

概述

此文档主要描述 SIO DUART 使用指南的补充说明

SPIN TROL

目录

1	SIO DUART 更新说明	7
1.1	SIO DUART 更新	7
1.1.1	SIO FIFO ReStart 函数	7
1.1.2	适用于特殊场景的 SIO FIFO Restart 函数	9

SPIN TROL

图片列表

未找到图形项目表。

SPIN TROL

表格列表

未找到图形项目表。

SPIN TROL

版本历史

版本	日期	作者	状态	变更
A/0	2022 年 8 月 3 日	HaiYangWang	Outdated	首次发布。
A/1	2023 年 12 月 5 日	HaiYangWang	Released	添加 FIFO 溢出后清除错误标志位的方法。 添加使用一个 UART 时清除校验位错误的方法。

SPIN
TROL

术语或缩写

术语或缩写	描述
PLA	ProgrammableLogicArray, 可编程逻辑阵列
SIO	Smart Input Output, SIO

SPIN TROL

1 SIO DUART 使用指南

1.1 SIO DUART 使用更新

1.1.1 SIO FIFO ReStart 函数

适用于 SPC1168/SPD1148/SPD1178/SPD1188/SPC1158 只包含一个 SIO 模块的芯片或 SPC1198/SPC2168 只使用一个 SIO 模块的芯片。当 FIFO 溢出后，推荐客户采用此函数进行 FIFO 复位，适用温度-40~125 度。

代码 1-2: SIO_ReStart_FIFO

```
void SIO_ReStart_FIFO(SIO_REGS* SIOx)
{
    uint32_t u32ClkDiv;
    uint32_t i;
    uint32_t j;
    uint32_t au32Data[8];

    /* Save current clock div */
    u32ClkDiv = CLOCK_GetModuleDiv(SIO0_MODULE);

    /* Set clock for write PLA timing requirment */
    CLOCK_SetModuleDiv(SIO0_MODULE, 10);
    Delay_Us(10);

    for (j = 0; j < 2; j++)
    {
        /* Stop SIO module */
        SIOx->SIOM[31].a11 = 0x100U;
        /* SIO FIF00 address : 0x11; SIO FIF01 address : 0x12 */
        if (j == 0)
            SIOx->SIOSADDR.a11 = 0x60U + 0x11;
        else
            SIOx->SIOSADDR.a11 = 0x60U + 0x12;
        /* Set write/read PLA timing parameter */
        SIOx->SIOPLDPARA.a11 = 0x7FFFU;
        /* Read FIFO PLA */
        (void) (SIOx->SIOSDATA.a11);
        au32Data[0] = SIOx->SIOSD[0].a11;
        au32Data[1] = SIOx->SIOSD[1].a11;
        au32Data[2] = SIOx->SIOSD[2].a11;
        au32Data[3] = SIOx->SIOSD[3].a11;
        au32Data[4] = SIOx->SIOSD[4].a11;
        au32Data[5] = SIOx->SIOSD[5].a11;
        au32Data[6] = SIOx->SIOSD[6].a11;
        au32Data[7] = SIOx->SIOSD[7].a11;
    }
}
```

```
/* Disable FIFO and then Enable FIFO */
for (i = 0; i < 2; i++)
{
    if (j == 0)
        SIOx->SIOSADDR.a11 = 0x60U + 0x11;
    else
        SIOx->SIOSADDR.a11 = 0x60U + 0x12;
    SIOx->SIOSD[0].a11 = au32Data[0];
    SIOx->SIOSD[1].a11 = au32Data[1];
    SIOx->SIOSD[2].a11 = au32Data[2];
    SIOx->SIOSD[3].a11 = au32Data[3];
    SIOx->SIOSD[4].a11 = au32Data[4];
    SIOx->SIOSD[5].a11 = au32Data[5];
    SIOx->SIOSD[6].a11 = au32Data[6] & ((i == 0) ? 0xFFFFFFFFBU :
0xFFFFFFFFFU);
    SIOx->SIOSD[7].a11 = au32Data[7];
    SIOx->SIOSDATA.a11 = 0x0U;
}
}
/* Switch the clock back */
CLOCK_SetModuleDiv(SIO0_MODULE, u32ClkDiv);
Delay_Us(10);

/* Enable SIO module */
SIOx->SIOM[31].a11 = 0x0101U;

/* Clear SIO all flag */
SIOx->SIOM[4].a11 = 0x8000U.
}
```


1.1.2 适用于特殊场景的 SIO FIFO Restart 函数

适用于 SPC1198/SPC2168 多个 SIO 使用且 FIFO 溢出需要复位时的应用场景，适用温度 20~125 度。

代码 1-2: SIO_ReStart_FIFO

```
void SIO_ReStart_FIFO(SIO_REGS* SIOx)
{
    uint32_t u32ClkDiv;
    uint32_t i;
    uint32_t j;
    uint32_t au32Data[8];

    for (j = 0; j < 2; j++)
    {
        /* Stop SIO module */
        SIOx->SIOM[31].a11 = 0x100U;
        /* SIO FIFO0 address : 0x11; SIO FIFO1 address : 0x12 */
        if (j == 0)
            SIOx->SIOSADDR.a11 = 0x60U + 0x11;
        else
            SIOx->SIOSADDR.a11 = 0x60U + 0x12;
        /* Set write/read PLA timing parameter */
        SIOx->SIOPLDPARA.a11 = 0x7FFFU;
        /* Read FIFO PLA */
        (void) (SIOx->SIOSDATA.a11);
        au32Data[0] = SIOx->SIOSD[0].a11;
        au32Data[1] = SIOx->SIOSD[1].a11;
        au32Data[2] = SIOx->SIOSD[2].a11;
        au32Data[3] = SIOx->SIOSD[3].a11;
        au32Data[4] = SIOx->SIOSD[4].a11;
        au32Data[5] = SIOx->SIOSD[5].a11;
        au32Data[6] = SIOx->SIOSD[6].a11;
        au32Data[7] = SIOx->SIOSD[7].a11;

        /* Disable FIFO and then Enable FIFO */
        for (i = 0; i < 2; i++)
        {
            if (j == 0)
                SIOx->SIOSADDR.a11 = 0x60U + 0x11;
            else
                SIOx->SIOSADDR.a11 = 0x60U + 0x12;
            SIOx->SIOSD[0].a11 = au32Data[0];
            SIOx->SIOSD[1].a11 = au32Data[1];
            SIOx->SIOSD[2].a11 = au32Data[2];
            SIOx->SIOSD[3].a11 = au32Data[3];
            SIOx->SIOSD[4].a11 = au32Data[4];
        }
    }
}
```

```
        SIOx->SIOSD[5].a11 = au32Data[5];
        SIOx->SIOSD[6].a11 = au32Data[6] & ((i == 0) ? 0xFFFFFFFFBU :
0xFFFFFFFFFU);
        SIOx->SIOSD[7].a11 = au32Data[7];
        SIOx->SIOSDATA.a11 = 0x0U;
    }
}

/* Enable SIO module */
SIOx->SIOM[31].a11 = 0x0101U;

/* Clear SIO all flag */
SIOx->SIOM[4].a11 = 0x8000U.
}
```

1.1.3 适用于一个 SIO UART 清除校验位失败的场景

当使用一个 SIO UART（V2 或 V4 版本）时，当校验错误后可能存在清除失败的场景，此时可以使用以下函数进行校验位清除。

代码 1-2: SIO_Clear_Parity_Error_Flag

```
void SIO_Clear_Parity_Error_Flag(SIO_REGS* SIOx)
{
    (void)SIOx->SIOM[0].a11;
    (void)SIOx->SIOM[1].a11;
    /* Clear SIO all flag */
    SIOx->SIOM[4].a11 = 0x8000U.
}
```