

专用电源嵌入式控制系统DSP端口问题研究

王伟强,赵楠,宋英利

(核工业理化工程研究院,天津 300180)

摘要:针对目前被广泛应用在电力电子和电源控制领域的TMS320F28335 DSP,结合在开发专用电源嵌入式控制系统过程中发现的GPIO端口问题开展研究工作。对产生的问题进行了复现试验,发现在使用同一分组GPIO的情况下,通过GPxDAT寄存器控制程序指令无法控制2个及以上端口产生相应动作的问题。经分析提出导致问题产生的3种假设并进行相应的验证试验,试验结果表明第3种假设情况为产生问题的原因。在此基础上提出了3种基于软件编程的解决方案,并对解决方案进行了验证。最后,对TMS320F28x系列其他5种常用型号的DSP进行拓展试验,均发现了相似的问题。采用提出的3种解决方案进行测试,结果证明提出的解决方案都是有效可行的。

关键词:数字信号处理器;嵌入式控制系统;通用输入输出端口;串行外设接口

中图分类号:TP273 **文献标识码:**A **DOI:**10.19457/j.1001-2095.dqed21738

Research on DSP Port Problem of Special Power Embedded Control System

WANG Weiqiang, ZHAO Nan, SONG Yingli

(Research Institute of Physical and Chemical Engineering of Nuclear Industry, Tianjin 300180, China)

Abstract: Aiming at the TMS320F28335 DSP, which is currently widely used in the field of power electronics and power supply control, the problem of general-purpose input/output (GPIO) port found in the development of embedded control system for special power supply was studied. Through the repetition tests, it was found that the GPxDAT register control program instruction can not control two or more ports to emerge corresponding actions when using the same GPIO packet. Three hypotheses were conjectured after analysis, and the third hypothesis was verified as the cause of the problem through verification tests. Three solutions based on software programming were proposed, and the solutions were verified. Similar problem was found in other five types of TMS320F28x series for DSP, and three solutions proposed were tested. The results show that the solutions proposed are effective and feasible.

Key words: digital signal processor (DSP); embedded control system; general-purpose input/output (GPIO); serial peripheral interface (SPI)

随着嵌入式技术和数字信号处理器(digital signal processor, DSP)技术的不断发展,DSP在工业领域的应用越来越广泛。特别是在电源控制技术领域,TMS320F28x系列DSP占据大部分市场份额,其外设接口齐全适合进行电源嵌入式控制系统开发。串行外设接口(serial peripheral interface, SPI)是一种高速、全双工、同步通信总线,只占用4根线可节约芯片引脚。SPI通信以主从方式进行,必须有一个主设备,可有多个从设备,通过片选信号来控制从设备。

在开发专用电源嵌入式控制系统中选用了

美国德州仪器(TI)公司的TMS320F28335 DSP作为控制核心;采用触摸显示屏作为人机交互界面,DSP与触摸显示屏之间采用SPI通信。

TMS320F28x系列DSP提供了单从设备模式下SPI通信所需的信号。但对多从设备模式,需要通过DSP的通用输入输出端口(general-purpose input/output, GPIO)进行片选控制,对从设备进行选通、读写控制。

TMS320F28335 DSP共有88个GPIO,对应芯片的88个引脚。由于芯片尺寸的限制,这88个引脚都是功能复用的,可根据用户需要进行灵活编程配

置。这88个GPIO分为A,B,C 3组,分别对应GPIO0~GPIO31,GPIO32~GPIO63,GPIO64~GPIO87。

本文主要介绍在开发专用电源嵌入式控制系统过程中发现的TMS320F28335 DSP端口控制存在的问题以及产生问题的条件,即在使用同一分组下的GPIO时,常用的修改GPxDAT寄存器来控制端口读写操作的程序指令无法控制两个及以上端口产生相应动作的问题。针对这个端口控制问题,本文开展了相应研究并提出3种可能引起问题原因的假设并设计了一系列的验证试验,进行了逐一的试验验证,寻求引起端口控制问题产生的原因。随后经过综合分析提出了3种基于软件编程的解决方案并进行试验验证。最后,对常用的TMS320F28x系列另外5种型号DSP进行了拓展验证试验,验证是否存在相似的端口控制问题,以及本文提出的3种解决方案是否可解决这5种型号DSP的端口控制问题。

1 端口问题的发现

1.1 专用电源嵌入式控制系统硬件设计简介

专用电源嵌入式控制系统主要用于专用电源逆变器的控制和核心电量参数的采集传输,采用TMS320F28335 DSP作为控制核心开展硬件设计,其系统数据显示和人机交互接口采用2.8寸配置SPI接口的触摸显示屏。将DSP设置为SPI主设备,将触摸屏上的显示屏驱动芯片ILI9341和触摸屏控制芯片XPT2046设置为SPI从设备,实现数据显示和触摸控制。DSP与触摸显示屏接口硬件原理图如图1所示。

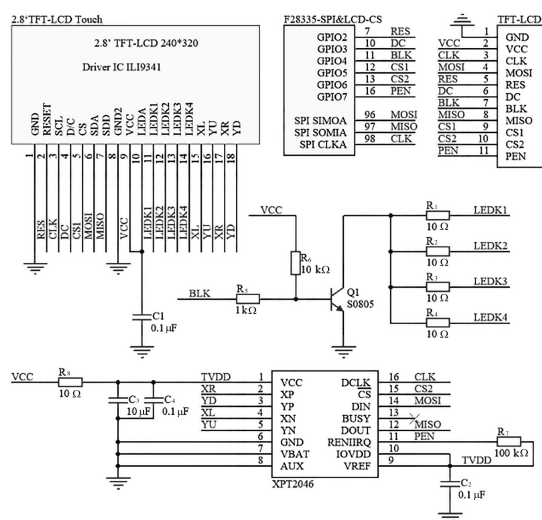


图1 DSP与触摸显示屏接口的硬件原理图
Fig.1 Hardware schematic of interface between DSP and touch screen

图1中使用了DSP的6个GPIO接口作为SPI从设备的控制信号,具体详见表1。

表1 DSP与触摸显示屏控制信号接口

DSP 端口	控制引脚	控制信号说明
GPIO2	RES	显示屏驱动复位信号
GPIO3	DC	显示屏驱动数据控制信号
GPIO4	BLK	显示屏背光控制信号
GPIO5	CS1	显示屏驱动选通信号
GPIO6	CS2	触摸屏控制选通信号
GPIO7	PEN	触摸屏笔中断控制信号

1.2 调试中发现的问题

按照设计要求编写触摸显示屏部分的程序代码,走查编译完成后,下载到DSP中进行调试测试。在调试过程中发现触摸显示屏没有任何反应。

使用示波器观察DSP的SPI端口波形,发现波形异常,片选引脚CS1和屏幕数据/指令引脚DC波形不正确。反复检查硬件电路和软件代码逻辑关系,并没有发现问题。同时,在集成开发环境的调试模式下,采取单步执行对程序进行调试,2个片选控制信号波形正确。

1.3 问题复现最小系统搭建和试验

选用TI公司的TMDSCNCD28335开发板,搭建问题复现最小系统,用于复现上述问题并利用该最小系统作为试验验证平台,寻找产生问题的原因及可行的解决方案。通过综合分析硬件电路的设计,使用两个GPIO(GPIO12和GPIO13)模拟SPI屏幕使用的屏幕片选引脚CS1和屏幕数据/指令选择引脚DC。

使用同样GpioDataRegs.GPADAT.bit.GPIOxC语言控制指令语句修改GPxDAT寄存器的值来控制GPIO12和GPIO13端口产生连续高低电平变化。通过示波器捕捉端口波形,试验波形如图2所示(图中上面波形为GPIO12、下面波形为GPIO13试验波形),可以看到GPIO12端口没有按照程序指令进行相应的动作,问题被复现。对试验现象进行总结,得出结论: DSP同一分组下的GPIO端口,通过修改GPxDAT寄存器方式来控制组内的任意2个端口时,会出现上述端口未按照程序指令动作的问题;大部分情况下只有最后一条修改GPxDAT寄存器的控制指令会被正确执行,相应端口会产生电平变化。

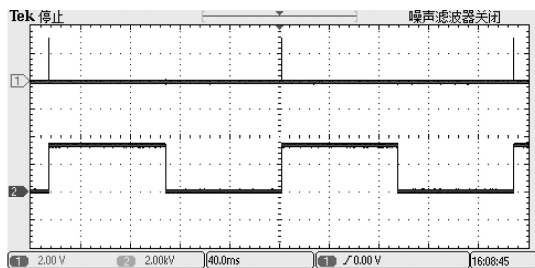


图2 问题复现最小系统试验示波器波形

Fig.2 Waveforms of problem repetition minimum test system

随后对同一分组内2个以上端口开展了进一步的测试试验,经试验发现同样存在上述同组端口的控制问题。为了简化分析过程,本文主要针对2个端口的情况进行研究,对这个问题的产生原因做进一步的分析和试验验证。

2 端口问题产生原因分析

2.1 端口控制问题初步分析

在图2中可以看到GPIO12的波形几乎没有电平变化,只是在某些位置产生了一些极短的脉冲。根据这一试验结果分析,在程序代码执行过程中,使用GpioDataRegs.GPADAT.bit.GPIOx指令语句对同组内端口进行控制,先被执行的控制GPIO12的指令并未使端口产生相应的电平变化,而后面被执行的控制GPIO13的指令可以使端口产生相应的电平变化。

随后进行了单步执行测试,控制指令语句都能够使GPIO12产生正确的电平变化。经初步分析该故障可能与CPU执行延迟有一定关系。于是在对应指令语句后插入足够的延迟,再进行测试研究。示波器波形如图3所示(图中上面波形为GPIO12、下面波形为GPIO13试验波形),此时可以看到GPIO12和GPIO13都产生了正确的电平变化。

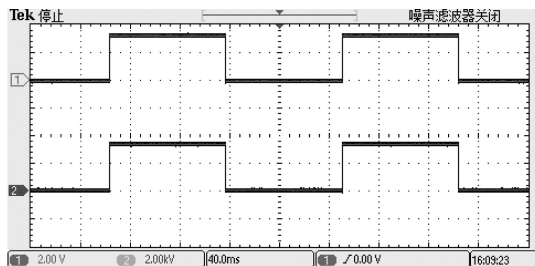


图3 问题复现最小系统增加延迟后试验示波器波形

Fig.3 Waveforms of problem repetition minimum test system after adding delay program

2.2 引起端口控制问题原因的假设提出

综合分析上面的试验结果,总结可能引起端

口控制问题的原因,在此基础上提出如下3种假设:

假设一:CPU流水线在处理相同内存读写时产生时序安排错误,造成第2次指令覆盖或者掩盖了第1条指令的执行;

假设二:CPU在该段内存总线上有一定的时序安排错误,使第2次读取修改写入覆盖或者掩盖了第1条指令的读取修改写入;

假设三:该寄存器存在一定问题,使第2次读取修改写入覆盖或者掩盖了第1条指令的读取修改写入。

2.3 假设一的验证

如果假设一的条件成立,那么对其它地址的内容进行相似的操作,也会出现一致的现象。使用汇编指令编写验证试验程序,去连续读取、修改、写入内存(也称为静态随机存储器(static random access memory, SRAM))中的某地址的16位内容,并在相应语句处设置断点。使用集成开发环境中的工具,观察该地址SRAM中的值。

经过测试试验,观察到SRAM中的值是正确的,因此“假设一”不成立。

2.4 假设二的验证

如果假设二的条件成立,那么对该段内存总线上其它地址的内容进行相似的操作,也会出现一致的现象。使用类似的操作方式,使用汇编指令去连续读取、修改、写入GpioCtrlRegs.GPADIR寄存器的内容,并在相应语句后设置断点,使用集成开发环境中的工具,观察该寄存器的值。

经过测试试验发现寄存器的值是正确的,因此“假设二”情况也不成立。

2.5 假设三的验证

经过上面的验证试验研究,排除了前两种假设。下面针对假设三开展验证试验研究,验证试验过程比较复杂,验证流程图如图4所示。

首先设计试验来验证该寄存器的这种问题是否与寻址方式有关。由于编译器在默认方式编译C代码时,产生了使用DP寻址,OR指令直接操作目的地址的方式进行操作。使用汇编语句实现与C语言控制指令语句相同的功能,采用XARn方式进行间接寻址来操作目的地址。经过测试发现,仍然存在端口控制问题。可以得出结论该寄存器的这种问题与寻址方式并无关联。

对汇编控制语句进行合并和分解操作,定位存在问题的语句执行位置。

汇编指令语句OR指令本身是读取、修改、写

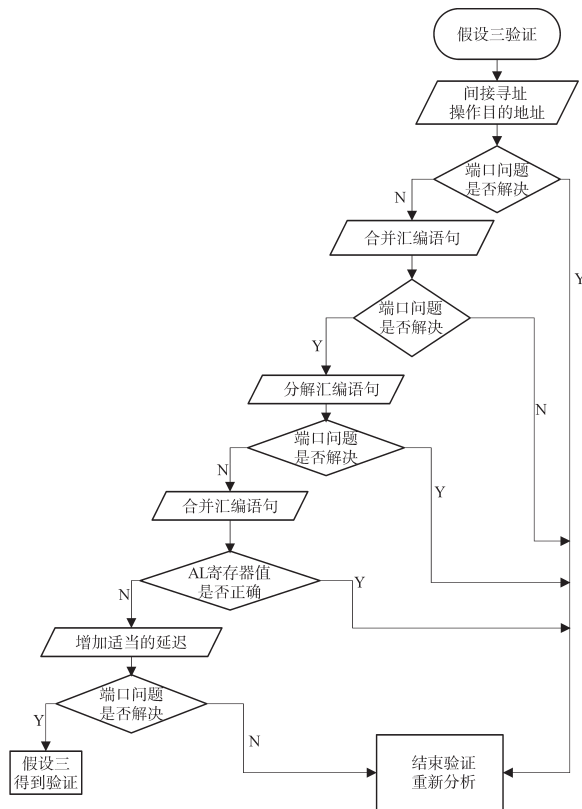


图4 假设三验证流程图

Fig.4 Verification flow chart of hypothesis 3

入的复合指令。通过合并汇编指令的操作,定位语句执行过程中存在问题的位置。经过测试发现,端口控制问题得到了解决。可以基本确认语句执行的问题可能出现在OR语句的执行中间。

将OR汇编复合指令进行分解,设置合适的断点并观察AL寄存器的值,精确定位出现问题的语句执行位置。经过测试发现,AL寄存器的值实际上没有因为控制指令语句的执行而发生相应的变化,即AL寄存器的值不正确。

通过上述2个测试试验,定位了出现问题的执行语句位置。在该语句后加入一定数量的“NOP”延迟指令,产生一定时间的延迟。匹配寄存器指令语句执行和硬件变化的时间。从而验证端口问题是否能得到解决。经过多次测试试验,在加入5个“NOP”指令语句情况下,寄存器的值正确。

至此,可以判定“假设三”是引起这些问题的原因所在。

2.6 补充验证试验和小结

综合分析总结上面的各项验证试验结果可以得出GpioDataRegs.GPxDAT寄存器的值在写入

后需要一定延时才能反应到端口的正确变化。

为了验证所需的延迟是否与CPU时钟有关联,更改CPU的运行时钟配置,测试使端口控制程序得到正确执行所需要的延迟指令时间,即在程序代码中加入的最少“NOP”指令数量。试验结果表明,当改变CPU运行时钟配置时,所需的延迟指令时间无明显增加,也就是说所需的延迟时间与CPU时钟没有关联。

将原程序由RAM转到Flash中执行,并关闭Flash运行加速,由于运行位置的变更使单条代码的执行周期数发生了变化,CPU执行单条指令所需的时钟周期将发生变化。测试使端口控制程序得到正确执行所需要的延迟指令时间。测试结果表明,所需的延迟指令时间无明显变化。

通过分析可以得出GpioDataRegs.GPxDAT寄存器的写入延迟与CPU运行频率无关,而是与指令的运行处理器周期数目相关。对该寄存器做修改后,必须经过一定的处理器周期才能够反应到对应硬件端口上的相应变化。因此,修改该寄存器的值后必须加入足够的延迟才能够避免产生端口控制的问题。

3 解决措施及措施验证

很多嵌入式处理器,尤其是使用精简指令集的处理器,使用软件控制GPIO时,操作往往会被分成3步:读取、修改、写回。如果这3步是非原子性的,那么由于外部中断或多线程执行等状况,处理器不能保证执行结果的正确性,所以很多嵌入式处理器提供专用的SET, CLEAR 和 TOGGLE 寄存器来对GPIO进行控制。由于这些寄存器的读取无效,只有写入才生效,因此嵌入式处理器只需要1个写入指令即可完成对GPIO的控制。不需要预先知道原GPIO状态,使GPIO的操作保持了原子性。TMS320F28x系列DSP尽管提供了用于原子操作的指令,但由于GpioDataRegs.GPxDAT寄存器修改的延迟,导致连续修改同一寄存器时出现了端口控制问题。因此需要使用GpioDataRegs.GPxSET/ CLEAR/ TOGGLE寄存器来操作GPIO或者在适当的位置加入一定的延迟指令等待寄存器完成修改,来解决端口控制问题。

根据试验结果综合分析研究,提出以下3种基于软件程序的解决措施方案:

解决措施方案一:在通过位域方式编程修改

寄存器 GpioDataRegs.GPxDAT 的值后须加入足够的延迟才可避免产生错误的执行逻辑。考虑到代码调试“RAM 执行”和实际应用“Flash 执行”的差异,可以经测试来验证需要插入延迟指令的数量来避免逻辑错误。

解决措施方案二:当编程连续操作同一个寄存器 GpioDataRegs.GPxDAT 时,应避免使用位域方式进行操作,而是将多个操作合并到同一条语句操作内完成,这样可以有效解决端口控制问题。

解决措施方案三:采用 GpioDataRegs.GPxSET/ CLEAR/ TOGGLE 3 个寄存器中的任意一个寄存器的控制方式,代替通过 GpioDataRegs.GPxDAT 寄存器进行端口控制的方式,可稳定实现预期功能。

在完成本文上述研究、试验的基础上,利用专用电源嵌入式控制系统进行了相应的测试试验和考核试验,验证了以上 3 种措施方案都可以有效地解决端口控制问题。

嵌入式控制系统对实时性的要求往往较高,对 3 种措施方案的控制实时性进行分析验证,这 3 种措施方案的实时性几乎是一致的,尽管第 1 种解决措施方案使用了一定量的“NOP”指令作为延迟函数,但是经过测试该指令使用数量一般低于 10 个,相当于处理器停滞了 10 个周期。以 TMS320F28335 为例,DSP 主频在 150 MHz 的使用条件下,延迟指令最多会带来 66.6 ns 的延迟。在实际的嵌入式控制系统中,直接操作 GPIO 通常用于“慢速”控制,该延迟对嵌入式控制系统的实时性几乎不会产生影响。

随后,在 TMS320F28x 系列其他 5 种型号的 DSP 进行了相似问题的测试以及 3 种解决措施的验证。主要验证型号和结果如表 2 所示。

表 2 拓展测试与措施验证结果

Tab.2 Additional testing and measure verification results		
DSP 型号	端口问题	解决方案验证
TMS320F28069	存在相似问题	3 种都有效
TMS320F28377S	存在相似问题	3 种都有效
TMS320F28377D	存在相似问题	3 种都有效
TMS320F28379S	存在相似问题	3 种都有效
TMS320F28379D	存在相似问题	3 种都有效

如需测试上述型号以外的 TMS320F28x 系列 DSP 或者其他系列的嵌入式处理器是否存在本文所述的 GPIO 端口控制问题,可以采用与本文类似的方法进行端口问题的测试验证。测试时在测试程序中,以位域方式先给目标芯片的 2 个同

组的 GPIO 控制寄存器赋值,然后调用延迟函数延迟一定的时间,并以位域方式将相同的 2 个 GPIO 控制寄存器赋相反的值,再延迟一定的时间,通过示波器观察 GPIO 翻转波形是否正确。

4 结论

本文通过对专用电源嵌入式控制系统开发过程中遇到的端口控制问题进行复现,经过综合分析、推测假设与试验验证,证明了 TMS320F28x 系列 DSP 有部分型号存在 GPIO 端口控制上的问题,并分析出引起端口控制问题的原因。提出 3 种可行的解决方案,经测试,这 3 种方案均可以解决端口控制问题。在今后的使用中,如果硬件资源允许的条件下,在使用 GPIO 端口做片选控制和输入输出端口控制时,应避免将相应端口配置在同一分组下。如果硬件资源不允许,可采用本文提出的 3 种解决方案对控制软件程序进行优化。另外,本文只是针对常用的 TMS320F28x 系列的 6 种型号 DSP 进行了测试验证,今后还需要进一步验证其他型号的 DSP 是否存在相似的问题。本文提出的相关研究思路和方法,可以作为分析及解决类似问题的参考。

参考文献

- [1] 刘陵顺,高艳丽,张树团,等. TMS28335 DSP 原理与开发编程[M]. 北京:北京航空航天大学出版社,2011.
- [2] 张卿杰,徐友,左楠,等. 手把手教你学 DSP——基于 TMS320F28335[M]. 北京:北京航空航天大学出版社,2018.
- [3] 江宏玲,周成,戴新荣,等. 基于 DSP 异步电动机矢量控制系统仿真与实验研究[J]. 电气传动,2017,47(6):14-18.
- [4] 孙丛君,张喜茂. 基于 TMS320F2812 的直接转矩数字控制系统[J]. 电气传动,2019,49(3):26-31.
- [5] Texas Instruments. TMS320F2833x, TMS320F2823x DSCs silicon errata silicon revisions A, 0[DB/OL]. 2019[2019-04] <http://www.ti.com/lit/er/sprz272l/sprz272l.pdf>.
- [6] Texas Instruments. TMS320F2833x, TMS320F2823x digital signal controllers (DSCs)[DB/OL]. 2019[2019-04] <http://www.ti.com/cn/lit/ds/symlink/tms320f28335.pdf>.
- [7] Texas Instruments. TMS320x2833x, 2823x system control and interrupts reference guide[DB/OL]. 2010[2010-03] <http://www.ti.com/lit/ug/spru0d/spru0d.pdf>.
- [8] Texas Instruments. TMS320x2833x, 2823x serial peripheral interface (SPI) reference guide[DB/OL]. 2009[2009-06] <http://www.ti.com/lit/ug/sprueu3a/sprueu3a.pdf>.

收稿日期:2020-04-08

修改稿日期:2020-04-21