Application Report
SPRA909A – June 2003

A DSP/BIOS AIC23 Codec Device Driver for the TMS320C6416 DSK

Software Development Systems

ABSTRACT

This document describes the usage and design of a device driver for the AIC23 audio codec on the TMS320C6416 DSK. This device driver is written in conformance to the DSP/BIOS™ IOM device driver model and uses the generic TMS320C6X1X EDMA McBSP driver to transfer samples to and from the serial port. For details on this generic driver, see the application note *A DSP/BIOS EDMA McBSP Device Driver for TMS320C6x1x DSPs* (SPRA846).

Contents

List of Figures

List of Tables

1 Usage

The device driver described here is 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 called the class driver and can be either the DSP/BIOS GIO, SIO/DIO, or PIP/PIO modules. The class driver 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. The diagram below 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 *DSP/BIOS Device Driver Developer's Guide* (SPRU616).

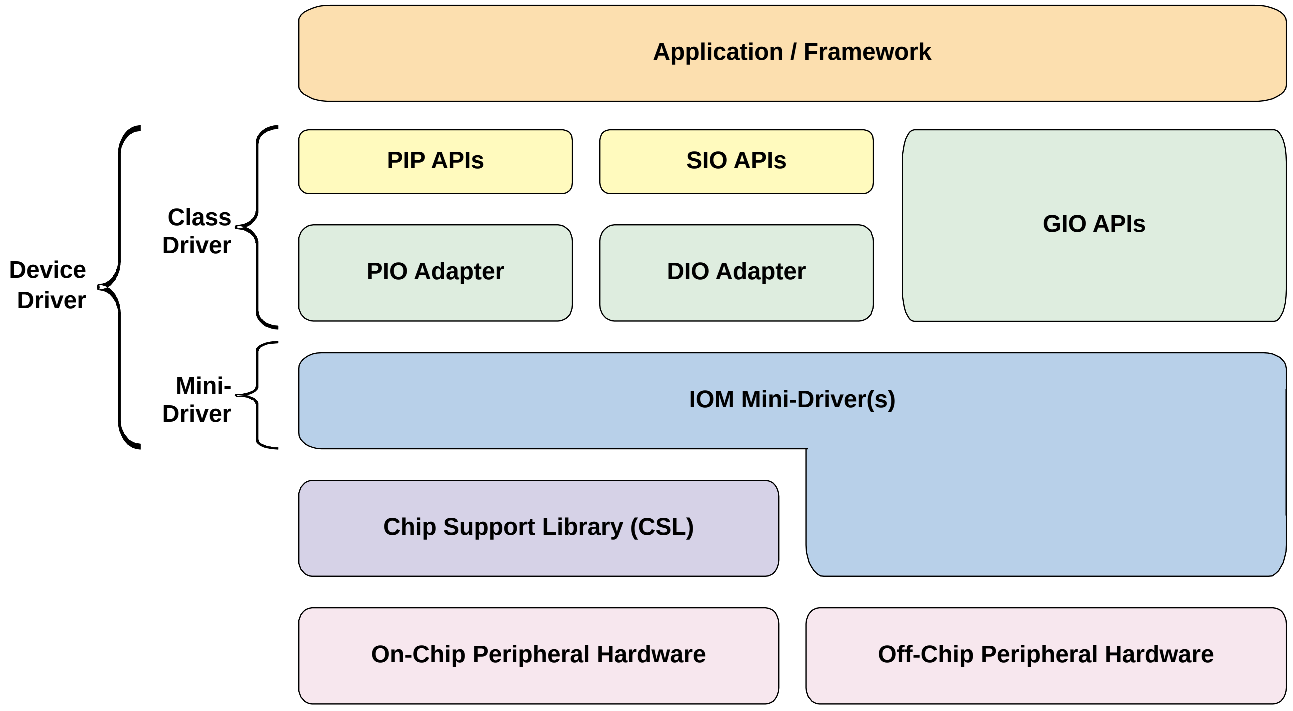

Figure 1. DSP/BIOS IOM Device Driver Model

Many mini-driver implementations split the code into a codec-specific portion and a generic portion that will work across many different codecs. Figure 2 shows the data flow between the components in a system in which the mini-driver is split into a generic part and a codec-specific part. This device driver uses the generic TMS320C6x1x EDMA McBSP device driver to transfer samples to and from the serial port. This means that to use this device driver, an application must not only link with this device driver library (*dsk6416_edma_aic23.l64*), but also with the generic device driver library (*c6x1x_edma_mcbsp.l64*). Other than this, the use of the generic device driver is hidden from the user. These device driver libraries are compiled for TMS320C641x, but can also be used with TMS320C671x. For example, if you are using TMS320C6713 and additional floating-point optimizations are needed, recompile the libraries for TMS320C6713. Note that this device driver uses both McBSP port 1 and port 2 to communicate with the codec, which implies that they cannot be used for any other purposes.

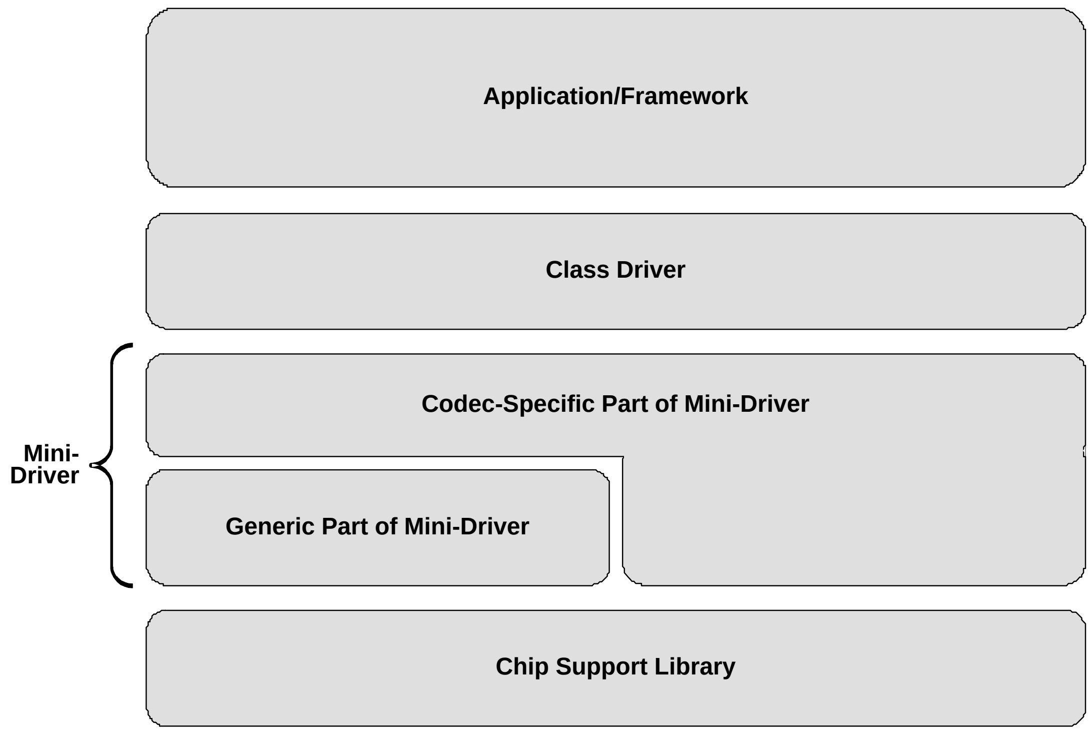

Figure 2. Codec Device Driver Partitioning

1.1 Configuration

To use this driver, a device entry has to be added and configured in the configuration tool. This device driver will set up the generic TMS320C6x1x EDMA McBSP driver to meet its needs.

- **Init function:** Type _DSK6416_EDMA_AIC23_init.
- **Function table ptr:** Type _DSK6416_EDMA_AIC23_Fxns.
- **Function table type:** Select IOM_Fxns.
- **Device id:** This property is ignored by this device driver, since there is only one AIC23 codec on the TMS320C6416 DSK.
- **Device params ptr:** A pointer to your instance of the device parameter structure. Set this property to 0x0 to use the default parameters. The parameter structure and its defaults are described below.
- **Device global data ptr:** This property must be set to 0x0.

1.2 Device Parameters

```
typedef struct DSK6416_EDMA_AIC23_DevParams {
    Int versionId;
    Bool cacheCalls;
    Int irqId;
    AIC23_Params aic23;
    Uns intrMask;
    Uns edmaPriority;
} DSK6416_EDMA_AIC23_DevParams;
```

- **versionId:** Version number of the driver.
- **cacheCalls:** If this parameter is set to TRUE, the device driver will treat buffers issued to any IOM channel associated with the device as if they are in cacheable memory and the L2 data cache is enabled. The default value of this parameter is TRUE.
- **irqId:** This parameter selects which IRQ number to use for the EDMA interrupt. The system default is 8. The default parameter of this value is 8.
- **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 0x0081.
  - **Register 9:** Digital interface activation. Default value is 0x0001.
  - **intrMask:** Interrupt mask, set in the ISR.
  - **edmaPriority:** Priority queue, to use for all EDMA transfers.

1.3 Channel Parameters

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

1.4 Control Commands

This device driver has no run-time control commands.

2 Architecture

The codec specific portion of the mini-driver inherits the features of the generic TMS320C6x1x EDMA McBSP driver. It uses two codec specific functions, mbBindDev() and mdCreateChan(), to do the 6416 DSK and AIC23 specific setup. These functions then call mbBindDev() and mdCreateChan() in the generic driver to complete generic portions of the driver initialization. The only thing the codec-specific part does is to set up the codec and leaves the transfers of samples to the generic device driver. The fact that this device driver uses the generic device driver is hidden from the user in all aspects except that the generic device driver library has to be linked into the application.

The AIC23 has a control channel which is used to configure the codec as well as a bidirectional data channel. On the DSK, McBSP1 is used to generate the SPI format control channel signals while McBSP2 is used for audio data.

The function mdBindDev() is responsible for configuring the codec through the control channel based on the DSK6416_EDMA_AIC23_DevParams structure that is passed in. It does some basic setup then calls a function called AIC23_setParams() function in aic23.c to do most of the real configuration work. The function mdCreateChan() generates the EDMA configuration used for the data transfers. The configuration is written to match the DSP data format mode of the codec with 16-bit stereo data samples.

3 Constraints

- Inherits the constraints of the generic TMS320C6x1x EDMA McBSP driver.
- By default the McBSP signals on the DSK are connected to the AIC23 codec. However, the McBSP signals can be re-routed to the expansion headers for use by daughtercards. The routing is set by bits in the MISC CPLD register on the DSK. The driver assumes that the default on-board routing is used and does not change the CPLD. See the DSK help file for more details on the CPLD registers.

4 References

All these documents are available on the TI Developer’s Village.

1. *A DSP/BIOS EDMA McBSP Device Driver for TMS320C6x1x DSPs* (SPRA846)
2. *TLV320AIC23 Stereo Audio Codec, 8 to 96 KHz, With Integrated Headphone Amplifier Data Manual*, SLWS106D
3. *DSP/BIOS Driver Developer’s Guide* (SPRU616).
4. *TMS320C6000 Chip Support Library API Reference Guide* (SPRU401)
5. *TMS320C6000 Peripherals Reference Guide* (SPRU190)
6. *TMS320C6000 DSP/BIOS Application Programming Interface (API) Reference Guide* (SPRU403)

Appendix A Device Driver Data Sheet

A.1 Device Driver Library Name

dsk6416_edma_aic23.l64

When building an application the generic c6x1x_edma_mcbsp.l64 library is required.

A.2 DSP/BIOS Modules Used

Same as for the generic TMS320C6x1x EDMA McBSP device driver.

A.3 DSP/BIOS Objects Used

Same as for the generic TMS320C6x1x EDMA McBSP device driver.

A.4 CSL Modules Used

Same as for the generic TMS320C6x1x EDMA McBSP device driver.

A.5 CPU Interrupts Used

Same as for the generic TMS320C6x1x EDMA McBSP device driver.

A.6 Peripherals Used

Same as for the generic TMS320C6x1x EDMA McBSP device driver.

A.7 Maximum Interrupt Latency

Same as for the generic TMS320C6x1x EDMA McBSP device driver.

A.8 Memory Usage

Includes the memory usage of the generic TMS320C6x1x EDMA McBSP device driver.

Table A–1. Device Driver Memory Usage

	Uninitialized memory	Initialized memory
CODE	—	1056 words
DATA	224 words	272 words

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