HC-B2020(AIO)卡寄存器定义

版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| FPGA版本 | 修改事项 | 修改时间 | 修改人 |
| 1.00 | 初始版本 | 2020-05-25 | 王勇 |
| 1.02 | 1、增加Over\_current\_width寄存器，用于过流保护滤波功能  2、增加Dac\_enable寄存器，使板卡上电默认无输出  3、增加Ai\_level\_avg\_exp寄存器，用于电平输入平均，提高测试精度  4、修改电平输入测试流程，见2.1  5、修改电平输出测试流程，见2.3 | 2020-07-09 | 王勇 |
| 1.04 | 1、增加PPS相关的寄存器  2、增加PPS秒脉冲工作流程，见2.5  3、增加在线升级寄存器说明 | 2020-07-15 | 王勇 |
| 1.07 | 1、增加PPS秒脉冲计数寄存器，代表系统总攻接收的PPS秒脉冲个数  2、修改在线升级SPI地址偏移 | 2020-07-18 | 王勇 |
| 1.11 | 增加系统时钟校准功能 | 2020-08-04 | 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 | |  |
| **AI相关寄存器** | | | | | | | | | | |
| Ai\_mode | | 模拟输入模式 | 824 | | RW | | 1 | 0 | | 输入模式  Bit值含义：  0:电平模式  1：波形模式 |
| Ai\_level0 | | 通道0输入电平 | 500 | | R | | 16 | 0 | | 通道0输入电平，bit[15:0]代表ADC采样值，需要结合ADC的PGIA（增益），以及前端的电阻分压，计算实际的电压值 |
| …… | |  |  | |  | |  |  | |  |
| Ai\_level31 | | 通道31输入电平 | 57C | | R | | 16 | 0 | | 通道31输入电平，bit[15:0]代表ADC采样值，需要结合ADC的PGIA（增益），以及前端的电阻分压，计算实际的电压值 |
| Ai\_level\_delay | | 电平输入通道切换延迟 | 580 | | RW | | 4 | 7 | | 电平输入通道切换延迟  Bit[3:0]含义：  模拟开关切换后，AI输入采样延迟计数。时间单位为采样周期。 |
| Ai\_level\_avg\_exp | | 电平输入平均系数 | 584 | | RW | | 3 | 3 | | 电平输入平均系数（2的幂次方）  Bit[2:0]含义：  平均系数（2的幂次方），例如默认值为3，代表对23=8个采样点进行平均，从而提高测试精度 |
| Adc\_reset | | ADC复位 | 800 | | RW | | 1 | 1 | | ADC复位信号  Bit值含义：  0：正常模式  1：复位 |
| Adc\_config\_en | | ADC配置使能 | 804 | | W | | 1 | - | | ADC配置使能信号，应用Adc\_config和Adc\_asr0的配置  Bit值含义：  1：配置生效  0：忽略 |
| Adc\_config | | 基础配置寄存器 | 808 | | RW | | 16 | F7EE | | 基础配置寄存器，见图 2。  以下设置，四个通道的PGIA设置相同：  F7FE:20.48V  F6FE:0.64V  F67E:1.28V  F5FE:2.56V  F57E:5.12V  F4FE:10.24V |
| Adc\_asr0 | | ADC ASR0配置寄存器 | 80C | | RW | | 16 | 7FFF | | ASR0配置寄存器，见图 3。 |
| Adc\_clk\_divider | | ADC时钟分频计数器 | 810 | | RW | | 32 | 80 | | ADC时钟分频计数器,需要在ADC复位信号为1时，才能配置 |
| Adc\_sw0 | | 通道切换开关0 | 814 | | RW | | 3 | 0 | | 输入通道0~7八选一切换开关。在波形模式下，需要手动切换开关。 |
| Adc\_sw1 | | 通道切换开关1 | 818 | | RW | | 3 | 0 | | 输入通道8~15八选一切换开关。在波形模式下，需要手动切换开关。 |
| Adc\_sw2 | | 通道切换开关2 | 81C | | RW | | 3 | 0 | | 输入通道16~23八选一切换开关。在波形模式下，需要手动切换开关。 |
| Adc\_sw3 | | 通道切换开关3 | 820 | | RW | | 3 | 0 | | 输入通道24~31八选一切换开关。在波形模式下，需要手动切换开关。 |
| Ddr\_head | | DDR环形缓冲区头部（写DDR地址） | 880 | | R | | 32 | 0 | | DDR环形缓冲区头部地址，当ADC数据往DDR写入时，该地址会增长 |
| Ddr\_tail | | DDR环形缓冲区尾部（读DDR地址） | 884 | | RW | | 32 | 0 | | DDR环形缓冲区尾部地址，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，又从0开始  DDR最大地址为0x7FF\_FFFF（1GB/8,每个地址代表8字节） |
| Ddr\_trans\_num | | DDR每次传输长度 | 888 | | RW | | 16 | 1024 | | DDR每次传输长度（单位8字节） |
| **AO相关寄存器** | | | | | | | | | | |
| Over\_current | | 过流保护 | 8C0 | | R | | 32 | 0 | | 过流保护状态，每个bit代表一个通道 |
| Over\_current\_clear | | 过流保护清楚 | 8C4 | | W | | 32 | 0 | | 清除过流保护状态，每个bit代表一个通道，写1清除过流保护；写0忽略； |
| Over\_current\_width | | 过流保护持续时间 | 8C8 | | RW | | 16 | 10 | | 过流保护持续时间  Bit[15:0]含义：  过流持续时间，单位16ns |
| Dac\_enable | | DAC输出使能 | 8CC | | RW | | 4 | 7 | | DAC输出使能，每个bit位代表一个DAC  Bit[n](n=0..3)含义：  0：禁止DAC输出  1：使能DAC输出 |
| **PPS相关寄存器** | | | | | | | | | | |
| Pps\_source | | PPS源 | 900 | | RW | | 1 | 0 | | PPS源  Bit[0]含义：  0：PXI Trigger  1：内部PPS |
| Pps\_edge\_sel | | PPS边沿选择 | 904 | | RW | | 1 | 0 | | PPS边沿选择  Bit[0]含义:  0:下降沿有效  1:上升沿有效 |
| Pps\_trig\_sel | | PXI TRIGGER选择 | 908 | | RW | | 3 | 0 | | PXI TRIGGER选择  Bit[2:0]含义：PXI Trigger 0~7 |
| Pps\_second\_set | | 系统秒计数设置 | 90C | | RW | | 32 | 0 | | 系统秒计数设置  Bit[31:0]含义：32位的秒计数值 |
| Pps\_second\_valid | | 系统秒计数设置有效 | 910 | | W | | 1 | 0 | | 系统秒计数设置有效  Bit[0]含义：  0：忽略  1：设置秒计数值有效 |
| Pps\_export\_en | | PPS输出使能 | 914 | | RW | | 8 | 0 | | PPS输出使能  Bit[7:0]含义：  每个bit代表一跟PXI Trigger线，0代表不输出PPS，1代表将内部PPS从PXI Trigger线输出 |
| Pps\_seconds | | 系统秒计数 | 918 | | R | | 32 | 0 | | 系统秒计数  Bit[31:0]含义：系统的秒计数绝对值  【注】用户设置秒计数会对系统秒计数值重新初始化；每一个PPS秒脉冲到来时，秒计数值加1； |
| Pps\_nano\_seconds | | 系统纳秒计数 | 91C | | R | | 32 | 0 | | 系统纳秒计数  Bit[31:0]含义：系统的纳秒计数绝对值。  【注】该值始终小于109，如果没有秒脉冲，该值将保持最大值不变。每次秒脉冲到来会重新复位该值。 |
| Pps\_counter | | PPS秒计数 | 920 | | R | | 32 | 0 | | PPS秒计数  Bit[31:0]含义：接收到的PPS秒脉冲个数。 |
| Wave\_seconds | | 波形采集起始时间秒计数 | 840 | | R | | 32 | 0 | | Bit[31:0]含义：模拟输入波形采集第一个采样点的秒计数绝对值  【注】当模拟输入模式更改为电平输入，或模拟输入复位时，该值将在下一次波形采集的第一个采样点时刻被更新 |
| Wave\_nano\_seconds | | 波形采集起始时间纳秒计数 | 844 | | R | | 32 | 0 | | Bit[31:0]含义：模拟输入波形采集第一个采样点的纳秒计数绝对值  【注】当模拟输入模式更改为电平输入，或模拟输入复位时，该值将在下一次波形采集的第一个采样点时刻被更新 |
| **系统时钟校准** | | | | | | | | | | |
| Cal\_counter | 系统时钟校准计数 | | 940 | R | | 32 | | 0 | 系统时钟校准计数，该计数是1秒内62.5MHz主时钟的计数值 | |
| Cal\_done | 系统时钟校准完成 | | 944 | R | | 1 | | 0 | 系统时钟校准完成，0为未校准完成；1标识校准完成 | |
| **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。 | | | | | | | | | | |
| **DAC IIC操作寄存器（配置输入参考电压）** | | | | | | | | | | |
| 地址偏移：0x4000，详细定义参考XILINX pg090-axi-iic.pdf，table2-4  DAC IIC操作方法，参考Microchip MCP4728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。 | | | | | | | | | | |

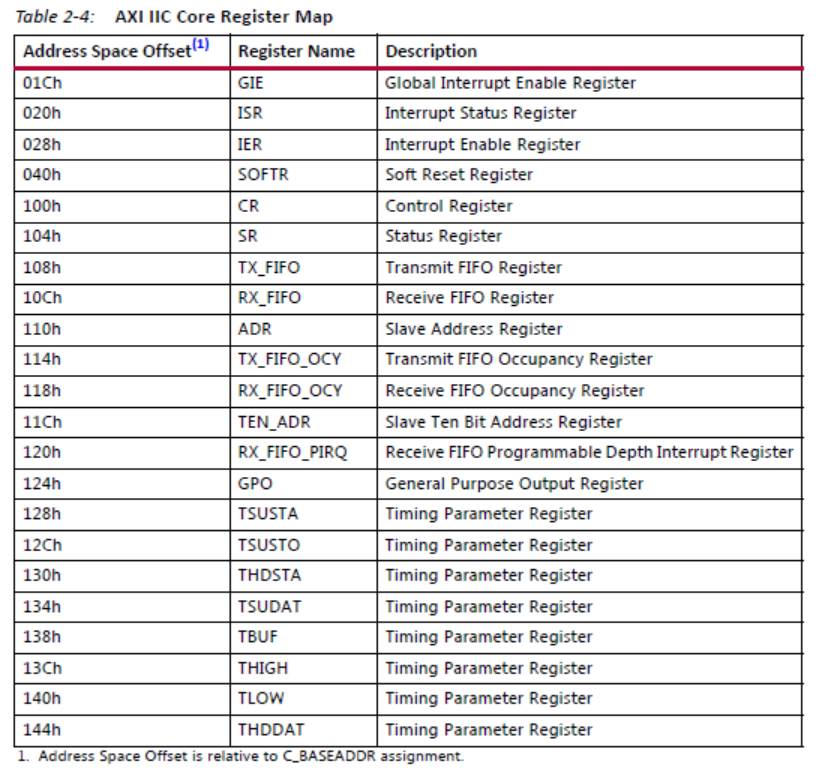
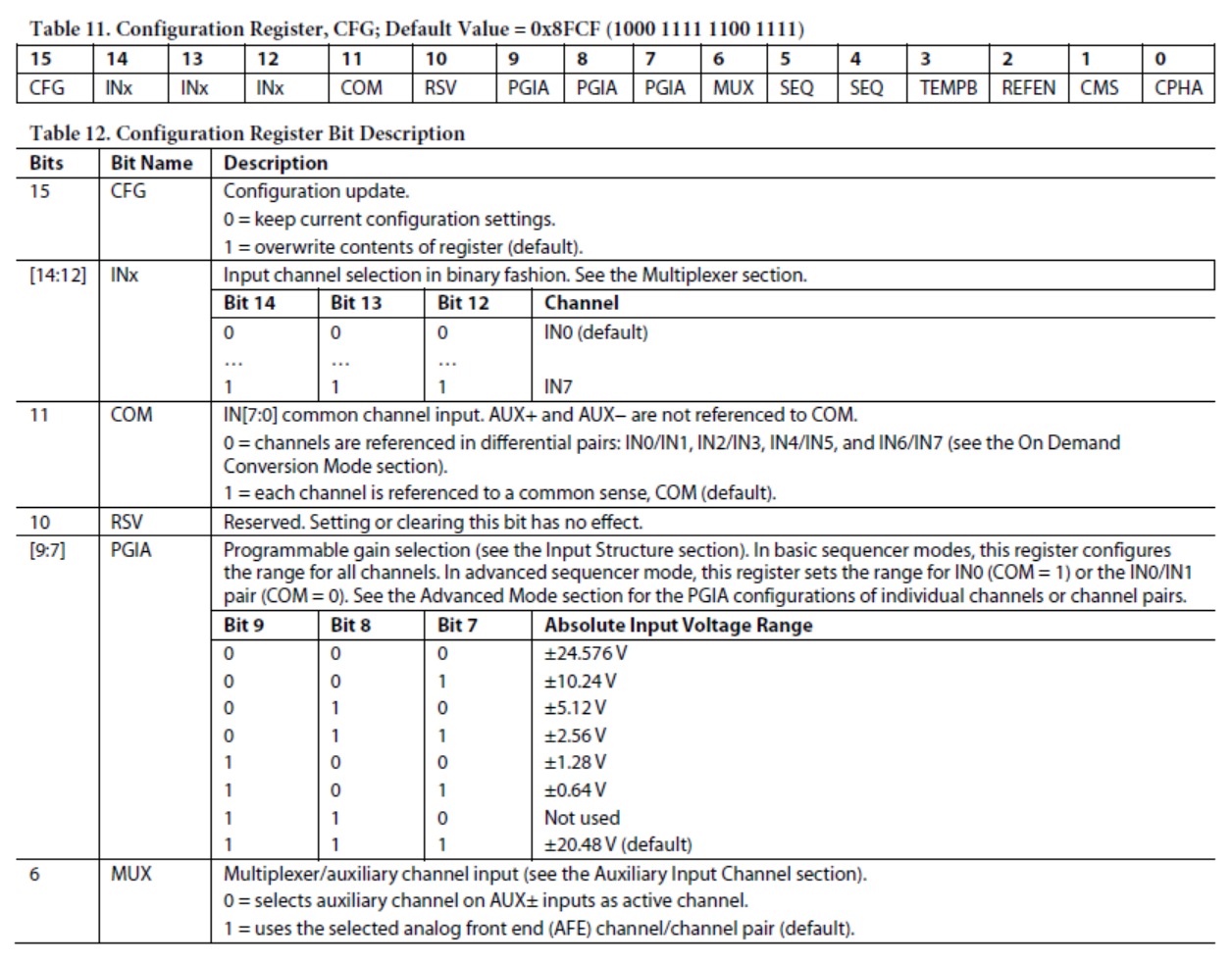


图 1 AXI IIC寄存器定义



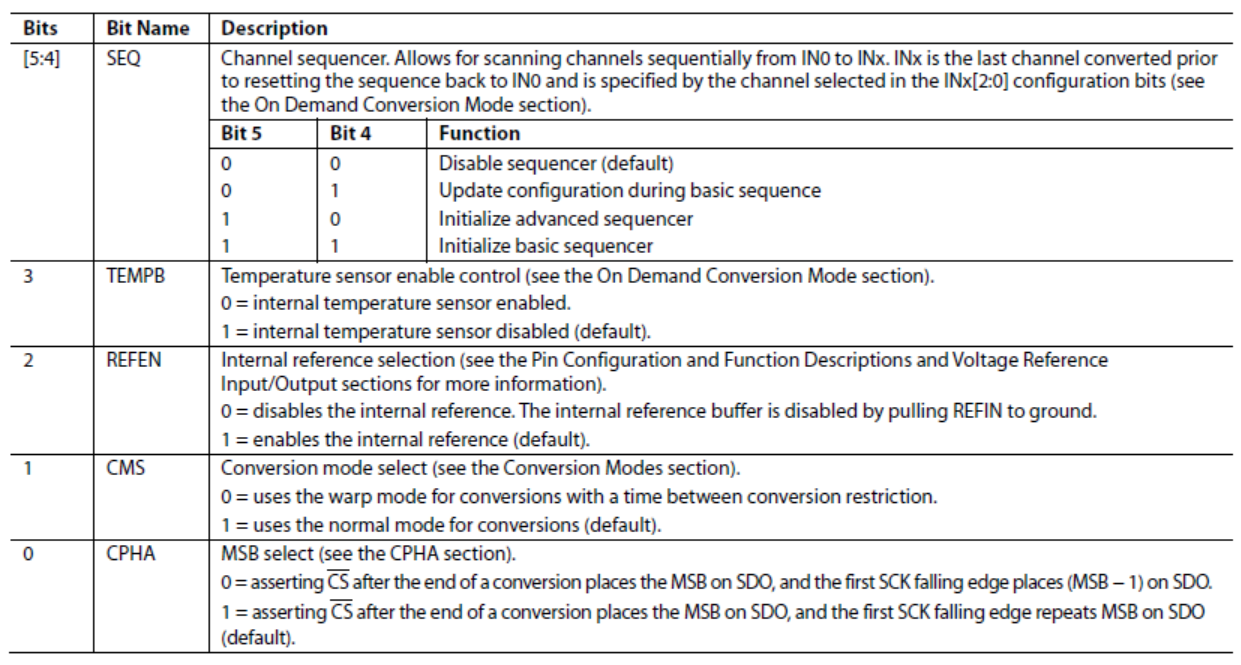


图 2 ADAS3022基础寄存器定义

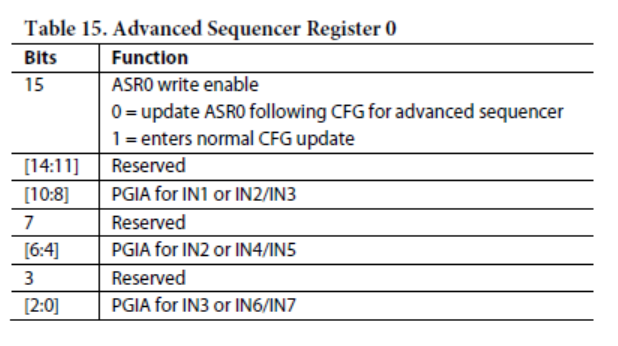


图 3 ADAS3022 ASR0寄存器定义

# 常见功能设置流程

## 时钟校准

系统62.5MHz主时钟存在固定的频偏，需要进行校准。

读取Cal\_counter寄存器，时钟校准系数***clk\_coef*** = 62500000/ Cal\_counter。如果该值＞1，表明实际的时钟频率＞62.5MHz；反之，如果该值＜1，则表明实际的系统时钟＜62.5MHz；

## 电平输入

操作步骤：

1. 设置Adc\_config寄存器，配置ADC通用配置（含第一路增益配置）；
2. 设置Adc\_asr0寄存器，配置ADC剩余三路的增益；
3. 设置Ai\_level\_delay寄存器（可选），配置每次模拟开关切换后，稳定几个采样周期后，开始采集电压；
4. 设置Ai\_level\_avg\_exp寄存器（可选），配置每次电平采样值，由几个采样点进行平均。平均次数为2 Ai\_level\_avg\_exp。
5. 设置Adc\_reset寄存器，写0，解除复位状态；
6. 设置Adc\_config\_en寄存器，写1，应用Adc\_config和Adc\_asr0寄存器配置；
7. 设置Ai\_mode寄存器，写0，设置为电平模式；
8. 读取Ai\_level0~Ai\_level31寄存器，分别获取通道0~通道31电平；

## 波形输入

1. 设置Adc\_config寄存器，配置ADC通用配置（含第一路增益配置）；
2. 设置Adc\_asr0寄存器，配置ADC剩余三路的增益；
3. 设置Adc\_sw0~sw3寄存器，配置需要采集波形的通道；
4. 设置Adc\_reset寄存器，写0，解除复位状态；
5. 设置Adc\_config\_en寄存器，写1，应用Adc\_config和Adc\_asr0寄存器配置；
6. 设置Ai\_mode寄存器，写1，设置为波形模式；
7. 读取Ddr\_head和Ddr\_tail寄存器，<DDR中有效数据个数> = (Ddr\_head - Ddr\_tail + 0x3FFF\_FFFF) & 0x3FFF\_FFFF，当DDR中有效个数大于0时，转步骤8）；
8. DMA读DDR，<起始地址>=Ddr\_tail，<字节数>=min(4\*1024\*1024，0x3FFF\_FFFF – Ddr\_head)，字节数需要为8的正整数倍；
9. 修改Ddr\_tail寄存器，<Ddr\_tail新值> = (<Ddr\_tail旧值> + <DMA读取字节数>) & 0x3FFF\_FFFF；
10. 重复步骤7）~步骤9）；

## 电平输出

操作步骤：

1. 设置Over\_current\_width寄存器（可选），更改过流保护持续时间（即进行过流保护滤波），单位为16ns；
2. 电平设置，使用axi-iic的IP CORE对DAC进行配置，偏移地址为0x4000。具体配置参考XILINX pg090-axi-iic.pdf和OPA1679 datasheet；
3. 设置Dac\_enable寄存器对应的通道为1，使能DAC输出；

## 过流保护

操作步骤：

1. 当出现过流保护时，板卡RUN/ALARM指示灯变为橙色闪烁状态；
2. 读取Over\_current寄存器，读取发生过流保护的通道；
3. 进行过流保护故障排除；
4. 设置Over\_current\_clear寄存器，清除过流保护状态，恢复正常状态；

## PPS秒脉冲

### 更新秒计数

1. 向Pps\_second\_set寄存器写入新的秒计数绝对值；
2. 向Pps\_second\_valid寄存器写1；

### 读取波形采集时刻

波形采集时刻返回一个64位数的微秒计数值，其操作步骤如下：

1. 读取Wave\_seconds寄存器；
2. 读取Wave\_nano\_seconds寄存器。由于纳秒计数采用了62.5MHz系统时钟，故需要进行校准，实际的纳秒计数为：min(999999999, Wave\_nano\_seconds \* ***clk\_coef***) ；
3. 微秒计数值为：Wave\_seconds\*106 + min(999999, Wave\_nano\_seconds \* ***clk\_coef*** / 1000)。

### 读取系统秒计数

直接读取Pps\_seconds寄存器并返回即可。