1 AD5761

**2.1** **AD5761 va 起始地址：0x6000\_0000**

1. **AD5761 CMD FIFO：0x60000000 W**

5761 输出FIFO

Bit23~bit0是写数据

此命令FIFO跟PWM波形关联，写进此FIFO的AD值在PWM低电平中间点输出

1. **AD5761 CMD FIFO STATE：0x60000004 R**

AD5761命令FIFO状态寄存器

AD5761命令寄存器实际是一个FIFO，软件可以连续往命令寄存器写入多条命令，但写之前需要检查命令FIFO状态，只有在非FULL的状态才能继续写命令。

Bit0：cmd fifo full

Bit1：cmd fifo empty

Bit2：ad5761 controller busy

1. **AD5761 CMD REG：0x60000008 W**

5761命令寄存器

Bit23~bit0是写数据

软件写此寄存器之后，硬件将软件写下来的bit23~bit0按照下图时序发送给AD5761

时序图如图1：

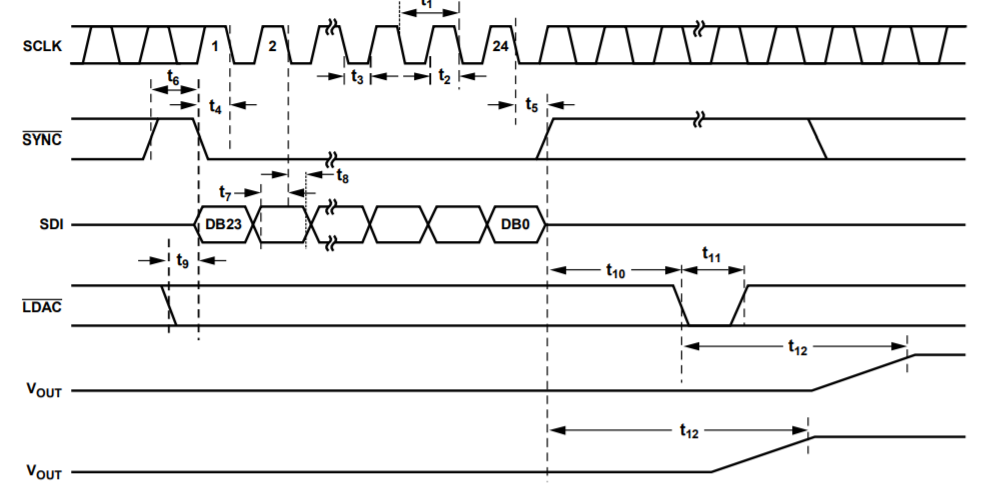


图1、AD5761时序图

1. **AD5761 CMD FIFO RESET：0x6000000C W**

复位AD5761命令FIFO

1. **AD5761 FIFO DATA : 0x60000010 R**

当前DAC从FIFO中读取的数据

**2.2** **AD5761 vb 起始地址：0x6010\_0000**

**2.3** **AD5761 ia 起始地址：0x6020\_0000**

**2.4** **AD5761 ib 起始地址：0x6030\_0000**

2、AD5318

1. **AD5318 CMD：0x60400000 W**

5318命令寄存器

Bit30~bit29控制选中哪个5318

Bit29：选中第1个

Bit30：选中第2个

可以同时选中多个5318，比如bit30~bit29写3，后续命令会同时发送给2个5318

Bit15~bit0是写数据

软件写此寄存器之后，硬件会将软件写下来的bit15~bit0按照图2时序发送给AD5318

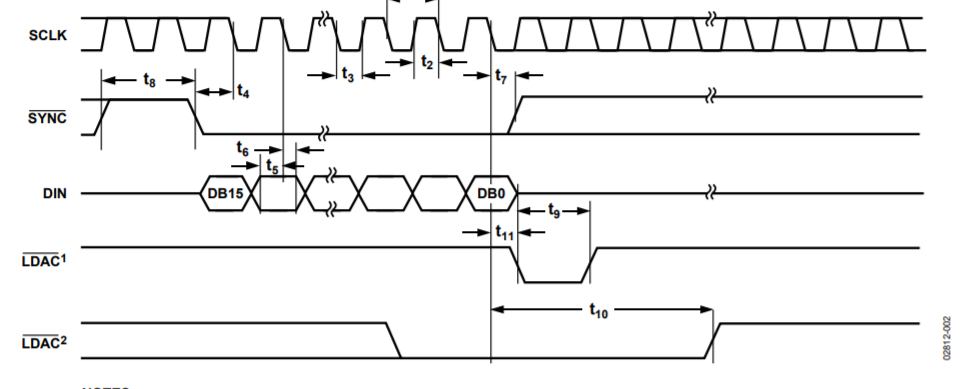


图2、AD5318\_0时序图

1. **AD5318 CMD FIFO STATE：0x60400004 R**

AD5318命令FIFO状态寄存器

AD5318命令寄存器实际是一个FIFO，软件可以连续往命令寄存器写入多条命令，但写之前需要检查命令FIFO状态，只有在非FULL的状态才能继续写命令。

Bit0：cmd fifo full

Bit1：cmd fifo empty

1. **AD5318 CMD FIFO WDATA: 0x60400008 R**

当前DAC从FIFO中读取的数据

1. PWM

**3.1** **PWM\_V**

1. **PWM高电平长度：0x60500000 W**
2. **PWM低电平长度：0x60500004 W**
3. **PWM个数：0x60500008 W**

高电平长度和低电平长度都是以周期为单位的，目前时钟40Mhz，周期25ns

1. **PWM输出使能：0x6050000C W**

Bit0 : 写1此寄存器之后PWM开始输出，AD在PWM低电平中间点输出（触发模式下配置完PWM后也需要开输出）

Bit1：写1此寄存器PWM引脚持续输出高电平

1. **关输出：0x60500010 RW**

Bit0：此寄存器写1，停止PWM输出，PWM计数器清零

1. **触发模式：0x60500014 RW**

BIT[1]:写0普通模式，写1触发模式

BIT[0]:写0作为主机，写1作为从机

触发模式整个采样结束后，需要关PWM输出才能清除TRIG线的状态，v路与i路都开输出触发线才会拉高

触发逻辑：从机配置完PWM——拉高trig\_out（从）——主机配置完PWM，检测trig\_in——主机检测到trig\_in拉高，主机开输出，并拉高trig\_out（主）——从机检测到主机trig\_in拉高，开输出

**3.2 PWM\_I**

起始地址:0x60600000

1. GPIO
2. **GPIO[31:0] OUT ENABLE：0x60500020 W**

对应Bit写1使能对应GPIO

1. **GPIO[63:32] OUT ENABLE：0x60500024 W**

对应Bit写1使能对应GPIO

1. **GPIO[31:0]：0x60500028 RW**

写对应Bit为0或1可输出高低电平

读对应Bit可获得当前对应GPIO是高或低电平

1. **GPIO[63:32]：0x6050002C RW**

写对应Bit为0或1可输出高低电平

读对应Bit可获得当前对应GPIO是高或低电平

1. **当前GPIO [5:2] 对应的是LED[5:2]**
2. AD7768 起始地址：0x6070\_0000
3. **AD7768 命令寄存器：0x60700000 WR**
4. **AD7768 0路采样数据FIFO读取：0x60700004 R**
5. **AD7768 0路实时采样数据：0x60700008 R**
6. **AD7768 1路采样数据FIFO读取：0x6070000C R**
7. **AD7768 1路实时采样数据：0x60700010 R**
8. **AD7768 2路采样数据FIFO读取：0x60700014 R**
9. **AD7768 2路实时采样数据：0x60700018 R**
10. **AD7768 3路采样数据FIFO读取：0x6070001C R**
11. **AD7768 3路实时采样数据：0x60700020 R**
12. **fifo状态寄存器：0x60700024 R**

BIT0:通道0 fifo空标志 BIT1:通道0 fifo满标志

BIT2:通道1 fifo空标志 BIT3:通道1 fifo满标志

BIT4:通道2 fifo空标志 BIT5:通道2 fifo满标志

BIT7:通道3 fifo空标志 BIT6:通道3 fifo满标志

1. **AD7768采样读取延时：0x60700028 W**

延时单位25ns

1. **AD7768采样读取个数：0x6070002C W**

采样点之间延时4微秒

1. **AD7768清空所有FIFO:0x60700030 W**

bit0：此位写1清空所有FIFO

1. **AD7768侦测寄存器:0x60700034 R**

Bit16:命令寄存器写有效位，当此寄存器为1时，写入的16位命令有效

Bit[15:0]:SPI配置寄存器回读数据

1. v路参数a配置：0x60700038 W
2. v路参数b配置：0x6070003C W
3. i路参数a配置：0x60700040 W
4. i路参数b配置：0x60700044 W
5. v路档位信息：0x60700048 W
6. i路档位信息：0x6070004C W

低位5bit写入寄存器

取出的数据结构是：{档位信息[3127]，ADC二进制原码数据[26:0]}

1. 点数平均个数寄存器: 0x60700050 W
2. AD数据fifo切换寄存器: 0x60700054 W

Bit0:写1数据平均后存入fifo中，写0只有脉冲模式才存

1. 平均后数据有效寄存器: 0x60700058 R

当该寄存器为1时，表示此时平均之后的数据有效，读数之后软核置0

1. LVDS接口：起始地址0x60800000
2. **往empu\_to\_lvds fifo中写数据：0x60800000 W**

写入数据位宽8bit即一字节，写入后的数据会在后续模块组帧发出。

写入的数据只有前八位有效，写数据之前需要先判断fifo 满标志。

1. **往上行常规帧 fifo中写数据：0x60800004 W**

写入数据位宽8bit即一字节，写入后的数据会在后续模块组帧发出。

写入的数据只有前八位有效，写数据之前需要先判断fifo 满标志。

1. **切换LVDS链路从哪一个FIFO取数据：0x60800008 W**

BIT0:写0选择从常规fifo取数，写1选择从empu\_to\_lvds fifo中取数

1. **从lvds\_to\_empu fifo中读取帧数据：0x6080000C R**

读出的数据位宽8bit，也就是一字节一字节的读出，读数据之前需要先判断fifo空标志。

1. **fifo状态寄存器： 0x60800010 R**

Bit0:empu\_to\_lvds fifo 空标志 Bit1:empu\_to\_lvds fifo 满标志

Bit2:lvds\_to\_empu fifo 空标志 Bit3:lvds\_to\_empu fifo 满标志

Bit4:常规fifo 空标志 Bit5:常规fifo 满标志

1. **100K采样fifo复位：0x60800014 W**

BIT0:写1复位