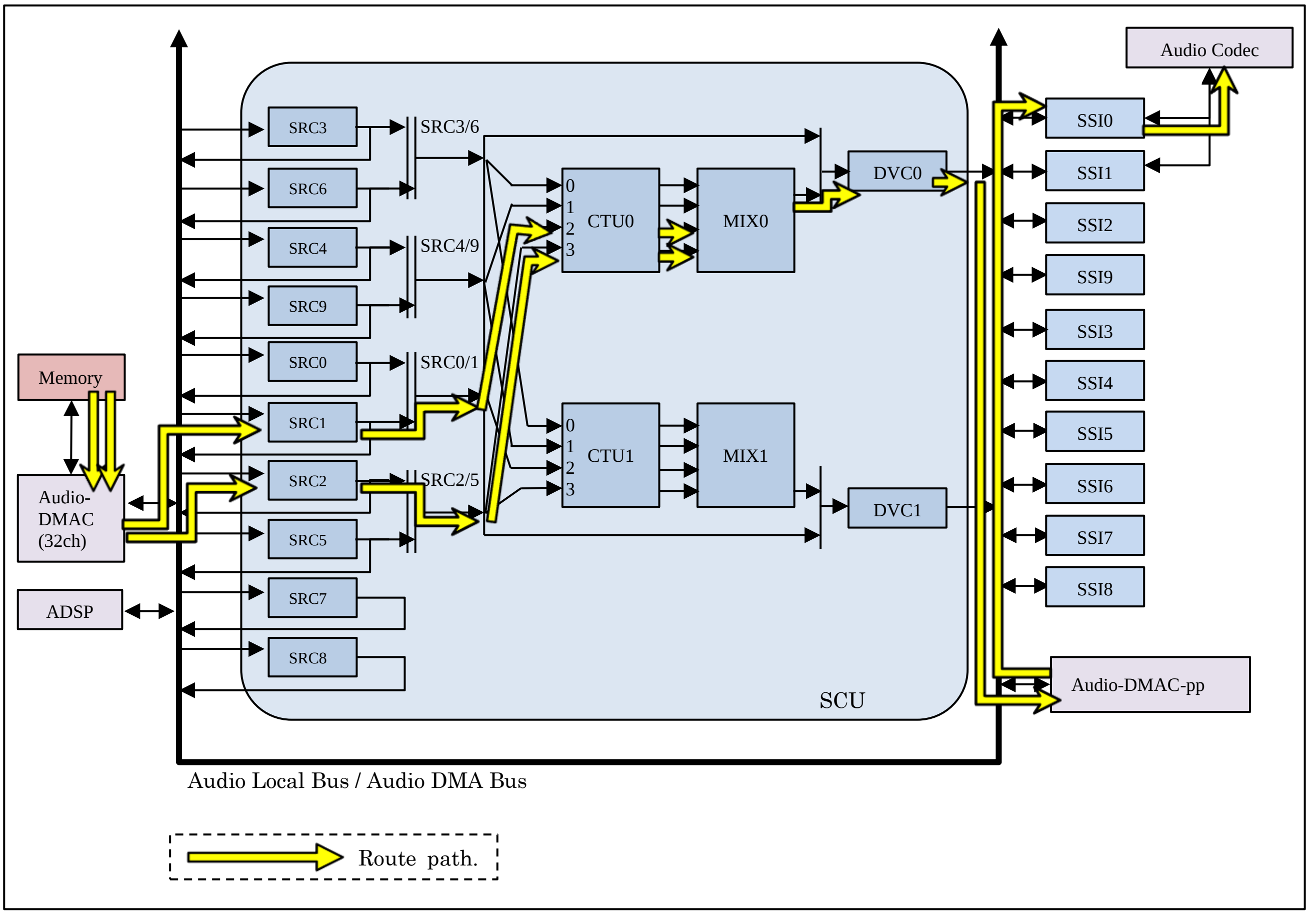


Figure 4.24 Memory->SRC1/2->CTU02/03->MIX0->DVC0->SSI0->CODEC data path

This route case's description example is shown below.

```
&rcar_sound {

        ...

        rsnd_port0: port@0 {
                rsnd_endpoint0: endpoint {
                        playback = <&src1>, <&ctu02>, <&mix0>, <&dvc0>, <&ssi0>;
                };
        };
        rsnd_port1: port@1 {
                 rsnd_endpoint1: endpoint {
                        playback = <&src2>, <&ctu03>, <&mix0>, <&dvc0>, <&ssi0>;
                };
        };
};
```

Figure 4.25 setting for Memory->SRC1/2->CTU02/03->MIX0->DVC0->SSI0->CODEC

4.3.3 Setting route for capture

[R-Car H3/M3/M3N/E3] (1) – (3), [R-Car D3] (4) – (6).

(1) **Setting case of “CODEC -> SSI1 -> Memory”**

Route path shows the case of "CODEC-> SSI1-> Memory".

Transfer settings of audio DMAC is set to transfer from the “SSI10” to the memory at the driver. Audio DMAC-pp is no use at this case.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

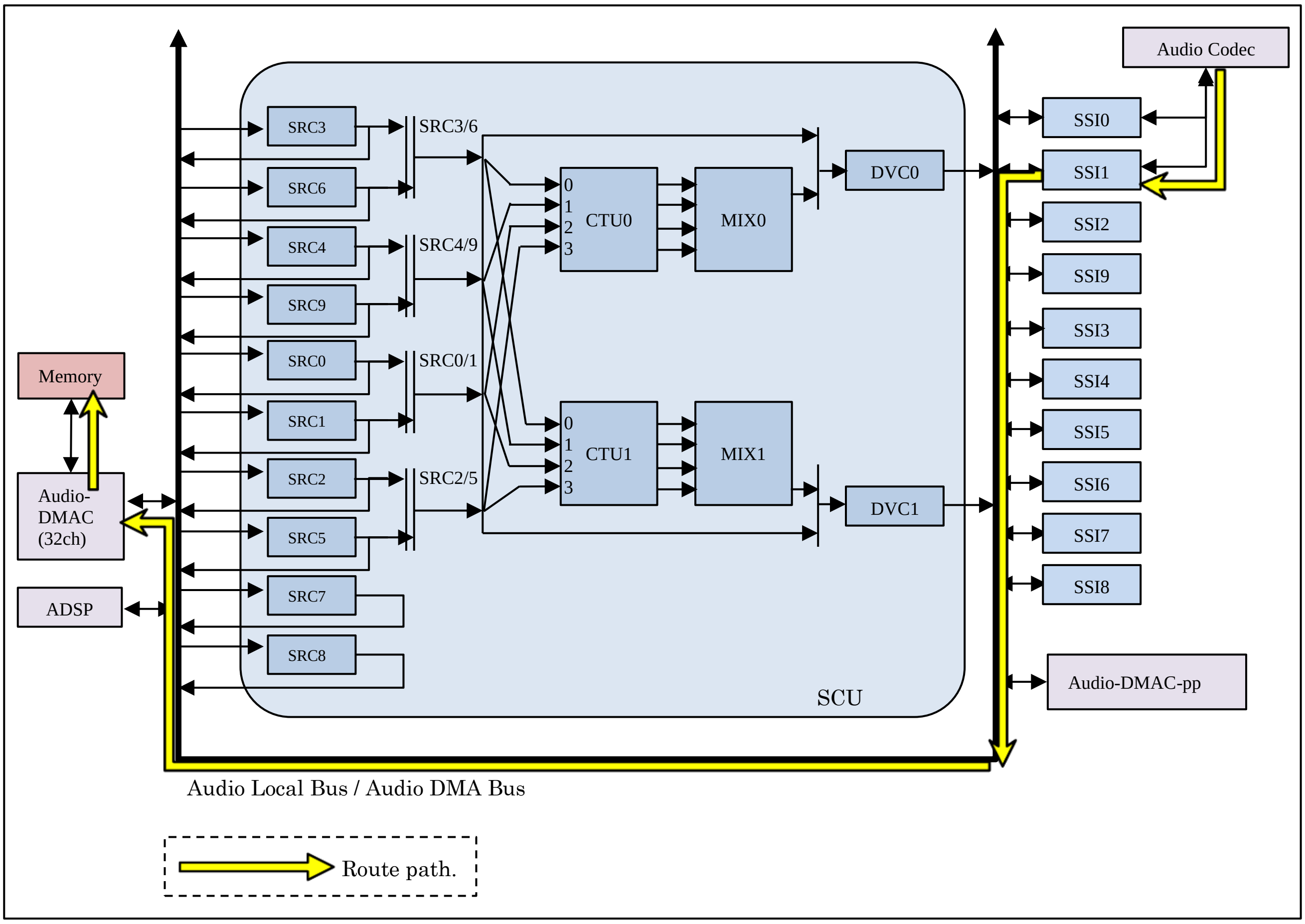


Figure 4-26 CODEC->SSI1->Memory data path (R-Car H3/M3/M3N)

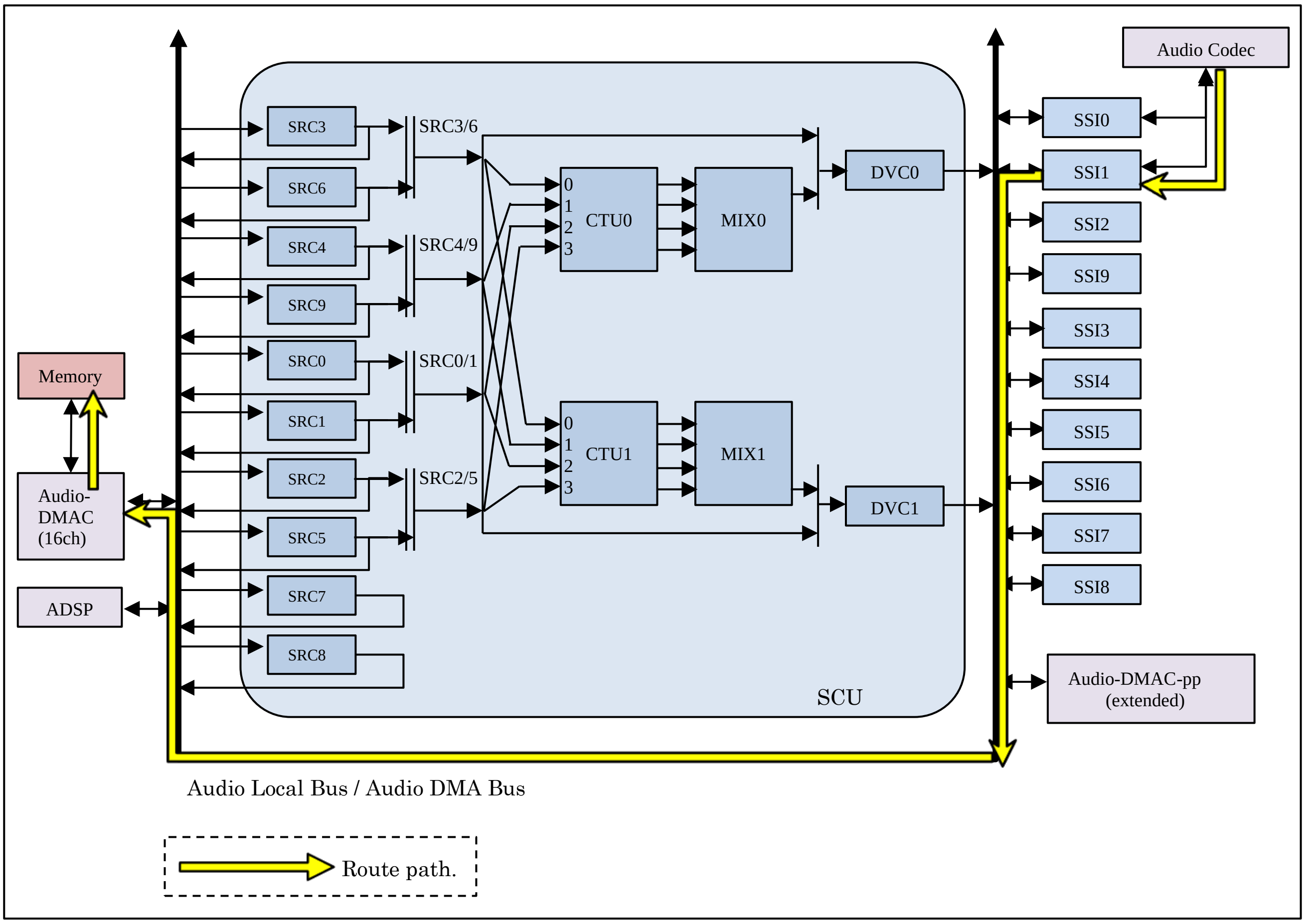


Figure 4-27 CODEC->SSI1->Memory data path (R-Car E3)

This route case’s description example is shown below.

```
sound_card: sound {
        compatible = "audio-graph-card";

        label = "rcar-sound";

        dais = <&rsnd_port0>;
};


&rcar_sound {

      ...

        ports {
                rsnd_port0: port@0 {
                        rsnd_endpoint0: endpoint {
                                remote-endpoint = <&ak4613_endpoint>;

                                dai-format = "left_j";
                                bitclock-master = <&rsnd_endpoint0>;
                                frame-master = <&rsnd_endpoint0>;

                                playback = <&ssi0>;
                                capture  = <&ssi1>;
                        };
                };
        };
};


&ssi1 {
      shared-pin;
};
```

Figure 4-28 setting for CODEC->SSI1->Memory

Each SSI of the R-Car H3/M3/M3N is possible to share the WS pin. At the Salvator-X board, "SSI0, SSI1, SSI2, SSI9" is configured as a shared pin (SSI_WS0129). At Ebisu board, "SSI0, SSI1, SSI2, SSI3, SSI9" is configured as a shared pin (SSI_WS01239). In such a configuration, it is necessary to describe the share pin configuration of the SSI.

(2) Setting case of “CODEC -> SSI1 -> SRC1 -> Memory”

Route path shows the case of "CODEC -> SSI1 -> SRC1 -> Memory”.

Transfer settings of audio DMAC is set to transfer from the “SCU_SRCI1” to the “memory” at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SSI10” to the “SCU_SRCI1” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

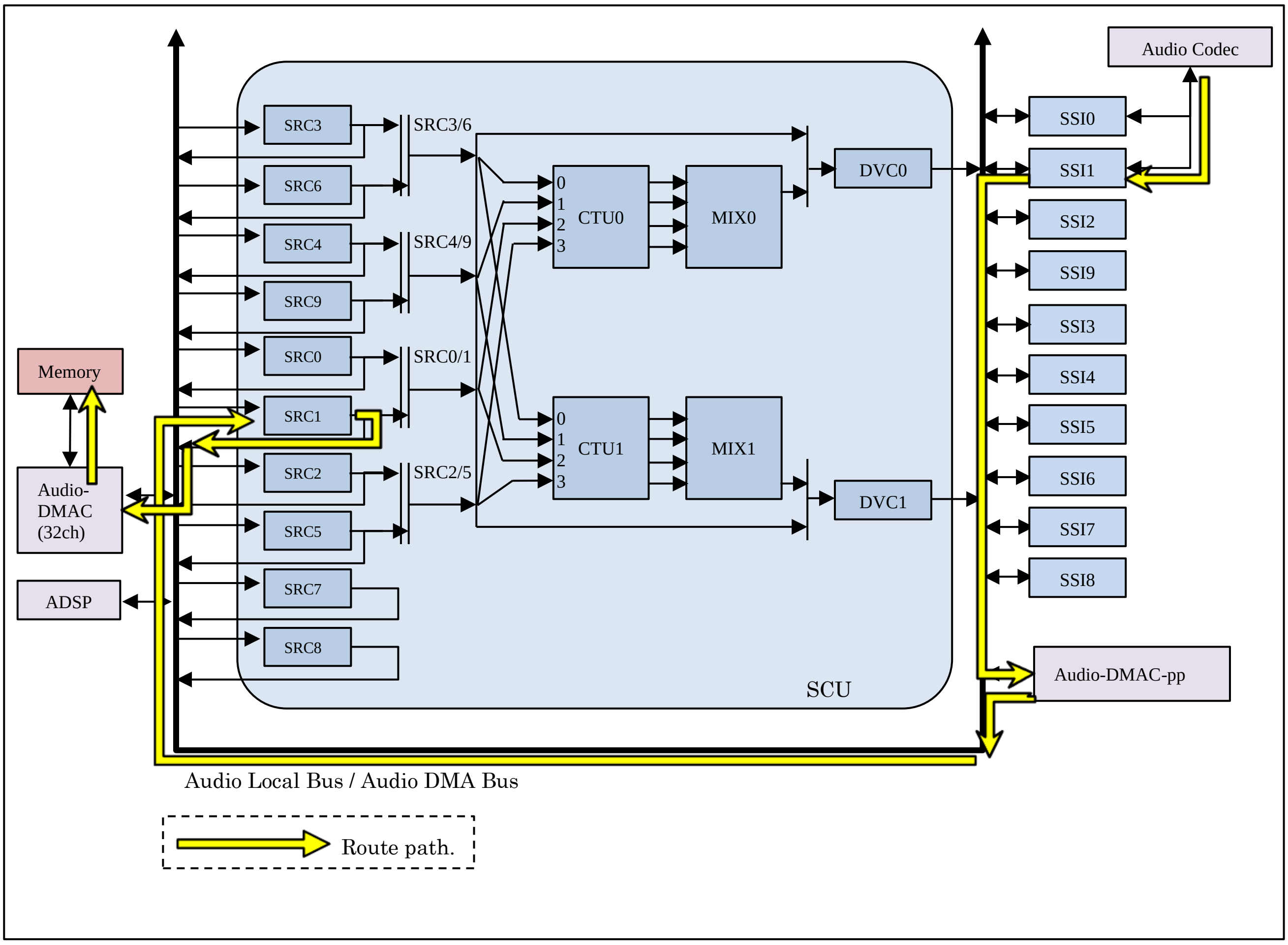


Figure 4-29 CODEC->SSI1->SRC1->Memory data path (R-Car H3/M3/M3N)

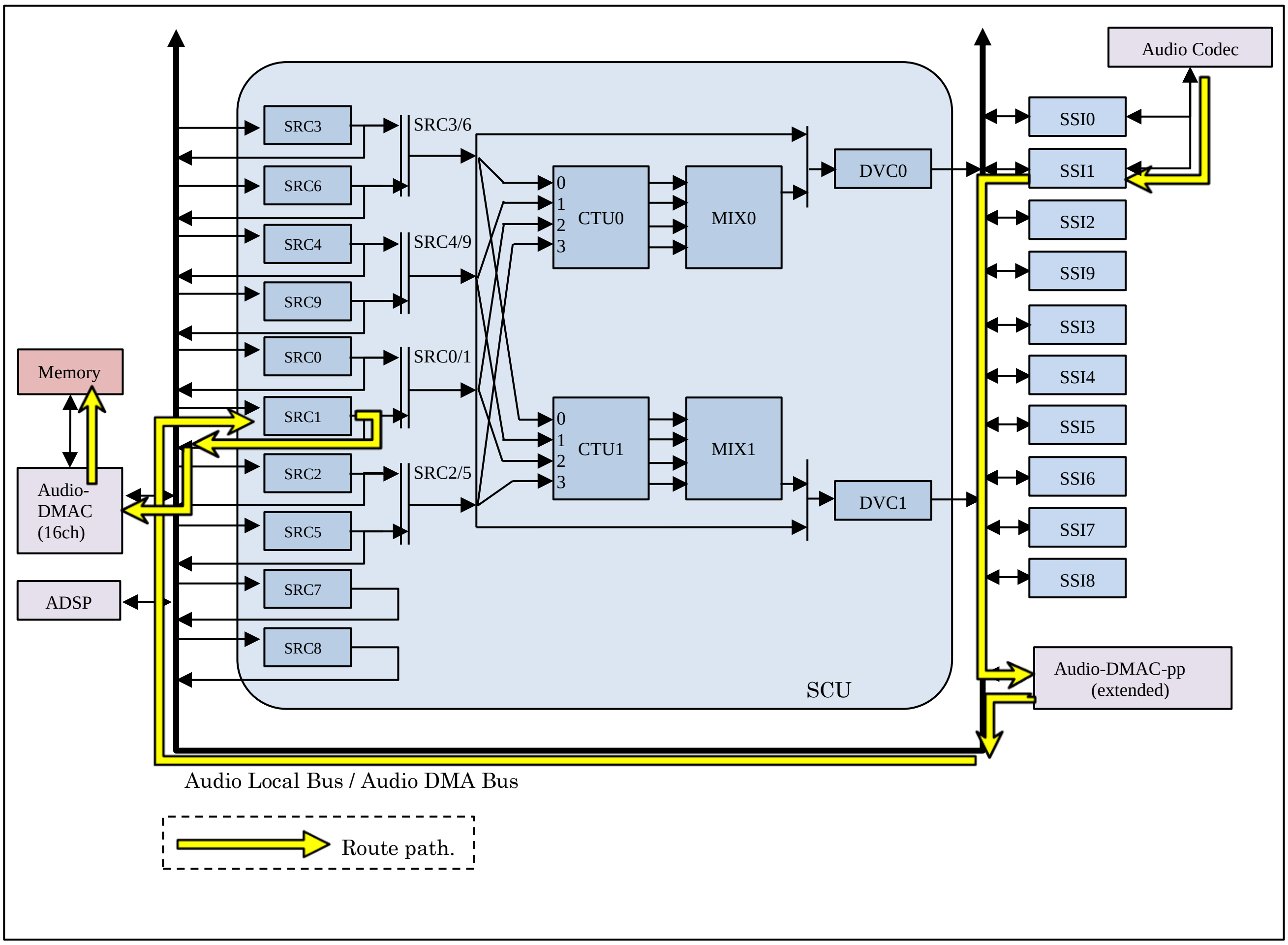


Figure 4-30 CODEC->SSI1->SRC1->Memory data path (R-Car E3)

This route case’s description example is shown below.

```
sound_card: sound {
        compatible = "audio-graph-card";

        label = "rcar-sound";

        dais = <&rsnd_port0>;
};

&rcar_sound {

       ...
        ports {
                rsnd_port0: port@0 {
                        rsnd_endpoint0: endpoint {
                                remote-endpoint = <&ak4613_endpoint>;

                                dai-format = "left_j";
                                bitclock-master = <&rsnd_endpoint0>;
                                frame-master = <&rsnd_endpoint0>;

                                playback = <&ssi0 &src0>;
                                capture  = <&ssi1 &src1>;
                        };
                };
        };
};


&ssi1 {
       shared-pin;
};
```

Figure 4-31 setting for CODEC->SSI1->SRC1->Memory

(3) Setting case of “CODEC -> SSI1 -> SRC1 -> DVC1 -> Memory”

Route path shows the case of "CODEC -> SSI1 -> SRC1 -> DVC1 -> Memory".

Notes) When the DVC is enabled, the sampling rate conversions at the SRC input cannot be used by hardware constraints. Therefore, some features, such as "SRC In rate" of amixer control interface is disabled.

Transfer settings of audio DMAC is set to transfer from the “SCU_CMD1” to the Memory at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SSI10” to the “SCU_SRCI1” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

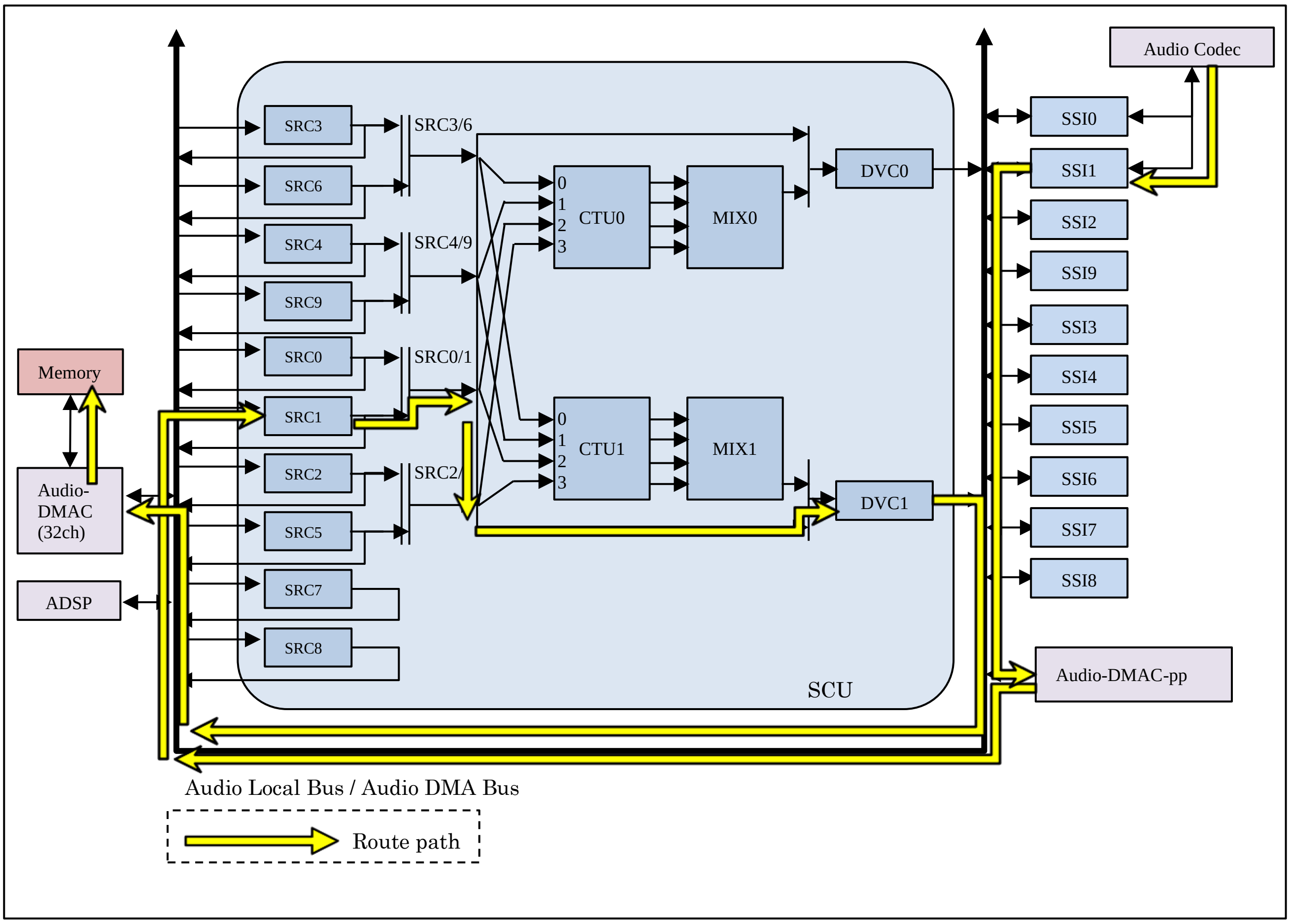


Figure 4-32 CODEC->SSI1->SRC1->DVC1->Memory data path (R-Car H3/M3/M3N)

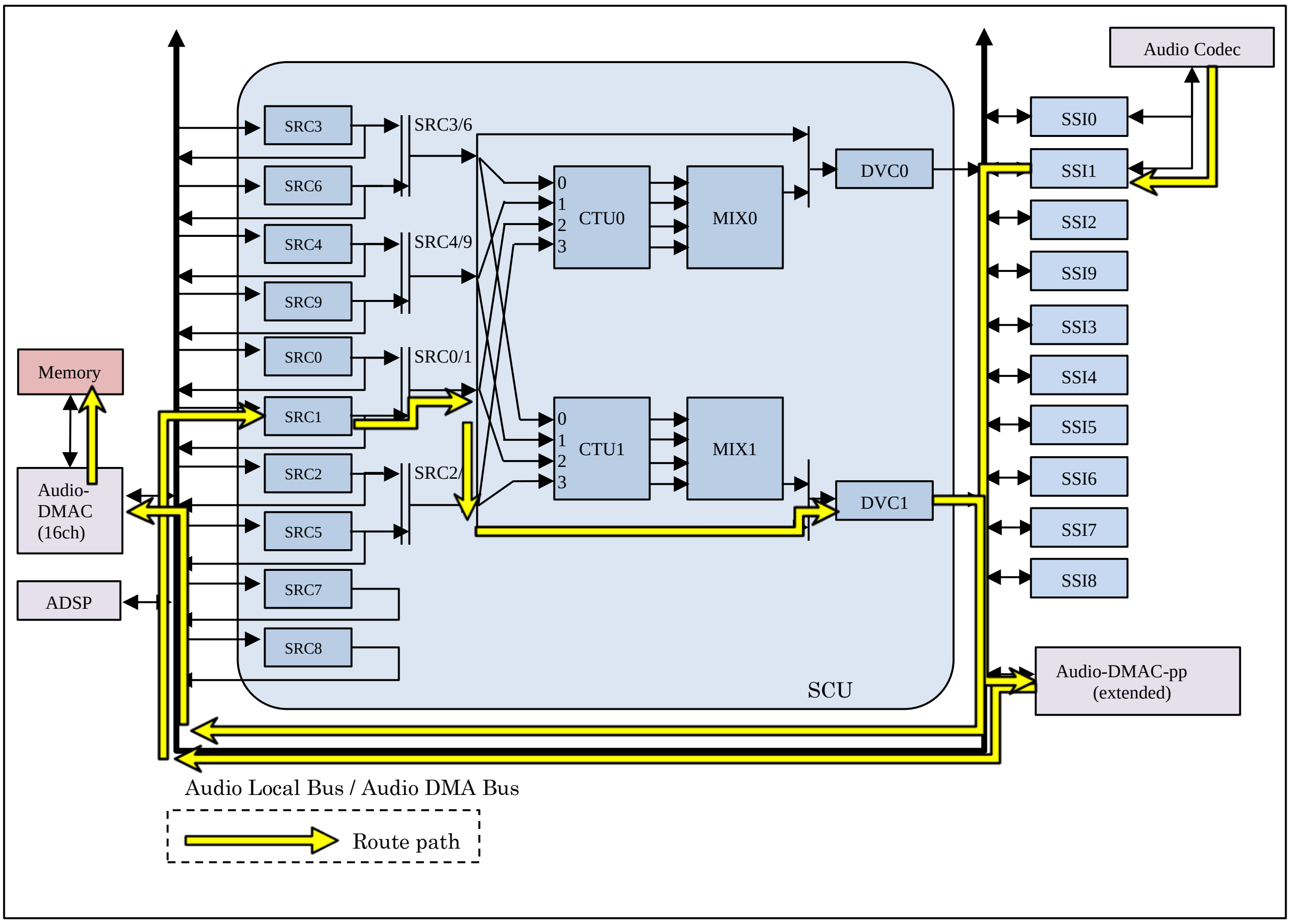


Figure 4-33 CODEC->SSI1->SRC1->DVC1->Memory data path (R-Car E3)

This route case’s description example is shown below.

```
sound_card: sound {
        compatible = "audio-graph-card";

        label = "rcar-sound";

        dais = <&rsnd_port0>;
};

&rcar_sound {

      ...

        ports {
                rsnd_port0: port@0 {
                        rsnd_endpoint0: endpoint {
                                remote-endpoint = <&ak4613_endpoint>;

                                dai-format = "left_j";
                                bitclock-master = <&rsnd_endpoint0>;
                                frame-master = <&rsnd_endpoint0>;

                                playback = <&ssi0 &src0 &dvc0>;
                                capture  = <&ssi1 &src1 &dvc1>;
                        };
                };
        };
};

&ssi1 {
      shared-pin;
};
```

Figure 4-34 setting for CODEC->SSI1->SRC1->DVC1->Memory)

(4) Setting case of “CODEC -> SSI4 -> Memory”

Setting case of “CODEC -> SSI4 -> Memory”

Route path shows the case of "CODEC-> SSI4-> Memory".

Transfer settings of audio DMAC is set to transfer from the “SSI40” to the memory at the driver. Audio DMAC-pp is no use at this case.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

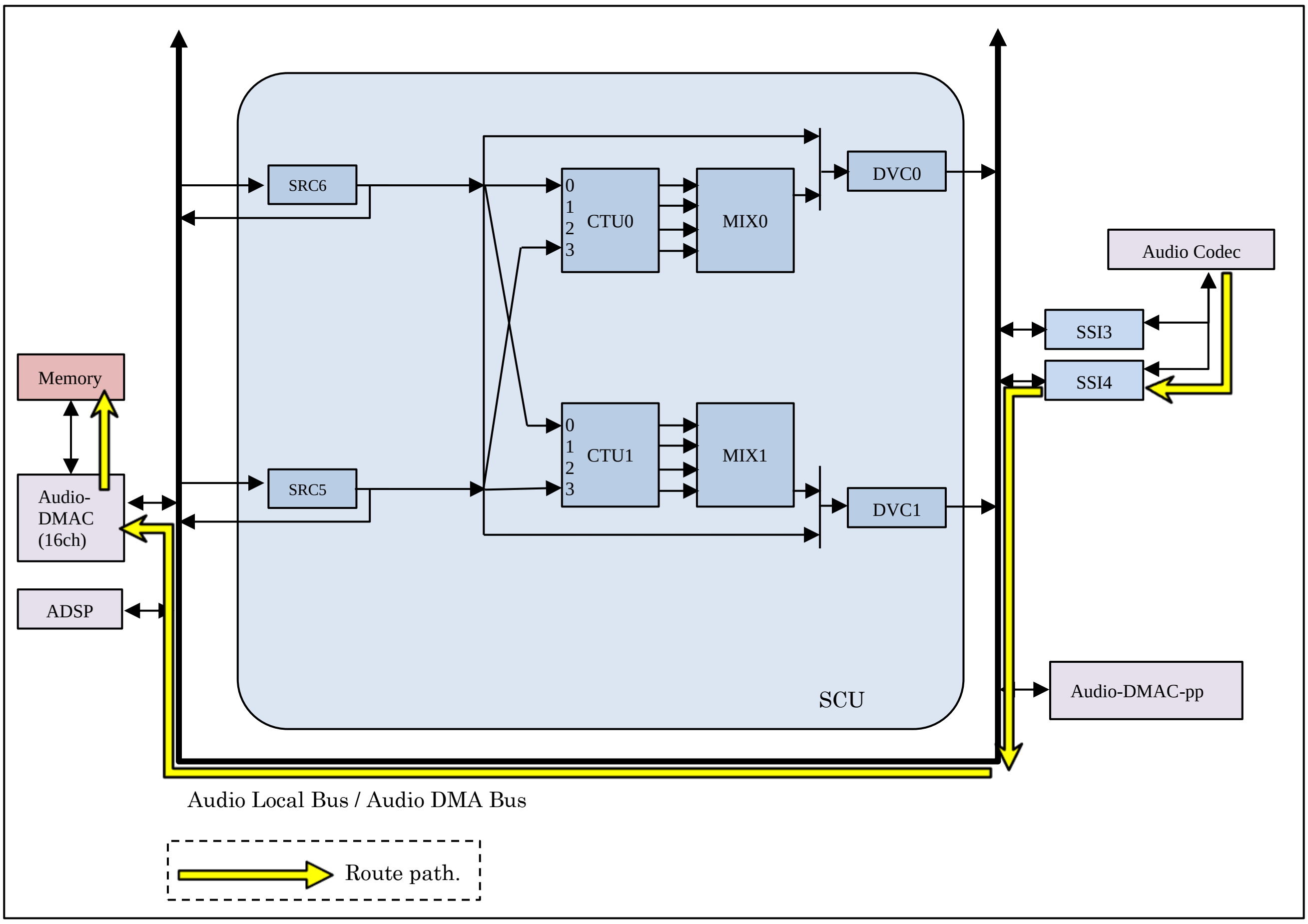


Figure 4.35 CODEC->SSI4->Memory data path

This route case’s description example is shown below.

```
&rcar_sound {

      ...


      rsnd_port0: port {
            rsnd_for_ak4613: endpoint {
                  ...
                  capture = <&ssi4>;
            };
      };
};

&ssi4 {
      shared-pin;
};
```

Figure 4.36 setting for CODEC->SSI4->Memory

Each SSI of the R-Car D3 is possible to share the WS pin. At the Draak board, "SSI3, SSI4 " is configured as a shared pin (SSI_WS). In such a configuration, it is necessary to describe the share pin configuration of the SSI.

(5) Setting case of “CODEC -> SSI4 -> SRC6 -> Memory”

Setting case of “CODEC -> SSI4 -> SRC6 -> Memory”

Route path shows the case of "CODEC -> SSI4 -> SRC6 -> Memory”.

Transfer settings of audio DMAC is set to transfer from the “SCU_SRCI6” to the “memory” at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SSI40” to the “SCU_SRCI6” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

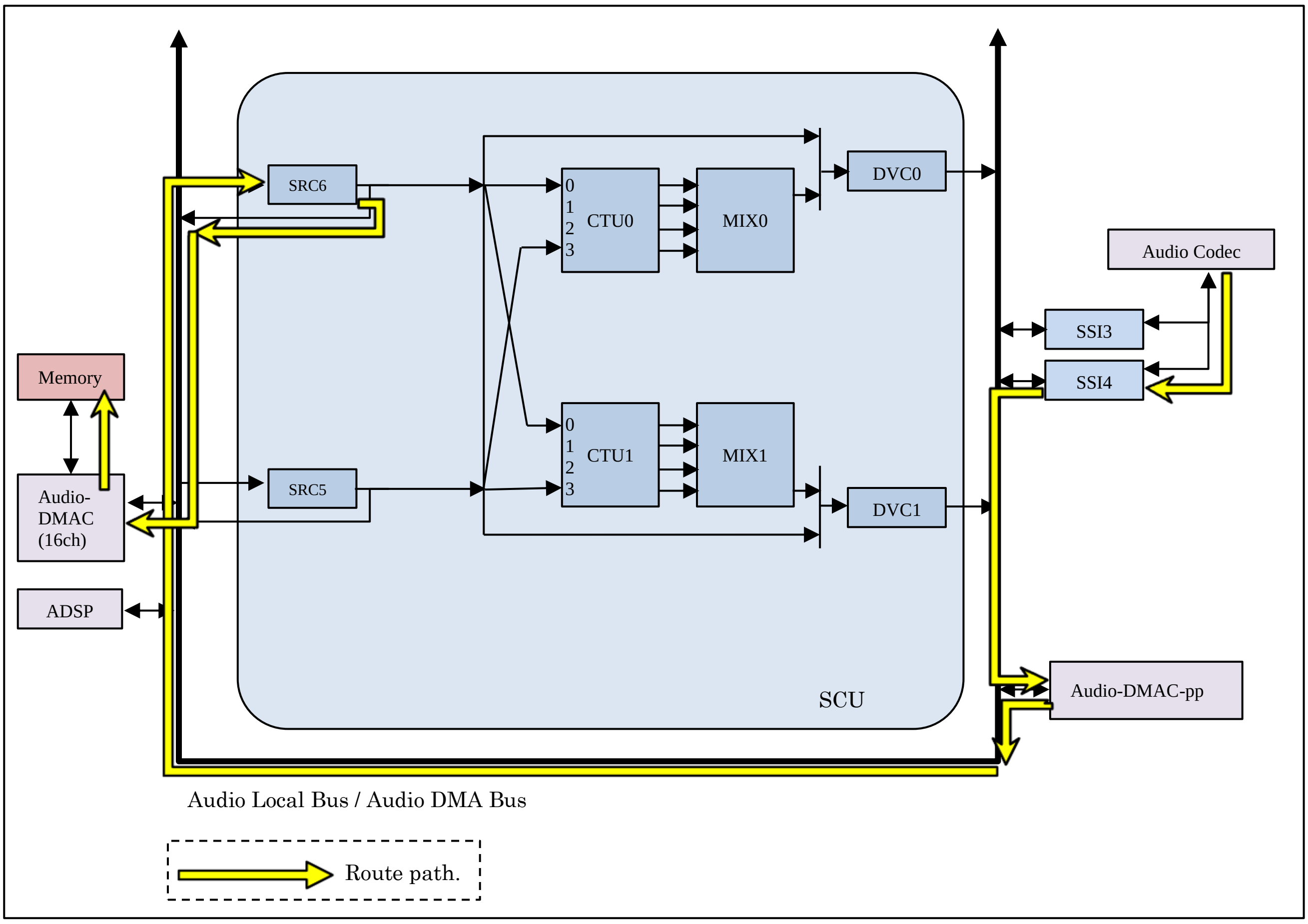


Figure 4.37 CODEC->SSI4->SRC6->Memory data path

This route case’s description example is shown below.

```
&rcar_sound {

      ...


      rsnd_port0: port {
            rsnd_for_ak4613: endpoint {
                  ...
                  capture = <&ssi4>, <&src6>;
            };
      };
};

&ssi4 {
      shared-pin;
};
```

Figure 4.38 setting for CODEC->SSI4->SRC6->Memory

(6) Setting case of “CODEC -> SSI4 -> SRC6 -> DVC1 -> Memory”

Setting case of “CODEC -> SSI4 -> SRC6 -> DVC1 -> Memory”

Route path shows the case of "CODEC -> SSI4 -> SRC6 -> DVC1 -> Memory".

Notes) When the DVC is enabled, the sampling rate conversions at the SRC input cannot be used by hardware constraints. Therefore, some features, such as "SRC In rate" of amixer control interface is disabled.

Transfer settings of audio DMAC is set to transfer from the “SCU_CMD1” to the Memory at the driver. In addition, the transfer setting of audio DMAC-pp is set to transfer from “SSI40” to the “SCU_SRCI6” at the driver.

Please refer to “R-Car Series, 3rd Generation User’s Manual: Hardware” about Audio-DMA/Audio-DMA-pp’s source and destination definitions.

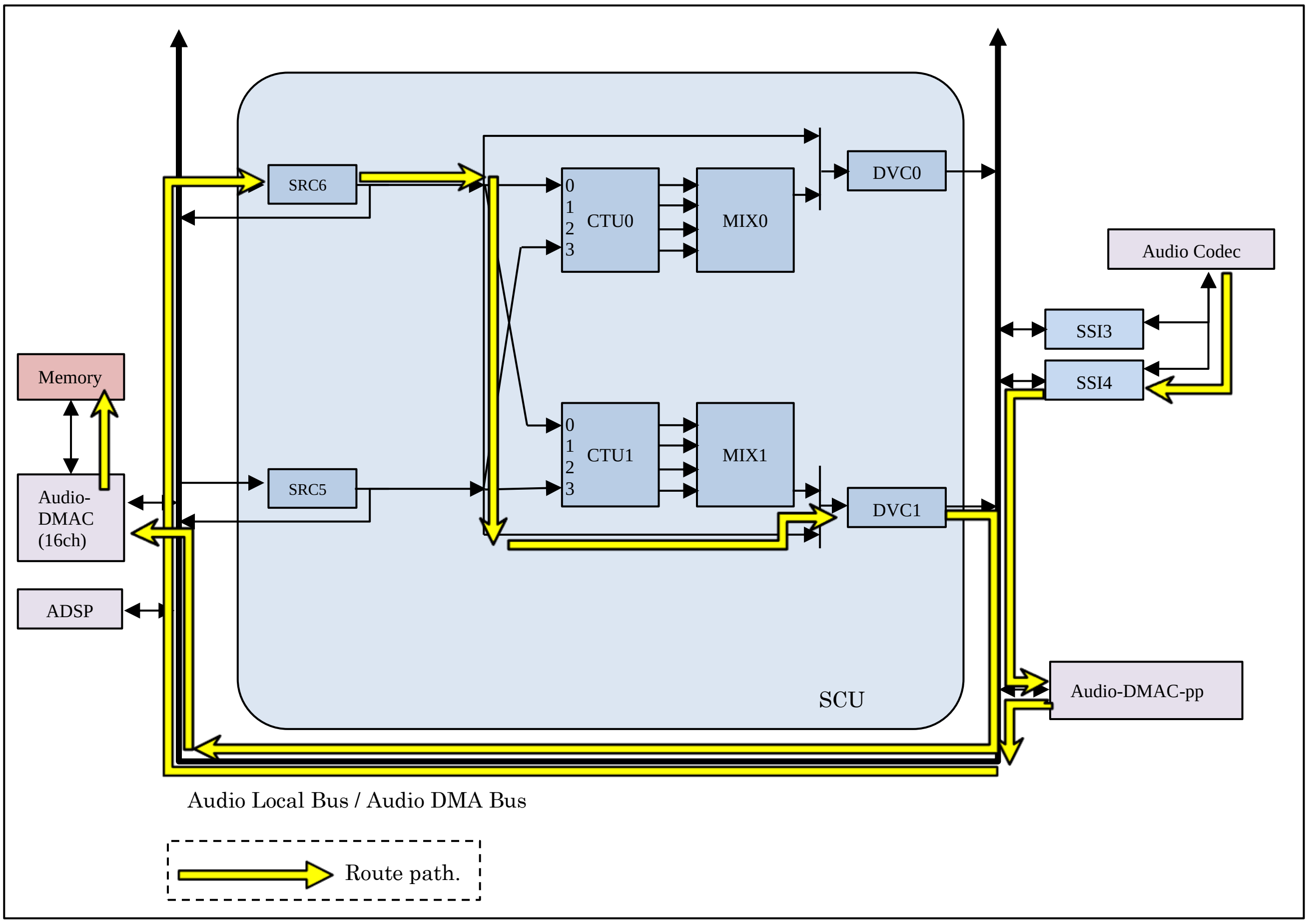


Figure 4.39 CODEC->SSI4->SRC6->DVC1->Memory data path

This route case’s description example is shown below.

```
&rcar_sound {

      ...


      rsnd_port0: port {
            rsnd_for_ak4613: endpoint {
                  ...
                  capture = <&ssi4>, <&src6>, <&dvc1>;
            };
      };
};

&ssi4 {
      shared-pin;
};
```

Figure 4.40 setting for CODEC->SSI4->SRC6->DVC1->Memory)

4.4 Sampling Rate Conversion

This module supports the sampling rate conversion function using the SRC.
To use it, please set enable “CONFIG_SND_ AUDIO_GRAPH_CARD” at kernel configuration (Refer to 5.2).
And please set with 'device tree file', or the control interface. If both are set, the control interface is given priority.

4.4.1 Asynchronous Mode

To activate the sampling rate conversion, requires a description of the “audio-graph-card”. The fixed sampling rate can be set by "convert-rate" in device tree file.

For example, if the codec can only be operated at 48kHz, the output side is fixed at 48kHz and the input side uses the Hz of the sound data.

"Figure 4-" shows example of 48 kHz. This example shows that all input data will be converted to 48 kHz at playback.
Inputted 48 kHz data will be converted to system specified Hz at capture.
In the case of the R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation board, please set fixed value in ak4613_dai_hw_params() of sound/soc/codecs/ak4613.c

(1) **Device tree file setting**

```
sound_card: sound {
        compatible = "audio-graph-card";

        convert-rate = <48000>;
        label = "rcar-sound";

        dais = <&rsnd_port0>;
};
```

Figure 4-41 Description example of sampling rate setting

The clock format can be written in the following format.
‘system-clock-frequency = <value>’
or ‘clocks = <&xxx>’
“Figure 4-” shows example of use ‘clocks =’.

```
audio_clkout: audio_clkout {
        compatible = "fixed-clock";
        #clock-cells = <0>;
        clock-frequency = <12288000>;
};


ak4613: codec@10 {
        ...

        clocks = <&audio_clkout>;
        ...

};
```

Figure 4-42 Description example of sampling rate setting

4.4.2 Synchronous Mode

By using the controls in the amixer, you can convert the sampling rate on runtime. But sound codec IC (AK4613) is maintained the sampling rate since the start. So, the sound will be fast/slow.

For example, to make fine adjustments of sound when playing TV or video, fine-tune the input / output Hz to speed up or slow down the playback.

In the case of a route set up to use the DVC, SRC feature is disabled at the capture.

(1) **Initial Conditions**

Initial setting value is "0". In this case, the rate is converted to same rate using the SRC. Settings can be confirmed by using the Mixer function. Its control name is "SRC Out Rate" and "SRC In Rate".

```
# amixer cget name='SRC Out Rate'
numid=13,iface=MIXER,name='SRC Out Rate'
  ; type=INTEGER,access=rw------,values=1,min=0,max=48000,step=0
  : values=0

# amixer cget name='SRC In Rate'
numid=20,iface=MIXER,name='SRC In Rate'
  ; type=INTEGER,access=rw------,values=1,min=0,max=48000,step=0
  : values=0
```

Figure 4-43 Sampling rate confirmation command

(2) Sampling rate setting

Mixer function can set sampling rate conversion. Control name is "SRC Out Rate" and "SRC In Rate". This function works only when changing settings during playback or recording. Rate conversions enabled by amixer will return with disabling upon completion of playback. Please enable again in necessary case. Available range is from 0 to 48000. When “0” is set, sampling rate converts to same rate. "Figure 4-", "Figure 4-" and "Figure 4-" show example of conversion to 48kHz.

```
# amixer cset name="SRC Out Rate Switch" on
```

Figure 4-44 Command example of enable the sampling rate conversion

And set the conversion rate example following command.

```
# <starting playback> &
# amixer cset name="SRC Out Rate" 48000
```

Figure 4-45 Command example of set conversion rate

```
# amixer cset name='SRC Out Rate' 48000
numid=13,iface=MIXER,name='SRC Out Rate'
  ; type=INTEGER,access=rw------,values=1,min=0,max=48000,step=0
  : values=48000

# amixer cset name='SRC In Rate' 48000
numid=20,iface=MIXER,name='SRC In Rate'
  ; type=INTEGER,access=rw------,values=1,min=0,max=48000,step=0
  : values=48000
```

Figure 4-46 Sampling rate setting command

4.5 Rate Continuous

By this setting, ALSA can support all sampling rate. But initial setting of this driver only supports specific sampling rate (Refer to Table 1-7), because this feature is disabled.

If you would like to enable this rate continuous function, please change source code (sound/soc/sh/rcar/core.c). The setting of "rates", "rate_min" and "rate_max" in snd_soc_dai_driver structure is needed. Changes are shown in Figure 4-. In case of R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board, please also change CODEC source code (sound/soc/codecs/ak4613.c) in the same way.

And more, description of device tree

sound/soc/codecs/ak4613.c

```
static struct snd_soc_dai_driver ak4613_dai = {
                        .stream_name     = "Playback",
                        .channels_min    = 2,
                        .channels_max    = 2,
-                       .rates           = AK4613_PCM_RATE,
+                       .rates           = SNDRV_PCM_RATE_CONTINUOUS,
+                       .rate_min        = 32000,
+                       .rate_max        = 192000,
                        .formats         = AK4613_PCM_FMTBIT,
            },
            .capture = {
                        .stream_name     = "Capture",
                        .channels_min    = 2,
                        .channels_max    = 2,
-                       .rates           = AK4613_PCM_RATE,
+                       .rates           = SNDRV_PCM_RATE_CONTINUOUS,
+                       .rate_min        = 32000,
+                       .rate_max        = 192000,
                        .formats         = AK4613_PCM_FMTBIT,
            },
        .ops = &ak4613_dai_ops,
```

sound/soc/sh/rcar/core.c

```
   *
   */
  #include <linux/pm_runtime.h>
+//#include <sound/pcm.h>
  #include "rsnd.h"

-#define RSND_RATES SNDRV_PCM_RATE_8000_192000
+#define RSND_RATES SNDRV_PCM_RATE_CONTINUOUS
  #define RSND_FMTS (SNDRV_PCM_FMTBIT_S24_LE | SNDRV_PCM_FMTBIT_S16_LE)
====
static void __rsnd_dai_probe(struct rsnd_priv *priv,
                             struct device_node *dai_np,
                             int dai_i, int is_graph)
```

```
{
        :
          snprintf(io->playback.name, RSND_DAI_NAME_SIZE,
                   "DAI%d Playback", dai_i);
          drv->playback.rates              = RSND_RATES;
+          drv->playback.rate_min          = 32000;
+          drv->playback.rate_max          = 48000;
          drv->playback.formats            = RSND_FMTS;
          drv->playback.channels_min       = 2;
          drv->playback.channels_max       = 8;
          drv->playback.stream_name        = io->playback.name;

          snprintf(io->capture.name, RSND_DAI_NAME_SIZE,
                   "DAI%d Capture", dai_i);
          drv->capture.rates               = RSND_RATES;
+          drv->capture.rate_min           = 32000;
+          drv->capture.rate_max           = 48000;
          drv->capture.formats             = RSND_FMTS;
          drv->capture.channels_min        = 2;
          drv->capture.channels_max        = 8;
          drv->capture.stream_name         = io->capture.name;
```

Figure 4-47 Setting of Rate Continuous

4.6 CTU Function

This function details see “R-Car Series, 3rd Generation User’s Manual: Hardware”.

Those sections are:

- Sampling Rate Converter Unit (SCU)
  - Register Description
    - CTUn Scale Value e00 ～ e37 register
  - Operation
    - CMD Block
    - Functional Blocks in CMD

4.6.1 CTU module setting

(1) **Device tree file setting**

This is example of all input data will be converted to 2ch as output data.

```
&sound_card {
        compatible = "audio-graph-scu-card";

        label = "rcar-sound";
        prefix = "ak4613";
        routing = "ak4613 Playback", "DAI0 Playback",
                        "ak4613 Playback", "DAI1 Playback";
        convert-channels = <2>;

        dais = <&rsnd_port0
                &rsnd_port1>;
};
```

Figure 4-48 Description example of CTU setting

(2) **Example of using**

The example of using CTU.

ex1) using matrix

output 0ch = (input 0ch x 0) + (input 1ch x 1)
output 1ch = (input 0ch x 1) + (input 1ch x 0)

```
$ amixer set "CTU Reset" on
$ amixer set "CTU Pass" 9,10
$ amixer set "CTU SV0" 0,4194304
$ amixer set "CTU SV1" 4194304,0
```

ex2) changing connection

```
$ amixer set "CTU Reset" on
$ amixer set "CTU Pass" 2,1
```

4.7 MIX Function

The Mixer function support the merges sounds path. Up to four sound interfaces can be set on one card device on the system, and these sounds are merged by MIX.

4.7.1 MIX module setting

(1) **Device tree file setting**

```
&sound_card {
        compatible = "audio-graph-scu-card";

        label = "rcar-sound";
        prefix = "ak4613";
        routing = "ak4613 Playback", "DAI0 Playback",
                        "ak4613 Playback", "DAI1 Playback",
                        "ak4613 Playback", "DAI2 Playback",
                        "ak4613 Playback", "DAI3 Playback";
        convert-rate = <48000>;
        convert-channels = <2>;

        dais = <&rsnd_port0
                &rsnd_port1
                &rsnd_port2
                &rsnd_port3>;
};

&i2c2 {
        ak4613: codec@10 {
                port {
                        /delete-node/ endpoint;

                        ak4613_endpoint0: endpoint@0 {
                                remote-endpoint = <&rsnd_endpoint0>;
                        };
                        ak4613_endpoint1: endpoint@1 {
                                remote-endpoint = <&rsnd_endpoint1>;
                        };
                        ak4613_endpoint2: endpoint@2 {
                                remote-endpoint = <&rsnd_endpoint2>;
                        };
                        ak4613_endpoint3: endpoint@3 {
                                remote-endpoint = <&rsnd_endpoint3>;
                        };
                };
        };
};

&rcar_sound {
        ...

        ports {
                rsnd_port0: port@0 {
                        reg = <0>;
                        rsnd_endpoint0: endpoint {
                                remote-endpoint = <&ak4613_endpoint0>;

                                dai-format = "left_j";
                                bitclock-master;
                                frame-master;

                                playback = <&src3 &ctu00 &mix0 &dvc0 &ssi0>;
                                capture  = <&ssi1 &src1 &dvc1>;
                        };
                };
                rsnd_port1: port@1 {
                        reg = <1>;
                        rsnd_endpoint1: endpoint {
                                remote-endpoint = <&ak4613_endpoint1>;

                                dai-format = "left_j";
                                bitclock-master;
                                frame-master;

                                playback = <&src4 &ctu01 &mix0 &dvc0 &ssi0>;
                        };
                };
                rsnd_port2: port@2 {
                        reg = <2>;
                        rsnd_endpoint2: endpoint {
                                remote-endpoint = <&ak4613_endpoint2>;

                                dai-format = "left_j";
                                bitclock-master;
                                frame-master;

                                playback = <&src0 &ctu02 &mix0 &dvc0 &ssi0>;
                        };
                };
                rsnd_port3: port@3 {
                        reg = <3>;
                        rsnd_endpoint3: endpoint {
                                remote-endpoint = <&ak4613_endpoint3>;

                                dai-format = "left_j";
                                bitclock-master;
                                frame-master;

                                playback = <&src2 &ctu03 &mix0 &dvc0 &ssi0>;
                        };
                };
        };
};
```

Figure 4-49 Description example of MIX setting (R-CarH3-SiP/M3-SiP/M3N-SiP)

```
rsnd_ak4613: sound {
       compatible = "simple-scu-audio-card";

       simple-audio-card,name = "rsnd-ak4613";
       simple-audio-card,format = "left_j";
       simple-audio-card,bitclock-master = <&sndcpu>;
       simple-audio-card,frame-master = <&sndcpu>;

       simple-audio-card,convert-rate = <48000>;

       simple-audio-card,prefix = "ak4613";
       simple-audio-card,routing = "ak4613 Playback", "DAI0 Playback",
       "ak4613 Playback", "DAI1 Playback";
       sndcpu: simple-audio-card,cpu@0 {
             sound-dai = <&rcar_sound 0>;
       };
       simple-audio-card,cpu {
             sound-dai = <&rcar_sound 1>;
       };


       sndcodec: simple-audio-card,codec {
              sound-dai = <&ak4613>;
       };
   };

&rcar_sound {
      ...
      /* Multi DAI */
      #sound-dai-cells = <1>;

      ...
      rcar_sound,dai {

       dai0 {
            playback = <&src6 &ctu00 &mix0 &dvc0 &ssi3>;
           };
       dai1 {
            playback = <&src5 &ctu03 &mix0 &dvc0 &ssi3>;
           };
   };
};
```

Figure 4.50 Description example of MIX setting(R-Car D3)

```
&sound_card {
        compatible = "audio-graph-scu-card";

        label = "rcar-sound";
        prefix = "ak4613";
        routing = "ak4613 Playback", "DAI0 Playback",
                        "ak4613 Playback", "DAI1 Playback",
                        "ak4613 Playback", "DAI2 Playback",
                        "ak4613 Playback", "DAI3 Playback";
        convert-rate = <48000>;

        dais = <&rsnd_port0
                &rsnd_port1
                &rsnd_port2
                &rsnd_port3>;
};

&i2c3 {
        ak4613: codec@10 {
                port {
                        /delete-node/ endpoint;

                        ak4613_endpoint0: endpoint@0 {
                                remote-endpoint = <&rsnd_endpoint0>;
                        };
                        ak4613_endpoint1: endpoint@1 {
                                remote-endpoint = <&rsnd_endpoint1>;
                        };
                        ak4613_endpoint2: endpoint@2 {
                                remote-endpoint = <&rsnd_endpoint2>;
                        };
                        ak4613_endpoint3: endpoint@3 {
                                remote-endpoint = <&rsnd_endpoint3>;
                        };
                };
        };
};

&rcar_sound {
        ...

        ports {
                rsnd_port0: port@0 {
                        reg = <0>;
                        rsnd_endpoint0: endpoint {
                                remote-endpoint = <&ak4613_endpoint0>;

                                dai-format = "left_j";
                                bitclock-master;
                                frame-master;

                                playback = <&src3 &ctu00 &mix0 &dvc0 &ssi0>;
                                capture  = <&ssi1 &src1 &dvc1>;
                        };
                };
                rsnd_port1: port@1 {
                        reg = <1>;
                        rsnd_endpoint1: endpoint {
                                remote-endpoint = <&ak4613_endpoint1>;

                                dai-format = "left_j";
                                bitclock-master;
                                frame-master;
                                playback = <&src4 &ctu01 &mix0 &dvc0 &ssi0>;
                        };
                };
                rsnd_port2: port@2 {
                        reg = <2>;
                        rsnd_endpoint2: endpoint {
                                remote-endpoint = <&ak4613_endpoint2>;

                                dai-format = "left_j";
                                bitclock-master;
                                frame-master;

                                playback = <&src0 &ctu02 &mix0 &dvc0 &ssi0>;
                        };
                };
                rsnd_port3: port@3 {
                        reg = <3>;
                        rsnd_endpoint3: endpoint {
                                remote-endpoint = <&ak4613_endpoint3>;

                                dai-format = "left_j";
                                bitclock-master;
                                frame-master;

                                playback = <&src2 &ctu03 &mix0 &dvc0 &ssi0>;
                        };
                };
        };
};
```

Figure 4-51 Description example of MIX setting (R-Car E3)

(2) Example of using

The example of using MIX.

```
$ aplay -D plughw:0,0 xxxx.wav &
$ aplay -D plughw:0,1 yyyy.wav
```

4.8 Amixer control interfaces

This module can adjust the volume and so on by Mixer function.

4.8.1 Control list

The content of the supported control is shown below.

Table 4-4 External function (Standard)[1/2]

Control Name	Overview	Parameter ([*] : initial value)		Remark
DVC In	Capture control DVC	(Left, Right) = (0, 0) [*]	0 - 8388607 (0 - 100 %)	
DVC In Mute	Capture mute	(Left, Right) = (0, 0) [*]	off, off	
		(Left, Right) = (0, 1)	off, on	
		(Left, Right) = (1, 0)	on, off	
		(Left, Right) = (1, 1)	on, on	
DVC In Ramp	Capture volume control	0 [*]	off	
		1	on	
DVC In Ramp Down Rate	Capture volume control	'128 dB/1 step' [*]	Item #0	
		(Other parameters refer to Table 4-10)	Item #1 - #23	
DVC In Ramp Up Rate	Capture volume control	'128 dB/1 step' [*]	Item #0	
		(Other parameters refer to Table 4-10)	Item #1 - #23	
DVC Out	Playback control DVC	(Left, Right) = (0, 0) [*]	0 - 8388607 (0 - 100 %)	
DVC Out Mute	Playback mute	(Left, Right) = (0, 0) [*]	off, off	
		(Left, Right) = (0, 1)	off, on	
		(Left, Right) = (1, 0)	on, off	
		(Left, Right) = (1, 1)	on, on	
DVC Out Ramp	Playback volume control	0 [*]	off	
		1	on	
DVC Out Ramp Down Rate	Playback volume control	128 dB/1 step [*]	Item #0	
		(Other parameters refer to Table 4-10)	Item #1 - #23	
DVC Out Ramp Up Rate	Playback volume control	128 dB/1 step [*]	Item #0	
		(Other parameters refer to Table 4-10)	Item #1 - #23	
Digital Playback Volume1	Playback volume control	255 [*]	0 - 255	
Digital Playback Volume2	Playback volume control	255 [*]	0 - 255	*1
Digital Playback Volume3	Playback volume control	255 [*]	0 - 255	*1
Digital Playback Volume4	Playback volume control	255 [*]	0 - 255	*1
Digital Playback Volume5	Playback volume control	255 [*]	0 - 255	*1
Digital Playback Volume6	Playback volume control	255 [*]	0 - 255	*1

Notes) *1 Target is not connected at R-CarH3-SiP/M3-SiP/M3N-SiP/D3/E3 System Evaluation Board.

Table 4-5 External function (Standard)[2/2]

Control Name	Overview	Parameter ([*] : initial value)		Remark
SRC In Rate	Input Rate of Sampling Rate Conversion	0 [*]	0 - 192000	*1
SRC Out Rate	Output Rate of Sampling Rate Conversion	0 [*]	0 - 192000	*1

*1: If more than one SRC device is valid, specify each with index=0 to 4. ex) ‘SRC Out Rate’,index=2

Table 4-6 External function (CTU)

Control Name	Overview	Parameter ([*] : initial value)		Remark
‘CTU Pass’	Pass mode setting	0	0 - 12	*1
‘CTU Reset’	reset the settings	off	off/on	*1
‘CTU SV0’	Scale value 0 setting	0	0 - 16777215	*1
‘CTU SV1’	Scale value 1 setting	0	0 - 16777215	*1
‘CTU SV2’	Scale value 2 setting	0	0 - 16777215	*1
‘CTU SV3’	Scale value 3 setting	0	0 - 16777215	*1

*1: If more than one CTU device is valid, specify each with index=0 to 4. ex) ‘CTU Pass’,index=2

Table 4-7 External function (MIX)

Control Name	Overview	Parameter ([*] : initial value)		Remark
‘MIX Playback Volume’	Playback Volume Control	1023	0 - 1023	*1
‘MIX Ramp Down Rate’	Ramp down rate control	128 dB/1 step [*] (Other parameters refer to Table 4-11)	Item #0 Item #1 - #10	- -
‘MIX Ramp Switch’	Enable and disable control.	off	off/on	-
‘MIX Ramp Up Rate’	Ramp up rate control.	128 dB/1 step [*] (Other parameters refer to Table 4-11)	Item #0 Item #1 - #10	- -

*1: If more than one MIX device is valid, specify each with index=0 to 4. ex) ‘MIX Playback Volume’,index=2

When audio-graph-scu-card configuration is enabled, codec name is added to the volume control name.

Table 4-8 Case of ak4613 on Salvator-X/ Draak /Ebisu

Control Name	Overview	Parameter ([*] : initial value)		Remark
'ak4613 Digital Playback Volume1',0	Playback volume control	255 [*]	0 - 255	-
'ak4613 Digital Playback Volume2',0	Playback volume control	255 [*]	0 - 255	*1
'ak4613 Digital Playback Volume3',0	Playback volume control	255 [*]	0 - 255	*1
'ak4613 Digital Playback Volume4',0	Playback volume control	255 [*]	0 - 255	*1
'ak4613 Digital Playback Volume5',0	Playback volume control	255 [*]	0 - 255	*1
'ak4613 Digital Playback Volume6',0	Playback volume control	255 [*]	0 - 255	*1

Notes) *1 Target is not connected at R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board.

4.8.2 DVC function

This module can adjust the volume by Mixer function (amixer command), it can be control names 'DVC Out', 'DVC In', 'DVC Out Playback Volume', 'DVC In Capture Volume'. The argument can be a percentage value or a positive integer value. See **Table 4-9** for the relationship between value and decibel.

Table 4-9 DVC volume control value

DVC percentage	dB	DVC (positive integer)	ratio	DVC percentage	dB	DVC (positive integer)	ratio
0%	-infinity	0	0.00	51%	12.21	4278190	4.08
1%	-21.94	83887	0.08	52%	12.38	4362076	4.16
2%	-15.92	167773	0.16	53%	12.55	4445962	4.24
3%	-12.40	251659	0.24	54%	12.71	4529848	4.32
4%	-9.90	335545	0.32	55%	12.87	4613734	4.40
5%	-7.96	419431	0.40	56%	13.03	4697620	4.48
6%	-6.38	503317	0.48	57%	13.18	4781506	4.56
7%	-5.04	587203	0.56	58%	13.33	4865393	4.64
8%	-3.88	671089	0.64	59%	13.48	4949279	4.72
9%	-2.85	754975	0.72	60%	13.62	5033165	4.80
10%	-1.94	838861	0.80	61%	13.77	5117051	4.88
11%	-1.11	922747	0.88	62%	13.91	5200937	4.96
12%	-0.35	1006633	0.96	63%	14.05	5284823	5.04
13%	0.34	1090519	1.04	64%	14.19	5368709	5.12
14%	0.98	1174405	1.12	65%	14.32	5452595	5.20
15%	1.58	1258292	1.20	66%	14.45	5536481	5.28
16%	2.14	1342178	1.28	67%	14.58	5620367	5.36
17%	2.67	1426064	1.36	68%	14.71	5704253	5.44
18%	3.17	1509950	1.44	69%	14.84	5788139	5.52
19%	3.64	1593836	1.52	70%	14.96	5872025	5.60
20%	4.08	1677722	1.60	71%	15.09	5955911	5.68
21%	4.51	1761608	1.68	72%	15.21	6039798	5.76
22%	4.91	1845494	1.76	73%	15.33	6123684	5.84
23%	5.30	1929380	1.84	74%	15.45	6207570	5.92
24%	5.67	2013266	1.92	75%	15.56	6291456	6.00
25%	6.02	2097152	2.00	76%	15.68	6375342	6.08
26%	6.36	2181038	2.08	77%	15.79	6459228	6.16
27%	6.69	2264924	2.16	78%	15.90	6543114	6.24
28%	7.00	2348810	2.24	79%	16.01	6627000	6.32
29%	7.31	2432697	2.32	80%	16.12	6710886	6.40
30%	7.60	2516583	2.40	81%	16.23	6794772	6.48
31%	7.89	2600469	2.48	82%	16.34	6878658	6.56
32%	8.16	2684355	2.56	83%	16.44	6962544	6.64
33%	8.43	2768241	2.64	84%	16.55	7046430	6.72
34%	8.69	2852127	2.72	85%	16.65	7130316	6.80
35%	8.94	2936013	2.80	86%	16.75	7214203	6.88
36%	9.19	3019899	2.88	87%	16.85	7298089	6.96
37%	9.43	3103785	2.96	88%	16.95	7381975	7.04
38%	9.66	3187671	3.04	89%	17.05	7465861	7.12
39%	9.88	3271557	3.12	90%	17.15	7549747	7.20
40%	10.10	3355443	3.20	91%	17.24	7633633	7.28
41%	10.32	3439329	3.28	92%	17.34	7717519	7.36
42%	10.53	3523215	3.36	93%	17.43	7801405	7.44
43%	10.73	3607102	3.44	94%	17.52	7885291	7.52
44%	10.93	3690988	3.52	95%	17.62	7969177	7.60
45%	11.13	3774874	3.60	96%	17.71	8053063	7.68
46%	11.32	3858760	3.68	97%	17.80	8136949	7.76
47%	11.50	3942646	3.76	98%	17.89	8220835	7.84
48%	11.69	4026532	3.84	99%	17.97	8304721	7.92
49%	11.87	4110418	3.92	100%	18.06	8388607	8.00
50%	12.04	4194304	4.00	-	-	-	-

4.8.3 Ramp function

This module supports the Ramp function of MIX and DVC. This function can be used when you define to use MIX module or DVC module for audio route setting. Control of the ramp function is controlled using the Amixer control interface.

Table 4-10 Ramp parameters for DVC

Item number	Ramp parameters
0	'128 dB/1 step'
1	'64 dB/1 step'
2	'32 dB/1 step'
3	'16 dB/1 step'
4	'8 dB/1 step'
5	'4 dB/1 step'
6	'2 dB/1 step'
7	'1 dB/1 step'
8	'0.5 dB/1 step'
9	'0.25 dB/1 step'
10	'0.125 dB/1 step'
11	'0.125 dB/2 steps'
12	'0.125 dB/4 steps'
13	'0.125 dB/8 steps'
14	'0.125 dB/16 steps'
15	'0.125 dB/32 steps'
16	'0.125 dB/64 steps'
17	'0.125 dB/128 steps'
18	'0.125 dB/256 steps'
19	'0.125 dB/512 steps'
20	'0.125 dB/1024 steps'
21	'0.125 dB/2048 steps'
22	'0.125 dB/4096 steps'
23	'0.125 dB/8192 steps'

Table 4-11 Ramp parameters for MIX

Item number	Ramp parameters
0	'128 dB/1 step'
1	'64 dB/1 step'
2	'32 dB/1 step'
3	'16 dB/1 step'
4	'8 dB/1 step'
5	'4 dB/1 step'
6	'2 dB/1 step'
7	'1 dB/1 step'
8	'0.5 dB/1 step'
9	'0.25 dB/1 step'
10	'0.125 dB/1 step'

4.8.4 Example of Control setting

The example of Control setting in R-CarH3-SiP/M3-SiP/M3N-SiP/E3/D3 System Evaluation Board is shown below. Please change the volume to suitable value for the sound source level.

In this example, <wavfile> is for 24-bit data. When using 16-bit data, please do not use "hw" option, please use "plughw" option.

Ex.1) Playback in Memory->SCU(DVC)->SSI->CODEC route

```
$ amixer set "Digital Playback Volume1" 80%
$ amixer set "DVC Out" 12%
$ aplay -D hw:0,0 <wavefile>
```

Note) “-D” option can use over 32kHz.

Ex.2) Playback in Memory->SSI->CODEC route

```
$ amixer set "Digital Playback Volume1" 80%
$ amixer set "DVC Out" 12%
$ aplay -D hw:0,0 <wavefile>
```

Ex.3) Capture in Memory<-SCU(DVC)<-SSI<-CODEC route

```
$ amixer set "DVC In" 12%
$ arecord -D hw:0,0 -t wav -d 5 -c 2 -r 44100 -f S24_LE <wavefile>
```

Ex.4) Slow down the playback of 48000Hz

```
$ amixer cset name='DVC Out' 12%
$ aplay <48KHz-wavefile> &
$ amixer cset name='SRC Out Rate' 52800
```

The amount of data increases, so it becomes slower playback.

Ex.5) Ramp control case of playback

```
$ amixer set "DVC Out" 0%
$ amixer set "Digital Playback Volume1" 100%
$ amixer set "DVC Out Ramp Up Rate"      "0.125 dB/64 steps"
$ amixer set "DVC Out Ramp Down Rate" "0.125 dB/512 steps"
$ amixer set "DVC Out Ramp" on
$ aplay <wavefile> &
$ amixer set "DVC Out" 75%
$ sleep 10
$ amixer set "DVC Out" 0%
```

4.9 Multi-channel Function

This driver supports Multi-channel by Multi-SSI, or TDM-SSI.

4.9.1 Multi-SSI Function

This function supports 6ch case. The SSI of stereo x3 is available.

(1) Device tree file setting

This example of SSI0/SSI1/SSI2 (= for 6ch).

```
&rcar_sound {
        pinctrl-0 = <&sound_pins &sound_clk_pins>;
        pinctrl-names = "default";

        /* Single DAI */
        #sound-dai-cells = <0>;

        ports {
                rsnd_port0: port@0 {
                        rsnd_endpoint0: endpoint {
                                playback = <&ssi0 &ssi1 &ssi2 &src0 &dvc0>;
                                /delete-property/ capture;
                        };
                };
        };
};
```

Figure 4-52 Description example of Multi-channel setting

4.9.2 TDM-SSI Function

SSI0/SSI1/SSI2/SSI3/SSI4/SSI9 supports the TDM format. Audio driver supports sound input/output of 2/6/8 channels.

(1) Device tree file setting

This is example of TDM 6ch.

```
&rcar_sound {
        ...

        ports {
                #address-cells = <1>;
                #size-cells = <0>;
                rsnd_port0: port@0 {
                        reg = <0>;
                        rsnd_endpoint0: endpoint {
                                remote-endpoint = <&ak4613_endpoint>;

                                dai-format = "left_j";
                                bitclock-master = <&rsnd_endpoint0>;
                                frame-master = <&rsnd_endpoint0>;

                                dai-tdm-slot-num = <6>;

                                playback = <&ssi0 &src0 &dvc0>;
                                capture  = <&ssi1 &src1 &dvc1>;
                        };
                };
```

```
            };
    };
```

Figure 4-53 Description example of TDM 6ch setting

4.10 Function Specification

4.10.1 ALSA API

The ALSA API support situation of this module is shown.

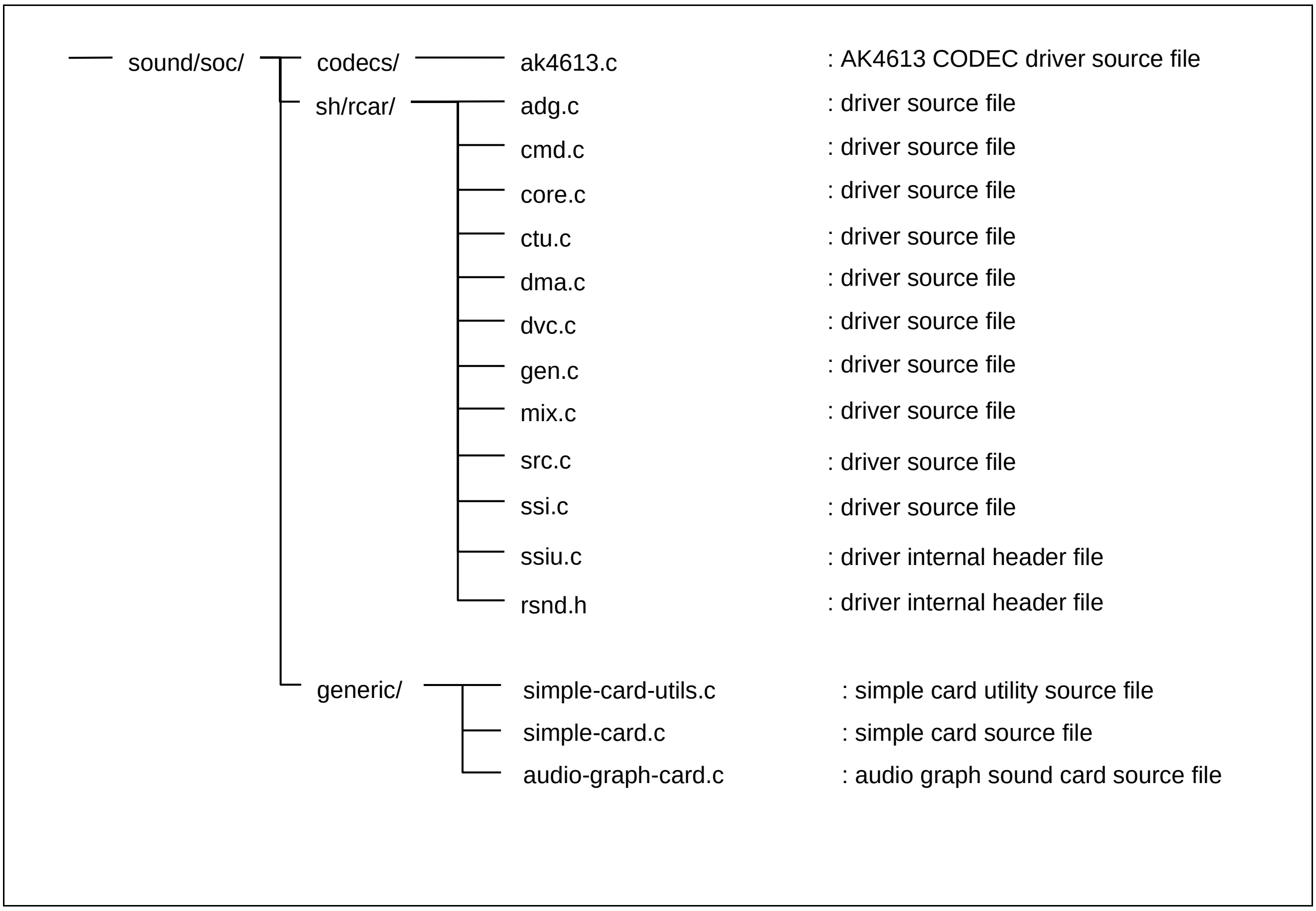
Table 4-12 ALSA API support

API(the C library reference Modules)	Support	Remark
Input Interface	yes	-
Output Interface	yes	-
Error handling	yes	-
Configuration Interface	yes	-
Control Interface	yes	-
PCM Interface: Stream Information	yes	-
PCM Interface: Hardware Parameters	yes	-
PCM Interface: Software Parameters	yes	-
PCM Interface: Access Mask Functions	yes	-
PCM Interface: Format Mask Functions	yes	-
PCM Interface: Status Functions	yes	-
PCM Interface: Description Functions	yes	-
PCM Interface: Debug Functions	yes	-
PCM Interface: Helper Functions	yes	-
PCM Interface: Deprecated Functions	yes	-
Timer Interface	yes	-
Hardware Dependent Interface	-	-
Global defines and functions	-	-
PCM Interface: Sub format Mask Functions	-	-
PCM Interface: Hook Extension	-	-
PCM Interface: Scope Plugin Extension	-	-
PCM Interface: Simple setup functions	-	-
Instrument Interface	-	-
PCM Interface: Direct Access (MMAP) Functions	-	-
Raw Midi Interface	-	-
MIDI Sequencer	-	-
External PCM plugin SDK	-	-
External Control Plugin SDK	-	-

5. Integration

5.1 Directory Configuration

The directory configuration is shown below.



```
sound/soc/ codecs/   ak4613.c                 : AK4613 CODEC driver source file
           sh/rcar/  adg.c                    : driver source file
                     cmd.c                    : driver source file
                     core.c                   : driver source file
                     ctu.c                    : driver source file
                     dma.c                    : driver source file
                     dvc.c                    : driver source file
                     gen.c                    : driver source file
                     mix.c                    : driver source file
                     src.c                    : driver source file
                     ssi.c                    : driver source file
                     ssiu.c                   : driver internal header file
                     rsnd.h                   : driver internal header file
           generic/  simple-card-utils.c      : simple card utility source file
                     simple-card.c            : simple card source file
                     audio-graph-card.c       : audio graph sound card source file
```

Figure 5-1 Directory configuration

5.2 Integration Procedure

To enable the function of this module, make the following setting with Kernel Configuration. AK4613 is automatically chosen at the time of board type selection. This setting also supports sampling rate convert. The fixed sampling rate can be set by "convert-rate" in device tree file.

```
Device Drivers  --->
    <*> Sound card support  --->
        <*> Advanced Linux Sound Architecture  --->
            <*> ALSA for SoC audio support --->
                <*>   ASoC Simple sound card support
                <*>   ASoC Audio Graph sound card support
```

Figure 5-2 Kernel configuration for audio devices

5.3 Option Setting

5.3.1 Module Parameters

There are no module parameters.

5.3.2 Kernel Parameters

There are no kernel parameters.

5.3.3 Device tree bindings

Audio driver supplies the function of statically setting. Please write these setting in a device tree (Refer Table 4-3). See Table 5-1 for binding properties.

Table 5-1 Device tree properties

properties	description
compatible	"renesas,rcar_sound-<soctype>", "renesas,rcar_sound-gen3" Examples with soctypes are: - "renesas,rcar_sound-r8a7795" (R-Car H3) - "renesas,rcar_sound-r8a7796" (R-Car M3) - "renesas,rcar_sound-r8a77965" (R-Car M3N) - "renesas,rcar_sound-r8a77990" (R-Car E3) - "renesas,rcar_sound-r8a77995" (R-Car D3)
reg	Should contain the register physical address. required register is SCU/ADG/SSIU/SSI/Audio-DMAC-pp
rcar_sound,ssi	Should contain SSI feature. The number of SSI sub node should be same as HW. sub node properties: - interrupts : Should contain SSI interrupt - shared-pin : If shared clock pin - dmas : Should contain Audio DMAC entry - dma-names : SSI case "rx" (=playback), "tx" (=capture). SSIU case "rxu" (=playback), "txu" (=capture). - status : SSI case "disabled" don't control module.
rcar_sound,src	Should contain SRC feature. The number of SRC sub node should be same as HW. sub node properties: - interrupts : - dmas : Should contain Audio DMAC entry - dma-names : SSI case "rx" (=playback), "tx" (=capture). SSIU case "rxu" (=playback), "txu" (=capture). - status : SSI case "disabled" don't control module.
rcar_sound,ctu	Should contain CTU feature The number of CTU sub node should be same as HW.
rcar_sound,mix	Should contain MIX feature The number of MIX sub node should be same as HW.
rcar_sound,dvc	Should contain DVC feature The number of DVC sub node should be same as HW. sub node properties: - dmas : Should contain Audio DMAC entry - dma-names : "tx" (=playback/capture)
rcar_sound,dai	DAI contents The number of DAI sub node should be same as HW. sub node properties: - playback : list of playback modules - capture : list of capture modules For audio output/input route path refer to "4.3 Setting route".
#sound-dai-cells	It must be 0 if your system is using single DAI. It must be 1 if your system is using multi DAI.

Table 5-2 Device tree optional properties

properties	description
#clock-cells	It must be 0 if your system has audio_clkout. It must be 1 if your system has audio_clkout0/1/2/3.
clock-frequency	frequency for all audio_clkout0/1/2/3
clkout-lr-asynchronous	Boolean property. It indicates that “audio_clkoutn” is asynchronies with lr-clock.