BR0101固件开发——模数转换器AD9715

（第三周）

# 1固件的改进与SPI接口的应用

为了尝试解决上板调试中遇到的问题，我们决定再次对固件进行改进，利用SPI接口读写AD9715内部寄存器的值，控制其采用内部参考电阻，并启用自动校准等功能。

## 1.1 SPI通信的实现

AD9715采用SPI进行通信，这是一种比较常见的通信协议，但是SPI缺少权威的规范标准，只有摩托罗拉的实用标准。所以各个厂商在芯片中使用的SPI信号定义和时序可能略有不同，比如AD9715采用的就是3线SPI，就是把常用的MISO、MOSI两个数据信号合成一个。这也造成了一些麻烦。

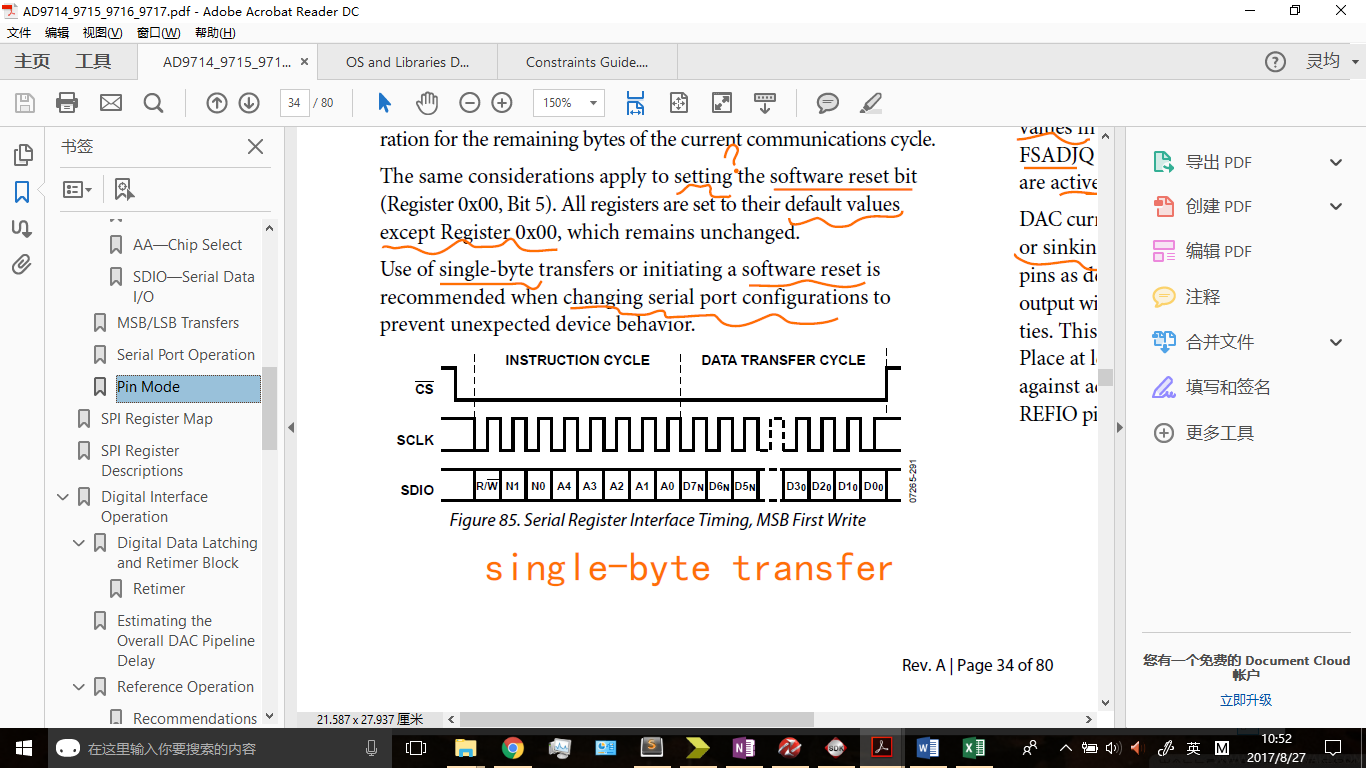


图 17 AD9715 SPI接口时序[4]

AD9715 SPI接口时序如上图所示，一个消息周期分为两个阶段，具体在第一周的报告中已经描述过。下面我们需要考虑的就是如何实现SPI的读写功能。

### 1.1.1 方案一 利用Xilinx提供的SPI IP核

Xilinx提供了一些SPI相关的IP核，不过在Virtex-4系列上能够使用的比较少。IP核添加向导中只能找到SPI 4.2的IP核。

H ierarchy 
test brOIOI 
xc4vsx55-IOff1148 
microblaze_top (microblaze_top.v 
microblaze_i - 
microblaze (micro 
rst_generator_O - rst_generator ( 
- clk_generator ( 
microblaze_top.ucf 
Processes: microblaze_top.ucf 
user Constraints 
Edit Constraints (Text) 
Li braries 
Console 
Xilinx Platform 
Xilinx EDK 14.7 Build EDK P. 20131013 
Copyright (c) I ggS—2012 Xilinx, Inc. 
Launching xps GUI. 
The IP Catalog has been reloaded. 
ISE Project Navigator (9.20131013) - .xise - 
[Design Summary (Programming File Generated)] 
Eile Edit yiew Pcoject Source erocess Tools Window La20ut Help 
Functi n 
• Pinal ri.ing sco 
1 n F 1 les Results 
New Sou rce Wiza rd 
Coregen or Archi tecture Wizard IF 
w by 
Na me 
q RXAUI 
S?' 4.2 
S?' 4.2 
q 
SPI-4.2 
q 
SPI-4.2 
q 
SPI-4.2 
Catalog: 
All 
All rights reserved. 
Find 
Version 
10.5 
11.2 
11.3 
11.4 
AX14 
only IF 
patible with chosen part 
status (08/21/2017 
ple.entation State: 
• 'arnings: 
• Routing 
Results: 
arni ngs 
ion 
49, 152 
Console 
Add a new source to the project 
W ngs 
File Generated 
81 Warnin 
S i ends 
Completelv Routed 
Constraints Met 
45 new 
14:58 
2017/8/21 

图 18 Virtex-4中SPI相关的IP核

SPI 4.2是一种高速的芯片间通信协议，与我们常用的低速SPI通信接口似乎不太一样。不过，在XPS中还可以找到Xilinx提供的SPI外设。

ISE Project Navigator (9.20131013) - 
File Edit View Project Source Process 
System 
t E m the "Available Peripher 
th or peripher 
console 
console 
Design 
H ierarchy 
test spi 
Simulati on 
xc4vsx55-IOff1148 
microblaze (microblaze.xmp) 
Processes: microblaze 
Design Summary/Reports 
Design Utilities 
Manage Processor Design (XPS) 
Generate Top HDL Source 
Export Hardware Design To SDK wi. 
Li brarie 
To implement this design 
01 run the Generate Top HDL 
which references this design 
implement 01 analyze p 
Launching Design Summary/ Report Vi 
W ngs 
Base System Builder 
peripheral Configurati on 
add a peripheral, dr g 
click on the peripheral 
i 1 able peripherals 
Peripheral Names 
10 Devices 
Internal Peripherals 
Imb bram if cntlr 
xps_bra r 
xps_timebase wet 
xps_tl mer 
e) Peripherals 
o change 
Pa ra meter 
x ps_spl 
All 
Co re 
Generic SPI 
Co re 
C NUM SS BITS 
C NUM TRANSFER BITS 
C SCK RATIO 
Include Receive and Transmit FIFO 
use Interrupt 
dlmb cntlr 
Core: Imb bram if cntlr 
ilmb cntlr 
Core: Imb bram if cntlr 
arni ngs 
arni ngs 
9:51 
2017/8/24 

图 19 XPS中的SPI外设

问题在于，这个外设是基于四线SPI的。要用于AD9715的SPI通信，必须做一些修改。

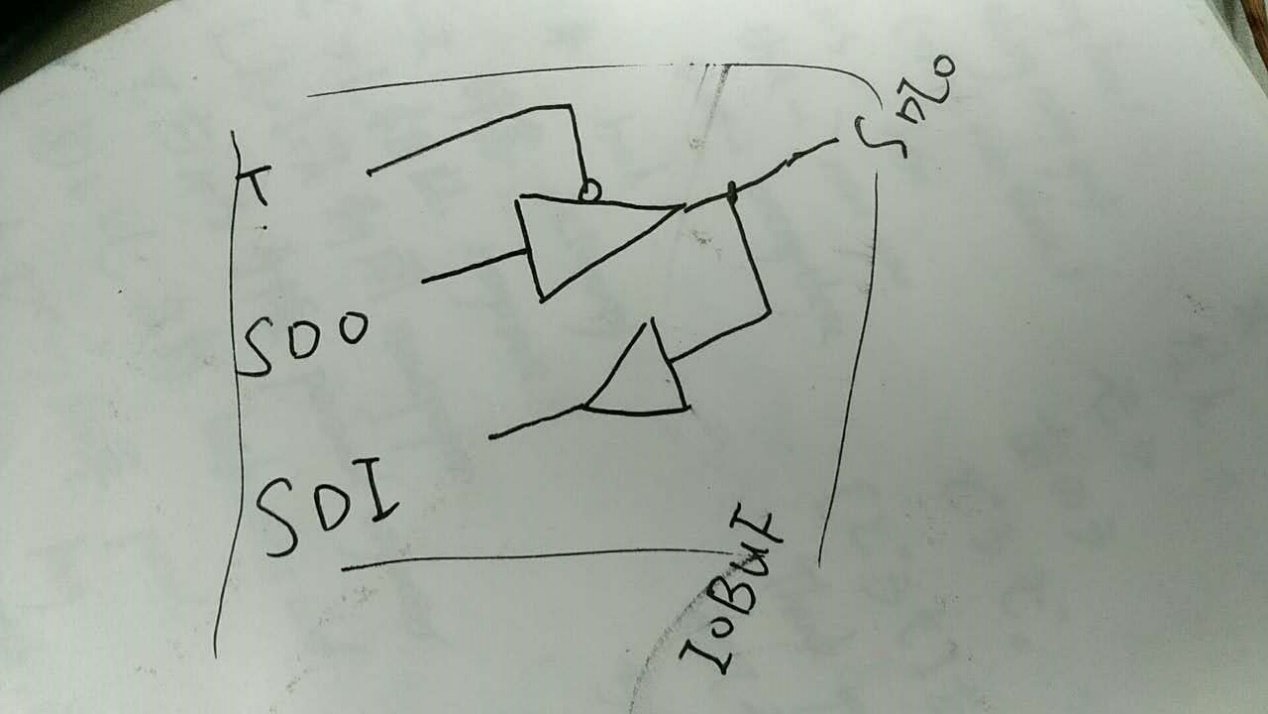


图 20 IOBUF结构

王老师提出可以用上图所示的IOBUF结构将四线SPI的SDO、SDI两个信号合并为SDIO一个信号。这确实是一种可行的方法，但是由于SPI的IP核是Xilinx提供的，我们无法直接修改，有很多不方便的地方。比如，AD99715的读时序要求先从SPI写入一个字节的控制信号，然后AD9715会立即将一个字节的数据通过SPI输出。这就要求SDIO的读写状态快速切换，必须要由硬件来完成。因为无法修改IP核，我们需要在顶层模块上对SDi、SDO等信号进行处理，控制信号T的产生也相当麻烦。此外，使用SPI IP核意味着我们需要用两个固件来控制AD9715，让人觉得比较奇怪。因此最后我们决定还是在原来的固件中重新实现一个SPI通信模块。

### 1.1.2 方案二 在原来的固件中实现SPI通信模块

为了实现SPI通信，我们需要对原来的固件进行升级。升级后的固件寄存器格式如下表所示。

其中，寄存器3（DAC\_SPI）是SPI控制寄存器。如果向DAC\_SPI中写入数据，固件会自动通过SPI接口读写相应的数据。其中，第15至第8 bit是SPI指令位，按照AD9715数据手册中的定义，决定了数据的读写方向和地址。不过我们的固件只支持一次读写一个字节。如果第15 bit是1，表示读，数据传输阶段固件会将SPI总线上传输来的数据存入DAC\_SPI寄存器的第7至第0 bit；如果第15 bit是0，表示写，数据传输阶段固件会将DAC\_SPI寄存器的第7至第0 bit 通过SPI总线发送出去。



图 21 固件升级后的寄存器格式

采用下图所示的状态机来控制固件和SPI接口时序。

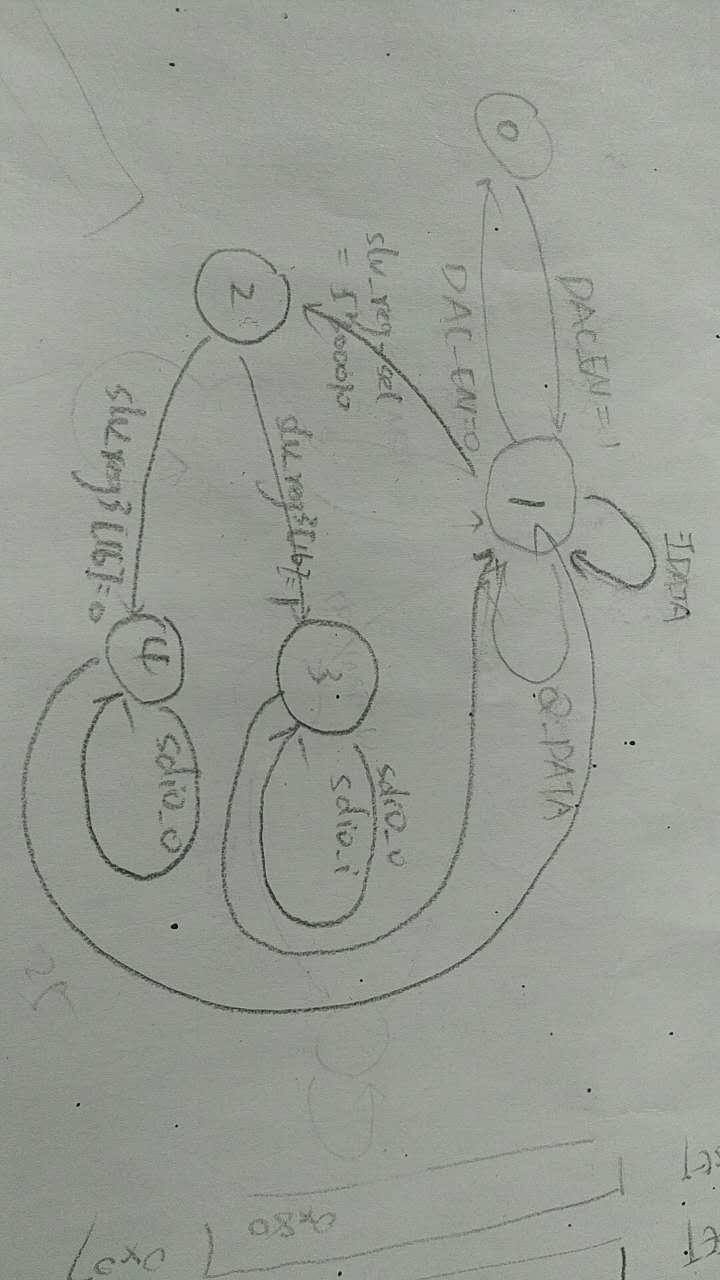


图 22 固件和SPI控制状态机

其中的状态定义如下：

0：空状态

1：待命状态

2：判断状态

3：读状态

4：写状态

不过将改进的固件添加到MicroBlaze系统上之后，却发现综合无法通过。这是因为SDIO是一个三态的端口，但在XPS中如果直接将它声明为inout型信号，综合工具会尝试将它拆分为SDIO\_I、SDIO\_O、SDIO\_T三个信号，导致无法和底层模块链接上。参考Xilinx论坛上关于XPS中三态端口的讨论[1] ，我们需要将输入、输出、控制三个信号拆分出来，再在MPD文件中进行如下声明：



图 23 关于三态端口的声明

这样就可以将三个信号合成一个三态端口，实现了三线SPI的数据交换。

## 1.2 改进后固件的仿真

先对用户逻辑进行单独仿真，得到波形如下：

Float (9.20131013) - [Default.wcfg*] 
File Edit View Simulation Window 
z 
Layout 
Help 
Name 
IP2DAC Dataco 
IP2DAC DCLKI( 
IP2DAC clkout 
IP2DAC PinMD 
IP2DAC PWRDI 
192CAC clkM 
IP2DAC Format 
IP2DAC opEnl 
IP2DAC opEnc 
IP28us Data[0:: 
IP28us RdAck 
IP28us_WrAck 
IP28us_Error 
aus21P clk 
aus21P Reset 
aus21P Data[o:: 
aus21P 
aus21P RdCE[0 
aus21P WrCE[0 
sdio reg 
write spi 
Value 
000 
00000003 
00000000 
10000 
00000 
1,200 ns 
400 ns 
1, 800 ns 
2, 000 ns 
2,200 ns 
Re—I aunch 
2, 400 ns 
AR 
2, 800 ns 
OOOOAbff 
17:11 
2017/8/24 
3, 000 
De Eau] E 

图 24 改进后的固件进行用户逻辑单独仿真得到的波形图

再将固件添加到MicroBlaze系统中，用ELF文件作为激励进行仿真，得到波形图如下：

Float (P.20131013) - [Default.wcfg*] 
File Edit View Simulation Window 
Name 
fpga O RS232 TX pin 
plb dac O S_PinMD_pin 
plb_dac O S_CIkout_pin 
plb dac O 
plb_dac O S_CIkMD_pin 
plb dac O S_Format pin 
plb dac O S_OpEnI_pin 
plb_dac O S_OpEnQ pin 
plb_dac I S_CIkMD_pin 
plb_dac I S_CIkout_pin 
plb dac 1 s_DCLK10 pin 
plb_dac I S_Data 
plb dac I S_Format pin 
plb dac I S_OpEnI_pin 
plb_dac I S_OpEnQ pin 
plb dac 1 s_PWRDN pin 
plb dac I S_PinMD_pin 
fp-ga O clk_l sys clk_pin_PERIOD 
fp-ga O rst I sys rst pin_LENGTH 
1' fpga O RS232 ax pin 
fpga O clk_l sys clk_pin 
fpga O rst I sys rst pin 
Layout Help 
Value 
0000000000 
20000 _ 000000 
320000 _ 000000 
ne Eau_1E 
20. 700 
Re—l aunch 
21, 000 
21. 100 
21, 200 
XI: 20.639. 7222 
21. 300 
2017/8/27 

图 25 改进后的固件在MicroBlaze系统中进行仿真得到的波形图（写时序）

Float (P.20131013) - [Default.wcfg] 
File Edit View Simulation Window 
Name 
fpga O RS232 TX pin 
plb dac O S_PinMD_pin 
plb_dac O S_CIkout_pin 
plb dac O s_DCLK10 pin 
plb_dac O S_Data 
plb dac O s_PWRDN pin 
plb_dac O S_CIkMD_pin 
plb dac O S_OpEnI_pin 
plb_dac O S_OpEnQ pin 
plb_dac I S_CIkMD_pin 
plb_dac I S_CIkout_pin 
plb dac 1 s_DCLK10 pin 
plb_dac I S_Data 
plb dac I S_Format pin 
plb dac I S_OpEnI_pin 
plb_dac I S_OpEnQ pin 
plb dac 1 s_PWRDN pin 
plb dac I S_PinMD_pin 
fp-ga O clk_l sys clk_pin_PERIOD 
fp-ga O rst I sys rst pin_LENGTH 
1' fpga O RS232 ax pin 
fpga O clk_l sys clk_pin 
fpga O rst I sys rst pin 
Layout Help 
Value 
0000000000 
0000000000 
20000 _ 000000 
320000 _ 000000 
neEau_1E w 
R e —1 aun c h 
21, 900 
XI: 22.529. 2222 
22. 100 
2017/8/27 

图 26 改进后的固件在MicroBlaze系统中进行仿真得到的波形图（读时序）

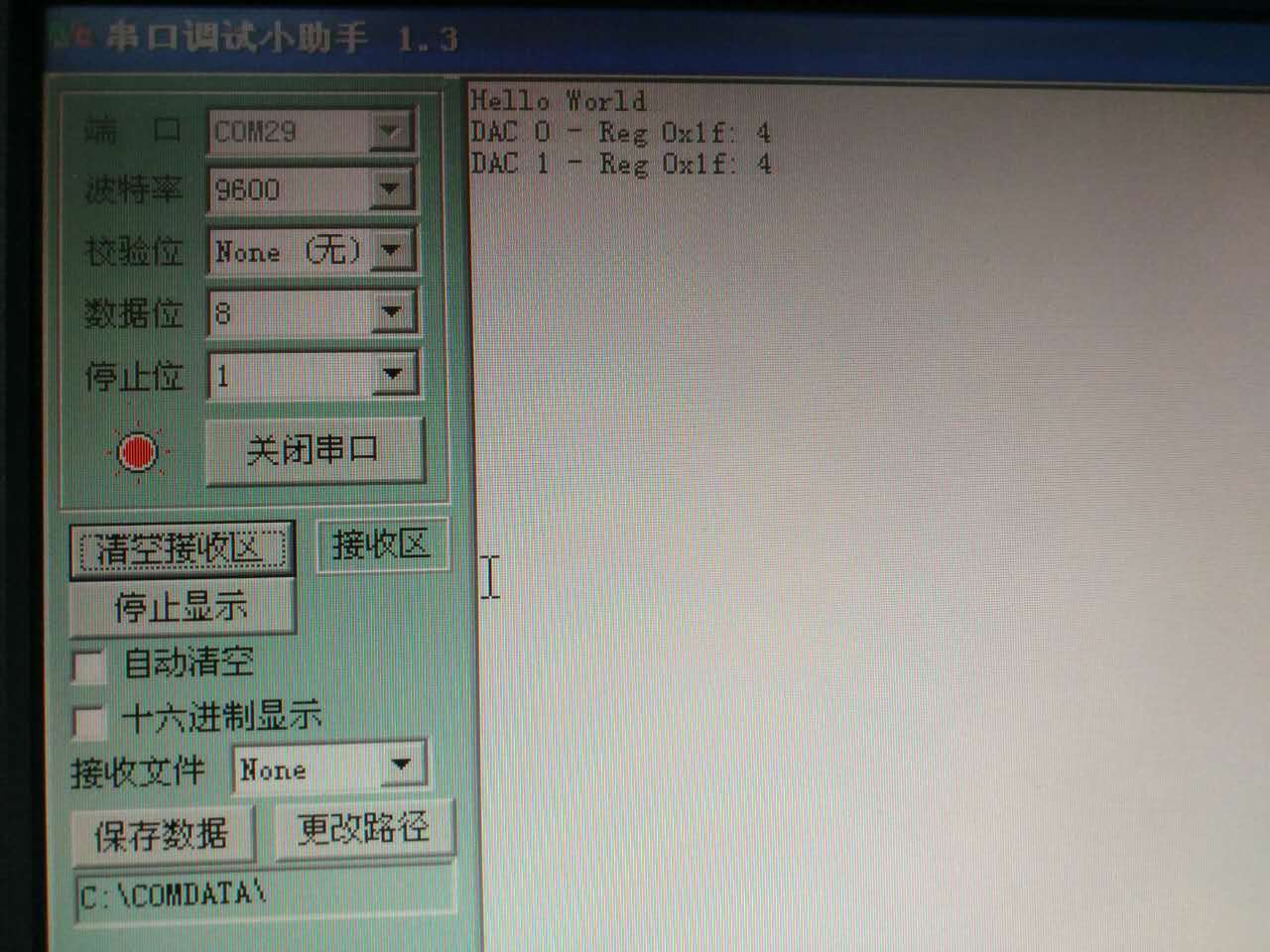
# 2 第三次上板调试

对固件进行改进并实现SPI接口之后，我们第三次进行上板调试，又发现了一些新的问题。

## 2.1 问题1：辅助DAC没有输出

## 2.2 问题2：输出信号的摆幅影响DAC性能

## 2.3 问题3：芯片的版本号不对



# 参考资料

1. Xilinx Forum, [Peripheral inout pin in xps](https://forums.xilinx.com/t5/Embedded-Development-Tools/peripheral-inout-pin-in-xps/td-p/31399)