文章编号: 1000-6893(2000)02-0134-04

嵌入式微处理器 NCS 多任务管理单元 MTU 的研究

李树国[1], 高德远[1], 聂培琴[2]
(1. 西北工业大学 航空微电子中心, 陕西 西安 710072)
(2. 山东建筑材料工业学院 信控系, 山东 济南 250022)

STUDY ON MULTITASK MANAGEMENT UNIT MTU OF EMBEDDED MICROPROCESSOR NCS

LI Shu-guo[1], GAO De-yuan[1], NIE Pei-qin[2]
(1. Aviation Microelectronic Center, Northwestern Polytechnical University, Xi′an 710072, China)
(2. Information & Control Department, Shandong University of Building Materials, Jinan 250022, China)

摘 要: 确立了多任务管理单元 MTU 的设计方案, 探讨了多任务管理的定义、数据结构, 给出了微处理器多任务切换算法, 提出了多任务管理单元 MTU 细胞群结构。细胞群结构由细胞组成, 每个细胞可进行一种测试并输出测试的布尔值给微程序控制器来控制微程序的转移。多任务管理单元 MTU 的 RTL 级 VHDL 描述经 EDA 工具 MENTOR GRAPHICS 的综合与仿真验证了多任务管理单元 RTL 级 VHDL 描述的正确性与有效性。

关键词: 多任务管理单元; 任务切换; 细胞; 指令周期
中图分类号: TP368; V247.1 文献标识码: A

Abstract: Multitask management unit (MTU) is a test unit which implements multitask management in a protected mode in microprocessors, and provides hardware support for tasks switch in microprocessor level. Firstly, a scheme for MTU is established, based on the scheme tasks switch algorithm given to switch tasks. Secondly, by analyzing the data structure and definition for multitask management, a kind of cell group architecture for MTU is put forward. The architecture consists of cells, each of which processes a kind of test and outputs Boolean value to microprogram controller. So an aim is achieved that the Boolean value of cell test in MTU may control microprogram branch. At last, the VHDL description of RTL level of MTU has been synthesized and simulated successfully in MENTOR GRAPHICS of EDA tools and its results of simulation prove MTU valid. \ = Key words: multitask management unit (MTU); task switch; cell; instruction cycle

结合航空应用的高可靠性、抗干扰性、实时检测、故障恢复等特点, 依照航空技术规格要求, 设计了机载嵌入式 16 位拥有保护机制的微处理器。该微处理器集多任务管理[1]、内存保护[2]、中断处理与响应、故障恢复等嵌入在芯片内, 实现了片上系统(system on chip)。

嵌入式微处理器 NCS 的多任务管理单元 MTU 是微处理器在保护模式[3,4]下实施多任务管理的测试单元。它完成任务切换的各种条件检查, 设置测试的状态位, 控制微程序的转移, 并给那些不能由中断处理程序恢复的故障提供了用任务恢复的可能性, 这对机载嵌入式微处理器来说意义重大。

收稿日期: 1999-01-22; 修订日期: 1999-05-28
基金项目: 航空基金(97F53133)资助项目
文章网址: http://www.hkxb.net.cn/hkxb/2000/02/0134/

1 多任务管理的定义

支持多任务管理的数据结构是任务状态段, 任务状态段描述符, 任务寄存器, 任务门描述符。使用这种结构, 处理器能从一个任务切换到另一个任务。

1.1 任务状态段 TSS (Task State Segment)定义

为恢复一个任务所需的微处理器状态信息被保存在指定的段中, 该段称为任务状态段 TSS。

TSS 动态字段是在每一次任务切换时处理器所需更新的 TSS 那些字段; 它包括通用寄存器, 段寄存器, 标志寄存器 Flags, 指令指针 IP, 前一任务的 TSS 指针(Back Link)。

TSS 静态字段是由任务创建时建立的, 任务切换时并不改变的 TSS 那些字段; 这些字段是任务 LDT 的选择子, 特权级 0, 1, 2 各自堆栈指针 SS: SP。任务状态段 TSS 见图 1。

图 1　任务状态段 TSS

1.2　任务状态段描述符 (TSS descriptor) 与任务门(Task Gate)描述符定义

用连续 8 个字节来表示任务状态段的控制信息, 这 8 个字节称任务状态段描述符; 它含状态段基址, 段限和属性见图 2。任务门描述符提供对任务状态段间接的、受保护的访问见图 3。任务门与任务状态段属性字段中的 P 位: 状态段/ 任务门的存在位; B 位: 状态段忙位, DPL 位: 程序访问状态段/ 任务门的最低特权级。如果在中断描述符表 IDT (Interrupt Descriptor Table) 的槽位中定义任务门, 则不执行中断处理, 而产生任务切换。这对那些不能用中断处理的故障和异常, 任务门提供了故障修复的可能性。

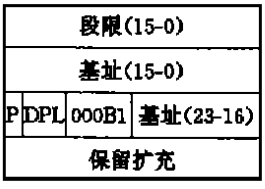

图 2　任务状态段 TSS 描述符

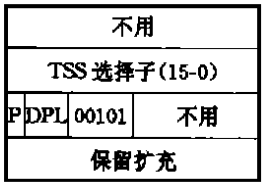

图 3　任务门描述符

1.3　任务寄存器 TR (Task Register)

任务寄存器(TR) 分两部分组成, 一部分存放 16 位选择子即 TR. selector(寻找 GDT 表中 TSS 段描述符的索引值); 另一部分是 TR 所对应的 cache 即 TR. cache, 它用于存放所索引的 TSS 段描述符, 见图 4。

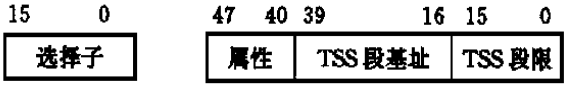

图 4　索引的 TSS 段描述符

2　任务切换算法

2.1　多任务管理表 GDT, LDT 和 IDT 及相应表寄存器 GDTR, LDTR 和 IDTR

保护模式下的每个段都对应一个段描述符, 每个段描述符依其属性不同分别存放在系统的 3 张表 GDT, LDT 和 IDT 中: GDT (Global Descriptor Table) 是全局描述符表; LDT (Local Descriptor Table) 是局部描述符表; IDT (Interrupt Descriptor Table) 是中断描述符表, 它们都被定义在存储器中。GDT 表整个系统仅有 1 张, 而 LDT 表每个任务 1 张; GDT/LDT 分别用来存放系统/局部代码段及数据段描述符、调用门及任务门等; IDT 整个系统仅有 1 张, 用于存放中断门、陷阱门和任务门, 以进行中断处理或故障恢复。

GDTR (Global Descriptor Table Register) 全局描述符表寄存器用于存放 GDT 的 24 位线性表基地址及其表限。IDTR (Interrupt Descriptor Table Register) 中断描述符表寄存器用于存放中断描述符表 IDT 的 24 位线性基地址及其表限。LDTR (Local Descriptor Table Register) 局部描述符表寄存器, LDTR. selector 用于存放寻址当前任务的选择子, LDTR. cache 用于存放由选择子在 GDT 中所指定的描述符, 该描述符由系统自动加载。见图 5。

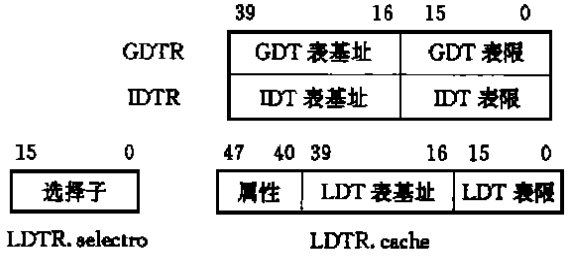

图 5　GDTR, IDTR, LDTR 寄存器

2.2　任务的建立

对微处理器 NCS 来说, 在执行任务以前, 先在存储器中定义 GDT, IDT, LDT 和 TSS, 再根据它们的基址和边界定义 GDTR, LDTR 及 TR。任务建立算法, 见图 6 的¹ ～À:

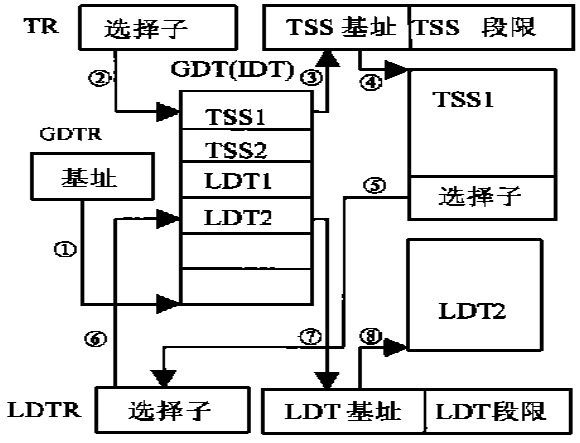

图 6　任务的建立

¹ 在GDT 中定义段描述符、LDT 描述符、门和 TSS 描述符。在 IDT 中定义中断门、陷阱门和任务门。并把 GDT 和 IDT 的基地址和边界分别送到GDTR 和 IDTR;

º 新任务的选择子装入 TR, 并将 TR 的选择子×8+ GDTR 基址得 TSS1 描述符的地址;

» 把TSS1 的描述符送入 TR. cache;

¼ 以TR. cache 的段基址得 TSS1 状态段;

½ 将TSS1 段中的 LDT 选择子装入 LDTR;

¾ LDTR 的选择子×8+ GDTR 的基址得 LDT2 描述符的地址;

¿ LDT2 描述符送到 LDTR. cache;

À 这时 LDTR. cache 高速缓存器指向新任务的局部描述符表 LDT2。任务建立完毕。

2.3 多任务切换算法

任务切换算法 Task-switch 如下:

step1: 检查任务状态段长度, 若大于 43, 顺序执行; 否则报告异常;

step2: 当前机器状态保存在老任务的 TSS 中;

step3: 新任务 TSS 描述符的选择子装入 TR, 并由系统自动加载新任务的 TSS 段的基址、属性及段限到 TR. cache;

step4: 把除 CS, SS, DS, ES 之外新任务 TSS 段的内容, 装入微处理器的硬件寄存器;

step5: 若链接, 老任务 TSS 选择子存入新任务 TSS 中, 置嵌套位, 新任务的描述符置 "忙"标记; 若解链, 老任务 TSS 的描述符标为 "非忙"; 若无链接, 老任务 TSS 的描述符标为 "非忙", 新任务的 TSS 描述符标为 "忙";

step6: 将 TSS 段的 LDT 选择子送入 LDTR, 并将该选择子×8+ GDTR 的表基址得 LDT 描述符地址, 并将其装入 LDTR. cache;

step7: 新任务的段选择子装入段寄存器, 选择子所对应的描述符由系统自动加载各自段的 cache。结束。

任务建立与切换算法的微程序固化于微处理器的控制ROM 中, 它完成了硬件芯片级切换; 这比过去由操作系统软件切换, 提高了微处理器响应速度。

3 多任务管理单元 MTU 的设计

3.1 单细胞

多任务管理单元是保护模式下实施多任务管理测试电路, 它主要在任务转换时进行特权级检查、段越界比较、链接字的判断、描述符的分类等, 并置测试的状态位控制微程序的转移。列举多任务管理单元所进行的各种测试, 给每种测试建立一个基本的测试单位称细胞。例如: 任务转换时, 任务门或 TSS 描述符特权级 DPL(见图 2、图 3 属性字段)必须满足 DPL≥MAX(CPL, RPL); CPL 为当前运行程序的特权级, RPL 为请求者的特权级。实现此关系式的细胞 cell 见图 7。

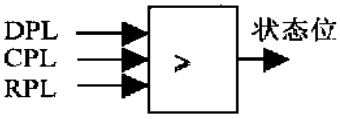

图 7 基本测试单位——细胞

3.2 多任务管理单元细胞群结构

对多任务管理中任一测试 $cell_i$, $1 \leqslant i \leqslant n$, n 为细胞数。根据功能不同也建立各自的细胞体, 则构成细胞群集合 C。

P cell 若 cell∈C, 合并同类项(细胞功能相同, 输入端数目相同)得子集 C′, 那么 C′A C, 则 C′为真子集即测试单元的终结细胞群集合。

在微处理器的每个控制步中, 细胞群中的个体仅能有一个处于激活状态(有效), 其它应为睡眠状态。这可采用多选一电路来控制, 以激活所选择的测试细胞。

见图 8 中①表示测试细胞。被测数据由 DBUS 送至暂存器 temp1, temp2, temp3 中, 控制信号由控制总线 CBUS 送至控制电路以选择某个测试细胞①, 经测试输出测试的布尔值。用该布尔值可确定多任务切换时微程序的走向。

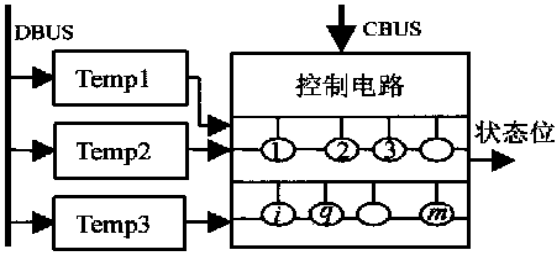

图 8 多任务管理单元细胞群结构

3.3 微处理器中的多任务管理单元 MTU

嵌入式机载 16 位微处理器 NCS 体系结构, 采用 3 级流水[4, 5], 即 指令预取] 指令译码] 指令执行。它由 6 个部件构成见图 9, 分别是多任务管理单元 MTU、地址单元 AU、算逻单元 ALU、微程序控制单元 MCU、指令译码单元 IU 和接口单元 I/O。多任务管理单元 MTU 与其它部件的

关系见图 9。

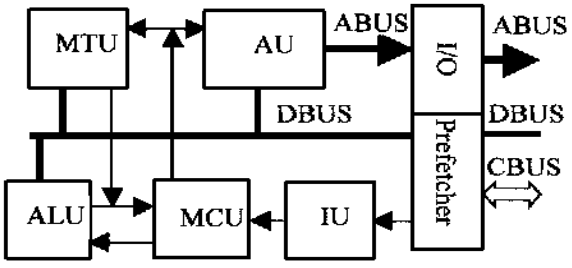

图 9　16 位嵌入式机载微处理器 NCS 体系结构

多任务管理单元 MTU 的数据通路和控制通路: 首先, 被测数据从 DBUS 传送至 MTU 的暂存器; 其次, 微程序控制单元 MCU 的微指令经控制总线 CBUS 送至 MTU 来选择某个测试细胞 cell 进行测试; 再次, 经测试所形成的状态位, 又送回微程序控制单元 MCU, MCU 根据微程序状态位布尔值确定微程序的走向, 从而达到了以细胞测试的布尔值控制微程序转移的目的。

4　仿真结果

16 位机载嵌入式微处理器 NCS 多任务管理单元 MTU 的设计方案已实现了 RTL 级的 VHDL 描述, 经 MENTOR 工具仿真成功。嵌入式微处理器 NCS 在用 CALL 和 JMP 指令通过任务门和 TSS 段进行间接和直接任务切换时指令周期与 intel 公司 16 位微处理器 iAPX[5,6] 指令周期 8 种情形比较见图 10, NCS 平均指令周期是

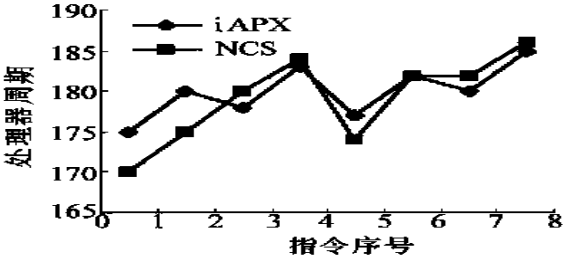

图 10　任务切换时指令周期比较

179.125 拍, 而 iAPX 平均指令周期是 180 拍。微处理器 NCS 任务切换速度高于 iAPX 处理器切换速度。用 MENTOR 工具综合 MTU 面积也符合预期的设计目标, 又将多任务管理单元与微处理器 NCS 的其它部件联调, 微处理器 NCS 运行成功从而验证了多任务管理单元 MTU 正确性与有效性。

5　结束语

多任务管理单元 MTU 是拥有保护模式微处理器不可缺少的组成部分, 它的功能是微处理器中的 ALU 单元所不能替代的。MTU 的实现为操作系统在硬件芯片级实施多用户, 多任务管理提供了硬件环境。多任务管理单元的细胞群结构的仿真成功, 验证了细胞群结构的正确性。这一结构为拥有保护机制的微处理器保护测试电路设计奠定了基础。多任务管理单元方案的提出、设计、实现及仿真的成功使拥有我国自主版权的微处理器设计[7] 又迈出了一步。

参　考　文　献

[1] Andrew V. Survey of advanced microprocessors[M]. New York: Van Nostrand Reinhold, 1991. 48～61.

[2] Heller P, Childs R. Memory protection moves onto 16-bit microprocessor chip[J]. Electronics, 1982, 55(4): 133～137.

[3] Ciminiera L, Valenzano A. Advanced microprocessor architectures[M]. New York: Addison wesley, 1987. 336～361.

[4] Wilkinson B. Computer architecture design and performance[M]. Englewood Cliffs: Prentice Hall, 1991. 90～99.

[5] Intel Corporation. Microprocessor and peripheral handbook [M]. Santa Clara: Intel Corporation. 1988. 347～354.

[6] Intel Corporation. Pentium Family User's Manual[M]. Santa Clara: Intel Corporation, 1994. 101～109.

[7] 任恭海. 32 位嵌入式航空机载 RISC 微处理器的研究及系统设计[D]. 西安: 西北工业大学, 1996.

作者简介:

李树国　男, 1963 年生, 西北工业大学航空微电子中心博士研究生, 主要研究方向: 计算机体系结构、ASIC 系统设计及电子设计自动化 EDA. Tel: (029) 8493967, Email: lisg@amec.nwpu.edu.cn。

高德远　男, 1946 年生, 教授, 博士生导师, 西北工业大学副校长, 主要研究方向: 计算机体系结构, VLSI 系统设计, ASIC 系统设计, 电子设计自动化 EDA, 计算机网络等。Tel: (029) 8492041。

聂培琴　女, 1965 年生, 山东建筑材料工业学院信控系讲师, 主要从事系统软件研制开发工作. Tel: (0531) 7963250-2264。