应用指南

32-bit Cortex™-M0 MCU
NuMicro® Family

NUC1xx中怎样使用I2S和编解码器

应用指南

目录

1 简介

本文档描述了示例程序“Smpl_DrvI2S”并说明了如何通过相关IP驱动使用I2S接口和音频编解码器

1.1 概览

本文对程序员说明了I2S IP在音频方面的应用。这里的音频编解码器使用的是WAU8822，假设读者已经熟悉WAU8822的相关性能和设定

1.2 性能

- I2S可以作为主设备或者从设备
- 可以处理8, 16, 24 及32位数据
- 支持单声道及立体声音频数据
- 支持标准 I2S 及 MSB 对准数据格式
- 提供两个8字长深度的FIFO，一个用来发送另一个用来接收
- 当FIFO水平穿越一个可编程的边界时会产生一个中断请求
- 两路DMA请求，一个用来发送另一个用来接收

1.3 结构

I2S包含MCLK ,LRCLK 以及 BCLK。MCLK产生时钟脉冲到音频编解码器，当I2S作为主模式时BCLK和LRCLK产生相应的采样率信号

1.3.1 I2S 框图

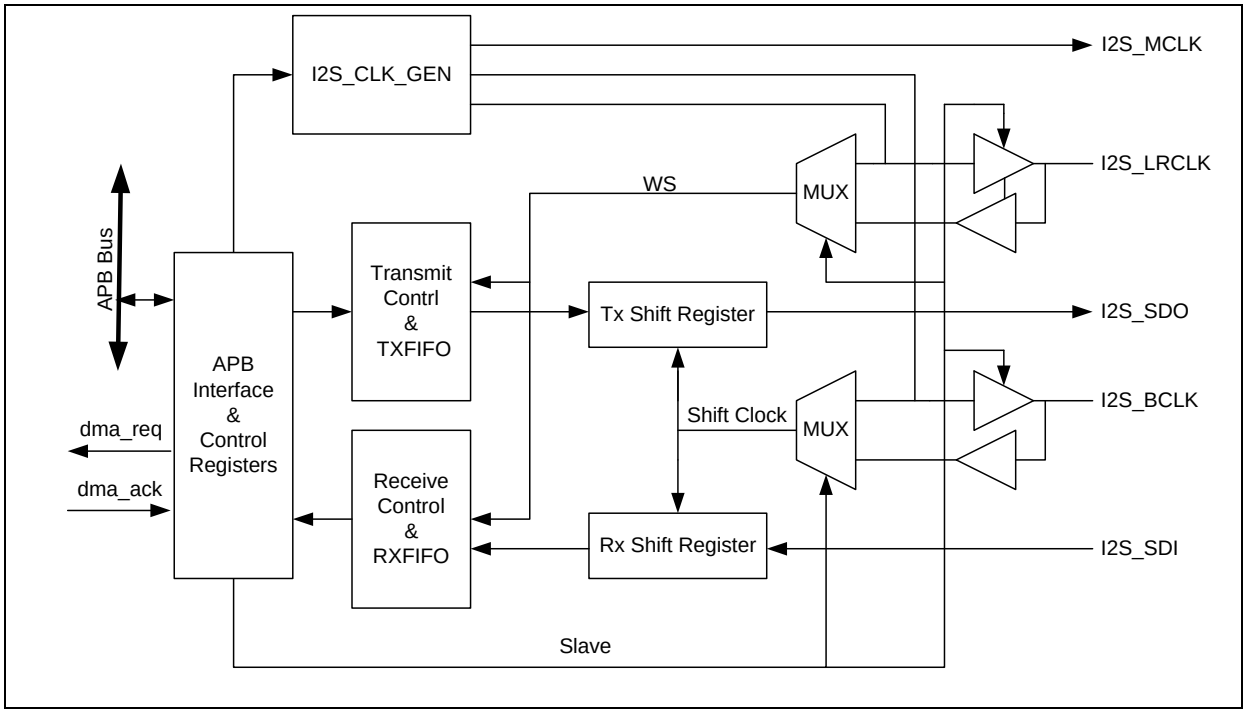

图1 I2S 控制器框图

1.3.2 I2S 操作时序

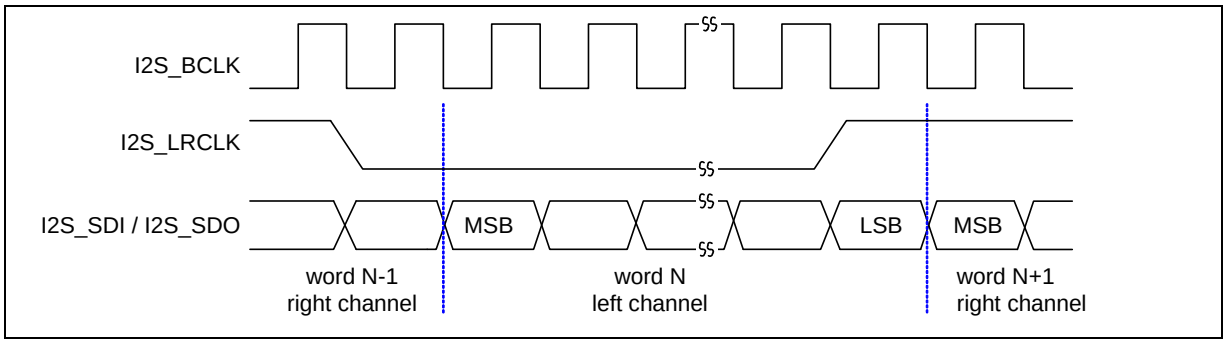

图2 I2S 总线时序图 (Format =0)

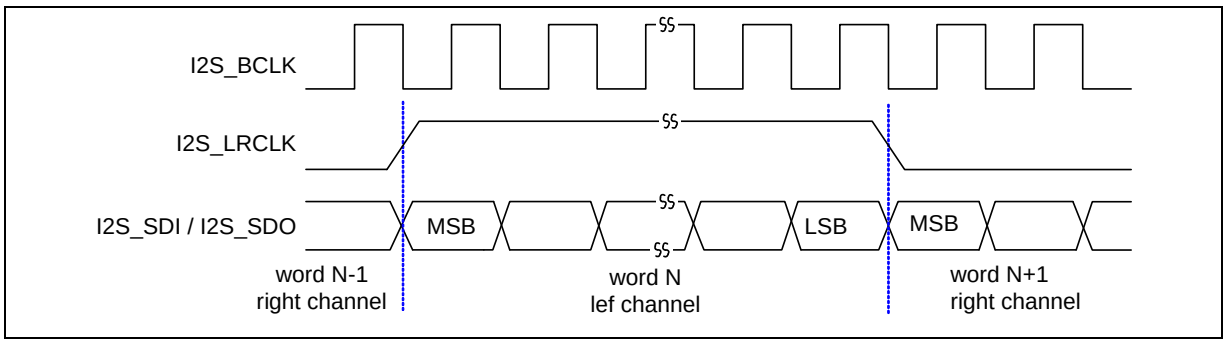

图3 MSB 对准时序图(Format=1)

1.3.3 FIFO 操作

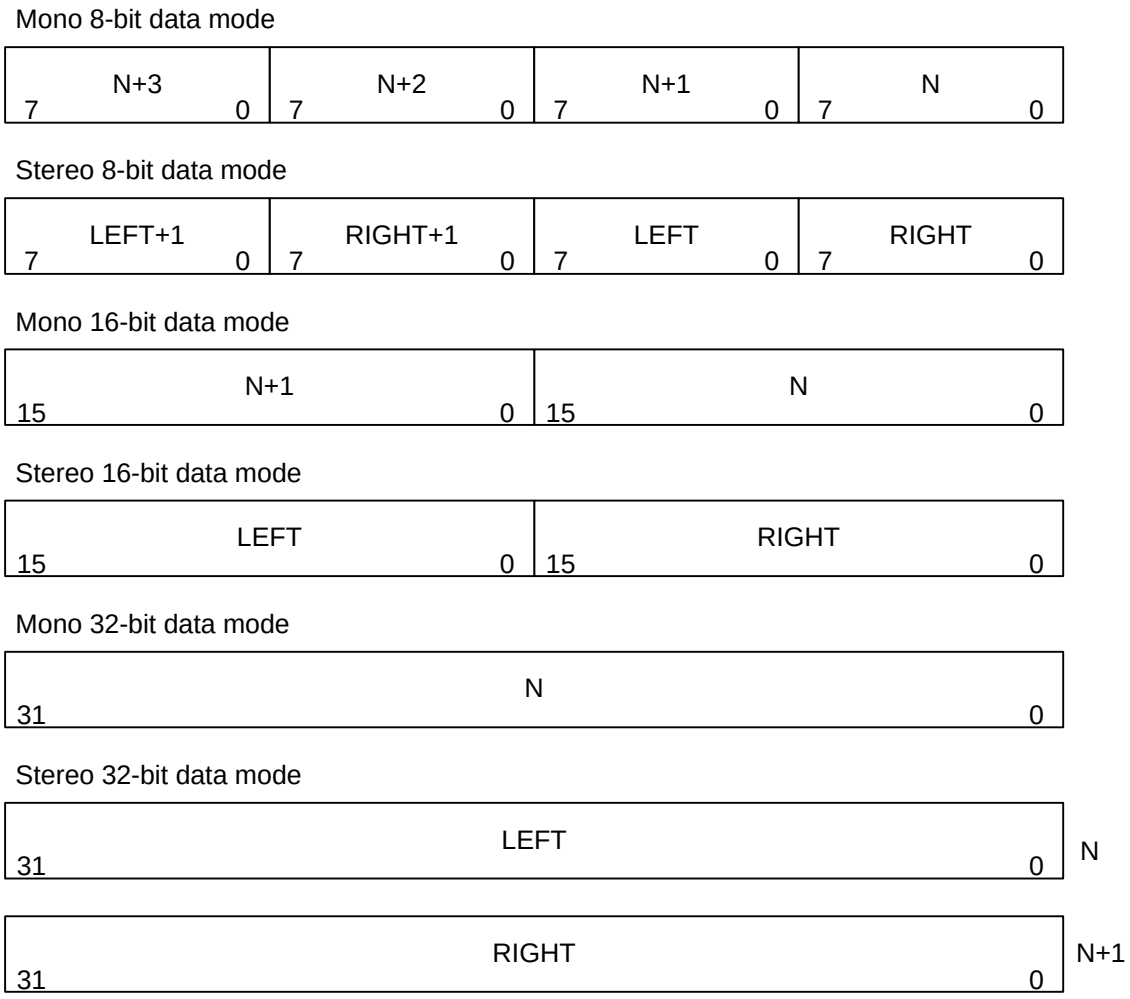

图4 各种I2S模式的FIFO内容

2 代码部分

2.1 主函数 (in Smpl_DrvI2S.c)

关于调用顺序请参考调用顺序章节

这个示例程序说明了I2S和编解码器在音频方面的应用，I2S从WAU8822的ADC接收16位的音频数据，并发送数据到Rx FIFO，当RX FIFO中的数据达到门槛值的时候，I2S中断将请求MCU传输数据到音频数据缓存，并且，当Tx FIFO中的数据低于门槛值的时候，I2S中断还要求MCU从音频数据缓存中传送数据到Tx FIFO，然后I2S发送16位音频数据到WAU8822得DAC.

在主函数中，配置WAU8822编解码器和I2C0的接口，硬件初始化后DrvI2S_Open()用来配置I2S模式，数据格式，数据宽度，FIFO门槛值等等。由于WAU8822可以驱动LRCLK和BCLK提供精确采样率信号，I2S将运行于从模式，当编解码器和I2S设定完成后，MCLK将被使能并给WAU8822产生主时钟

```
int32_t main (void)
{
        uint32_t u32clock, u32HCLK;
        __IO uint32_t u32startFlag = 1;

        STR_UART_T sParam;
        S_DRVI2S_DATA_T st;

        /* Set I2S I/O */
    DrvGPIO_InitFunction(FUNC_I2S);

        /* Set I2C I/O */
    DrvGPIO_InitFunction(FUNC_I2C0);

        /* Set UART I/O */
        DrvGPIO_InitFunction(FUNC_UART0);

        /* UART Setting */
    sParam.u32BaudRate              = 115200;
    sParam.u8cDataBits              = DRVUART_DATABITS_8;
    sParam.u8cStopBits          = DRVUART_STOPBITS_1;
    sParam.u8cParity            = DRVUART_PARITY_NONE;
    sParam.u8cRxTriggerLevel = DRVUART_FIFO_1BYTES;
```

```
        /* Set UART Configuration */
        DrvUART_Open(UART_PORT0, &sParam);

        printf("+-----------------------------------------------------------------------+\n");
printf("|                  I2S Driver Sample Code with WAU8822                  |\n");
printf("+-----------------------------------------------------------------------+\n");

        //u32HCLK = DrvSYS_GetHCLK() * 1000;
        u32HCLK = 12000000;

        /* Open I2C0 and I2C1, and set clock = 100Kbps */
        DrvI2C_Open(I2C_PORT0, u32HCLK, 100000);

        /* Get I2C0 clock */
        u32clock = DrvI2C_GetClock(I2C_PORT0, u32HCLK);
        printf("I2C0 clock %d Hz\n", u32clock);

        /* Enable I2C0 interrupt and set corresponding NVIC bit */
        DrvI2C_EnableInt(I2C_PORT0);

        st.u32SampleRate        = 16000;
st.u8WordWidth                  = DRVI2S_DATABIT_16;
st.u8AudioFormat      = DRVI2S_STEREO;
        st.u8DataFormat         = DRVI2S_FORMAT_I2S;
st.u8Mode                       = DRVI2S_MODE_SLAVE;
st.u8TxFIFOThreshold = DRVI2S_FIFO_LEVEL_WORD_0;
st.u8RxFIFOThreshold = DRVI2S_FIFO_LEVEL_WORD_8;

        DrvI2S_Open(&st);

        WAU8822_Setup();

        /* Set MCLK and enable MCLK */
        DrvI2S_SetMCLK(12000000);
```

```
        u32clock = DrvI2S_GetMCLK();
        printf("MCLK %d\n", u32clock);
        DrvI2S_EnableMCLK(TRUE);

        /* Enable Rx threshold level interrupt and install its callback function */
        DrvI2S_EnableInt(I2S_RX_FIFO_THRESHOLD, Rx_thresholdCallbackfn);

        /* Enable I2S Rx function to receive data */
        DrvI2S_EnableRx(TRUE);

        while(1)
        {
                if (u32startFlag)
                {
                        /* Enable I2S Tx function to send data when data in the buffer
is more than half of buffer size */
                        if (u32BuffPos >= BUFF_LEN/2)
                        {
                                DrvI2S_EnableInt(I2S_TX_FIFO_THRESHOLD,
Tx_thresholdCallbackfn);
                                DrvI2S_EnableTx(TRUE);
                                u32startFlag = 0;
                        }
                }
        }
}
```

2.2 I2C 配置

请参考I2C示例程序和相关文档

2.3 编解码器 WAU8822 配置

详细设定请参考WAU8822 设计指南

```
void WAU8822_Setup()
{
        printf("WAU8822 Setup\n");

        I2C_WriteWAU8822(0,  0x000);   /* Reset all registers */
        Delay(0x200);

        I2C_WriteWAU8822(1,  0x02F);
        I2C_WriteWAU8822(2,  0x1B3);   /* Enable L/R Headphone, ADC Mix/Boost, ADC */

        I2C_WriteWAU8822(3,  0x00F);   /* Enable L/R main mixer, DAC */

        I2C_WriteWAU8822(4,  0x010);   /* 16-bit word length, I2S format, Stereo */

        I2C_WriteWAU8822(5,  0x000);   /* Companding control and loop back mode (all disable) */

        I2C_WriteWAU8822(6,  0x1AD);   /* Divide by 6, 16K */

        I2C_WriteWAU8822(7,  0x006);   /* 16K for internal filter cofficients */

        I2C_WriteWAU8822(10, 0x008);   /* DAC softmute is disabled, DAC oversampling rate is 128x */

        I2C_WriteWAU8822(14, 0x108);   /* ADC HP filter is disabled, ADC oversampling rate is 128x */

        I2C_WriteWAU8822(15, 0x1EF);   /* ADC left digital volume control */
        I2C_WriteWAU8822(16, 0x1EF);   /* ADC right digital volume control */

        I2C_WriteWAU8822(44, 0x000);   /* LLIN/RLIN is not connected to PGA */

        I2C_WriteWAU8822(47, 0x050);   /* LLIN connected, and its Gain value */
        I2C_WriteWAU8822(48, 0x050);   /* RLIN connected, and its Gain value */
```

```
        I2C_WriteWAU8822(50, 0x001);   /* Left DAC connected to LMIX */
        I2C_WriteWAU8822(51, 0x001);   /* Right DAC connected to RMIX */
}
```

3 调用顺序

3.1 I2S和编解码器的集成

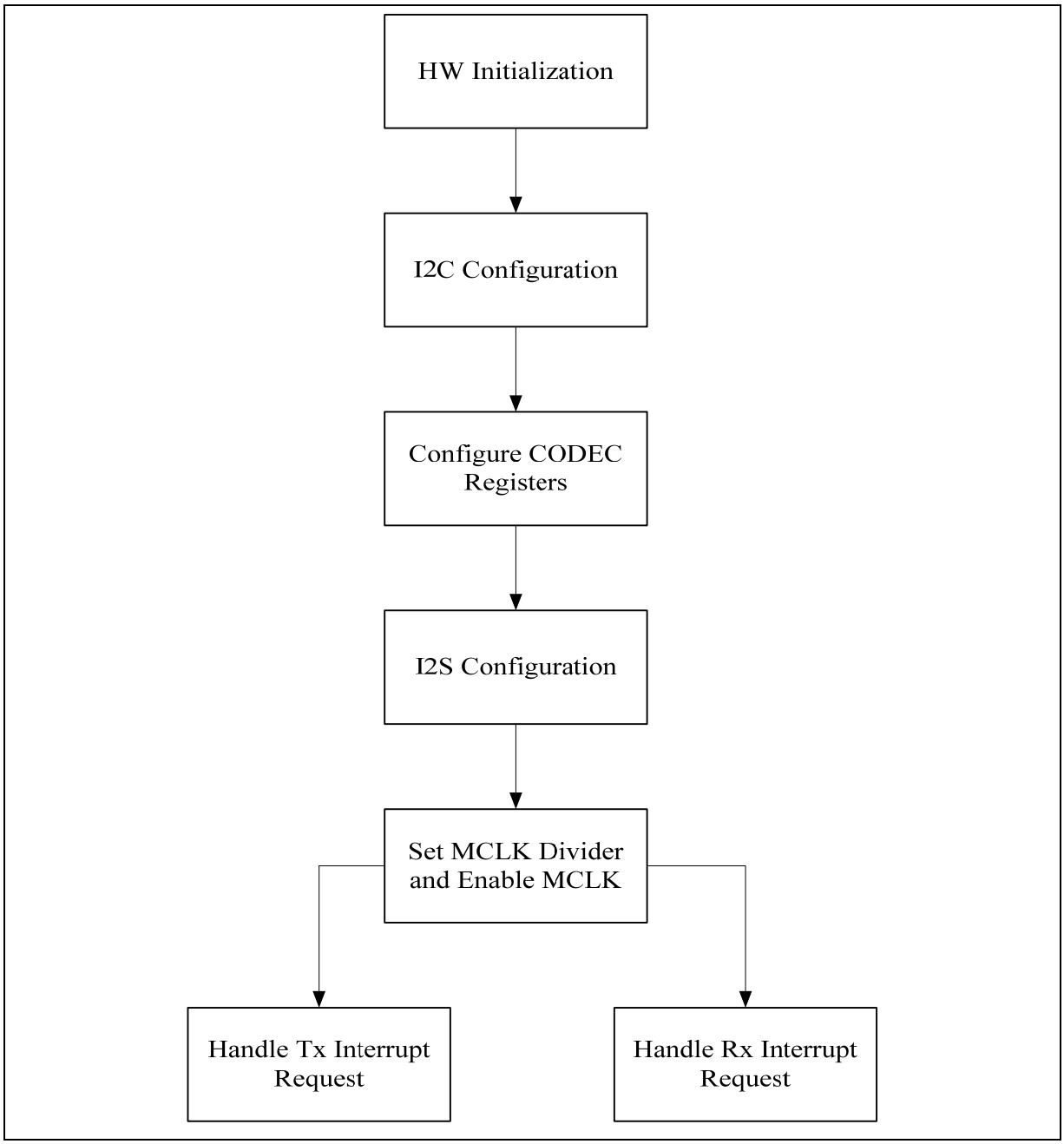

图 5 I2S 和编解码器集成的控制流程

3.2 NUC1xx 设置

1. 硬件初始化，设置PLL控制寄存器、HCKL时钟源，以及I2C和I2S的引脚功能

2. 打开I2C驱动初始化I2C接口，设置I2C总线时钟，设定I2C回调函数，请参考I2C驱动指南
3. 通过I2C接口配置编解码器相关设定，编解码器的主时钟源是外部引脚，使能编解码器输出LRCLK和BCLK，并且设置输出时钟频率和音频格式
4. 打开I2S驱动来初始化I2S接口，设置I2S为从模式，设置音频数据格式以及发送/接收FIFO门槛值
5. 为编解码器时钟源设置MCLK 除频器并使能MCLK输出到编解码器

3.3 Rx请求处理

1. I2S从WAU8822的ADC接收音频数据，并发送数据到Rx FIFO，当RX FIFO中的数据等于或大于门槛值的时候，将产生I2S接收中断
2. 从RX FIFO中读出数据并发送到音频数据缓存区

3.4 Tx请求处理

1. 当Tx FIFO中的数据等于或小于Tx FIFO门槛值的时候将产生I2S发送中断
2. 从音频数据缓存区读出数据并送入Tx FIFO
3. I2S 发送数据到编解码器的DAC

3.5 相关API参考

- I2S Driver Reference Guide.doc
- I2C Driver Reference Guide.doc

4 运行换进设置及结果

4.1 测试 Smpl_DrvI2S

I2S示例程序，Smpl_DrvI2S，可以使用Keil MDK工具编译并通过ICE下载到NUC1xx系列开发板，然后用户可以在ICE环境下执行代码，或者复位开发板执行已经被写到内置程序Flash中的代码，接着输入声音源到WAU8822的ADC并接喇叭到WAU8822 的DAC

4.2 结果

通过喇叭听到正确的声音

5 修订历史

版本.	日期	描述
0.01	2010-3	创建

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

Please note that all data and specifications are subject to change without notice. All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.