

高性能 PLC 专用指令集处理器设计与仿真

曾舒婷^{1,2}, 杨志家¹

(1 中国科学院沈阳自动化研究所 工业信息学重点实验室, 辽宁 沈阳 110016; 2 中国科学院研究生院, 北京 100039)

摘 要: 该高性能 PLC 专用指令集处理器采用自主设计的 PLC 专用指令集, 符合 PLC 指令特征, 可减少该 PLC 专用指令集处理器执行的指令数, 并采用 32 位 RISC 体系结构加快 PLC 程序的执行速度. 该高性能 PLC 专用指令集处理器采用哈佛总线结构, 寄存器组采用位编址模式, 位处理器可加速 PLC 布尔运算, 功能块单元可提高功能块指令执行的精度, 并采用四级流水线提高 PLC 指令的执行速度. 现已完成了该高性能 PLC 专用指令集处理器的系统功能仿真, 经测试仿真结果正确.

关键词: PLC; 专用指令集; 专用指令集处理器; RISC 体系结构; 四级流水线

中图分类号: TP36

文献标识码: A

文章编号: 1000-7180(2011)07-0076-06

Design and Simulate High Performace ASIP for PLC

ZENG Shu-ting^{1,2}, YANG Zhi-jia¹

(1 Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; 2 Graduate School of the Chinese Academy of Sciences, Beijing 100039, China)

Abstract: The PLC ASIP adpots PLC special instruction set, which is consistent with the character of PLC instructions. The PLC special instruction set can reduce the number of instructions and accelerate the executing speed of PLC program by 32 bit RISC architecture. The designed architecture of the high performance ASIP for PLC adopts harvord bus architecture, the registers adopts bit addressing mode, the bit processor can accelerate bool operation, the function block unit can improve the precision of function block execution, the high performance ASIP for PLC adopts four pipelines to speed up the PLC instruction execution. The systemic functional simulation of the high performance ASIP for PLC has been finished, and the result of simulation is correct after testing.

Key words: PLC; instruction set; ASIP; RISC architecture; four pipelines

1 引言

可编程控制器(PLC)是以微处理器为核心,把自动化技术、通信技术融为一体的新型工业自动控制装置. 可编程控制器以其可靠性高、抗干扰能力强、实时性强、控制程序可变、体积小、功耗低、使用方便、易维护等特点,广泛应用于工业行业的各个领域. 随着信息技术的迅猛发展,工业环境对可编程控制器的网络通信能力及实时性要求越来越高. 现有的可编程控制器微处理器,大多采用通用微处理器或者通用处理器和布尔协处理器相结合^[1-2]的形式. 通用处理器主要面向字节、字的处理,而 PLC 指令

主要面向比特位操作,占 PLC 指令使用频率的 69%^[3]. 国内现有的 PLC 微处理器模式大多仅仅增加布尔协处理器单元处理 PLC 位逻辑指令,但没有实现位逻辑指令的加速执行,亦没有针对 PLC 功能块指令的处理,很难提高处理 PLC 指令的性能.

据此研究设计了一款 PLC 专用指令集处理器,该 PLC 专用指令集处理器有七个特点:拥有符合 PLC 指令特征的指令集;采用 32 位 RISC 处理器体系结构^[4-5]满足 PLC 指令集的执行,通过位处理器和功能块单元分别实现了对布尔指令的加速执行和功能块指令的精确执行;采用哈佛总线结构,将指令存储器与数据存储器分开,符合 PLC 对数据频繁需

求的特征;由于 PLC 数据量大,为节约数据存取时间,采用存取器直接寻址和寄存器直接寻址方式;寄存器组采用位编址模式,符合 PLC 指令面向布尔运算的特征;包含功能块寄存器组,其编址模式符合 PLC 功能块指令所需数据存储特征;采用四级流水线,将执行阶段和存储访问阶段合并,符合 PLC 程序多为布尔指令,且其执行速度快的特征。

2 PLC 专用指令集处理器的体系结构

该 PLC 专用指令集处理器包括以下单元:指令存储器、指令计数器、指令寄存器、指令译码器、寄存器组、位处理器、功能块单元、跳转调用\存取单元、功能块寄存器组、数据存储器、I/O 数据存储器、控制单元等,如图 1 所示。

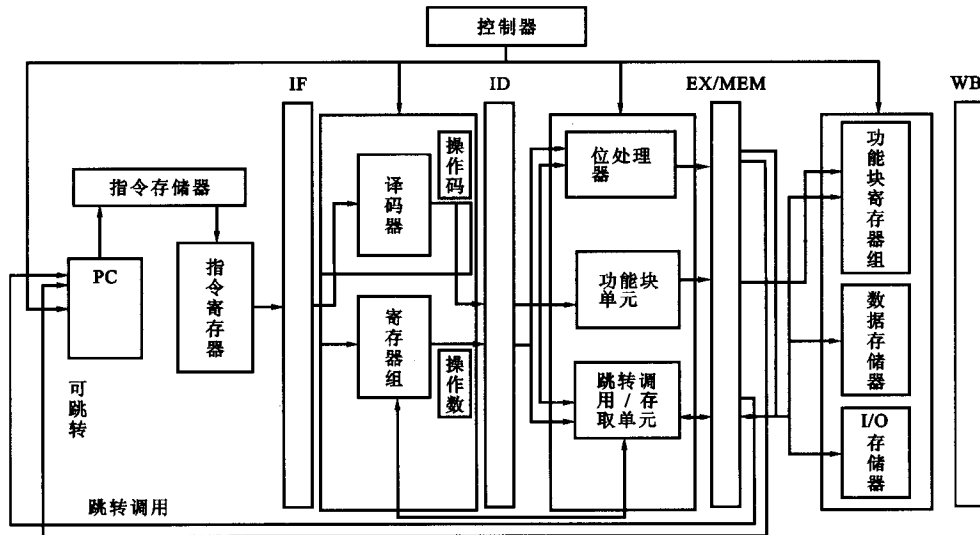


图 1 PLC 专用指令集处理器的体系结构

该 PLC 专用指令集处理器采用 32 位 RISC 体系结构;由于 PLC 程序多为布尔指令,其执行时间短,可将执行阶段和存储访问阶段合并,采用四级流水线,以加快 PLC 程序的执行。PLC 对数据存取频繁,其总线采用哈佛总线结构,其数据存取方式采用存取器直接寻址和寄存器直接寻址方式。其中寄存器组采用位编址模式,符合 PLC 指令主要面向布尔运算的特征。

位处理器可执行位逻辑运算指令和可跳转位逻辑运算指令,通过处理可跳转位逻辑运算指令加速 PLC 布尔运算的执行。

功能块单元功用于执行 PLC 功能块指令,如置位复位优先指令、边沿检测器指令、计数器指令、计时器指令;并可分别同时执行 16 个同类功能块指令,基本满足小型 PLC 功能块指令执行的需求;功能块寄存器组,其编址模式依据功能块指令数据的特征,用于存储功能块输入输出数据。

3 PLC 专用指令集

根据指令格式中组码及操作码的不同,将指令集分为位逻辑运算指令集、可跳转位逻辑运算指令集、跳转调用指令集、存取指令集、功能块指令集。

3.1 指令格式

指令格式为位逻辑运算指令格式、可跳转位逻辑运算指令格式、跳转调用及存取指令格式以及功能块指令格式共四种,如表 1 所示。

位逻辑运算指令格式针对布尔指令而设计,其操作数来源于累加器堆栈 CR、输入寄存器及内部寄存器,并将结果保存在累加器堆栈 CR 中。

可跳转位逻辑运算指令格式针对可跳转位逻辑运算指令而设计,其跳转条件取决于当前累加器堆栈值。若可跳转位逻辑运算指令满足跳转条件,则发生跳转。

跳转调用及存取指令格式针对 PLC 指令中跳转调用指令而设计,若其指令格式第 24 位值为 1,则该跳转调用指令有条件位,其条件位的值来自输入寄存器 RS 或累加器堆栈 CR,并依据跳转调用偏移量跳转。当指令格式中第 25 位值为 1 时,该指令为存取指令,其操作数来自数据存储器、I/O 数据存储器及功能块寄存器。

PLC 功能块分为触发器、边沿检测、计数器及定时器,它们都有自身的功能块单元寄存器组。功能块指令格式中,当第 21 位值为 1 时,则该功能块指令有初始值,则下一条指令存储该功能块指令的 32

位初始值。

表 1 PLC ASIP 的指令格式

指令格式	组码 31~30	操作码 OP29~26	25	24	23..16		15..8		7..0	
位逻辑运算 指令格式	00		判断源操作数 1 是否 来自累加器堆栈 内 CRYN	未定义	目的地址——内部寄存 器 RM 或累加器堆栈 CR 的数据地址		源操作数 1 地址——输 入寄存器 RS、RM、CR 地址或功能块寄存器 Q 地址		源操作数 2 地址——输 入寄存器 RS、RM、CR 地址或功能块寄存器 Q 地址	
可跳转位逻辑 运算指令格式	01		跳转条件位	未定义	跳转偏移量				当前累加器堆栈地址	
跳转调用及 存取指令格式	10		跳转调用指令 i=0	是否有条件位	输入寄存器 RS 或累加 器堆栈 CR 的数据地址		跳转调用偏移量或操作数存取地址			
			存取指令 i=1							
功能块 指令格式	11	复位优先或加计数器 输入的上升沿触发位	置位优先或减 计数器输入的 上升沿触发位	23	22	21		20..11	10..0	
				复位或时钟 电平	初始值输入 使能位	是否有初 始值		未定义	计数器初始值或计 时器初始值地址	

3.2 指令集

根据指令格式中组码及操作码的不同,将指令集分为位逻辑运算指令集、可跳转位逻辑运算指令集、跳转调用指令集、存取指令集、功能块指令集。

其中可跳转位逻辑运算指令集包括:位累与指令、位累与非指令、位累或指令、位累或非指令、可跳转位逻辑运算指令跳转结束符、累加器堆栈栈顶操作数出栈并求或指令。

存取指令集包括:功能块寄存器组与寄存器组之间的存取指令;存储单元与寄存器组之间的存取

指令;\I\O 数据存储器与寄存器组之间的存取指令;寄存器组内部寄存器与累加器堆栈之间的存取指令;

功能块指令集包括:置位优先功能块指令、复位优先功能块指令、上升沿检测功能块指令、下降沿检测功能块指令、加计数器功能块指令、减计数器功能块指令、加减计数器功能块指令、脉冲功能块指令、接通延时功能块指令、断开延时功能块指令、实时时钟功能块指令。部分 PLC 指令集如表 2 所示。

表 2 PLC ASIP 的部分指令集

指令集分类	指令
位逻辑运算指令集	ANDBIT、ORBIT、XORBIT、ANDNBIT
可跳转位逻辑运算指令集	ANDCRBIT、CR、ORCRBIT、PUSHOR
跳转调用指令集	JMPBIT、JMPCBIT、CALBIT、CALCBIT、RETBIT、RETCBIT
存取指令集	LOADQBIT、STOREQEBIT、LOADMBIT、STOREMBIT、LOADIOBIT、STOREIOBIT、LOADRMBIT、STORERMBIT
功能块指令集	FB SR、FB RS、FB CTU、FB TON

4 PLC 专用指令集处理器四级流水线设计

4.1 取指阶段

该取指阶段执行步骤如下:

(1) 指令计数器依据当前指令地址,从指令存储器取指令至指令寄存器当中;

(2) 若无来自跳转调用\存取单元中跳转调用的中断或来自位处理器单元中可跳转位逻辑运算指令的跳转中断,指令计数器下一条指令地址等于当前指令地址加 4;

(3) 若有跳转调用指令的中断,则指令计数器

依据跳转调用偏移量从指令存储器中取指;若为可跳转位逻辑运算指令的跳转中断,则指令计数器依据跳转偏移量从指令存储器中取指;若有返回指令中断,则返回偏移量为下一条 PLC 指令地址,指令计数器按照其返回偏移量取指。

4.2 译码阶段

指令译码器从指令寄存器取出指令进行译码,通过指令中操作数地址,取出操作数;并依据指令中组码,将相应的操作码及操作数送至对应的处理单元中;该译码阶段前半半个时钟周期用于译码,后半个

时钟周期取出操作数。

4.3 执行阶段

依据指令译码器发出的操作码及操作数,位处理器、功能块单元、跳转调用\存取单元给予相应处理;另外存储器访问阶段与执行阶段合为一个流水线阶段,跳转调用\存取单元执行存储器访问。

4.3.1 布尔指令执行单元

位处理器(图2)用于执行 PLC 布尔指令,包括位逻辑运算指令及可跳转位逻辑运算指令;采用累加器堆栈的形式存储位处理器所需操作数及其运算结果;其累加器堆栈的栈顶值来自寄存器组、功能块寄存器组、I/O 数据存储器及位处理器运算结果;位处理器的操作数来自累加器堆栈、源寄存器、内部寄存器;位处理器运算结果保存到累加器堆栈、内部寄存器、数据存储器及 I/O 数据存储器;为后期能够与通用处理器通信,建立状态寄存器,存储位处理器产生的状态信号。

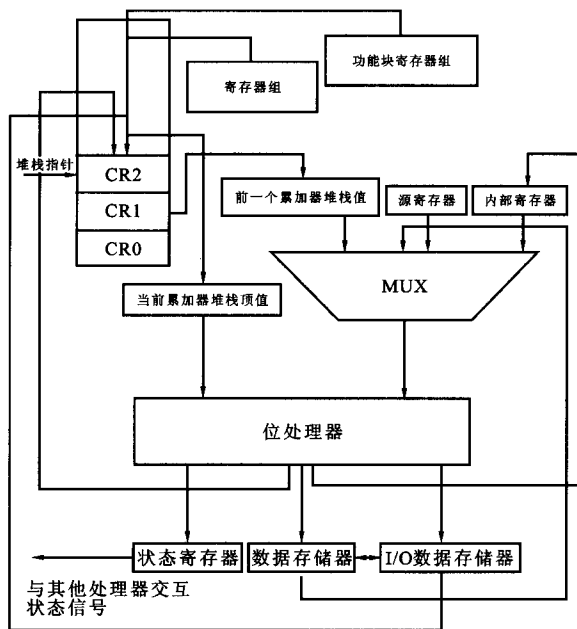


图2 位处理器结构图

当位处理器执行位逻辑运算指令时,从累加器堆栈的栈顶或寄存器组中取操作数后,进行位逻辑运算操作,将运算结果存储到累加器堆栈的栈顶或寄存器组中;执行完成后存储位处理器的运算结果;

当位处理器执行可跳转位逻辑运算指令时,按可跳转位逻辑运算指令的执行流程(图3)执行,执行完成后存储位处理器的运算结果;

可跳转位逻辑运算指令的执行过程为:

(1) 如果是可跳转位逻辑指令,则指令译码器向位处理器输入操作码;

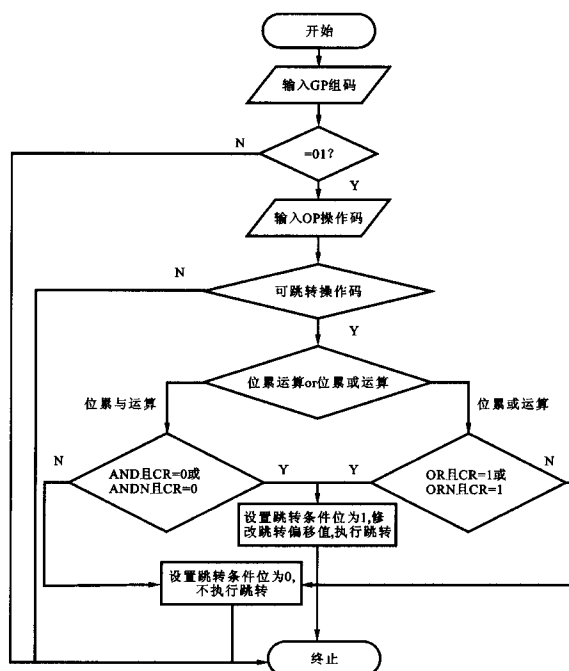


图3 可跳转位逻辑运算流程图

(2) 由位处理器判断该可跳转位逻辑指令是否为位累与指令、位累与非指令、位累或指令、位累或非指令:

① 若为位累与指令,且累加器堆栈的栈顶值为0,则发生跳转;若累加器堆栈的栈顶值为1,则只执行与操作,其跳转条件位为0,不执行跳转操作;

② 若为位累与非指令,且累加器堆栈的栈顶值为0,则发生跳转,并将跳转条件位置为1;若累加器堆栈的栈顶值为1,则只执行与非操作,其跳转条件位为0,不执行跳转操作;

③ 若为位累或指令,且累加器堆栈的栈顶值为1,则发生跳转,并将跳转条件位置为1;若累加器堆栈的栈顶值为0,则只执行或操作,其跳转条件位为0,不执行跳转操作;

④ 若为位累或非指令,且累加器堆栈的栈顶值为1,则发生跳转,并将跳转条件位置为1;若累加器堆栈的栈顶值为0,则只执行或非操作,其跳转条件位为0,不执行跳转操作。

4.3.2 功能块指令执行单元

PLC 专用指令集处理器中的功能块单元包括触发器、边沿检测、计数器及计时器四类功能块单元,可分别同时执行 16 个同类功能块指令,基本满足小型 PLC 功能块指令执行的需求;针对功能块指令数据的特征,另外设计了功能块寄存器组,用于存储功能块输入输出数据,相对片外计数器与计时器等功能块单元,加速对功能块输入输出数据的存取,

提高了 PLC 专用指令集处理器的实时性与精确度。

4.4 回写阶段

将 PLC 专用指令集处理器中位处理器执行结果,依据存取指令类型写回寄存器组中的累加器堆栈、寄存器组中的内部寄存器、数据存储器或 I/O 数据存储器当中;将 PLC 专用指令集处理器中功能块单元执行结果,存储在功能块寄存器组内。

5 仿真与实现

图 4 给出了该 PLC 专用指令集处理器运行定时器程序一个循环时位处理器的仿真结果。在仿真过程中设 I/O 端口“X0”、“X1”、“X2”、“X4”、“X5”值为 1。从仿真波形图可以看出,当执行第一条“LOADIOBIT”指令时,译码器输出信号“out_

BITJumpEnable”使能位处理器及“out_BITType”操作码,并在下一周期执行可跳转位逻辑运算指令“ANDCRBIT”;由于 I/O 端口“X0”输入值为“1”,则对应累加器堆栈的栈顶值为“1”,执行指令“ANDCRBIT”后,对应其“out_jumpcondition”值为“0”,则该可跳转位逻辑运算指令不发生跳转;一周后译码“X1 AND X2”指令,在译码阶段后半周期,依据译码的左操作数地址“out_BITLeftRegisterAdr”及右操作数地址“out_BITRightRegisterAdr”取操作数,并输出“out_BITTargetRegisterAdr”结果存储地址;由于 I/O 端口“X1”、“X2”输入值取反,则该“X1 AND X2”指令执行结果“out_BITTargetRegister”值为“0”。

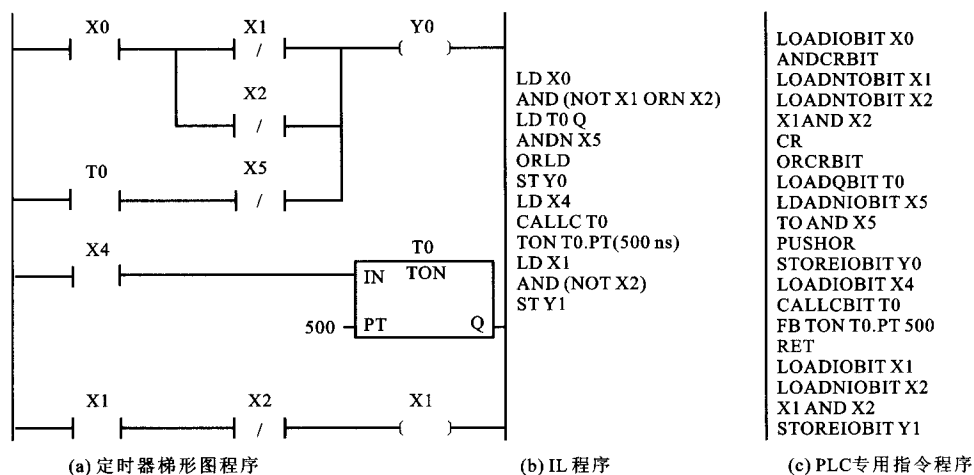


图 4 PLC 运行定时器程序

图 5 给出了该 PLC 专用指令集处理器执行定时器程序八个循环时功能块单元的仿真结果,即接通延时定时器 TON 的仿真结果。在整个仿真过程中,设 I/O 端口前七个循环“X0”、“X1”、“X2”、“X4”、“X5”值为 1,并设置接通延时定时器的时延的初始值为,设 I/O 端口最后一个循环“X0”、“X1”、“X2”、“X5”值为 1,“X4”值为 0,即接通延时定时器 TON 关闭。从仿真波形图可以看出,当执行接通时延大于或等于 500ns 时,“out_Q”输出值为 1,即该定时器输出值为 1。当执行到 IO 输入“X4”值为 0 时,该定时器断开,“out_Q”输出值为 0。

经过各模块仿真及系统仿真后,该 PLC 专用指令集处理器的仿真结果正确,并可提高该 PLC 专用指令集处理器的性能。为验证该 PLC 专用指令集处理器的性能,本实验利用 PIC16F877、ATmega16 及 PLC 专用指令集处理器分别对含开关量 PLC 程序

进行编译,并比较其编译后指令数目,其实验结果如表 3 所示。

表 3 各处理器编译 PLC 程序后的指令数目比较

处理器 指令数目	PIC16F877	ATmega16	PLC 专用指令 集处理器
定时器程序	98	256	21
交通灯程序	618	944	193

实验结果证明,PLC 专用指令集处理器相比 PIC16F877、ATmega16 可减少编译后的指令数,对于定时器程序其指令数约减少到 PIC16F877 编译指令数的 21.4%及 ATmega16 编译指令数的 8.2%;对于交通灯程序其指令数约减少到 PIC16F877 编译指令数的 31.2%及 ATmega16 编译指令数的 20.4%;该 PLC 专用指令集处理器编译指令数减少,执行的速度也相对加快,可提高该 PLC 专用指

