

引用格式:钟纬坤,李景虎,付方发.一种快速 MCU 修调测试方法[J].微电子学与计算机,2021,38(7):67-72. [ZHONG W K, LI J H, FU F F, A fast method of MCU trimming and testing[J]. Microelectronics & Computer, 2021, 38(7): 67-72.]

一种快速 MCU 修调测试方法

钟纬坤¹,李景虎¹,付方发²

(1 福建农林大学,福建 福州 360000;

2 哈尔滨工业大学,黑龙江 哈尔滨 150001)

摘 要:为降低具有高精度要求和功能丰富的 MCU 的修调测试成本,提出一种快速高效的 MCU 修调测试方法。此方法是将 MCU 芯片成品通过 Flash 自动修调补偿来控制电路的输出基准和频率,或者是调整电流,从而实现自动化修调 MCU 芯片的效果,同时控制 MCU 模拟从不同 pin 脚上输出到测试仪器,然后再由软件自动记录测试结果。修调测试使用的是一款工业级 8051 芯片。测试结果表明:经过这种修调测试方法可以使 MCU 满足输出精度要求且能快速得到 MCU 模拟信号输出,并有效地降低 MCU 的修调测试成本。

关键词:工业级 MCU;Flash 自动修调;程序自动测试;算法设计与优化

中图分类号: TP272

文献标识码: A

文章编号: 1000-7180(2021)07-0067-06

A fast method of MCU trimming and testing

ZHONG Weikun¹, LI Jinghu¹, FU Fangfa²

(1 Fujian Agriculture and Forestry University, Fuzhou 360000, China;

2 Harbin Institute of Technology, Harbin 150001, China)

Abstract: In order to reduce the cost of MCU trim test with high precision requirements and rich functions, a fast and efficient MCU trim test method is proposed. This method controls the output reference and frequency of the circuit through Flash automatic adjustment and compensation of the produced MCU chip, or adjusts the current to realize the effect of automatic adjustment of the MCU chip, and controls the MCU analog output to be output from different pins to the test. Furthermore, the instrument automatically records the test results by the software. The trim test uses an industrial grade 8051 chip. The test results show that through this trimming test method, the MCU can not only meet the output accuracy requirements but also quickly obtain the MCU analog signal output, which effectively reduces the MCU trim test cost.

Key words: industrial-grade MCU; Flash automatic trimming; automatic program testing; algorithm design and optimization

1 引言

目前芯片的修调技术主要有四种方式:金属或多晶硅熔丝修调、二极管反熔丝修调、电阻薄膜激光修调和内嵌存储单元修调。但是由于 MCU 芯片对精度要求较高,金属或多晶硅熔丝修调方式存在较难控制的问题,所以这种方式难以实现 MCU 的修

调。而二极管反熔丝修调可以在封装之后进行修调,虽然能够实现更高精度,但是该技术受二极管反熔丝影响占用面积比较大。激光修调技术需要专用的激光设备和测试仪器,存在成本昂贵的问题,所以 MCU 的修调也不考虑这种方式^[1]。所以对于 MCU 最常用的修调方式就是用 EEPROM 写特定地址的数据达到修调的目的。通常 MCU 修调就是使用 C2

收稿日期:2020-10-20;修回日期:2021-01-30

基金项目:国家重点研究计划项目(2018YFB2201000);福建省自然科学基金项目(2019J01662)

接口直接写相应的修调所用寄存器,然后测得相应的模拟信号,多次擦写修调找到满足测试精度的数值后统一写到修调所用寄存器中.这种修调测试方式往往只适用于那些模拟输出精度要求不高且需要修调的信号比较少的芯片.但是现在随着 MCU 功能的不断地丰富,并且对于模拟信号的输出精度要求越来越高,所需修调的地方也越来越多,如果还是采用传统的修调测试方法,那么 MCU 修调测试的工作量就会呈几何级的增大,花费的人力和时间成本是 MCU 修调测试人员亟需解决的问题^[2-3].

针对这个问题,本文提出了一种快速高效的 MCU 测试修调方法,考虑到 MCU 都有用于程序运行的 Flash,如果可以通过编程的方式,通过向 MCU 中烧写编程好的自动修调测试程序,并将 MCU 的模拟输出通过测试仪器读取并自动传送到计算机上,由计算机记录 MCU 模拟信号的结果,这样就由人工手动修调测试转变成 MCU 的自动修调测试,时间成本也能转化为软件运行的时间成本,这其中只需要人工将程序烧写进 MCU 芯片中,并将 MCU 芯片、测试仪器和计算机正确连接,之后靠程序自动完成 MCU 芯片的修调测试.这种方法让修调测试所需花费的主要时间成本变成了程序运行所需花费时间,通过开发指令级并行,自动修调值获取算法的优化既能有效的降低程序运行时间,又进一步减少了芯片修调测试所需花费对的时间成本.根据这个思路编写测试修调程序,应用在一款 8 位的 MCU 芯片上,最后修调测试结果显示效果良好.

2 MCU 修调测试

2.1 MCU 芯片

图 1 是采用 QFN 封装的 8 位 MCU 结构示意图,这款 MCU 芯片可用于数据通信光收发器和工业控制以及智能传感器.它以小封装提供高精度 14 位 ADC,温度传感器($\pm 3^{\circ}\text{C}$),四个 12 位 DAC. MCU 内核的工作频率为 72 MHz,可提供各种接口以及专为嵌入式应用设计的可编程逻辑.

2.2 MCU 修调测试工作

MCU 具有丰富的外设模块,这些外设模块有些需要精准的模拟信号输入作为基准,或是本身能够输出精准模拟信号.由于工艺上存在非理性因素的影响,MCU 设计了修调寄存器用于控制电路的模拟信号输出.

这款 MCU 的修调测试包括了精密基准电压 V_{REF} ,可用于为 ADC,DAC 提供电压基准,或由连接至

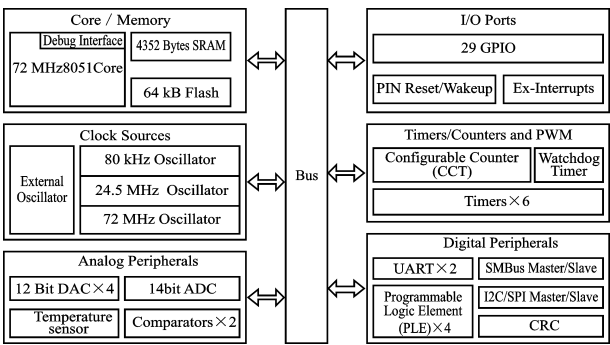


图 1 MCU 结构示意图

V_{REF} 引脚的其他电路使用.有基准电压 1.2 V 和 2.4 V 两种输出,要求修调后 1.2 V 输出精度在 $\pm 5\text{ mV}$,2.4 V 输出精度在 $\pm 10\text{ mV}$.线性稳压器 LDO,芯片内部的大部分电路都由它供电.输出为 1.8 V,要求修调后的精度在 $\pm 20\text{ mV}$.包括三个振荡器,分别为 HFOSC0、HFOSC1 和 LFOSC0,HFOSC0 和 HFOSC1 都是可编程的内部高频振荡器,要求分别校准为 24.5 MHz 和 72 MHz.当请求时,振荡器自动启用.要求修调后精度在 $\pm 2\%$.LFOSC0 是一种可编程的低频振荡器,要求校准为 80 kHz.要求修调后精度在 $\pm 2\%$.还包括一个温度传感器 Temperature Sensor,要求修调后精度在 $\pm 3^{\circ}\text{C}$.一个逐次逼近型 A/D 转换器,具有 10 位、12 位和 14 位 3 种模式.要求修调后精度可以达到最高 14 位.四个 D/A 转换器,要求修调后精度可以达到 12 位.

2.3 MCU 一般修调测试流程

由于芯片在生产过程中可能受到工艺或者环境等非理性因素的影响,往往刚生产出来的芯片都无法满足设计师的精度要求,所以在设计时通常会在电路设计中设计修调电路.本次修调测试使用的 8051 芯片在设计时预留了隐藏寄存器,通过这些隐藏寄存器可以直接输出到电路的模拟信号模块上,控制模拟电路中的修调电路,调整模拟信号的输出.如图 2 所示为本次测试的 8051 芯片中隐藏寄存器对电路的修调原理.

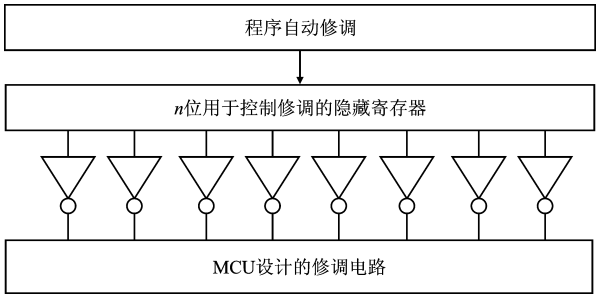


图 2 修调原理

一般芯片的修调测试就是测试人员对隐藏寄存

器的值进行修改,然后将芯片的对应模拟信号输出与期待值进行对比得到修调值.如图 3 所示在所有修调测试完成后将这些修调值统一写到 MCU 的 Flash 中,在芯片到达用户手中使用时,每次 MCU 上电时 CPU 自动从 Flash 中读取修调参数,并将修调参数写入到隐藏寄存器,这样用户使用的 MCU 就是已经修调完成的芯片.

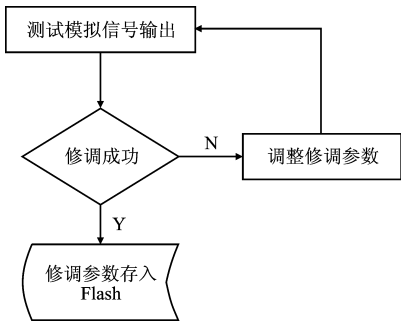


图 3 MCU 一般修调测试流程

3 快速高效的 MCU 修调测试方法

本文提出的是一种快速高效的 MCU 修调测试方法,这种方法的流程图如图 4 所示.它从三个方面提高修调测试的效率,大大降低了修调测试的时间成本.

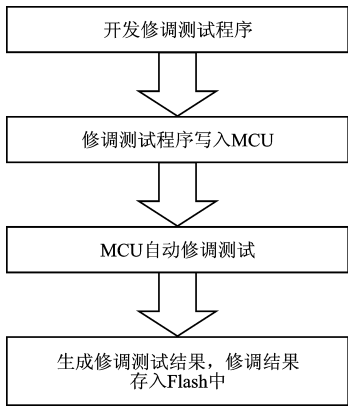


图 4 MCU 快速修调测试流程

3.1 程序实现自动修调测试

MCU 功能丰富,设计有多个用于微调模拟信号输出的寄存器.这些模拟信号如果可以通过一个程序找到所有修调寄存器的期待值那么就可以很大程度上提高修调效率.之后将这些待测试的模拟信号在没有相互影响的前提下通过 C 语言编译好的 hex 文件烧写进 MCU 中,同时在芯片的不同 pin 脚上输出待测模拟信号,这样就完成了一颗芯片的所有模拟测试^[4].

以一款 MCU 芯片为例,该芯片有 6 个需要修调的模块,每个模块需要对一个或多个寄存器进行修调.为了快速修调,在 MCU 修调时常采用内部输出的模拟信号作为参考源.而参考源通常选取精度

最高的模块输出的模拟信号作为参考源. MCU 的 VDD 电压 3.3 V, ADC 最高为 14 位,所以精度为 0.2 mV,是 6 个模块中精度最高的模块.可以首先对 ADC 模块进行修调,然后其他模块以 ADC 的模拟输出作为参考源进行修调,可以减少额外设备.

自动修调模拟一般手动修调过程,用编程的方式让 MCU 自己完成对模拟信号的修调.首先是对 ADC 的修调,需要两台精度高于 0.2 mV 的电压源,一台 3.300 0 V 作为 MCU 的供电,另外一台输出 1.650 0 V 作为 ADC 的模拟输入,软件自动调整修调寄存器值得到 14 位 ADC 结果 2 000 h,将此时寄存器值保存于 Flash,完成 ADC 模块修调^[5].其余模块中,除了振荡器其余输出的都是模拟信号,可以配置这些模块的模拟电压输出作为 ADC 的输入,之后通过 ADC 结果与期待值直接的关系来修调相应寄存器,使这些模块满足精度要求并保存这些寄存器值于 Flash 中完成对应模块的修调^[6-8].振荡器的修调可以使用 MCU 芯片中的定时器,配置测试板 TIMER₀ 为 16 位 72 MHz 系统时钟定时器,定时器初值 0000h,记到 FFFFh 后产生中断.配置测试板 TIMER₁ 为 16 位 T1(P0.3)下降沿计数器,与 TIMER₀ 同时启动, TIMER₀ 中断产生后停止 TIMER₁ 计数.配置待测芯片 P0.1 为 HFOSC0 2 分频系统时钟输出,此处设成 2 分频是因为 T₁ 输入需要 2 个周期同步,要求 T₁ 频率不能超过系统时钟频率的四分之一也就是 18 MHz,所以设成 12.25 MHz(81.633 ns).理论上 TIMER₀ 计数时间为 13.888 × 65 536=910 222.216 ns,对应 TIMER₁ 计数次数为 910 222.216/81.633=2B8E,扣除 TIMER₀ 中断产生到 TIMER₁ 关闭的时间,仿真得到 TIMER₁ 的期待值为 2B90,通过修改隐藏寄存器得到期待值,将寄存器值存入 Flash 完成振荡器的修调^[9].

自动测试则是将测试模块输出分别输出到不同的 pin 脚上,这些 pin 脚连接测试仪器进行数据的记录,对于模块测试时会相互影响或不建议同时测试的模块可以在程序中加入对应的判断,两个相互干扰的测试可以通过分时测试的方式完成,这样就能通过一个程序完成所有测试.测试结果将通过计算机软件保存起来.

3.2 开发指令级并行

MCU 中的 CPU 通过使用流水线作业的方式, CPU 运行的指令可以并行执行,开发指令级并行可以提高 CPU 的性能,这样可以减少程序自动修调所花费的时间.指令级并行的开发大体分为两种,一种是依靠硬件的帮助动态发现和开发并行,另一种

是通过软件技术在编译时静态发现并行^[10].

修调测试所用芯片 CPU 部分指令所需时钟周期如图 5 所示,其中 FLRT 是不同时钟源频率下指令所需时钟周期. 以修调 VREF 和 LDO 为例, VREF 修调中的有(MOV direct, #data)类指令,而下一指令与这条指令所用数据存在相互依赖的关系,则需要等待 3 个时钟周期后执行下一指令. 但是此时如果把 LDO 修调程序中的(MOV @Ri, A)类的指令插入进来,可以在时钟周期为 4 时同时完成 VREF 和 LDO 的两条指令,实现指令级并行.

Mnemonic	Description	Bytes	Clock Cycles		
			FLRT = 0	FLRT = 1	FLRT = 2
MOV direct, #data	Move immediate to direct byte	3	3	3	3
MOV @Ri, A	Move A to indirect RAM	1	2	2	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3	7	9
MOVC A, @A+PC	Move code byte relative PC to A	1	3	7	9
MOVX A, @Ri	Move external data (8-bit address) to A	1	3	3	3

图 5 MCU 部分指令周期

HFOSCO			LDO			Vref			DAC0					
HFOCAL	P0.1/MHz	Δ (后-前)	Page1 B3	P2.5/V	Δ (后-前)	Page1 89	P0.0/V	Δ (后-前)	FFA2	P2.5/V	Δ (后-前)	FFA4	P3.0/V	Δ (后-前)
0	15.51		80	1.2770		0	2.9115		0	1.6173		0	1.6298	
8	14.96	-0.55	81	1.3469	0.0699	8	2.8674	-0.0441	4	1.6216	0.0043	4	1.6331	0.0033
10	14.57	-0.39	82	1.4145	0.0676	10	2.8559	-0.0115	8	1.626	0.0044	8	1.6366	0.0035
18	14.10	-0.47	83	1.4827	0.0682	18	2.8124	-0.0435	B	1.6295	0.0035	B	1.6389	0.0023
20	13.86	-0.24	84	1.5497	0.067	20	2.8026	-0.0098	10	1.6256	0.0061	10	1.6431	0.0042
28	13.41	-0.45	85	1.6181	0.0684	28	2.7590	-0.0436	14	1.6398	0.0042	14	1.6464	0.0033
30	13.07	-0.34	86	1.6854	0.0673	30	2.7473	-0.0117	18	1.6442	0.0044	18	1.6498	0.0034
38	12.67	-0.40	87	1.7537	0.0683	38	2.7042	-0.0431	1B	1.6477	0.0035	1B	1.6522	0.0024
40	12.53	-0.14	88	1.8189	0.0652	40	2.6943	-0.0099	20	1.6547	0.007	20	1.6572	0.005
48	12.16	-0.37	89	1.8870	0.0681	48	2.6510	-0.0433	24	1.6594	0.0047	24	1.6605	0.0033
50	11.93	-0.23	8A	1.9543	0.0673	50	2.6393	-0.0117	28	1.6639	0.0045	28	1.6639	0.0034
58	11.58	-0.35	8B	2.0226	0.0683	58	2.5960	-0.0433	2B	1.6672	0.0033	2B	1.6663	0.0024
60	11.41	-0.17	8C	2.0902	0.0676	60	2.5866	-0.0094	30	1.6726	0.0054	30	1.6074	0.0042
68	11.12	-0.29	8D	2.1590	0.0688	68	2.5430	-0.0436	34	1.6773	0.0047	34	1.6739	0.0035

图 6 MCU 部分模拟信号修调

根据该图可以看出,修调寄存器值的变化与模拟信号的变化是正比或反比关系的. 但是步长变化并非都是不变的. 这样就可以把修调寄存器值对应的模拟输出看成是有序表,通过查找算法找到第一个满足精度阈值的结果. 前文所设计的修调算法停止条件是两次比较结果变号,在两次结果中选择更接近期望值的参数作为修调结果,这样保证了能够找到的修调结果是最优结果. 但是芯片的修调往往不需要最优结果,只要可以满足精度要求就是成功的修调,所以我们只需要将测试结果与精度阈值作比较,当测试结果满足精度阈值条件后就将修调值作为修调结果存入 Flash 中,这样每个模块的修调至少可以减少一次结果与期望值的比较次数. 修调算法流程图优化前后的对比如图 7 所示. 对于这款 MCU 芯片,优化后模拟测试结果与期待值对比判

3.3 算法设计与优化

这款 MCU 的修调在算法的实现上可以有很多种,不同的需求选择不一样的算法实现. 如果着重于芯片的性能则需要修调时找到最优的修调值,着重于修调效率可以在第一次找到满足要求的修调值就结束修调. 本文的目的在于实现一种快速 MCU 修调,所以更加注重修调效率,在算法的选择和实现上采用尽可能快速的算法.

MCU 的一些模拟信号输出是比较容易修调的,这些模拟信号在修调寄存器值增大时模拟输出也增大,或者修调寄存器值增大模拟输出减小,且修调寄存器改变量和模拟输出变化量大致是成正比. 但是往往 MCU 模拟信号的修调并没有那么简单,需要先对不同修调寄存器值和对应的模拟信号输出进行采样,根据修调寄存器和模拟信号的变化关系和变化步长确定合适的算法. 图 6 是 MCU 改变修调寄存器值对模拟信号的改变,并记录步长的变化.

断选择采用折半查找算法,这种方式比顺序增大或减小修调参数更具优势.

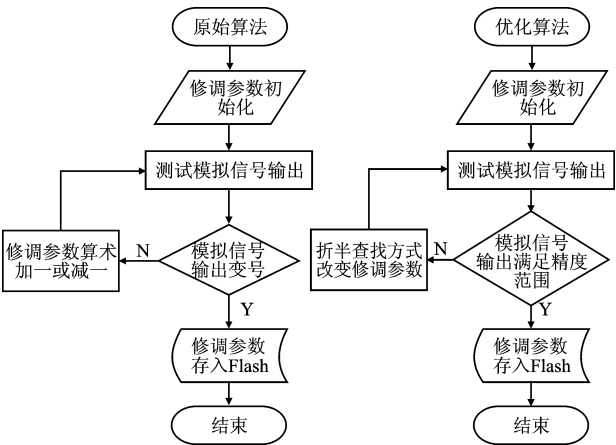


图 7 修调算法算法对比

4 修调测试对比

4.1 自动修调结果与手动修调结果对比

前期已对一批此型号的 MCU 芯片进行手动修调测试,自动修调测试程序对芯片重新进行修调测

试,程序成功找到所有模块修调值并存入 Flash,并测试输出测试结果.表 1 是部分模块自动修调测试和手动修调测试结果,从表中可知自动修调可以达到修调的目的,并且只需要修改自动修调的算法,自动修调可以和手动修调一样找到芯片的最优修调结果.

表 1 芯片修调对比

修调模块	修调寄存器地址	手动修调值	自动修调值	期待值	手动修调结果	自动修调结果
VREF	IRAM40h	0xA5	0xA0	2.390~2.410V	2.4007 V	2.3902 V
LDO	IRAM41h	0x88	0x86	1.780~1.820V	1.8015 V	1.7825 V
HFOSC0	IRAM42h-IRAM43h	0x49,0x47	0x57,0x49	12.1275~12.375 MHz	12.268 MHz	12.325 MHz
HFOSC1	IRAM44h-IRAM45h	0x13,0x46	0x13,0x44	8.9775~9.0225 MHz	8.9945 MHz	8.9825 MHz
LFOSC0	IRAM46h	0xE3	0xE3	78.4~81.6 kHz	79.66 kHz	79.66 kHz

MCU 芯片 12 位精度 DAC 通过手动修调和自动修调得到两组寄存器参数,分别是:手动修调参数 IRAM50h = 1E, IRAM51h = 3F; 自动修调参数 IRAM50h=1E,IRAM51h=18.芯片自动测试结果如图 8 所示,可以看出芯片自动修调和手动修调的 DAC 输出基本重合,手动修调是调整到了 DAC 最优的参数,自动修调的参数也能满足 DAC 修调要求.

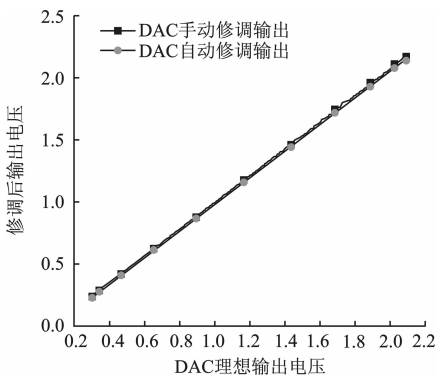


图 8 DAC 修调结果对比

图 9 表示的是温度传感器传递函数,其中 Offset 表示温度传感器在 0℃下的失调量, V_{TEMP} 是 MCU 芯片的温度传感器在用于 ADC 多路复用器输入时 ADC 的正输入.通过手动修调和自动修调得到相同的寄存器参数 IRAM50h=8F,温度传感器软件自动测试结果如表 2 所示,满足±3℃的精度要求.

4.2 自动修调测试与手动修调测试时间成本对比

在未开发自动修调测试程序前对一批芯片共计 50 颗做修调测试,测试修调工作持续了三个月的时间,平均每颗 MCU 芯片完成修调测试需要花费 15 小时的时间成本.而开发自动修调测试程序后,芯片的修调只需要将 MCU 芯片放到开发板上,对芯片的不同 pin 脚进行接线,通过程序烧写自动完成对 MCU 的修调,最后通过自动测试程序对 MCU 进行测试并输出测试结果文档.通常一颗芯片只需要 1

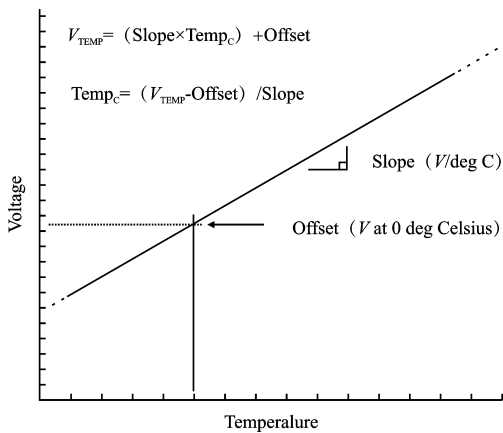


图 9 Temperature Sensor 传递函数

表 2 Temperature Sensor 测试结果

PCB_FIB2		temperature	
设置温度	传感器温度	P2.5/V	ADC值
25	25.9	1.16519	594-5B4
20	21.5	1.5061	57C-597
10	12.1	1.1113	549-56B
0	2.8	1.0710	520-535
-10	-6.7	1.0309	4F6-510
-20	-16.2	0.98982	4B1-4CF
-30	-25.8	0.94959	48B-49B
-40	-35.4	0.90829	45A-46A
-30	-26.7	0.94712	48A-49C
-20	-17.5	0.98682	4BA-4CD
-10	-8.2	1.0225	4EF-507
0	1.2	1.0603	527-536
10	10.7	1.0995	544-561
20	21.0	1.14153	57A-5A0
30	30.1	1.17978	5B1-5CA
40	39.4	1.2182	5D2-5EC
50	48.8	1.2579	614-625
60	58.3	1.2961	633-66A
70	67.6	1.3352	66D-686
80	78.1	1.3734	690-6D6
90	85.9	1.4110	6D1-6F2
100	95.2	1.4496	71O-726
105	100.1	1.4704	72F-730
100	96.1	1.4533	6F3-731
90	87.8	1.4166	6C6-6E9
80	78.8	1.3788	698-6BE
70	69.7	1.3409	66A-694
60	60.5	1.3024	642-671
50	51.6	1.2645	607-640
40	42.4	1.2270	5D7-5F9
30	33.2	1.18910	5A1-5E1

小时的时间成本就可以完成这一过程,大大提高了 MCU 的修调测试效率。

5 结束语

本文提供了一种 MCU 测试修调方法,并为某款 8051MCU 芯片开发了实际可行的修调测试程序,成功应用在 MCU 芯片的测试修调上,大大降低了 MCU 芯片的修调测试成本。在程序的实现上可以根据不同性能要求修改算法,通过不同算法可以实现更优参数的获取或更快的算法运行结果。根据测试结果显示,这种自动修调测试的方式是可行的,可以实现 MCU 芯片的快速修调测试。

参考文献:

- [1] 王欢. 一款高性能开关电源芯片的修调方案分析与设计[D]. 成都: 电子科技大学, 2014.
WANG H. Design and analysis of trimming solutions for a high-performance switching mode power supply chip[D]. Chengdu: University of Electronic Science and Technology of China, 2014.
- [2] 邹敏瀚, 柯学, 金玉丰, 等. 一种片后数字修调技术及其应用[J]. 微电子学, 2006, 36(5): 666-669. DOI: 10.3969/j.issn.1004-3365.2006.05.033.
ZOU M H, KE X, JIN Y F, et al. A digital trim technology and its application[J]. Microelectronics, 2006, 36(5): 666-669. DOI: 10.3969/j.issn.1004-3365.2006.05.033.
- [3] 沈红莉, 徐静平, 钟德刚, 等. 一种用于温补晶振的 EEPROM 修调电路设计[J]. 微电子学与计算机, 2011, 28(6): 100-104. DOI: 10.19304/j.cnki.issn1000-7180.2011.06.024.
SHEN H L, XU J P, ZHONG D G, et al. Design of the EEPROM trimming circuit for the temperature compensated crystal oscillator[J]. Microelectronics & Computer, 2011, 28(6): 100-104. DOI: 10.19304/j.cnki.issn1000-7180.2011.06.024.
- [4] 葛南, 陈东坡. ATE 测试中的 Bandgap Trim 技术研究[J]. 微电子学与计算机, 2015, 32(4): 70-74. DOI: 10.19304/j.cnki.issn1000-7180.2015.04.015.
GE N, CHEN D P. The research of Bandgap Trim for ATE test[J]. Microelectronics & Computer, 2015, 32(4): 70-74. DOI: 10.19304/j.cnki.issn1000-7180.2015.04.015.
- [5] 李涛. 高精度多通道逐次逼近型模数转换器测试与修调[D]. 成都: 电子科技大学, 2014.
LI T. The testing and trimming of high-precision multi-channel successive approximation ADC [D]. Chengdu: University of Electronic Science and Technology of China, 2014.
- [6] 张敏森, 余庆, 杨勇. 电源类芯片测试修调方法及问题分析[J]. 电子质量, 2018(3): 11-14.
ZHANG M S, YU Q, YANG Y. Power class chip trimming problem of parametric testing[J]. Electronics Quality, 2018(3): 11-14.
- [7] 谢伟. Flyback 型数字电源模拟关键电路的研究与设计[D]. 成都: 西南交通大学, 2018.
XIE W. Research and design for Flyback digital power critical analog circuits[D]. Chengdu: Southwest Jiaotong University, 2018.
- [8] 赵来钊, 刘明磊. MCU 内嵌高精度 OSC 的补正方法[J]. 中国集成电路, 2015, 24(8): 30-33. DOI: 10.3969/j.issn.1681-5289.2015.08.004.
ZHAO L Y, LIU M L. Embodied high accuracy OSC trimming method of MCU[J]. China Integrated Circuit, 2015, 24(8): 30-33. DOI: 10.3969/j.issn.1681-5289.2015.08.004.
- [9] 闫书芳. 一种采用 UART 接口对智能电表 MCU 自动频率修调方法[J]. 集成电路应用, 2020, 37(2): 48-49.
YAN S F. Method of automatic frequency adjustment of MCU by UART interface for intelligent meter[J]. Application of IC, 2020, 37(2): 48-49.
- [10] HENNESSY J L, PATTERSON D A. Computer architecture: a quantitative approach[M]. Beijing: China Machine Press, 2007.

作者简介:

钟纬坤 男, (1996-), 硕士研究生. 研究方向为网络与信息安全. E-mail: 785403402@qq.com.

李景虎 男, (1975-), 教授, 硕士生导师. 研究方向为模拟和混合信号集成电路设计.

付方发 男, (1984-), 博士, 硕士生导师. 研究方向为数字集成电路设计.