国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第11卷第4期　1989年12月　Vol.11 No.4

多微机系统VME总线控制器的设计与实现

龚雪春　邹逢兴
（自动控制系）

摘　要　文中给出了一个机械人力反馈依从控制用多微机系统的VME总线控制器的基本结构，给出了采用循环选择(RRS)优先策略裁决器的设计细节，指出了在设计过程中遇到的问题及其解决办法，最后以MSI、SSI给出了总线控制器的具体实现。

关键词　多机系统，裁决器，VME总线，循环选择
分类号　TP36

在多微机系统中，对共享资源访问的控制是最关键的技术之一。共享资源包括共享存储器，公用I/O设备或I/O子系统，公用分时总线等。

多微机系统发展迅速，大量地采用新技术，结构不断变化。尽管如此，由于总线型多微机系统结构简单，灵活性好，易扩充，因而它仍是多微机系统采用最多的形式。本文作者在设计用于机器人力反馈依从控制的多微机系统时便采用了图1所示的总线结构。

从图1可以看出，多个处理单元和I/O处理机都连在VME总线上，而VME总线在任何时刻都只能为一个主设备（此处为PE或IOP）所占有。当某一时刻多个主设备请求VME总线服务时，需要有一个控制机构，登记各主设备的访问请求，按一定的优先次

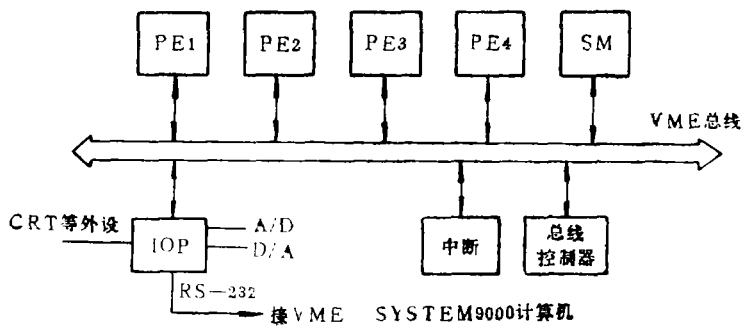

图 1　基于VME总线的多微机系统

PE1为由MC68020，MC68881，MC68851和1MRAM组成的处理单元
SM为－4MB存储器
IOP为由MC68000组成的I/O处理机，可带多个

学术论文　1988年12月6日收稿

序将总线分配给各主设备使用。这一控制机构便是图1中的总线控制器。

1 VME总线控制器的基本结构

根据VME总线规范，总线控制器包含四大部分：系统时钟驱动器，总线裁决器，系统复位逻辑和总线监视计时器。其方框图如图2所示。

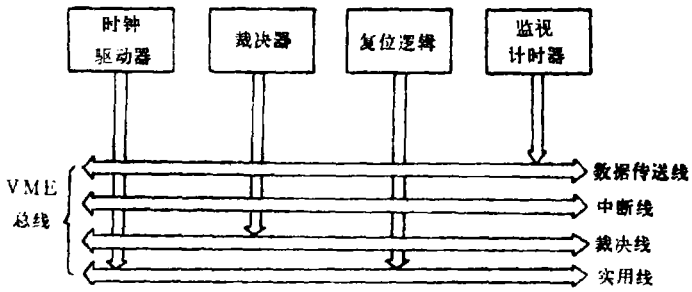

图2 VME 总线控制器的基本结构

图2中，将VME总线分成4组线。总线控制器涉及到的线包括：4条总线请求线BR_0^*、BR_1^*、BR_2^*、BR_3^*，4条总线允许线BG_0^*、BG_1^*、BG_2^*、BG_3^*，1条总线忙线BBSY*，系统时钟线SYSCLK，系统失效线SYSFAIL*，交流失效线ACFAIL*，系统复位线SYSRESET*，数据选通线DSO*、DS1*，总线错误线BERR*。它们分别来自VME总线4组线中除中断线组外的其它3组。信号名右上角标星号（*）的表示低电平有效。

2 VME总线裁决器

总线裁决是总线控制器要实现的一个重要功能。

2.1 裁决算法的选择

目前可用的裁决算法很多，如先来先服务、静态优先级、动态优先级、时间片轮转、循环选择等。究竟选用哪一种裁决算法呢？裁决算法各有其特点，无法找到一个统一的标准来衡量。通常视具体应用场合来定。

若把总线看作一个服务员，而把要访问该总线的主设备看成顾客，处理机使用总线当作服务员为顾客服务。设主设备因访问请求未被批准而加入的等待队列的队长为L，主设备的平均等待时间（包括延误时间和访问总线的时间）为W。由Little公式可知：$L=\lambda W$，其中λ为主设备访问请求的到达率。

有的主设备有固定百分比的占用总线的要求，否则要造成信息丢失等严重后果。为避免这种现象发生，要求访问请求率λ较大，或要求每次占用总线的时间（含在W中）较长。这时，关键要看系统中这种设备所占比率。若比率低于100%，可采用静态优先级方法，将这类设备的优先级定为最高级。

用于机械人力反馈依从控制的多微机系统，各主设备不要求固定的总线时间。如果总线已被某主设备占用，其它设备仅是等待而已。考虑到负载均衡性、公平性和VME

总线规范所提供的三种裁决算法：单级菊花链(SGL)、固定优先级(PRI)和循环选择(RRS)，作者选择了RRS法。

2.2 逻辑表达式

设有：主设备N个；请求信号分别为$BR_0, BR_1, \cdots, BR_{N-1}$；响应信号分别为$BG_0, BG_1, \cdots, BG_{N-1}$。所谓RRS裁决就是：若第$i$号主设备刚服务完，则在下一个总线周期里，处理机$i+1$(modN)的优先级最高，处理机$i+2$(modN)次之，处理机$i+N$(modN)，即处理机$i$成为优先级最低的。将$N$个主设备的总线允许信号的次态$BG_i'$表示为其现态$BG_i$与请求$BR_i$($i=0,1,\cdots,N-1$)的函数：

$$\begin{aligned}BG_i' = BR_i(&BG_{i-1} + BG_{i-2}\cdot\overline{BR_{i-1}} + BG_{i-3}\cdot\overline{BR_{i-2}}\cdot\overline{BR_{i-1}} + \cdots \\ &BG_{i-N+1}\cdot\overline{BR_{i-N+2}}\cdot\overline{BR_{i-N+3}}\cdot\cdots\cdot\overline{BR_{i-1}} + \\ &BG_{i-N}\cdot\overline{BR_{i-N+1}}\cdot\overline{BR_{i-N+2}}\cdot\cdots\cdot\overline{BR_{i-1}}) \quad (\text{modN})\end{aligned} \tag{1}$$

其中$i=0,1,\cdots,N-1$.

(1)式可以表示成如下的压缩形式：

$$BG_i' = BR_i\cdot\sum_{j=1}^{N}\left[BG_{i-j}\left(\prod_{k=1}^{j-1}\overline{BR_{i-k}}\right)\right] \quad (\text{modN}) \tag{2}$$

由于VME总线只有4根请求线和4根允许线，下面只讨论$N=4$的情况。据(1)式，有：

$$\begin{cases}BG_0' = BR_0(BG_3 + BG_2\overline{BR_3} + BG_1\overline{BR_2}\ \overline{BR_3} + BG_0\overline{BR_1}\ \overline{BR_2}\ \overline{BR_3}) \\ BG_1' = BR_1(BG_0 + BG_3\overline{BR_0} + BG_2\overline{BR_3}\ \overline{BR_0} + BG_1\overline{BR_2}\ \overline{BR_3}\ \overline{BR_0}) \\ BG_2' = BR_2(BG_1 + BG_0\overline{BR_1} + BG_3\overline{BR_0}\ \overline{BR_1} + BG_2\overline{BR_3}\ \overline{BR_0}\ \overline{BR_1}) \\ BG_3' = BR_3(BG_2 + BG_1\overline{BR_2} + BG_0\overline{BR_1}\ \overline{BR_2} + BG_3\overline{BR_0}\ \overline{BR_1}\ \overline{BR_2})\end{cases} \tag{3}$$

如果(3)式直接用于实现电路，则存在下面三个问题：

(1) 当系统中没有任何主设备提出访总申请时，由于BR_0～BR_3全为"0"，从(3)式可知，过一个裁决周期后BG_0'～BG_3'全为"0"。之后，即使有主设备提出访总申请，BG_0'～BG_3'都将一直保持为"0"。

(2) 当系统初始化时，可能出现一个无效状态。即当系统加电时，可能出现不止一个保持BG_i状态的触发器被置成"1"或者所有BG_i全部为"0"的情形。

(3) 由于VME总线规范中4根请求线和4根允许线传送低电平信号时表示有效，因此，要进行正负逻辑变换。

解决第一个问题的方法可能有多种，但最简单的方法是在(3)式的各子式BG_i'的右边加入

$$BG_i\overline{BR_{i-1}}\ \overline{BR_{i-2}}\ \overline{BR_{i-3}}\ \overline{BR_i} \quad (\text{mod4}) \tag{4}$$

其中$i=0,1,2,3$. 这时(3)式变成

$$\begin{cases}BG_0' = BR_0(BG_3 + BG_2\overline{BR_3} + BG_1\overline{BR_2}\ \overline{BR_3}) + BG_0\overline{BR_1}\ \overline{BR_2}\ \overline{BR_3} \\ BG_1' = BR_1(BG_0 + BG_3\overline{BR_0} + BG_2\overline{BR_3}\ \overline{BR_0}) + BG_1\overline{BR_0}\ \overline{BR_2}\ \overline{BR_3} \\ BG_2' = BR_2(BG_1 + BG_0\overline{BR_1} + BG_3\overline{BR_0}\ \overline{BR_1}) + BG_2\overline{BR_0}\ \overline{BR_1}\ \overline{BR_3} \\ BG_3' = BR_3(BG_2 + BG_1\overline{BR_2} + BG_0\overline{BR_1}\ \overline{BR_2}) + BG_3\overline{BR_0}\ \overline{BR_1}\ \overline{BR_2}\end{cases} \tag{5}$$

增加项(4)意味着在没有任何请求的情况下，裁决器保持原状态不变。

第二个问题只要将(5)式中的BG_i代之以A_i即可，这里：

$$\begin{cases} A_0 = BG_0 \\ A_1 = \overline{BG_0} BG_1 \\ A_2 = \overline{BG_0\ BG_1} BG_2 \\ A_3 = BG_0\ \overline{BG_1}\ \overline{BG_2} \end{cases} \tag{6}$$

易见：若系统处于有效状态，即仅有一个BG_i为"1"时，则A_i等于BG_i. 若不止一个BG_i为"1"，则只有与所有为"1"的BG_i中最小下标i相对应的A_i才为"1"，其余A_i皆为"0"。若所有BG_i全为"0"，则$A_3=1$. 从而确保了有且仅有一个"1"的要求。系统启动完毕后，裁决器即进入到一个有效状态，此后一直有效地工作着。

经过上面两次修改后，裁决器的输出函数变成如下形式：

$$\begin{aligned} BG_0' &= BR_0(A_3 + A_2\overline{BR_3} + A_1\overline{BR_2}\ \overline{BR_3}) + A_0\overline{BR_1}\ \overline{BR_2}\ \overline{BR_3} \\ BG_1' &= BR_1(A_0 + A_3\overline{BR_0} + A_2\overline{BR_3}\ \overline{BR_0}) + A_1\overline{BR_0}\ \overline{BR_2}\ \overline{BR_3} \\ BG_2' &= BR_2(A_1 + A_0\overline{BR_1} + A_3\overline{BR_0}\ \overline{BR_1}) + A_2\overline{BR_0}\ \overline{BR_1}\ \overline{BR_3} \\ BG_3' &= BR_3(A_2 + A_1\overline{BR_2} + A_0\overline{BR_1}\ \overline{BR_2}) + A_3\overline{BR_0}\ \overline{BR_1}\ \overline{BR_2} \end{aligned} \tag{7}$$

当(7)式变成低电平有效时，得：

$$\begin{aligned} BG_0^{*\prime} &= (BR_0^* + A_3^*)(BR_0^* + A_2^* + \overline{BR_3^*})(BR_0^* + A_1^* + \overline{BR_2^*} + \overline{BR_3^*}) \\ BG_1^{*\prime} &= (BR_1^* + A_0^*)(BR_1^* + A_3^* + \overline{BR_0^*})(BR_1^* + A_2^* + \overline{BR_3^*} + \overline{BR_0^*}) \\ BG_2^{*\prime} &= (BR_2^* + A_1^*)(BR_2^* + A_0^* + \overline{BR_1^*})(BR_2^* + A_3^* + \overline{BR_0^*} + \overline{BR_1^*}) \\ BG_3^{*\prime} &= (BR_3^* + A_2^*)(BR_3^* + A_1^* + \overline{BR_2^*})(BR_3^* + A_0^* + \overline{BR_3^*} + \overline{BR_2^*}) \\ &\cdot (A_0^* + \overline{BR_1^*} + \overline{BR_2^*} + \overline{BR_3^*}) \\ &\cdot (A_1^* + \overline{BR_0^*} + \overline{BR_2^*} + \overline{BR_3^*}) \\ &\cdot (A_2^* + \overline{BR_0^*} + \overline{BR_1^*} + \overline{BR_3^*}) \\ &\cdot (A_3^* + \overline{BR_0^*} + \overline{BR_1^*} + \overline{BR_2^*}) \end{aligned} \tag{8}$$

其中A_0^*, A_1^*, A_2^*和A_3^*的表达式如下：

$$\begin{cases} A_0^* = BG_0^* \\ A_1^* = \overline{BG_0^*} + BG_1^* \\ A_2^* = \overline{BG_0^*} + \overline{BG_1^*} + BG_2^* \\ A_3^* = \overline{BG_0^*} + \overline{BG_1^*} + \overline{BG_2^*} \end{cases} \tag{9}$$

2.3 电路实现

(8)式可用与非门加上D触发器来实现。若选择具有互补输出的4D触发器LS175来实现，则将其4个D端对应于(8)式中各式的右端，Q端对应于BG_0～BG_3即可。

裁决器实现的另一个问题是何时进行裁决，即LS175的触发脉冲何时加入。由于裁决器为事件驱动，所以不能直接取自SYSCLK. 据VME总线规范，处理机每次访问总线都有下列过程：提出访问申请，获准访问总线，升起总线忙信号BBSY*，完成总线数据传送，最后释放BBSY*. 因此，裁决器时钟Car为BBSY*与BR_0^*～BR_3^*的函数。最后的

实现如图 3 所示。将Car连到LS175的CK端即可。

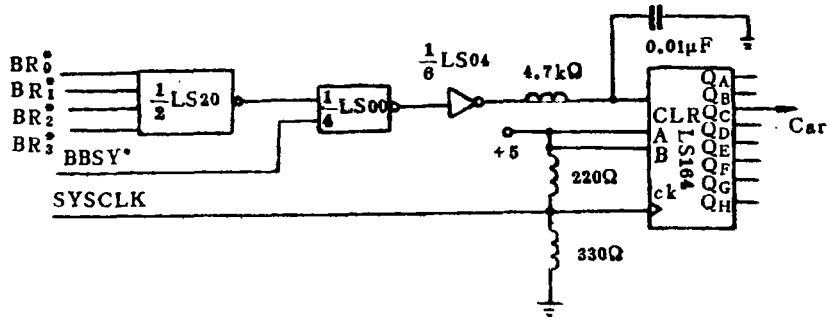

图 3　裁决器时钟信号Car的获取

3　总线控制器的其宅模块

由图 2 知，除了裁决器外，还设计了时钟发生器，复位逻辑和总线监视计时器。

3.1　时钟发生器

VME总线规范指出：总线时钟必须为16MHZ. 为此，选用一个16MHZ的晶振和其它辅助电路实现，因电路比较简单，此处从略。

3.2　系统复位逻辑

系统复位包括人工复位、电源故障时的复位和刚加电时的复位。人工（手动）复位比较简单，只要接一个开关和相应的去抖电路，因 SYSRESET*为集电极 开路信号，故输出接在LS38上。当检测到交流电发生故 障时开 始 计 时，经 过 约 2ms 之后产生一个复位信号。另外，系统通电时，为使系统进入一已知的非随机初始状态，应能使系统自动复位，这一复位信号至少要维持200ms. 这部分的电路实现选用了一片 NE555，两片LS393计数器和一些与非门、电阻电容。

3.3　总线监视计时器

在总线控制器检测到有数据选通信号DSO* 或DSI* 或二者都有效后，便 启动 计时

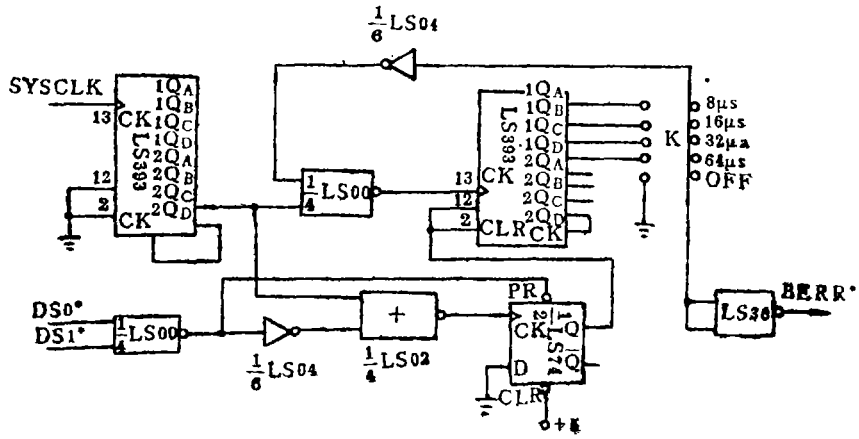

图 4　总线监视计时器

器。如果过了较长的一段时间后，数据选通信号仍处在有效状态，则认为此次总线数据交换已发生错误，并发出BERR*信号。这段较长的时间是可选的，即8μs, 16μs, 32μs, 64μs或∞(根本不产生BERR*)。在最后的情况下，认为从设备都能发出BERR*信号，电路如图4所示。

4 结 论

按上述方法设计的总线控制器已安装调试完毕，不久将用于机械人力反馈依从控制器。该电路结构简单，易于制作。

参 考 文 献

[1] Motorola. The VME bus specification,1985

[2] Merchant S S. The design and performance analysis of an arbiter for a multiprocessor shared-memory system,LIDS-TH-1396,M.I.T.,1984

[3] Paker Y.Multi-microprocessor systems,Academic Press,1983

[4] Kleinrock L. Queueing systems,John Wiley & Sons,1976, 2

The Design and the Implementation of the Bus-controller in the VME-based Multimicroprocessor System

Gong Xuechun Zou Fengxing

(Department of Automatic Control)

Abstract

First a block diagram of the bus-controller in the VME-based multiprocessor system which is to be used for force compliance control of robot is drawn in the paper. Then, the design of the Round-Robin-Select (RRS) priority policy arbiter is given in detail. The problems met in the design process are analysed and solved. Implementation of the controller using only MSI and SSI is described

Key Words: mutiprocessor system; arbiter, VME bus, round-robin-select