HC-B2030(AUDIO)卡寄存器定义

版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改事项 | 修改时间 | 修改人 |
| 1.00 | 初始版本 | 2020-06-04 | 王勇 |
| 1.01 | 1、完善音频采集流程，见2.1；  2、采集每个通道由32MB更改为64MB；  3、增加音频输出配置相关寄存器，地址：48C~490；  4、增加音频输出流程，见2.2； | 2020-06-10 | 王勇 |
| 1.02 | 1、增加A2B主模式配置流程，见2.3.1； | 2020-06-23 | 张博 |
| 1.03 | 1、增加A2B文件读写相关寄存器定义和读写流程，见2.3.3、2.3.4；  2、增加A2B从模式配置流程，见2.3.2； | 2020-07-03 | 王勇 |
| 1.04 | 1、增加音频播放采样率设置寄存器；  2、音频播放流程中增加采样率设置步骤； | 2020-07-03 | 王勇 |
| 1.06 | 1、每个通道的音频输入增加浮点系数功能（整形归一化为浮点数），并修改音频采集流程，见2.1；  2、每个通道的音频输出增加浮点系数功能（归一化的浮点数转换为整形），并修改音频播放流程，见2.2；  3、A2B接收、发送通道增加浮点系统功能  4、A2B增加从模式繁忙状态指示寄存器，以及作为从模式时的采样率和BIT率测量寄存器  5、音频输入增加偏置电流输出使能、禁止开关  6、A2B主模式配置增加“检查配置状态”和“配置完成后检查”流程，见2.3.1.1、2.3.1.7；  7、更新A2B从模式配置流程，见2.3.2 | 2020-07-08 | 王勇 |
| 1.10 | 1、增加循环播放寄存器  2、增加在线升级功能 | 2020-07-18 | 王勇 |
| 1.13 | 增加播放采集暂停功能 | 2020-08-03 | 王勇 |
| 1.17 | 1、音频采集增加整形存储功能  2、音频播放增加整形数据播放功能  3、A2B增加整形采集和播放的功能 | 2020-08-11 | 王勇 |

# 寄存器定义

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **名称** | **说明** | **地址(Hex)** | **类型** | **长度(bit)** | **复位值** | **描述** |
| **通用寄存器** | | | | | | |
| Slot | 槽位号 | 000 | R | 5 |  |  |
| Datetime | 程序日期 | 004 | R | 32 |  |  |
| Version | 程序版本 | 008 | R | 32 |  |  |
| Rw\_test | 读写测试 | 400 | RW | 32 | 0x1234ABCD |  |
| **音频采集相关寄存器** | | | | | | |
| Trans\_num | DDR每次传输长度 | 480 | RW | 16 | 8 | DDR每次传输长度  Bit[15:0]：长度 |
| Adc\_reg\_data | ADC寄存器值 | 484 | RW | 16 | 0 | Adc寄存器值  Bit[15:14]:0  Bit[13]:1  Bit[12:8]:寄存器地址  Bit[7:0]：寄存器值 |
| Adc\_reg\_trig | ADC寄存器写触发 | 488 | W | 1 | 0 | Adc寄存器写触发 |
| Bias\_curr\_en | 输入通道偏置电流使能 | 4C0 | RW | 4 | 0 | 输入通道偏置电流使能，bit0~bit3分别代表输入通道1~通道4  Bit[ch]含义（ch为通道号）  0：禁止偏置电流  1：使能偏置电流  【注】输入通道接IEPE麦克风输入时，需要使能偏置电流；其他情况禁止偏置电流 |
| Bias\_curr\_err | 输入通道偏置电流错误 | 4C4 | W | 4 | 0 | 输入通道偏置电流错误，bit0~bit3分别代表输入通道1~通道4  Bit[ch]含义（ch为通道号）  0：无错误  1：有错误 |
| In\_enable | 输入使能 | 800 | RW | 4 | 0 | 输入通道使能，bit0~bit3分别代表通道1~通道4  Bit[ch]含义（ch为通道号）：  0：禁止输入；  1：使能输入 |
| In\_mode | 输入模式 | 804 | RW | 8 | 0 | 输入模式，bit0~bit3分别代表通道1~通道4的单声道/立体声；bit4~bit7代表通道1~通道4的浮点/整形采集模式；  Bit[n]含义（n>=0 && n<=3）：  0：单声道输入  1：立体声输入  Bit[n]含义（n>=4 && n<=7）：  0：浮点输入  1：整形输入  【注】只有当对应通道的in\_enable为0时，才能更改对应通道的in\_mode |
| In\_atten | 输入衰减 | 808 | RW | 4 | 0 | 输入衰减，bit0~bit3分别代表通道1~通道4  Bit[ch]含义（ch为通道号）：  0:1/5衰减；  1：无衰减 |
| In\_mult\_coef1 | 输入通道1乘系数 | 80C | RW | 32 | 2-31 | 输入通道1乘系数  Bit[31:0]为单精度浮点数，默认值为2-31  【注】：输入通道（1~4）乘系数默认作用是将32bit的整形PCM采样值进行浮点归一化运算，使得播放器可以正确播放。如果需要进行音量调节，可以改变该系数，但需要确保浮点数不要超过(-1~1)取值范围，否则会导致声音失真 |
| In\_mult\_coef2 | 输入通道2乘系数 | 810 | RW | 32 | 2-31 | 输入通道2乘系数  Bit[31:0]为浮点数，默认值为2-31 |
| In\_mult\_coef3 | 输入通道3乘系数 | 814 | RW | 32 | 2-31 | 输入通道3乘系数  Bit[31:0]为浮点数，默认值为2-31 |
| In\_mult\_coef4 | 输入通道4乘系数 | 818 | RW | 32 | 2-31 | 输入通道4乘系数  Bit[31:0]为浮点数，默认值为2-31 |
| In\_pause | 输入采集暂停 | 81C | RW | 4 | 0 | 输入采集暂停功能，bit0~bit3分别对应输入通道1~4  Bit[n]（n=0..3）含义  0：正常采集  1：暂停采集 |
| Ddr\_head\_in1 | 通道1DDR缓冲区头指针 | 840 | R | 32 | 0 | 通道1 DDR缓冲区头指针  Bit[15:0]: (64MB)  0 ~ 0x3FFFFFF  【注】：通道1DMA地址偏移：0x0 |
| Ddr\_tail\_in1 | 通道1DDR缓冲区尾指针 | 844 | RW | 32 | 0 | 通道1 DDR缓冲区尾指针。  Bit[15:0]: (64MB)  0 ~ 0x3FFFFFF  【注】：当从DDR读取数据后，应当修改该指针，释放空间。当超过最大地址时，重新从0开始。 |
| Ddr\_full\_in1 | 通道1DDR缓冲区满 | 848 | R | 1 | 0 | 通道1 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| Ddr\_head\_in2 | 通道2DDR缓冲区头指针 | 84C | R | 32 | 0 | 通道2 DDR缓冲区头指针  Bit[15:0]: (64MB)  0 ~ 0x3FFFFFF  【注】：通道2 DMA地址偏移：0x4000000 |
| Ddr\_tail\_in2 | 通道2DDR缓冲区尾指针 | 850 | RW | 32 | 0 | 通道2 DDR缓冲区尾指针  Bit[15:0]: (64MB)  0 ~ 0x3FFFFFF  【注】：当从DDR读取数据后，应当修改该指针，释放空间。当超过最大地址时，重新从0开始。 |
| Ddr\_full\_in2 | 通道2DDR缓冲区满 | 854 | R | 1 | 0 | 通道2 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| Ddr\_head\_in3 | 通道3DDR缓冲区头指针 | 858 | R | 32 | 0 | 通道3 DDR缓冲区头指针  Bit[15:0]: (64MB)  0 ~ 0x3FFFFFF  【注】：通道3 DMA地址偏移：0x8000000 |
| Ddr\_tail\_in3 | 通道3DDR缓冲区尾指针 | 85C | RW | 32 | 0 | 通道3 DDR缓冲区尾指针  Bit[15:0]: (64MB)  0 ~ 0x3FFFFFF  【注】：当从DDR读取数据后，应当修改该指针，释放空间。当超过最大地址时，重新从0开始。 |
| Ddr\_full\_in3 | 通道3DDR缓冲区满 | 860 | R | 1 | 0 | 通道3 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| Ddr\_head\_in4 | 通道4DDR缓冲区头指针 | 864 | R | 32 | 0 | 通道4 DDR缓冲区头指针  Bit[15:0]: (64MB)  0 ~ 0x3FFFFFF  【注】：通道4 DMA地址偏移：0xC000000 |
| Ddr\_tail\_in4 | 通道4DDR缓冲区尾指针 | 868 | RW | 32 | 0 | 通道4 DDR缓冲区尾指针  Bit[15:0]: (64MB)  0 ~ 0x3FFFFFF  【注】：当从DDR读取数据后，应当修改该指针，释放空间。当超过最大地址时，重新从0开始。 |
| Ddr\_full\_in4 | 通道4DDR缓冲区满 | 86C | R | 1 | 0 | 通道4 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| **音频播放相关寄存器** | | | | | | |
| Dac\_reg\_data0 | 第一片DAC寄存器值 | 48C | RW | 16 | 0 | Dac寄存器值  Bit[15:14]:0  Bit[13]:1  Bit[12:8]:寄存器地址  Bit[7:0]：寄存器值 |
| Adc\_reg\_trig0 | 第一片DAC寄存器写触发 | 490 | W | 1 | 0 | Dac寄存器写触发 |
| Dac\_reg\_data1 | 第二片DAC寄存器值 | 494 | RW | 16 | 0 | Dac寄存器值  Bit[15:14]:0  Bit[13]:1  Bit[12:8]:寄存器地址  Bit[7:0]：寄存器值 |
| Dac\_reg\_trig1 | 第二片DAC寄存器写触发 | 498 | W | 1 | 0 | Dac寄存器写触发 |
| Out\_enable | 输出使能 | 900 | RW | 4 | 0 | 输出通道使能，bit0~bit3分别代表通道1~通道4  Bit[ch]含义（ch为通道号）：  0：禁止输出；  1：使能输出 |
| Out\_mode | 输入模式 | 904 | RW | 8 | 0 | 输出模式，bit0~bit3分别代表通道1~通道4的单声道/立体声模式；bit4~bit7分别代表通道1~通道4的整形/浮点模式  Bit[n]含义（n=0..3）：  0：单声道播放  1：立体声播放  Bit[n]含义（n=4..7）：  0：浮点播放  1：整形播放  【注】只有当对应通道的out\_enable为0时，才能更改对应通道的out\_mode |
| Out\_amp | 输出增益 | 908 | RW | 4 | 0 | 输出增益，bit0~bit3分别代表通道1~通道4  Bit[ch]含义（ch为通道号）：  0:无增益；  1：5倍增益 |
| Out\_sample\_rate0 | 第一片DAC采样率 | 90C | RW | 2 | 0 | 第一片DAC采样率  Bit[1:0]定义  0:48kHz  1:96kHz  2:192kHz  【注】：  输出通道0、1的采样率设置使用Out\_sample\_rate0寄存器，两个通道采样率相同 |
| Out\_sample\_rate1 | 第二片DAC采样率 | 910 | RW | 2 | 0 | 第二片DAC采样率  Bit[1:0]定义  0:48kHz  1:96kHz  2:192kHz  【注】：  输出通道2、3的采样率设置使用Out\_sample\_rate1寄存器，两个通道采样率相同 |
| Out\_mult\_coef1 | 输出通道1乘系数 | 914 | RW | 32 | 231 | 输出通道1乘系数  Bit[31:0]为单精度浮点数，默认值为231  【注】：输出通道（1~4）乘系数默认作用是将归一化的浮点音频数据转为为32bit的整形PCM，便于DAC输出。如果需要进行音量调节，可以改变该系数，但需要确保转换后的整数不要超过(-231~231)取值范围，否则会导致声音失真 |
| Out\_mult\_coef2 | 输出通道2乘系数 | 918 | RW | 32 | 231 | 输出通道2乘系数  Bit[31:0]为浮点数，默认值为231 |
| Out\_mult\_coef3 | 输出通道3乘系数 | 91C | RW | 32 | 231 | 输出通道3乘系数  Bit[31:0]为浮点数，默认值为231 |
| Out\_mult\_coef4 | 输出通道4乘系数 | 920 | RW | 32 | 231 | 输出通道4乘系数  Bit[31:0]为浮点数，默认值为231 |
| Out\_repeat | 循环播放 | 924 | RW | 4 | 0 | 循环播放，每个bit代表一个通道，bit位为0代表正常播放，bit位为1代表循环播放 |
| Out\_pause | 播放暂停功能 | 928 | RW | 4 | 0 | 播放暂停功能，bit0~bit3分别对应输出通道1~4  Bit[n]（n=0..3）含义  0：正常播放  1：暂停播放 |
| Ddr\_head\_out1 | 通道1DDR缓冲区头指针 | 940 | RW | 32 | 0 | 通道1 DDR缓冲区头指针  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF  【注】：  1）通道1DMA地址偏移：0x10000000  2）当向DDR写取数据时，应当修改该指针，当超过最大地址时，重新从0开始。 |
| Ddr\_tail\_out1 | 通道1DDR缓冲区尾指针 | 944 | R | 32 | 0 | 通道1 DDR缓冲区尾指针。  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF |
| Ddr\_full\_in1 | 通道1DDR缓冲区满 | 948 | R | 1 | 0 | 通道1 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| Ddr\_head\_out2 | 通道2DDR缓冲区头指针 | 94C | RW | 32 | 0 | 通道2 DDR缓冲区头指针  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF  【注】：  1）通道1DMA地址偏移：0x18000000;  2）当向DDR写取数据时，应当修改该指针，当超过最大地址时，重新从0开始。 |
| Ddr\_tail\_out2 | 通道2DDR缓冲区尾指针 | 950 | R | 32 | 0 | 通道2 DDR缓冲区尾指针  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF |
| Ddr\_full\_out2 | 通道2DDR缓冲区满 | 954 | R | 1 | 0 | 通道2 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| Ddr\_head\_out3 | 通道3DDR缓冲区头指针 | 958 | RW | 32 | 0 | 通道3 DDR缓冲区头指针  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF  【注】：  1）通道1DMA地址偏移：0x20000000  2）当向DDR写取数据时，应当修改该指针，当超过最大地址时，重新从0开始。 |
| Ddr\_tail\_out3 | 通道3DDR缓冲区尾指针 | 95C | R | 32 | 0 | 通道3 DDR缓冲区尾指针  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF |
| Ddr\_full\_out3 | 通道3DDR缓冲区满 | 960 | R | 1 | 0 | 通道3 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| Ddr\_head\_out4 | 通道4DDR缓冲区头指针 | 964 | RW | 32 | 0 | 通道4 DDR缓冲区头指针  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF  【注】：  1）通道1DMA地址偏移：0x28000000  2）当向DDR写取数据时，应当修改该指针，当超过最大地址时，重新从0开始。 |
| Ddr\_tail\_out4 | 通道4DDR缓冲区尾指针 | 968 | R | 32 | 0 | 通道4 DDR缓冲区尾指针  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF |
| Ddr\_full\_out4 | 通道4DDR缓冲区满 | 96C | R | 1 | 0 | 通道4 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| **A2B相关寄存器** | | | | | | |
| Dds\_out\_enable | DDS模拟波形输出使能 | 49C | RW | 1 | 0 | DDS模拟波形输出使能  Bit[0]含义：  0：不使能DDS模拟波形功能（使用DDR环形缓冲发送数据）  1：使能DDS模拟波形功能 |
| Dds\_cfg\_data | DDS相移参数 | 4A0 | RW | 16 | 0 | DDS相移参数：  Bit[15:0]含义：  相移参数，用于控制输出频率。  【注】输出频率计算公式：  f=48000\* Dds\_cfg\_data /65536 |
| Dds\_cfg\_trig | DDS相移参数使能 | 4A4 | W | 1 | 0 | DDS相移参数使能触发信号  Bit[0]含义：  0：忽略  1：应用DDS相移参数，更改DDS输出频率 |
| A2b\_mode | A2B主从模式 | A00 | RW | 2 | 0 | A2B模式  Bit[0]含义：  0：从模式  1：主模式  Bit[1]含义  0：浮点数据采集、播放  1：整形数据采集、播放  【注】  1、bit[0]位随A2B配置流程修改，不能单独修改  2、bit[1]位必须在A2B收发都禁止时才能更改 |
| A2b\_tx\_en | A2B DDR发送使能 | A04 | RW | 1 | 0 | A2B DDR发送使能  Bit[0]含义：  0：禁止DDR发送  1：使能DDR发送 |
| A2b\_rx\_en | A2B DDR接收使能 | A08 | RW | 1 | 0 | A2B DDR接收使能  Bit[0]含义：  0：禁止DDR接收  1：使能DDR接收 |
| A2b\_irq | A2B是否产生中断 | A0C | R | 1 | 0 | A2B是否产生中断，表示AD2428W输出的中断信号  Bit[0]含义：  0：无中断  1：有中断 |
| A2b\_ddr\_head\_rx | A2B接收通道头指针 | A10 | RW | 32 | 0 | A2B接收 DDR缓冲区头指针  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF  【注】：  1）A2B接收DMA地址偏移：0x30000000  2）当向DDR写取数据时，应当修改该指针，当超过最大地址时，重新从0开始。 |
| A2b\_ddr\_tail\_rx | A2B接收通道尾指针 | A14 | RW | 32 | 0 | A2B接收 DDR缓冲区尾指针。  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF |
| A2b\_ddr\_full\_rx | A2B接收通道满标记 | A18 | R | 1 | 0 | A2B接收 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| A2b\_ddr\_head\_tx | A2B发送通道头指针 | A1C | RW | 32 | 0 | A2B发送 DDR缓冲区头指针  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF  【注】：  1）通道1DMA地址偏移：0x38000000  2）当向DDR发送数据时，应当修改该指针，当超过最大地址时，重新从0开始。 |
| A2b\_ddr\_tail\_tx | A2B发送通道尾指针 | A20 | RW | 32 | 0 | A2B发送 DDR缓冲区尾指针。  Bit[15:0]: (128MB)  0 ~ 0x7FFFFFF |
| A2b\_ddr\_full\_tx | A2B发送通道满标记 | A24 | R | 1 | 0 | A2B发送 DDR缓冲区满标记  Bit[0]含义：  0：DDR环形缓冲区未满；  1：DDR环形缓冲区已满； |
| A2b\_sample\_rate | A2B发送采样率（主模式） | A28 | RW | 2 | 0 | A2B发送采样率（主模式）  Bit[1:0]含义：  0:48kHz  1:96kHz  2:192kHz |
| A2b\_slave\_busy | A2B从模式繁忙 | A2C | R | 1 | 0 | A2B从模式繁忙。当处于这种模式时，不能将当前板卡切换为主模式，以避免时钟冲突。将现有A2B总线上的主节点移除后，才能将当前板卡切换为主模式  Bit[0]含义：  0：主模式或未配置状态  1：当前板卡处于从模式工作状态 |
| A2b\_bclk\_cnt | A2B BCLK时钟计数 | A30 | R | 32 | 0 | A2B BCLK时钟计数  Bit[31:0]含义：  10ms内的周期计数  【注】：   1. A2B总线配置了主节点后，A2b\_bclk\_cnt才有效 2. I2S的位数，可以通过A2b\_bclk\_cnt/(A2b\_sync\_cnt\*2)公式计算得到，当前系统仅支持32bit的I2S 3. 主模式和从模式下，都可以通过A2b\_bclk\_cnt和A2b\_sync\_cnt计算I2S位数 |
| A2b\_sync\_cnt | A2B SYNC时钟计数 | A34 | R | 32 | 0 | A2B SYNC时钟计数  Bit[31:0]含义：  10ms内的SYNC周期计数  【注】：   1. 1、A2B总线配置了主节点后，A2b\_sync\_cnt才有效 2. A2B采样率为该值乘以100，例如该寄存器结果为480，则实际采样率为480\*100=48K，误差为±100Hz 3. 主模式和从模式下，都可以通过A2b\_sync\_cnt计算采样率 |
| A2b\_rx\_mult\_coef | A2B接收乘系数 | A38 | RW | 32 | 2-31 | A2B输入通道乘系数  Bit[31:0]为单精度浮点数，默认值为2-31  【注】：A2B输入通道乘系数默认作用是将32bit的整形PCM采样值进行浮点归一化运算，使得播放器可以正确播放。如果需要进行音量调节，可以改变该系数，但需要确保浮点数不要超过(-1~1)取值范围，否则会导致声音失真 |
| A2b\_tx\_mult\_coef | A2B发送乘系数 | A3C | RW | 32 | 231 | A2B输出通道乘系数  Bit[31:0]为单精度浮点数，默认值为231  【注】：A2B输出乘系数默认作用是将归一化的浮点音频数据转为为32bit的整形PCM，便于DAC输出。如果需要进行音量调节，可以改变该系数，但需要确保转换后的整数不要超过(-231~231)取值范围，否则会导致声音失真 |
| A2b\_tx\_repeat | A2B循环播放功能 | A40 | RW | 1 | 0 | A2B循环播放功能  Bit[0]含义：  0：正常播放  1：循环播放 |
| A2b\_tx\_pause | A2B播放暂停 | A44 | RW | 1 | 0 | A2B播放暂停  Bit[0]含义：  0：正常播放  1：暂停播放 |
| A2b\_rx\_pause | A2B采集暂停 | A48 | RW | 1 | 0 | A2B采集暂停  Bit[0]含义：  0：正常采集  1：暂停采集 |
| **EEPROM IIC操作寄存器（校准数据存取）** | | | | | | |
| 地址偏移：0x2000，详细定义参考XILINX pg090-axi-iic.pdf，table2-4  EEPROM IIC操作方法，参考复旦微FMC24C64D数据手册。 | | | | | | |
| **A2B IIC操作寄存器（设置AD2428W）** | | | | | | |
| 地址偏移：0x4000，详细定义参考XILINX pg090-axi-iic.pdf，table2-4  A2B IIC操作方法，参考ADI公司AD2420(W)/6(W)/7(W)/8(W)/9(W) Automotive Audio Bus A2B Transceiver Technical Reference数据手册。 | | | | | | |
| **在线升级相关寄存器** | | | | | | |
| 地址偏移：0xA000，详细定义参考XILINX pg153-axi-quad-spi.pdf。  Flash的读写操作参考S25FL256SAGNFI00配置芯片数据手册。 | | | | | | |

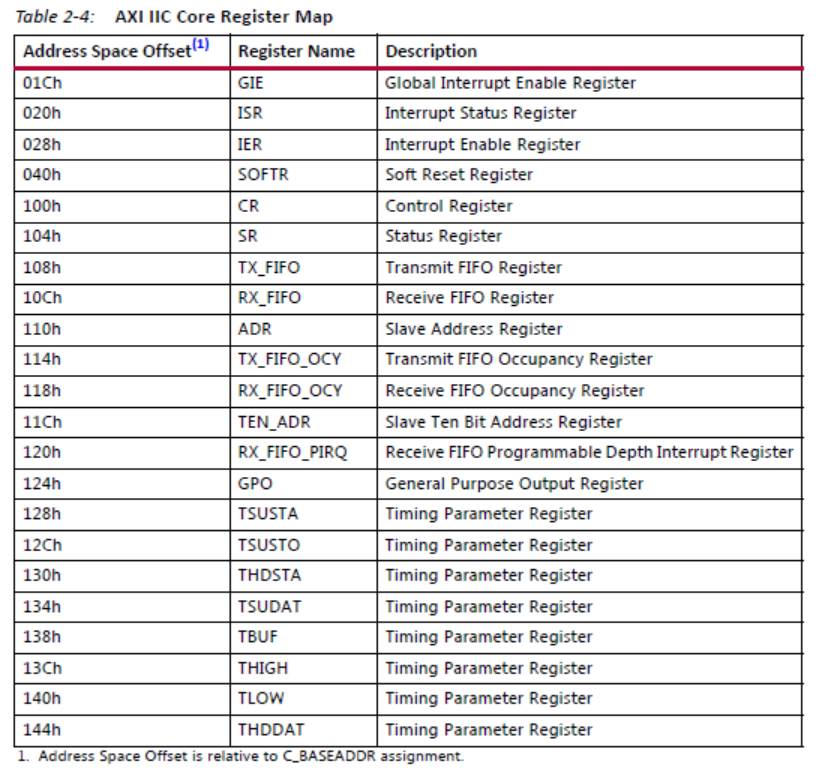


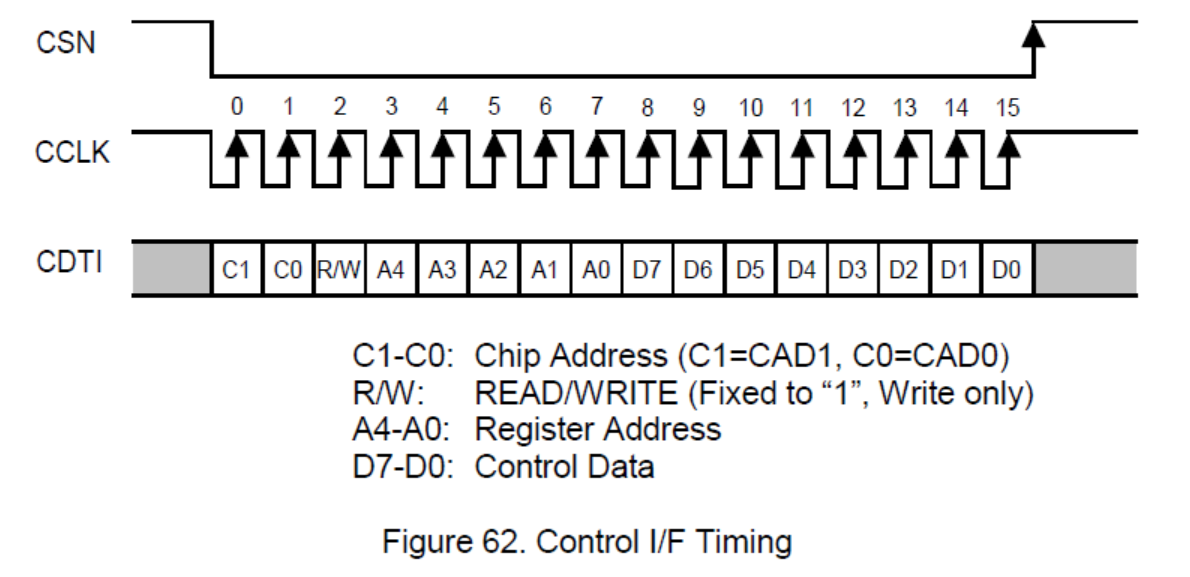
图 1 AXI IIC寄存器定义

# 常见功能设置流程

## 音频采集

操作步骤：

1. 设置In\_enable寄存器，对应通道的bit位设置为0，先禁止通道数据接收；
2. 设置Adc\_reg\_data寄存器值，如下图所示，Bit[15:14]为0；Bit[13]为1；Bit[12:8]为寄存器地址；Bit[7:0]为寄存器值；



主要对0x2地址寄存器进行配置，寄存器具体含义参考AK5574芯片手册。不同采样率，寄存器的设置如下：

48kHz：0x2237；

96kHz：0x221F；

192kHz：0x2207；

1. 设置Adc\_reg\_trig寄存器，写1；
2. 设置Adc\_mode寄存器，设置通道为单声道和立体声；
3. 设置In\_mult\_coef<N>（N=1..4）寄存器（可选），修改增益系数（浮点数），默认增益系数为2-31（浮点数）。修改该系数，需要确保音频数据从整形转换为归一化的浮点数后，不要超过(-1.0~-1.0)范围，否则可能导致声音失真；
4. 如果当前输入是IEPE麦克风，则设置Bias\_curr\_en寄存器，对应通道的bit为设置为1，打开偏置电流输出；如果当前输入不是IEPE麦克风，直接转步骤6；
5. 延时若干秒（例如3秒），确保ADC和IEPE偏置电流源均处于稳定工作状态；
6. 设置In\_enable寄存器，对应通道的bit位设置为1，使能对应的通道数据接收；
7. 读取Ddr\_head\_in<N>和Ddr\_tail\_in<N>寄存器，计算待采集数据长度。每次读取的字节数为以下结果的最小值：

* DDR环形缓冲中有效数据个数：(Ddr\_head\_in<N> - Ddr\_tail\_in<N> + 0x4000000) & 0x3FFFFFF;
* 单次DMA的最大长度：0x400000;
* 当ddr\_head\_in<N>小于ddr\_tail\_in<N>时，DDR环形缓冲区尾部的长度：0x4000000 – ddr\_head\_in<N>；
* 剩余待采集长度（需32字节凑整）：(剩余长度 + 0x1F) & ~0x1F；

如果可读取字节数为0，则延迟1ms，并重复当前步骤，否则转步骤8；

1. 通过DMA读DDR，<起始地址> = Ddr\_tail\_in<N> + 通道DMA偏移地址，DMA长度为上一步骤的计算结果。

每个输入通道DDR环形缓冲区大小为64MB，偏移地址分别如下：

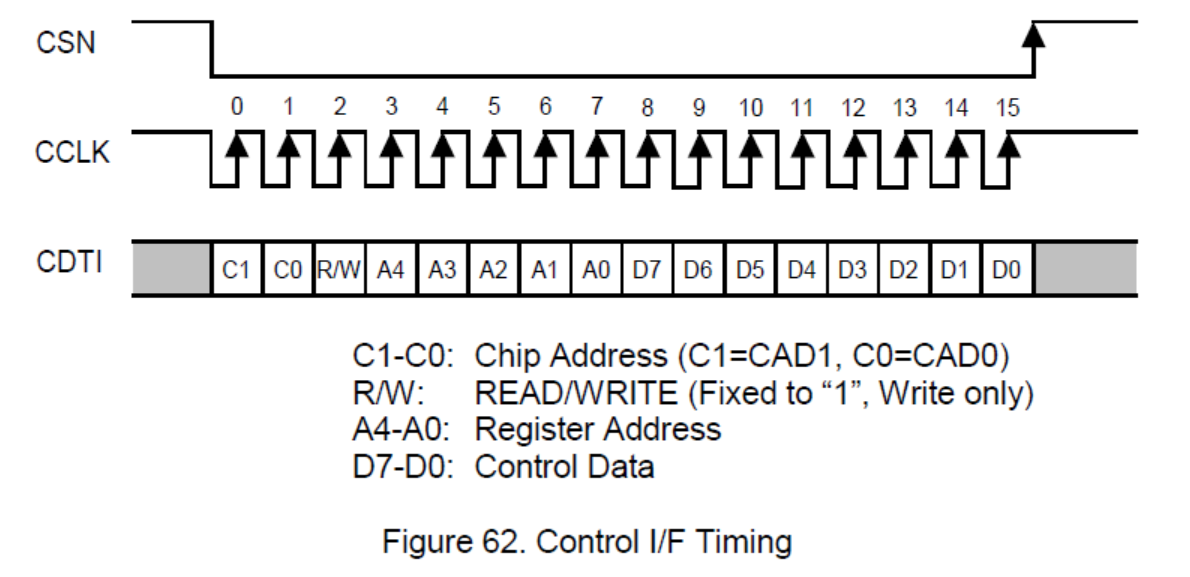
* 通道0：0x0000000；
* 通道1：0x4000000；
* 通道2：0x8000000；
* 通道3：0xC000000；

1. 更新Ddr\_tail\_in<N>寄存器。<Ddr\_tail\_in<N>新值> = (Ddr\_tail\_in<N>旧值 + DMA读取字节数) & 0x3FF\_FFFF；
2. 更新已采集长度，如果已采集长度≥期望采集长度，则转步骤12；否则重复步骤9~11。
3. 设置Bias\_curr\_en寄存器，对应通道的bit为设置为0，关闭偏置电流输出；
4. 将采集的音频数据增加wave文件头（位数、采样率等），并保存为wave文件。

## 音频播放

操作步骤：

1. 设置Out\_enable寄存器，对应通道的bit位设置为0，先禁止通道数据发送；
2. 设置Out\_mode寄存器，设置立体声输出或单声道输出；
3. 设置Out\_amp寄存器，配置输出增益；
4. 设置Out\_sample\_rate<N>(N=0..1)寄存器，设置DAC输出采样率；
5. 设置Out\_mult\_coef<N>(N=1..4)寄存器（可选），修改增益系数（浮点数），默认增益系数为231（浮点数）。修改该系数，需要确保音频数据从归一化的浮点数转换为整数后，不要超过(-231~-231)范围，否则可能导致声音失真；
6. 设置Dac\_data\_reg<N>(N=0..1)寄存器，通道0、1设置Dac\_data\_reg0寄存器，通道2、3设置Dac\_data\_reg1寄存器；Bit[15:14]为0；Bit[13]为1；Bit[12:8]为寄存器地址；Bit[7:0]为寄存器值；



DAC主要对0号寄存器进行配置，寄存器具体定义参考AK4493手册。该寄存器现在默认配置为0x200F；

1. 设置Dac\_reg\_trig<N>(N=0..1)寄存器，设置为1；
2. 延迟若干秒（例如3秒），确保DAC处于稳定工作状态；
3. 打开待发送的wave文件，并检查位数和采样率是否与当前设置一致。如果不一致，则重新选择合适的wave文件。如果一致，则转步骤9；
4. 设置Out\_enable寄存器，对应通道的bit位设置为1，使能通道数据发送；
5. 读取ddr\_full\_out<N>寄存器，如果ddr\_full\_out<N>的值为1，则延迟1ms，重复当前步骤；
6. 读取ddr\_head\_out<N>、ddr\_tail\_out<N>，计算待发送数据的长度：

待发送数据长度为以下各项的最小值：

* 文件剩余字节数：(Ddr\_head\_out<N> - Ddr\_tail\_out<N> + 0x7FFFFFF) & 0x7FFFFFF；
* 单次DMA最大长度：0x400000；
* 当Ddr\_head\_out<N>小于Ddr\_tail\_out<N>，DDR环形缓冲尾部的长度：0x8000000 – Ddr\_head\_out<N>；
* 音频文件剩余待发送长度（需32字节凑整）：(剩余长度 + 0x1F) & ~0x1F；；

1. 从当前位置读取待发送的音频文件，提取左右声道的数据。每个采样点包含8个字节，如果发送立体声，则前4个字节为左声道数据，后四个字节为右声道数据；如果发送单声道数据，则前4个字节为单声道数据，后四个字节补0。注意待发送数据长度为32字节的整数倍，长度不足32字节的，补0对齐；
2. 将上一步读取的音频数据通过DMA写到DDR环形缓冲区。写地址为Ddr\_head\_out<N> + 通道DMA偏移地址。

每个输出通道DDR的环形缓冲区长度为128MB，偏移地址分别如下：

* 通道0：0x10000000；
* 通道1：0x18000000；
* 通道2：0x20000000；
* 通道3：0x28000000；

【注】：如果发送立体声，通道1的音频需要与通道0的音频合并，并向通道0的DDR环形缓冲写数据；同理，通道3的音频数据作为右声道数据需要与通道2的左声道数据合并，并向通道2的DDR环形缓冲写数据；

1. 更新Ddr\_head\_out<N>寄存器值。新Ddr\_head\_out<N> = (旧Ddr\_head\_out<N> + DMA长度) & 0x7FFFFFF；
2. 更新已发送音频文件长度，如果已发送长度≥音频文件长度，则结束，否则重复步骤11~15。

## A2B

### A2B主模式配置

A2B总线具有传输数据，总线供电，为slave提供同步时钟等功能。A2B总线通过superframe进行通信。

A2B总线配置的操作步骤为：1、Master唤醒2、Slave唤醒3、Slave配置4、Master配置。如果有多个slave node，则需要顺序依次唤醒slave之后，倒序依次配置slave，最后配置master。

板卡base\_addr: 0x6E bus\_addr:0x6F

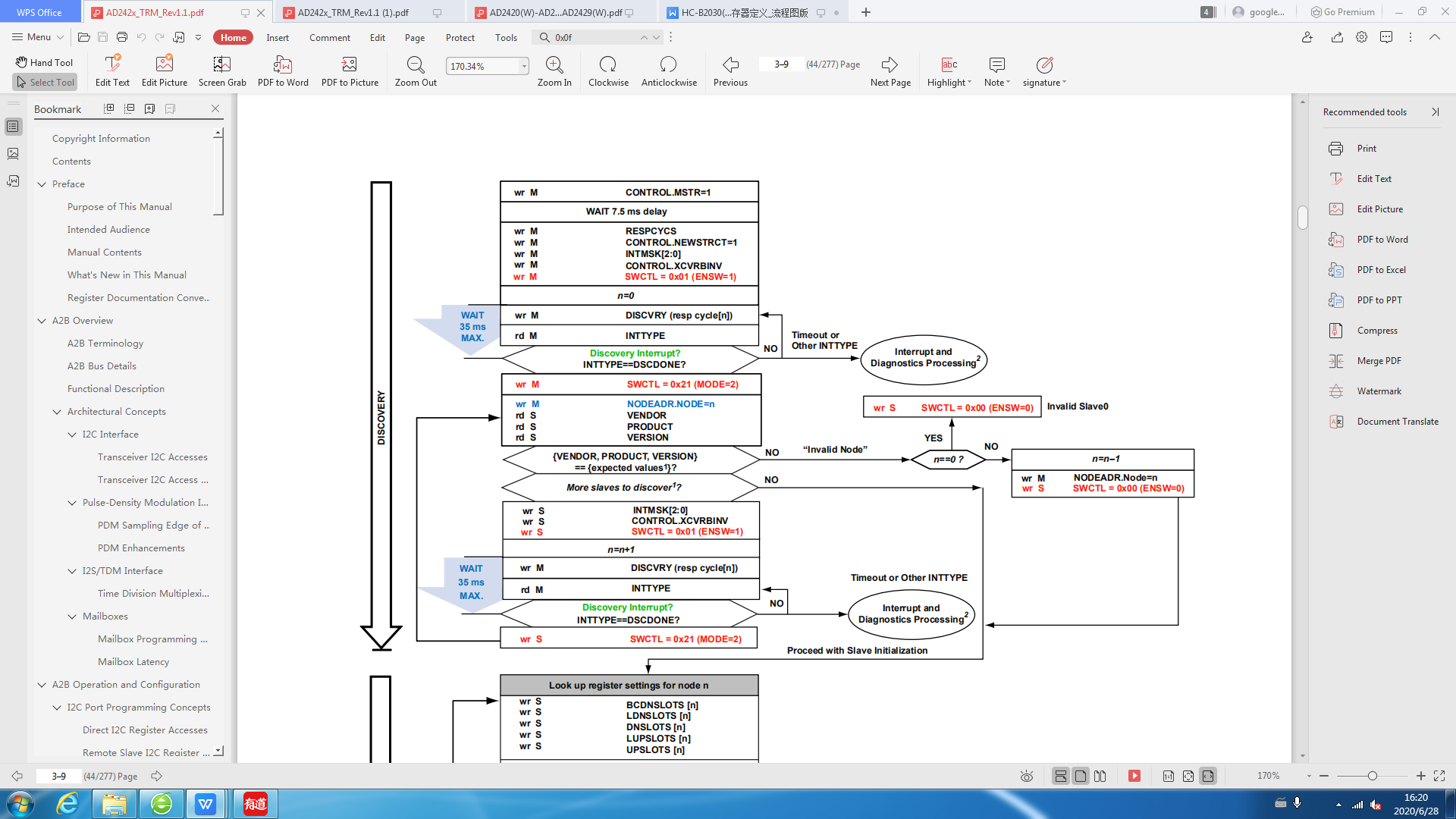
Wm – 使用base地址写 Rm – 使用base地址读

Ws – 使用bus地址写 Rs – 使用bus地址读

#### 检查配置状态

1. 设置A2b\_mode寄存器为0，先进入从模式状态，确保不主动往外送BCLK和SYNC时钟信号；
2. 配置为主模式前，需要确保当前A2B节点没有处于从节点工作状态，以避免配置为主模式后BCLK和SYNC时钟冲突。通过读取A2b\_slave\_busy寄存器，可检查是否处于从节点工作状态；
3. 如果A2b\_slave\_busy寄存器为1，则当前节点不能马上配置为主节点，转步骤4），如果A2b\_slave\_busy寄存器为0，则转2.3.1.2节，进入Master唤醒步骤；
4. 如果主节点在本机箱上，需要将对应节点的A2b\_mode寄存器改为0，去除主节点功能，并转步骤2）。如果主节点在机箱外部，则转步骤5）；
5. 手动将外部的主节点移除，或将外部主节点更改为从节点，并转步骤2）；

#### Master 唤醒

主从机激活唤醒流程图

1. Rm A2B\_vendor 0x02 0xAD //读取值为AD
2. Rm A2B\_product 0x03 0x28//读取值为28，对器件进行识别
3. Wm A2B\_control 0x12 0x84// 配置Master模式且进行软复位
4. 设置工作于主模式的音频卡的A2b\_sample\_rate寄存器，设置采样率（48kHz/96kHz/192kHz）；
5. 设置主模式的音频卡的A2b\_mode寄存器为1，驱动SYNC管脚；
6. Rm A2B\_intsrc 0x16 0x80//判断中断源为Master interrupt
7. Rm A2B\_inttype 0x17 0xFF//判断中断类型为MSTR\_RUNNING，至此，Master唤醒完成

#### Slave0 唤醒

1. Wm A2B\_ intmask0 0x1B 0x77// SRF帧校验等源的中断使能
2. Wm A2B\_ intmask1 0x1C 0x78// I03\4\5\6管脚中断
3. Wm A2B\_ intmask2 0x1D 0x0F// slave中断和Dscdone 中断使能
4. Wm A2B\_becctl 0x1E 0xEF//使能SRF帧，数据流等CRC校验错误计数功能
5. Wm A2B\_intpnd2 0x1A 0x01// 清零Discovery Done bit
6. Wm A2B\_respcycs 0x0F 0X\*\*//配置相邻SCF之间的响应周期，计算方法见附录1
7. Wm A2B\_control 0x12 0x01// 设置A2B\_CONTROL.NEWSTRCT=1，使得Master相关配置生效
8. Wm A2B\_i2sgcfg 0x41 0x\*\*// 配置I2S接口

Note：此寄存器一定要在幻象供电之前配置，且配置后不允许再修改。

1. Wm A2B\_swctl 0x09 0x01//使能 A2B bus power switching，即使能幻象供电
2. Wm A2B\_discvry 0x13 0x\*\*//为将被发现的从节点配置响应时间值,计算方法见附录1

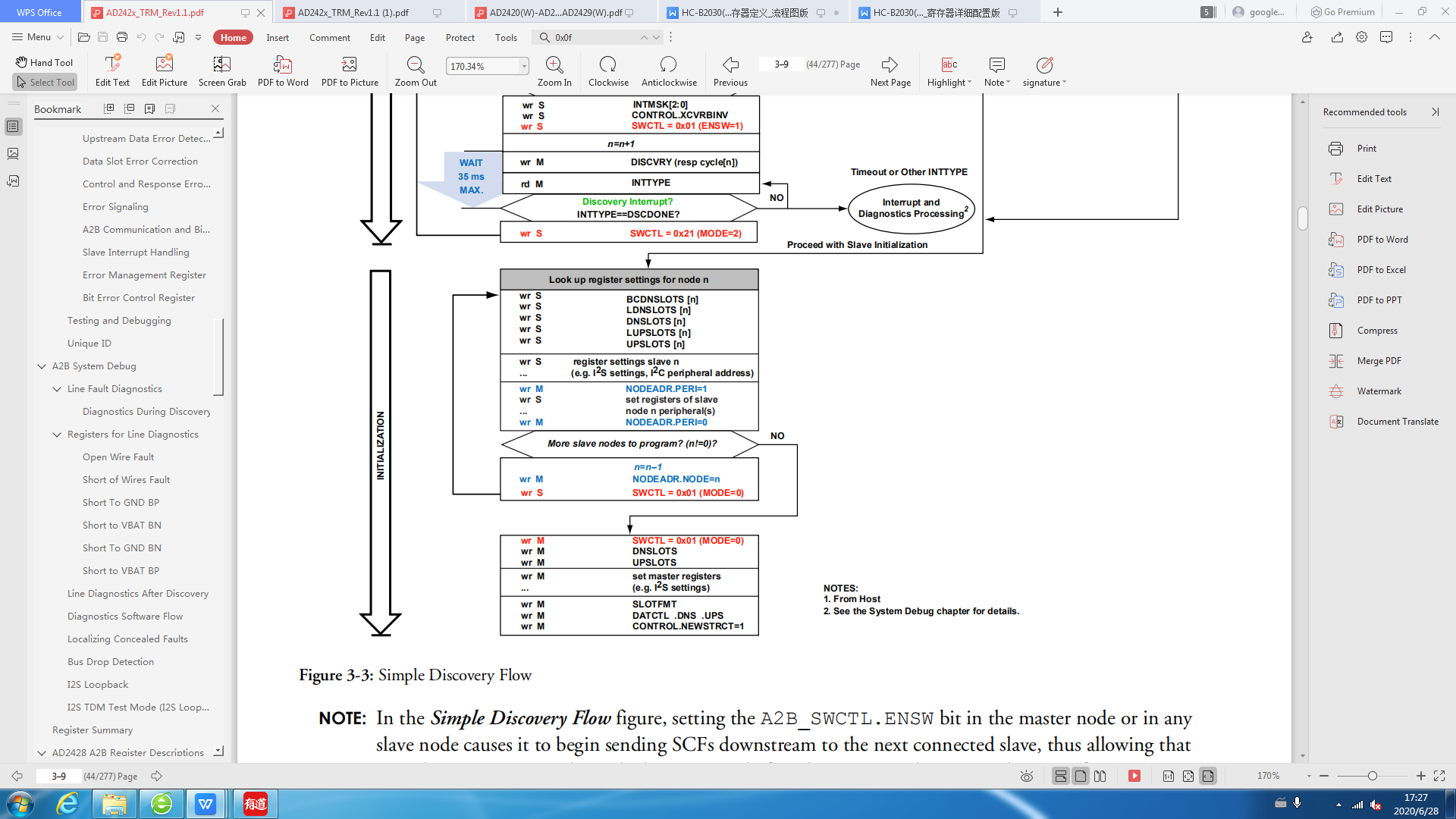
Note:第10步完成后，等待最多35ms即可读取寄存器0x16

1. Rm A2B\_intsrc 0x16 0x80//判断中断源为Master interrupt
2. Rm A2B\_inttype 0x17 0x18//判断中断类型为Discovery done interrupt
3. Wm A2B\_swctl 0x09 0x21//调整mode =2
4. Wm A2B\_nodeadr 0x01 0x00//给第一个发现的slave编号为slave 0
5. Rs A2B\_vendor 0x02 0xAD//读取slave 0的vendor number
6. Rs A2B\_product 0x03 0x28//读取slave 0的product number
7. Rs A2B\_version 0x04 0x00//读取slave 0的version number
8. Rs A2B\_CAPABILITY 0x05 0x01//表示slave0节点的IIC接口可用

Note:如果读取的vendor、product、version number都和期望值一致标志着slave0发现完成，slave0已唤醒，可以进行初始化配置。slave0唤醒后，如果A2B总线仍有多的slave需要唤醒初始化，则继续执行步骤19 –30，slave\*唤醒后执行slave\*的初始化操作。如果没有多的slave需要唤醒初始化则直接执行slave0的初始化操作。

1. Ws A2B\_ intmask0 0x1B 0x10
2. Ws A2B\_ intmask1 0x1C 0x00
3. Ws A2B\_swctl 0x09 0x01//使能 A2B bus power switching，即使能幻象供电
4. Wm A2B\_discvry 0x13 0x\*\*//为将被发现的从节点配置响应时间值,计算方法见附录1
5. Rm A2B\_intsrc 0x16 0x80//判断中断源为Master interrupt
6. Rm A2B\_inttype 0x17 0x18//判断中断类型为Discovery done interrupt
7. Ws A2B\_swctl 0x09 0x21//调整mode =2
8. Wm A2B\_nodeadr 0x01 0x01//给第2个发现的slave编号为slave 1
9. Rs A2B\_vendor 0x02 0xAD//读取slave 0的vendor number
10. Rs A2B\_product 0x03 0x28//读取slave 0的product number
11. Rs A2B\_version 0x04 0x00//读取slave 0的version number
12. Rs A2B\_CAPABILITY 0x05 0x01//表示slave0节点的IIC接口可用

#### Slave1 初始化



主从机初始化流程图

1. Ws A2B\_LDNslots 0x0b 0x\*\*// 表示当前节点在下行数据帧添加数据槽的数量
2. Ws A2B\_LUPSLOTS 0x0c 0x\*\*// 表示当前节点在上行数据帧添加数据槽的数量
3. Ws A2B\_ DNSLOTS 0x0d 0x\*\*// 表示当前节点后的下行数据槽数量
4. Ws A2B\_UPSLOTS 0x0e 0x\*\*// 表示当前节点前的上行数据槽数量
5. Ws 0x3f 0x01// 设置I2C Data Rate (Slave Only)为400Khz
6. Ws A2B\_i2sgcfg 0x41 0x\*\*// 配置 I2S接口
7. Ws A2B\_i2scfg 0x42 0x\*\*// 配置数据时钟使能I2S接口
8. Ws A2B\_pdmctl 0x47 0x18// 禁用PDM0、PDM1
9. Ws A2B \_chip 0x00 0x\*//给外设配置IIC地址
10. Wm A2B \_nodeadr 0x01 0x21//使能slave1外设

Note:此步骤后可以直接用bus地址对外设进行访问配置，无需配置的外设可忽略此提示

1. Wm A2B \_nodeadr 0x01 0x01//禁用slave1外设
2. Wm A2B \_nodeadr 0x01 0x00//如果还有多的节点需要初始化，本例中还有slave0，则将master中node地址定位到0节点，否则直接对master进行配置
3. Ws A2B\_swctl 0x09 0x01//调整mode=0

#### Slave0 初始化

1. Wm A2B \_nodeadr 0x01 0x00//将master中node地址定位到0节点
2. Ws A2B\_LDNslots 0x0b 0x\*\*// 表示当前节点在下行数据帧添加\*个 slots
3. Ws A2B\_DNslots 0x0d 0x\*\*// 表示经过当前节点后下行数据帧还有\*个slots
4. Ws A2B\_i2ccfg 0x3f 0x01// 设置I2C Data Rate (Slave Only)为400Khz
5. Ws A2B\_i2sgcfg 0x41 0x\*\*// 配置I2S接口
6. Ws A2B\_i2scfg 0x42 0x\*\*// 配置数据时钟使能I2S接口
7. Ws A2B\_pdmctl 0x47 0x18// 禁用PDM0、PDM1
8. Ws A2B\_swctl 0x09 0x01//调整mode =0
9. Wm A2B \_nodeadr 0x01 0x00//禁用slave0外设
10. Ws A2B \_chip 0x00 0x\*\*//给外设配置IIC地址
11. Wm A2B \_nodeadr 0x01 0x20//使能slave0外设

Note:此步骤后可以直接用bus地址对外设进行访问配置，无需配置的外设可忽略此提示

1. Wm A2B \_nodeadr 0x01 0x00//禁用slave0外设访问

#### Master 初始化

1. Wm A2B\_swctl 0x09 0x01//调整mode=0
2. Wm A2B\_DNslots 0x0d 0x\*\*// 表示master的下行DNSLOTS
3. Wm A2B\_UPslots 0x0e 0x\*\*// 表示master的上行UPSLOTS
4. Wm A2B\_i2sgcfg 0x41 0x\*\*// 配置I2S接口
5. Wm A2B\_i2scfg 0x42 0x\*\*// 配置数据时钟使能I2S接口
6. Wm A2B\_respcycs 0x0f 0x\*\*//设置主机的响应时间
7. Wm A2B \_slotfmt 0x10 0x\*\*//配置上下行数据宽度
8. Wm A2B \_datctl 0x11 0x03//使能UP和DN
9. Wm A2B\_control 0x12 0x81// 设置A2B\_CONTROL.NEWSTRCT=1，使得Master相关配置生效
10. Wm A2B\_control 0x12 0x82// 结束discovery

#### 配置完成后检查

读取A2b\_bclk\_cnt寄存器和A2b\_sync\_cnt，并计算I2S位数和采样率。

1. I2S位数：A2b\_bclk\_cnt / (2 \* A2b\_sync\_cnt)；（将结果四舍五入后取整数）;
2. 采样率：A2b\_cync\_cnt \* 100Hz（误差±100Hz）；

当前系统I2S位数为32，采样率只支持48kHz、96kHz、192kHz。如果不满足要求，需提示I2S位数配置错误或采样率配置错误，需要检查前面的配置步骤是否有错误，并重新配置；

### A2B从模式配置

操作步骤：

1. 设置A2b\_mode寄存器为0，进入从模式状态，确保不主动往外送BCLK和SYNC时钟信号；
2. Wm A2B\_control 0x12  0x00 // 将A2B节点配置为从模式
3. 如果主节点在本机箱上，则转2.3.1配置主节点；如果主节点在机箱外部，则根据外部节点主模式的方法进行配置；
4. 读取A2b\_slave\_busy寄存器，如果结果为1，表明从节点配置成功，否则提示配置失败错误信息，转步骤3确保主节点正确配置；
5. 读取A2b\_bclk\_cnt寄存器和A2b\_sync\_cnt，并计算I2S位数和采样率
6. I2S位数：A2b\_bclk\_cnt / (2 \* A2b\_sync\_cnt)；（将结果四舍五入后取整数）;
7. 采样率：A2b\_cync\_cnt \* 100Hz（误差±100Hz）；

当前系统I2S位数为32，采样率只支持48kHz、96kHz、192kHz。如果不满足要求，需提示I2S位数配置错误或采样率配置错误；

### A2B接收

操作步骤：

1. 读取A2b\_bclk\_cnt和A2b\_sync\_cnt寄存器，并根据2.3.2的计算公式，计算I2S位数和采样率，用于后续步骤将接收到数据按照正确的采样率和位数进行保存；
2. 设置A2b\_rx\_mult\_coef寄存器（可选），修改增益系数（浮点数），默认增益系数为2-31（浮点数）。修改该系数，需要确保音频数据从整形转换为归一化的浮点数后，不要超过(-1.0~-1.0)范围，否则可能导致声音失真；
3. 设置A2b\_rx\_en寄存器为0，禁止A2B数据接收；
4. 设置A2b\_rx\_en寄存器为1，使能A2B数据接收；
5. 读取A2b\_ddr\_head\_rx和A2b\_ddr\_tail\_rx寄存器，计算待采集数据长度。每次读取的字节数为以下结果的最小值：

* DDR环形缓冲中有效数据个数：(A2b\_ddr\_head\_rx - A2b\_ddr\_tail\_rx + 0x8000000) & 0x7FFFFFF;
* 单次DMA的最大长度：0x400000;
* 当A2b\_ddr\_head\_rx小于A2b\_ddr\_tail\_rx时，DDR环形缓冲区尾部的长度：0x8000000 – A2b\_ddr\_head\_rx；
* 剩余待采集长度（需8字节凑整）：(剩余长度 + 0x7) & ~0x7；

如果可读取字节数为0，则延迟1ms，并重复当前步骤，否则转步骤6；

1. 通过DMA读DDR，起始地址 = A2b\_ddr\_tail\_rx + 0x30000000，DMA长度为上一步骤的计算结果。
2. 更新A2b\_ddr\_tail\_rx寄存器。A2b\_ddr\_tail\_rx新值 = (A2b\_ddr\_tail\_rx旧值 + DMA读取字节数) & 0x7FF\_FFFF；
3. 更新已采集长度，如果已采集长度≥期望采集长度，则转步骤8；否则重复步骤5~7；
4. 将采集数据，根据采样率和I2S位数保存为wave文件；

### A2B发送

#### DDS模拟调试发送

本功能只用作内部调试时使用，不对外发布。

接收方操作步骤：

设置A2b\_rx\_mult\_coef寄存器，修改增益系数（浮点数），增益系数设置为2-25（浮点数）。因DDS输出数据为26位，去掉符号位还剩25位；

发送方操作步骤：

1. 设置DDS相移参数Dds\_cfg\_data。根据期望输出频率，计算相移参数。f=48000\* Dds\_cfg\_data /65536；
2. 设置Dds\_cfg\_trig寄存器为1；
3. 设置Dds\_out\_enable寄存器为1。

【注】：DDS默认输出频率为1kHz，如果不需要更改输出频率，可跳过步骤1、2。

#### DDR缓冲区发送

操作步骤：

1. 读取A2b\_bclk\_cnt和A2b\_sync\_cnt寄存器，并根据2.3.2的计算公式，计算I2S位数和采样率，用于后续步骤选择正确位数和采样率的文件进行发送；
2. 设置A2b\_tx\_mult\_coef寄存器（可选），修改增益系数（浮点数），默认增益系数为231（浮点数）。修改该系数，需要确保音频数据从归一化的浮点数转换为整数后，不要超过(-231~-231)范围，否则可能导致声音失真；
3. 设置A2b\_tx\_en寄存器为0，禁止数据发送；
4. 设置A2b\_sample\_rate\_tx寄存器，设置输出采样率（只有主模式可以设置输出采样率）；
5. 读取A2b\_ddr\_full\_tx寄存器，如果A2b\_ddr\_full\_tx的值为1，则延迟1ms，重复当前步骤；
6. 读取A2b\_ddr\_head\_tx、A2b\_ddr\_tail\_tx，计算待发送数据的长度：

待发送数据长度为以下各项的最小值：

* 环形缓冲剩余字节数：(A2b\_ddr\_head\_tx - A2b\_ddr\_tail\_tx + 0x7FFFFFF) & 0x7FFFFFF；
* 单次DMA最大长度：0x400000；
* 当A2b\_ddr\_head\_tx小于A2b\_ddr\_tail\_tx，DDR环形缓冲尾部的长度：0x8000000 - A2b\_ddr\_head\_tx；
* 剩余待发送长度（需8字节对齐）：(剩余长度 + 0x7) & ~0x7；

1. 从当前位置读取待发送的音频文件，提取左右声道的数据。每个采样点包含8个字节，如果发送立体声，则前4个字节为左声道数据，后四个字节为右声道数据；如果发送单声道数据，则前4个字节为单声道数据，后四个字节补0。注意待发送数据长度为8字节的整数倍，长度不足8字节的，补0对齐；
2. 将上一步读取的音频数据通过DMA写到DDR环形缓冲区。写地址为A2b\_ddr\_head\_tx + 0x38000000。
3. 更新A2b\_ddr\_head\_tx寄存器值。新A2b\_ddr\_head\_tx = (旧A2b\_ddr\_head\_tx + DMA长度) & 0x7FFFFFF；
4. 更新已发送音频文件长度，如果已发送长度≥音频文件长度，则结束，否则重复步骤4~8。

附录1：Response Cycle Formula

1. 主节点响应时间设置
2. 计算DNSLOT\_ACTIVITY、UPSLOT\_ACTIVITY

NUM\_DNSLOTS：下行数据数目 DNSLOT\_SIZE：下行数据位宽

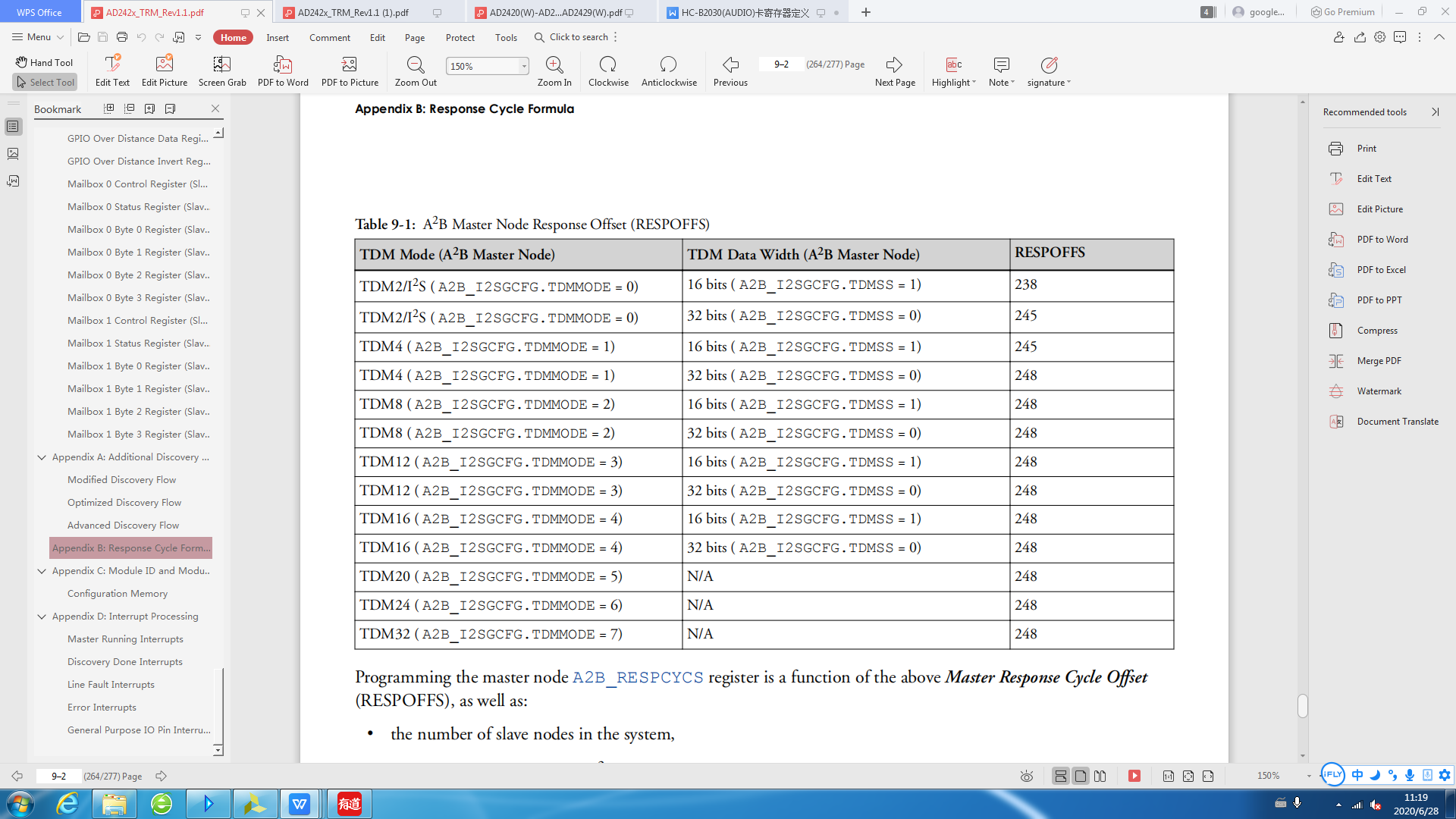
N：当前节点在拓扑结构中的节点值。

DNSLOT\_ACTIVITY[n] = NUM\_DNSLOTS \* (DNSLOT\_SIZE + 1)

UPSLOT\_ACTIVITY[n] = NUM\_UPSLOTS \* (UPSLOT\_SIZE + 1)

1. 查找主节点响应偏移

根据I2S通信中TDM格式查找对应的主机RESPOFFS



例：I2S接口配置为TDM2, TDMSS为32bit。则对应的Master RESPOFFS为 245

1. 计算上行下行的响应周期要求

RESPCYCS\_DN[n] = ((64 + DNSLOT\_ACTIVITY[n])/4) + 4n + 2

RESPCYCS\_UP[n] = RESPOFFS - (((64 + UPSLOT\_ACTIVITY[n])/4) + 1)

1. 计算主节点的响应周期

A2B\_RESPCYCS = (MAX(RESPCYCS\_DN[n]) + MIN(RESPCYCS\_UP[n])) / 2

1. 从节点响应时间设置

SLV\_RESPCYCS[n] = MSTR\_RESPCYCS - 4n