A DSP/BIOS AIC23 Codec Device Driver for the TMS320C5509 EVM

Software Development Systems

ABSTRACT

This document describes the implementation of a DSP/BIOS device driver for the AIC23 codec on the TMS320C5509 EVM. This device driver is written in conformance to the DSP/BIOS IOM device driver model and APIs. The main functionality of this device driver is to set up the codec. The actual data samples transformation is performed by the generic C5509_DMA_MCBSP driver. This device driver is an interface for the core generic C5509_DMA_MCBSP driver.

Contents

List of Figures

List of Tables

1 Usage

The device driver described here is actually part of an IOM mini-driver. That is, it is implemented as the lower layer of a 2-layer device driver model. The upper layer is the class driver, and can be the DSP/BIOS GIO, SIO/DIO, or PIP/PIO modules, or even some other class driver. The upper layer provides an independent and generic set of APIs and services for a wide variety of mini-drivers and allows the application to use a common interface for I/O requests. Figure 1 shows the overall DSP/BIOS device driver architecture. For more information about the IOM device driver model as well as the GIO, SIO/DIO, and PIP/PIO modules, see the References section.

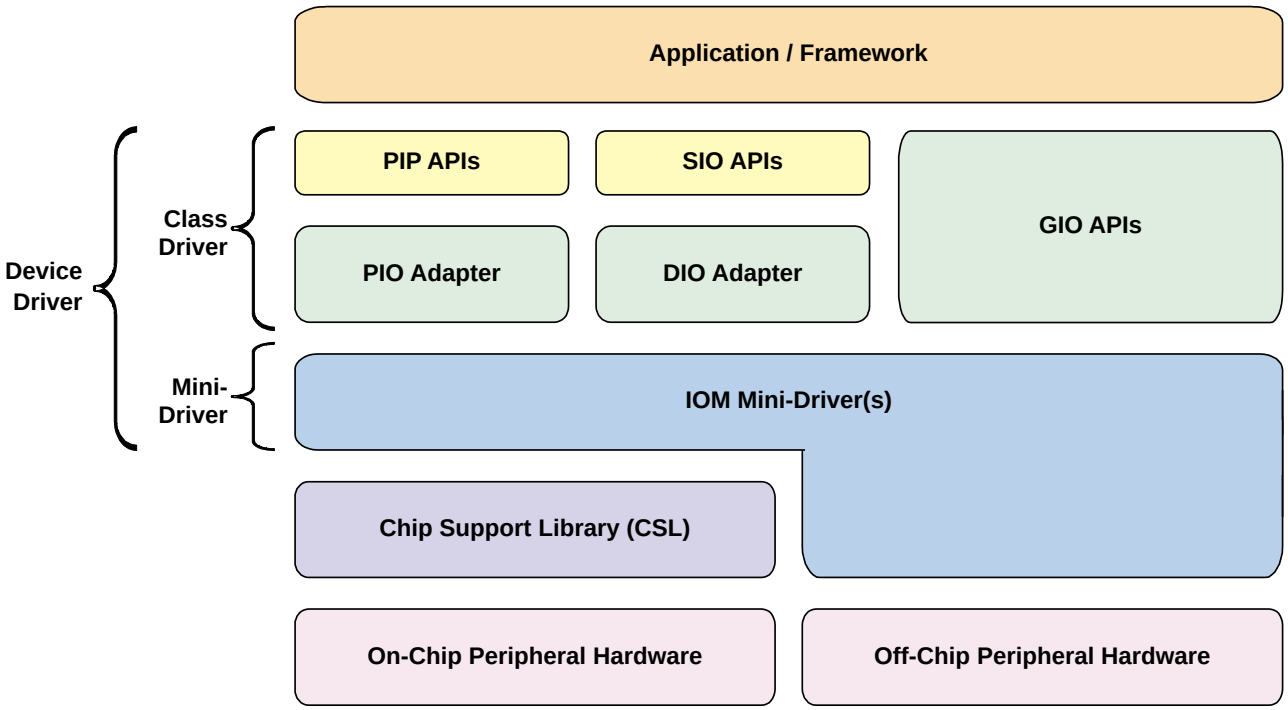

Figure 1. DSP/BIOS IOM Device Driver Model

As shown in Figure 2, mini-drivers are often split into a generic part and a codec-specific part in order to maximize code reusability. This application note only describes the codec-specific part of the mini-driver, which uses the generic C5509_DMA_MCBSP to transfer data to and from the serial ports. Because this codec-specific part is dependent on the generic C5509_DMA_MCBSP, an application must link both the libraries for it to function correctly. These two libraries are called evm5509_dma_aic23.l55 and c5509_dma_mcbsp.l55

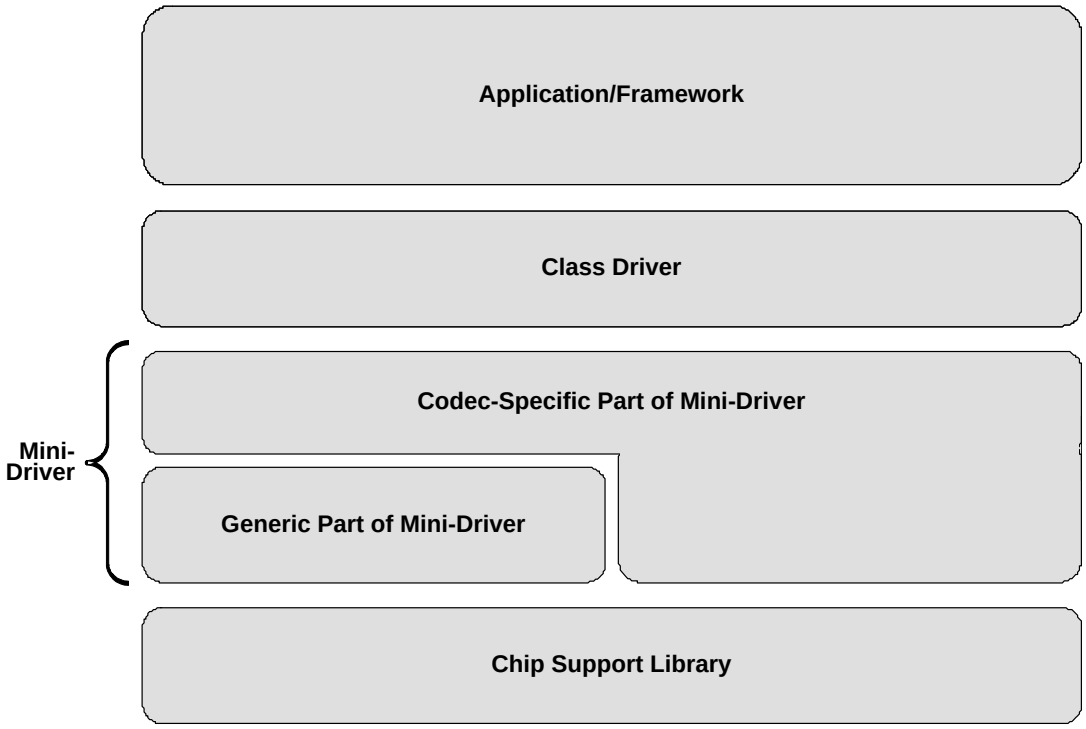

Figure 2. Codec Device Driver Partitioning

1.1 Configuration

To use this device driver, a device entry has to be added and configured in the DSP/BIOS configuration tool. Refer to the *DSP/BIOS Device Driver Developer's Guide* (SPRU616) for more information on how to use the DSP/BIOS configuration tool to configure device drivers. This device driver will set up the generic C5509_DMA_MCBSP driver to meet its needs. The following are the device configuration settings required to use this driver:

- **Init function:** _EVM5509_DMA_AIC23_init.
- **Function table ptr:** _EVM5509_DMA_AIC23_Fxns.
- **Function table type:** IOM_Fxns.
- **Device id:** This property is ignored by this device driver since there is only one AIC23 codec on the TMS320C5509 EVM and it is hard-coded within the driver.
- **Device params ptr:** An optional pointer to an object of type EVM5509_DMA_AIC23_DevParams as defined in the header file evm5509_dma_aic23.h. This pointer will point to a device parameter structure. Set this pointer to NULL to use the default parameters. The parameter structure and defaults are described below.
- **Device global data ptr:** N/A, not used by this driver.

1.2 Device Parameters

```
typedef struct EVM5509_DMA_AIC23_DevParams {
    Int versionId;
    Uns rxDmaId;
    Uns txDmaId;
    Uns dmaPortType;
    AIC23_Params aic23;
    Uns rxIerMask[2];
    Uns txIerMask[2];
} EVM5509_DMA_AIC23_DevParams;
```

- **versionId:** Version number of the driver.
- **rxDmaId:** Receive DMA channel.
- **txDmaId:** Transmit DMA channel.
- **dmaPortType:** Specifies the type of memory to be used by DMA. By default, it uses DARAM. The possible dmaPortType values are shown in Table 1.

Table 1. dmaPortType Values

dmaPortType	Value
EVM5509_DMA_AIC23_PORTTYPE_DARAM	DARAM
EVM5509_DMA_AIC23_PORTTYPE_SARAM	SARAM
EVM5509_DMA_AIC23_PORTTYPE_EMIF	EMIF
EVM5509_DMA_AIC23_PORTTYPE_DEFAULT	DARAM

- **aic23:** The codec registers setup. If the device parameters pointer is NULL, the default parameters are used. Here are the default setups for the registers.
  - **Register 0:** Left input channel volume control. Default value is 0x0017.
  - **Register 1:** Right input channel volume control. Default value is 0x0017.
  - **Register 2:** Left channel headphone volume control. Default value is 0x01F9.
  - **Register 3:** Right channel headphone volume control. Default value is 0x01F9.
  - **Register 4:** Analog audio path control. Default value is 0x0011.
  - **Register 5:** Digital audio path control. Default value is 0x0000.
  - **Register 6:** Power down control. Default value is 0x0000.
  - **Register 7:** Digital audio interface format control. Default value is 0x0043.
  - **Register 8:** Sample rate control. Default value is 0x00C1, which is 48 kHz.
  - **Register 9:** Digital interface activation. Default value is 0x0001.
- **rxIerMask:** Interrupt Enable Register mask, set in receiver ISR.
- **txIerMask:** Interrupt Enable Register mask, set in transmitter ISR.

1.3 Channel Parameters

This device driver does not have any channel parameters. Any values that are passed as channel parameters will be ignored. Specifying NULL is suggested.

1.4 Control Commands

This device driver has no run-time control commands.

2 Architecture

This codec-specific portion of the mini-driver is a layer to the generic C5509_DMA_MCBSP part that transfers the data samples, opens the channels, and so on. The codec-specific part uses two basic functions, mdBindDev() and mdCreateChan(), to bind with the functions from the generic part. The mdBindDev() function uses the I^2C port to set up the codec and passes McBSP port 0 to the generic C5509_DMA_MCBSP mdBindDev() function to handle data control. The mdCreateChan() function specifies the mode of the channel (i.e., input or output) to be opened and then call the generic C5509_DMA_MCBSP mdCreateChan() function to carry out the work of creating the channel.

It is important to note that every sample sent to the codec that has the LSB set will be interpreted as a command. This device driver does not strip the LSB when sending the samples to the codec, so the application layer has to strip the LSB from every sample it sent to the codec device driver.

3 Constraints

- Inherits the constraints of the generic C5509_DMA_MCBSP part of the mini-driver.
- Application must strip the least significant bit (LSB) from the data.

4 References

All these documents are available at the TI Developer's Village.

1. *TMS320C55x Chip Support Library API User's Guide* (SPRU433).
2. *TMS320C5000 DSP/BIOS Application Programming Interface (API) Reference Guide* (SPRU404C).
3. *DSP/BIOS Device Driver Developer's Guide* (SPRU616).
4. *TLV320AIC23 Stereo Audio Codec, 8- to 96-kHz, With Integrated Headphone Amplifier Data Manual*, SLWS106D
5. *A DSP/BIOS Generic DMA McBSP Device Driver for TMS320C5000 DSPs* (SPRA858).

Appendix A Device Driver Data Sheet

A.1 Device Driver Library Name

Evm5509_dma_aic23.l55

c5509_dma_mcbsp.l55 is required for building application.

A.2 DSP/BIOS Modules Used

Refer to the generic C5509_DMA_MCBSP documentation.

A.3 DSP/BIOS Objects Used

Refer to the generic C5509_DMA_MCBSP documentation.

A.4 CSL Modules Used

Refer to the generic C5509_DMA_MCBSP documentation.

A.5 CPU Interrupts Used

Refer to the generic C5509_DMA_MCBSP documentation.

A.6 Peripherals Used

McBSP port 0 is used for data control.

A.7 Interrupt Disable Time

Maximum time that hardware interrupts can be disabled by the driver: refer to the generic C5509_DMA_MCBSP documentation. This measurement is taken using the compiler option –O3.

A.8 Memory Usage

Table A–1. Device Driver Memory Usage

	Uninitialized memory	Initialized memory
CODE	—	324 (8-bit bytes)
DATA	158 (8-bit bytes)	204 (8-bit bytes)

NOTE: This data was gathered using the sectti command utility.
Uninitialized data: .bss
Initialized data: .cinit + .const
Initialized code: .text + .text:init

The sizes in this table were measured on the driver library build for small model on the C5509 EVM platform.

Note that to calculate the total driver overhead, you must include the sizes of the generic C5509_DMA_MCBSP part of the mini-driver as well. These size overheads are listed in the generic part's application note, which is listed in the References section.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265