Rockchip Developer Guide Linux FLEXBUS ADC and DAC Mode

ID: RK-KF-YF-C12

Release Version: V1.0.0

Release Date: 2024-06-11

Security Level: ☐Top-Secret ☐Secret ☐Internal ■Public

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document introduces how to use FLEXBUS ADC mode and DAC mode on Linux.

Product Version

Chipset	Kernel Version
RK3576	6.1

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	Wesley Yao	2024-06-11	Initial version

Contents

1. FLEXBUS ADC mode

1.1 Overview

FLEXBUS ADC mode is **FLEXBUS1 interfacing high-speed parallel ADC devices** such as TI ADS6144 (https://www.ti.com/product/ADS6144) and ADI LTC2207 (https://www.analog.com/en/products/ltc2207.html). **ADC devices with interface types such as SPI and I2C are not supported**.

- Clock frequency up to 100MHz
- Resolution up to 16-Bit

1.2 Configuration

1.2.1 Hardware configuration

- The LSB of the ADC device needs to be connected to FLEXBUS1_D0, such as 10-Bit ADC to FLEXBUS1_D[9:0], 16-Bit ADC to FLEXBUS1_D[15:0]
- Slave mode indicates that the device provides CLK to FLEXBUS1, and master mode indicates that FLEXBUS1 provides CLK to the device

1.2.2 Kernel configuration

FLEXBUS ADC mode relies on the iio/adc framework.

Device Drivers -> Multifunction device drivers -> Rockchip Flexbus

Device Drivers -> Industrial I/O support -> Analog to digital converters -> Rockchip Flexbus ADC opmode driver

1.2.3 dtsi configuration

Take RK3576 platform and RK3576 TEST1 board, FLEXBUS1 for ADC as an example.

In rk3576.dtsi:

```
flexbus: flexbus@2a2f0000 {
    ......

    flexbus_adc: adc {
        compatible = "rockchip,flexbus-adc";
        #io-channel-cells = <0>;
        rockchip,slave-mode;    // Configuring this is the slave mode, not
configuring this is the master mode
```

```
        rockchip,free-sclk; // Configuring this is clock keeps output, not
configuring this is clock follows data
        rockchip,auto-pad;  // Default configuration
        rockchip,cpol;  // Configuring this is CPOL = 1, not configuring this is
0
        rockchip,cpha;  // Configuring this is CPHA = 1, not configuring this is
0
        rockchip,dfs = <16>;    // data frames
        status = "disabled";
    };

    flexbus_dac: dac {
        ......
    };
};
```

- rockchip,slave-mode: With configuring this is slave mode, the device provides CLK to FLEXBUS1; Without configuring this is master mode, FLEXBUS1 provides CLK to the device
- rockchip,free-sclk: With configuring this, clock keeps output; Without configuring this, clock follow data (this configuration is only valid in master mode)
- rockchip,cpol, rockchip,cpha: Same as SPI protocol definitions, configured according to the timing diagram in the device datasheet
- rockchip,dfs: data frames, only support configuration of 4, 8, 16; If the resolution of the device is not 4, 8 or 16, it needs to be rounded up; For example, when using ADC devices with 14-Bit resolution, rockchip,dfs needs to be configured to 16, and only 14bit data is obtained through software processing when receiving data

In arch/arm64/boot/dts/rockchip/rk3576-test1.dtsi:

```
&flexbus {
    rockchip,flexbus0-opmode = <ROCKCHIP_FLEXBUS0_OPMODE_xxx>;
    rockchip,flexbus1-opmode = <ROCKCHIP_FLEXBUS1_OPMODE_ADC>;  // FLEXBUS1
selects ADC mode
    status = "okay";    // Enable FLEXBUS
};

&flexbus_adc {
    pinctrl-names = "default";
    pinctrl-0 = <&flexbus1m4_csn &flexbus1_clk
             &flexbus1_d0 &flexbus1_d1 &flexbus1_d2 &flexbus1_d3
             &flexbus1_d4 &flexbus1_d5 &flexbus1_d6 &flexbus1_d7
             &flexbus1_d8 &flexbus1_d9 &flexbus1_d10 &flexbus1_d11
             &flexbus1m1_d12 &flexbus1m1_d13 &flexbus1m1_d14 &flexbus1m1_d15>;
// Configure IOMUX for FLEXBUS1 ADC mode
    status = "okay";    // Enable ADC mode
};
```

1.2.4 Driver file

The driver file is drivers/iio/adc/rockchip-flexbus-adc.c.

rockchip_flexbus_adc_read_block() is a function of reading ADC device data. The main operations are as follows:

1. rockchip_flexbus_writel(rkfb, FLEXBUS_RX_NUM, num_of_dfs);

   Configure the number of RX in dfs (rockchip,dfs in dtsi)

2. rockchip_flexbus_writel(rkfb, FLEXBUS_DMA_DST_ADDR0, (ulong)dst_phys >> 2);

   Configure physical address of dst buffer

3. rockchip_flexbus_writel(rkfb, FLEXBUS_DMA_DST_LEN0, dst_len);

   Configure length of dst buffer

4. rockchip_flexbus_writel(rkfb, FLEXBUS_ENR, FLEXBUS_RX_ENR);

   Enable RX transfer

5. wait_for_completion_timeout(&rkfb_adc->completion, FLEXBUS_ADC_TIMEOUT)

   Wait for the RX transfer to complete and generate an interrupt (the interrupt handler is rockchip_flexbus_adc_isr())

6. rockchip_flexbus_writel(rkfb, FLEXBUS_ENR, FLEXBUS_RX_DIS);

   Disable RX transfer

1.3 Common interface

1.3.1 Check the device corresponding to FLEXBUS ADC

For example:

```
root@rk3576-buildroot:/# cat /sys/bus/iio/devices/iio\:device0/name
2a2f0000.flexbus:adc
```

FLEXBUS ADC corresponds to iio:device0.

1.3.2 Get ADC value

```
root@rk3576-buildroot:/# cd /sys/bus/iio/devices/iio\:device0
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device0# cat in_voltage_raw
33004
```

1.3.3 Get and change clock frequency

Get clock frequency:

```
root@rk3576-buildroot:/# cd /sys/bus/iio/devices/iio\:device0
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device0# cat
in_voltage_sampling_frequency
99000000
```

The current clock frequency is 99MHz.

Change clock frequency:

```
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device0# echo 25000000 >
in_voltage_sampling_frequency
```

The clock frequency is changed to 25MHz。

Note:

- Only master mode supports getting and changing clock frequency. The CLK of slave mode comes from the ADC device
- Clock frequency up to 100MHz

2. FLEXBUS DAC mode

2.1 Overview

FLEXBUS DAC mode is **FLEXBUS0 interfacing high-speed parallel DAC devices** such as ADI max5885 (https://www.analog.com/en/products/max5885.html) and ADI AD9744 (https://www.analog.com/en/products/ad9744.html). **DAC devices with interface types such as SPI and I2C are not supported**.

- Clock frequency up to 100MHz
- Resolution up to 16-Bit

2.2 Configuration

2.2.1 Hardware configuration

- The LSB of the DAC device needs to be connected to FLEXBUS0_D0, such as 10-Bit DAC to FLEXBUS0_D[9:0], 16-Bit DAC to FLEXBUS0_D[15:0]

2.2.2 Kernel configuration

FLEXBUS DAC mode relies on the iio/dac framework.

Device Drivers -> Multifunction device drivers -> Rockchip Flexbus

Device Drivers -> Industrial I/O support -> Digital to analog converters -> Rockchip Flexbus DAC opmode driver

2.2.3 dtsi configuration

Take RK3576 platform and RK3576 TEST1 board, FLEXBUS0 for DAC as an example.

In rk3576.dtsi:

```
flexbus: flexbus@2a2f0000 {
    ......

    flexbus_adc: adc {
        ......
    };

    flexbus_dac: dac {
        compatible = "rockchip,flexbus-dac";
        #io-channel-cells = <0>;
        rockchip,free-sclk; // Configuring this is clock keeps output, not
configuring this is clock follows data
        rockchip,cpol;  // Configuring this is CPOL = 1, not configuring this is
0
        rockchip,cpha;  // Configuring this is CPHA = 1, not configuring this is
0
        rockchip,dfs = <16>;    // data frames
        status = "disabled";
    };
};
```

- rockchip,free-sclk: With configuring this, clock keeps output; Without configuring this, clock follow data
- rockchip,cpol, rockchip,cpha: Same as SPI protocol definitions, configured according to the timing diagram in the device datasheet
- rockchip,dfs: data frames, only support configuration of 4, 8, 16; If the resolution of the device is not 4, 8 or 16, it needs to be rounded up; For example, when using DAC devices with 14-Bit resolution, rockchip,dfs needs to be configured to 16, and bit15 and bit14 of the sent data are replaced by 0.

In arch/arm64/boot/dts/rockchip/rk3576-test1.dtsi:

```
&flexbus {
    rockchip,flexbus0-opmode = <ROCKCHIP_FLEXBUS0_OPMODE_DAC>;  // FLEXBUS0
selects DAC mode
    rockchip,flexbus1-opmode = <ROCKCHIP_FLEXBUS1_OPMODE_xxx>;
    status = "okay";    // Enable FLEXBUS
```

```
};

&flexbus_dac {
    pinctrl-names = "default";
    pinctrl-0 = <&flexbus0m4_csn &flexbus0_clk
             &flexbus0_d0 &flexbus0_d1 &flexbus0_d2 &flexbus0_d3
             &flexbus0_d4 &flexbus0_d5 &flexbus0_d6 &flexbus0_d7
             &flexbus0_d8 &flexbus0_d9 &flexbus0_d10 &flexbus0_d11
             &flexbus0_d12 &flexbus0m0_d13 &flexbus0m0_d14 &flexbus0m0_d15>;
// Configure IOMUX for FLEXBUS0 DAC mode
    status = "okay";    // Enable DAC mode
};
```

2.2.4 Driver file

The driver file is drivers/iio/dac/rockchip-flexbus-dac.c.

rockchip_flexbus_dac_write_block() is a function of sending data to DAC device. The main operations are as follows:

1. rockchip_flexbus_writel(rkfb, FLEXBUS_TX_NUM, num_of_dfs);

   Configure the number of TX in dfs (rockchip,dfs in dtsi)

2. rockchip_flexbus_writel(rkfb, FLEXBUS_TXWAT_START, val);

   Configure waterline

3. rockchip_flexbus_writel(rkfb, FLEXBUS_DMA_SRC_ADDR0, (ulong)src_phys >> 2);

   Configure physical address of src buffer

4. rockchip_flexbus_writel(rkfb, FLEXBUS_DMA_SRC_LEN0, src_len);

   Configure length of src buffer

5. rockchip_flexbus_writel(rkfb, FLEXBUS_ENR, FLEXBUS_TX_ENR);

   Enable TX transfer

6. wait_for_completion_timeout(&rkfb_dac->completion, FLEXBUS_DAC_TIMEOUT)

   Wait for the TX transfer to complete and generate an interrupt (the interrupt handler is rockchip_flexbus_dac_isr())

7. rockchip_flexbus_writel(rkfb, FLEXBUS_ENR, FLEXBUS_TX_DIS);

   Disable TX transfer

2.3 Common interface

2.3.1 Check the device corresponding to FLEXBUS DAC

For example:

```
root@rk3576-buildroot:/# cat /sys/bus/iio/devices/iio\:device2/name
2a2f0000.flexbus:dac
```

FLEXBUS DAC corresponds to iio:device1.

2.3.2 Send data to DAC

```
root@rk3576-buildroot:/# cd /sys/bus/iio/devices/iio\:device0
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device2# echo 16383 >
out_voltage_raw
```

2.3.3 Get and change clock frequency

Get clock frequency:

```
root@rk3576-buildroot:/# cd /sys/bus/iio/devices/iio\:device2
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device2# cat
out_voltage_sampling_frequency
99000000
```

The current clock frequency is 99MHz.

Change clock frequency:

```
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device2# echo 25000000 >
out_voltage_sampling_frequency
```

The clock frequency is changed to 25MHz。

Note:

- Clock frequency up to 100MHz