

## **ADE7880与ADE7878的区别**

作者: Petre Minciunescu

### **简介**

本应用笔记说明[ADE7878](#)与[ADE7880](#)的区别, 讨论二者在硬件和软件上的不同, 并提供了一个用于ADE7880的头文件。

### **硬件区别**

[ADE7880](#)与[ADE7878](#)引脚兼容。

### **新抗混叠滤波器**

然而, 由于ADC带宽已从2 kHz(-3 dB点)提高到3.3 kHz(-3 dB点), 因此必须更换ADC输入路径中使用的抗混叠滤波器。以前, [ADE7878](#)评估板上使用1 k $\Omega$ /22 nF(7.2 kHz转折频率)抗混叠滤波器。现在, [ADE7880](#)评估板上使用1 k $\Omega$ /10 nF(15.9 kHz转折频率)抗混叠滤波器。

### **检测零线电流和相电流所用的传感器可以不同**

可以使用不同类型的传感器来分别检测零线电流和相电流。例如, 相电流可以利用罗氏线圈来检测, 零线电流可以利用电流互感器(CT)来检测, 反之亦然。

[ADE7880](#)的CONFIG寄存器的位0 (INTEN)用于使能/禁用相电流通道中的积分器。

[ADE7880](#)的CONFIG3寄存器的位3(ININTEN)用于使能/禁用零线电流通道中的积分器。

在[ADE7880](#)中, 位于地址0xE60F的增益寄存器的定义保持不变。

## 目录

硬件区别 .....	1	附录： ADE7880.H头文件 .....	5
检测零线电流和相电流所用的传感器可以不同 .....	1		
修订历史 .....	2		
软件区别 .....	3		

## 修订历史

2011年11月—修订版0：初始版

## 软件区别

### 寄存器地址

寄存器映射已更改。ADE7878的许多寄存器现在都有新地址。由于新增谐波计算，ADE7880增加了一些寄存器。寄存器信息请参见“附录：ADE7880.H头文件”部分。

### ADE7880不计算总无功功率和总无功电能

ADE7878计算总/基波无功功率/电能。ADE7880则仅计算基波无功功率/电能。

ADE7878将各相总无功功率的瞬时值存储在AVAR、BVAR和CVAR寄存器中。这些寄存器已不再存在。

当HSDC\_CFG寄存器的位[4:3](HXFER)设为10时，ADE7878 HSDC端口传输各相总无功功率。但是，当HXFER设为10时，ADE7880则传输基波无功功率。ADE7880的各相基波无功功率寄存器AFVAR、BFVAR和CFVAR未映射到寄存器空间中的某个地址，只能通过HSDC端口访问。

### xPGAIN寄存器取代xWGAIN、xVARGAIN、xVAGAIN寄存器(x = A, B, C)

在ADE7878中，有功、无功和视在功率数据路径中的增益寄存器为AWGAIN、BWGAIN、CWGAIN、AVARGAIN、BVARGAIN、CVARGAIN、AVAGAIN、BVAGAIN和CVAGAIN。建议使用相同的值来初始化各相位的寄存器。

在ADE7880中，APGAIN、BPGAIN和CPGAIN寄存器取代了各相的xWGAIN、xVARGAIN和xVAGAIN寄存器。APGAIN管理相位A的所有功率增益，BPGAIN管理相位B的所有功率增益，CPGAIN管理相位C的所有功率增益。

### WTHR、VARTHR和VATHR寄存器定义发生改变

在ADE7878中，48位寄存器WTHR、VARTHR和VATHR定义为：

$$WTHR = VARTHR = VATHR = \frac{P_{MAX} \times f_s \times 3600 \times 10^n}{U_{FS} \times I_{FS}}$$

其中：

$P_{MAX} = 33,516,139 = 0x1FF6A6B$ ，即ADC输入为满量程时计算出的瞬时功率。

$f_s = 8 \text{ kHz}$ ，即DSP用于计算瞬时功率的频率。

$n$ 为整数，决定xWATTHR/xVARHR/xVAHR寄存器的1 LSB相当于多少wh [ $10^n \text{ wh}$ ]。

$U_{FS}$ 和 $I_{FS}$ 分别为ADC输入为满量程时相电压和电流的有效值。

在ADE7880中，WTHR、VARTHR和VATHR现为8位无符号寄存器，定义为：

$$WTHR = VARTHR = VATHR = \frac{P_{MAX} \times f_s \times 3600 \times 10^n}{U_{FS} \times I_{FS} \times 2^{27}}$$

其中：

$P_{MAX} = 27,059,678 = 0x19CE5DE$ ，即ADC输入为满量程时计算出的瞬时功率。

$f_s = 1.024 \text{ MHz}$ ，即DSP以8 kHz速率计算的每个瞬时功率的累计频率。

$n$ 为整数，决定xWATTHR/xVARHR/xVAHR寄存器的1 LSB相当于多少wh [ $10^n \text{ wh}$ ]。

$U_{FS}$ 和 $I_{FS}$ 分别为ADC输入为满量程时相电压和电流的有效值。

针对总有功/无功和基波有功/无功功率的空载管理发生改变在ADE7880中，总有功/无功功率和基波有功/无功功率的空载条件已改变，更多信息参见ADE7880数据手册。

### HPF由CONFIG3寄存器中的位0 (HPFEN)管理(ADE7880)

在ADE7878中，电流和电压通道数据路径所用的高通滤波器(HPF)由24位寄存器HPFDIS管理。如果该寄存器为0(默认值)，则HPF使能。如果该寄存器用非零值初始化，则HPF禁用。

在ADE7880中，HPF由CONFIG3寄存器中的位0(HPFEN)管理。如果HPFEN = 0，则HPF禁用。如果HPFEN = 1(默认值)，则HPF使能。

### ADE7880计算相电流和的RMS值

ADE7878和ADE7880计算相电流和的瞬时值，并将其存储在ISUM寄存器中。ADE7878不计算ISUM的rms值。

如果CONFIG3寄存器的位2(INSEL)设为1(参见ADE7880数据手册)，则ADE7880计算ISUM的rms值并将其存储在NIRMS寄存器中。如果INSEL = 0(默认值)，则NIRMS寄存器包含INP和INN引脚处检测到的零线电流的rms值。

### ADE7880计算三相三线配置中第三电压的RMS值

在三相三线配置中(ACCMODE寄存器的CONSEL位设为01时)，仅检测A相和C相，B相用作参考。ADE7878和ADE7880均会计算A相与B相之间以及C相与B相之间的线路电压的rms值，并将其存储到AVRMS和CVRMS寄存器中。ADE7880还会计算A相与C相之间的线路电压的rms值，并将其存储到BVRMS寄存器中。

# AN-1127

## ADE7880可以计算更平滑的瞬时有功功率数据

在ADE7880中，CONFIG3寄存器的位1(LPFSEL)管理总有功功率计算的建立时间。如果LPFSEL清0(默认值)，则建立时间为650 ms。如果LPFSEL置1，则建立时间为1300 ms，可以使瞬时总有功功率数据更平滑。

## ADE7880引入了通信验证寄存器

ADE7880包括三个用于对I2C或SPI通信进行验证的寄存器：LAST\_OP、LAST\_ADD和LAST\_RWDATA，它们分别记录上一次成功通信的性质、地址和数据。

## ADE7880改进了CHECKSUM计算

在ADE7878中，CHECKSUM计算包括13个配置寄存器和6个内部寄存器。

在ADE7880中，CHECKSUM计算包括13个配置寄存器(全部位于DSP数据存储RAM的地址0x4380与地址0x43BE之间)和8个内部寄存器。此外，每次写入寄存器或意外更改数值时，STATUS1寄存器的位25(CRC)就会置1。如果MASK1寄存器的位25(CRC)置1，则 $\overline{IRQ1}$ 中断引脚会变为低电平。

## 电能累计模式和电能频率转换模式发生改变

在ADE7878中，电能频率转换器在引脚CF1、CF2和CF3/HSCLK上产生脉冲，这些引脚对应于ACCMODE寄存器的位[1:0](WATTACC)和位[3:2](VARACC)。瞬时功率始终以带符号形式在电能寄存器中累计，与WATTACC和VARACC位的状态无关。

在ADE7880中，ACCMODE寄存器的WATTACC和VARACC位所决定的模式已改变(参见表1和表2)。现在，瞬时功率是否累计到电能寄存器中要依据其状态。电能频率转换器在CF1、CF2/HREADY和CF3/HSCLK引脚上产生脉冲也是根据WATTACC和VARACC位的状态，但有一个例外：在仅正值模式(WATTACC = 01)下累计瞬时总/基波有功功率时，电能频率转换器仍会在带符号累计模式下产生脉冲。

## 当一个电路板同时支持ADE7880和ADE7878时，如何辨别？

ADE7880和ADE7878的CONFIG寄存器位于同一地址0xE618。ADE7878的默认值为0x0，ADE7880的默认值为0x2。

要辨别电路板上安装的是ADE7878还是ADE7880，请在上电后读取地址0xE618。如果值为0x0，则安装的是ADE7878；如果值为0x2，则安装的是ADE7880。

表1. 总有功功率累计模式和基波有功功率累计模式

WATTACC位 (ACCMODE寄存器的位[1:0])	总/基波有功电能寄存器累计模式	CF脉冲产生模式
00	带符号累计	同电能寄存器
01	仅正值累计	带符号累计
10(保留)	ADE7880的表现与WATTACC[1:0] = 00时一致	同电能寄存器
11	绝对值累计	同电能寄存器

表2. 基波无功功率累计模式

VARACC位 (ACCMODE寄存器的位[3:2])	基波无功电能寄存器累计模式	CF脉冲产生模式
00	带符号累计	同电能寄存器
01(保留)	ADE7880的表现与VARACC位[1:0] = 00时一致	同电能寄存器
10	基波无功功率的累计取决于基波有功功率的符号。如果有功率为正值，则以原样来累计无功功率，而如果有功率为负值，则以相反符号形式累计无功功率。	同电能寄存器
11	绝对值累计	同电能寄存器

## 附录： ADE7880.H头文件

```

#ifndef __ADE7880_H__
#define __ADE7880_H__

#define AIGAIN 0x4380
#define AVGAIN 0x4381
#define BIGAIN 0x4382
#define BVGAIN 0x4383
#define CIGAIN 0x4384
#define CVGAIN 0x4385
#define NIGAIN 0x4386
#define DICOEFF 0x4388
#define APGAIN 0x4389
#define AWATTOS 0x438A
#define BPGAIN 0x438B
#define BWATTOS 0x438C
#define CPGAIN 0x438D
#define CWATTOS 0x438E
#define AIRMSOS 0x438F
#define AVRMSOS 0x4390
#define BIRMSOS 0x4391
#define BVRMSOS 0x4392
#define CIRMSOS 0x4393
#define CVRMSOS 0x4394
#define NIRMSOS 0x4395
#define HPGAIN 0x4398
#define ISUMLVL 0x4399
#define VLEVEL 0x439F
#define AFWATTOS 0x43A2
#define BFWATTOS 0x43A3
#define CFWATTOS 0x43A4
#define AFVAROS 0x43A5
#define BFVAROS 0x43A6
#define CFVAROS 0x43A7
#define AFIRMSOS 0x43A8
#define BFIRMSOS 0x43A9
#define CFIRMSOS 0x43AA
#define AFVRMSOS 0x43AB
#define BFVRMSOS 0x43AC
#define CFVRMSOS 0x43AD
#define HXWATTOS 0x43AE
#define HYWATTOS 0x43AF
#define HZWATTOS 0x43B0
#define HXVAROS 0x43B1
#define HYVAROS 0x43B2
#define HZVAROS 0x43B3

#define HXIRMSOS 0x43B4
#define HYIRMSOS 0x43B5
#define HZIRMSOS 0x43B6
#define HXVRMSOS 0x43B7
#define HYVRMSOS 0x43B8
#define HZVRMSOS 0x43B9
#define AIRMS 0x43C0
#define AVRMS 0x43C1
#define BIRMS 0x43C2
#define BVRMS 0x43C3
#define CIRMS 0x43C4
#define CVRMS 0x43C5
#define NIRMS 0x43C6
#define ISUM 0x43C7
#define RUN 0xE228
#define AWATTHR 0xE400
#define BWATTHR 0xE401
#define CWATTHR 0xE402
#define AFWATTHR 0xE403
#define BFWATTHR 0xE404
#define CFWATTHR 0xE405
#define AFVARHR 0xE409
#define BFVARHR 0xE40A
#define CFVARHR 0xE40B
#define AVAHR 0xE40C
#define BVAHR 0xE40D
#define CVAHR 0xE40E
#define IPEAK 0xE500
#define VPEAK 0xE501
#define STATUS0 0xE502
#define STATUS1 0xE503
#define AIMAV 0xE504
#define BIMAV 0xE505
#define CIMAV 0xE506
#define OILVL 0xE507
#define OVLVL 0xE508
#define SAGLVL 0xE509
#define MASK0 0xE50A
#define MASK1 0xE50B
#define IAWV 0xE50C
#define IBWV 0xE50D
#define ICWV 0xE50E
#define INWV 0xE50F

```

# AN-1127

```
#define VAWV 0xE510 #define FVAR 0xE883
#define VBWV 0xE511 #define FVA 0xE884
#define VCWV 0xE512 #define FPF 0xE885
#define AWATT 0xE513 #define VTHDN 0xE886
#define BWATT 0xE514 #define ITHDN 0xE887
#define CWATT 0xE515 #define HXVRMS 0xE888
#define AFVAR 0xE516 #define HXIRMS 0xE889
#define BFVAR 0xE517 #define HXWATT 0xE88A
#define CFVAR 0xE518 #define HXVAR 0xE88B
#define AVA 0xE519 #define HXVA 0xE88C
#define BVA 0xE51A #define HXPF 0xE88D
#define CVA 0xE51B #define HXVHD 0xE88E
#define CHECKSUM 0xE51F #define HXIHD 0xE88F
#define VNOM 0xE520 #define HYVRMS 0xE890
#define LAST_RWDATA_24bit 0xE5FF #define HYIRMS 0xE891
#define PHSTATUS 0xE600 #define HYWATT 0xE892
#define ANGLE0 0xE601 #define HYVAR 0xE893
#define ANGLE1 0xE602 #define HYVA 0xE894
#define ANGLE2 0xE603 #define HYPF 0xE895
#define PHNOLOAD 0xE608 #define HYVHD 0xE896
#define LINECYC 0xE60C #define HYIHD 0xE897
#define ZXTOUT 0xE60D #define HZVRMS 0xE898
#define COMPMODE 0xE60E #define HZIRMS 0xE899
#define Gain 0xE60F #define HZWATT 0xE89A
#define CFMODE 0xE610 #define HZVAR 0xE89B
#define CF1DEN 0xE611 #define HZVA 0xE89C
#define CF2DEN 0xE612 #define HZPF 0xE89D
#define CF3DEN 0xE613 #define HZVHD 0xE89E
#define APHCAL 0xE614 #define HZIHD 0xE89F
#define BPHCAL 0xE615 #define HCONFIG 0xE900
#define CPHCAL 0xE616 #define APF 0xE902
#define PHSIGN 0xE617 #define BPF 0xE903
#define CONFIG 0xE618 #define CPF 0xE904
#define MMODE 0xE700 #define APERIOD 0xE905
#define ACCMODE 0xE701 #define BPERIOD 0xE906
#define LCYCMODE 0xE702 #define CPERIOD 0xE907
#define PEAKCYC 0xE703 #define APNOLOAD 0xE908
#define SAGCYC 0xE704 #define VARNLOAD 0xE909
#define CFCYC 0xE705 #define VANLOAD 0xE90A
#define HSDC_CFG 0xE706 #define LAST_ADD 0xE9FE
#define Version 0xE707 #define LAST_RWDATA_16bit 0xE9FF
#define LAST_RWDATA_8bit 0xE7FD #define CONFIG3 0xEA00
#define FVRMS 0xE880 #define LAST_OP 0xEA01
#define FIRMS 0xE881 #define WTHR 0xEA02
#define FWATT 0xE882 #define VARTHR 0xEA03
# #define VATHR 0xEA04
```

```
#define HX_reg          0xEA08
#define HY_reg          0xEA09
#define HZ_reg          0xEA0A
#define LPOILVL         0xEC00
#define CONFIG2         0xEC01

#endif /* __ADE7880_H__ */
```

**注释**

I<sup>2</sup>C指最初由Philips Semiconductors（现为NXP Semiconductors）开发的一种通信协议。