

MCU 存储器双倍速率控制技术研究

郑杰辉

(厦门海洋学院 信息技术系, 福建 厦门 361000)

摘要: 由于传统技术缺少控制指令设置步骤, 导致控制精度较低, 为了解决该问题, 提出了基于拟合的存储双倍速率速率精确控制技术; 根据速率计算公式, 实行并行总线来获取双倍存储速率, 采用分时加载方法对双倍存储速率数据进行拟合, 方便控制指令的设置; 分析 MCU 存储器控制性能, 并对双倍存储速率控制设置指令, 根据该控制指令分别对分时存储和空存储操作进行精确校正, 实现 MCU 存储器双倍速率的有效控制; 由实验结果表明, 该技术控制精度最高可达到 98%, 可以达到实际存储过程中对于双倍速率精准控制标准。

关键词: MCU 存储器; 双倍速率; 控制指令; 拟合; 并行总线; 分时加载

Research on Double Rate Control Technology of MCU Memory

Zheng Jiehui

(Information Technology Department of Xiamen Institute of Oceanology, Xiamen 361000, China)

Abstract: Due to the lack of control instruction setting steps in traditional technology, the control accuracy is relatively low. In order to solve this problem, a double rate rate accurate control technology based on fitting is proposed. According to the rate formula, parallel bus is used to get double storage rate. Time sharing loading method is used to fit double storage rate data to facilitate the setting of control instructions. The control performance of MCU memory is analyzed, and the instruction set of double storage rate control is set. According to the control instruction, the time division storage and empty storage operation are corrected accurately, and the MCU memory's double rate is effectively controlled. The experimental results show that the precision of the technology can reach up to 98%, and the precision control standard for double rate can be achieved in the actual storage process.

Keywords: MCU memory; double rate; control instruction; fitting; parallel bus; time sharing loading

0 引言

MCU 存储器即为微控制单元存储器, 该存储器芯片不但可以集成计算机中的 CPU, 也可以集成与外部直接通信的多种接口, 比如串行外设接口 spi。MCU 存储器是一种在外部同步控制条件下实现数据读写与存储的, 需要不断刷新地址才能允许用户访问^[1]。由于各种处理器工作频率加快, 使存储器读写与存储性能成为了约束存储器性能的主要原因^[2]。对于人们应用需求的不断变化和协议不断升级, MCU 存储器的双倍存储速率成为了当下最重要的存储工具, 利用该存储器不但能够提高存储速率, 还能减少人工劳动力, 但同时对于速率的不精准控制问题也逐渐凸显出来。采用传统控制技术存在控制精度低的问题, 不能满足实际存储过程中对于双倍速率精准控制的需求, 为此, 提出了基于拟合的存储双倍速率速率精确控制技术。

1 基于拟合的存储双倍速率速率精确控制技术

对于 MCU 存储器双倍速率精确控制设计思想如下:

首先采用分时加载存储技术对 MCU 存储器的存储数据进行拟合, 然后设置 MCU 存储器控制性能, 并对双倍存储速率控制下达指令, 最后在 Xscale 处理器上对存储速率进

行精确校正^[3]。

1.1 存储数据拟合

对 K9K8G08UOA 型号的 MCU 存储器芯片进行操作时, 编程会消耗大量时间, 若想控制存储速率需从两个方面展开研究^[4]。一是对多个芯片并行的存储器进行数据存储, 二是将编程所消耗的时间充分利用, 采用分时加载存储技术计算存储速度:

$$V_{存} = \frac{m_1 \times m_2 \times n_1}{t \times n_2} \quad (1)$$

公式(1)中: m_1 为存储的数据量; m_2 为并行操作芯片数量; n_1 为流水级数; n_2 为分时节数; t 为加载时间, 由公式(1)可获取 MCU 存储器单倍存储速率, 将该速率乘以 2, 即为双倍存储速率。

根据上述获取的双倍存储速率, 使用主频率为 500 MHz 的 IX P2400 的微引擎所发送的存储指令作为一个时钟周期, 可满足测试流量对存储速率精度的需求^[5]。数据帧结构包括帧首部、数据帧和循环冗余校验三部分, 其中帧首部是由一个前同步码 7 字节和帧开始符 1 字节、源物理 6 字节、目的物理 6 字节组成的; 数据帧是由长度为 50—1500 字节组成的; 循环冗余校验是由长度为 4 字节校验结果组成的^[6]。不同帧数之间的连接至少需要 12 字节的信息数据, 分时加载存储技术的实现关键在于能够使用拟合方法准确推算出数据存储速率。因此, 需对存储器中的并行

收稿日期: 2018-07-23; 修回日期: 2018-09-07。

作者简介: 郑杰辉(1982-), 男, 福建龙海人, 讲师, 本科, 主要从事网络数据分析方向的研究。

总线进行分析,采用分时加载存储技术对 MCU 存储器的存储数据进行拟合。

1.1.1 并行总线

在 MCU 存储器中实行并行总线方式,能够拓宽数据总线的宽度,进而方便海量数据的存储。该拓宽方式是将 m_2 片的并行操作芯片组成一个长为 m_2 , 宽为 8 的存储模块,该模块用来存储具有相同地址的信号。将这个模块当作一个完整的存储器模式完成操作,并对不同数据进行加载^[7]。经过加载后的数据存储量为原始单片芯片存储的 m_2 倍,存储速度也是原始数据存储速度的 m_2 倍。

根据上述并行原理,将 2 片芯片组成一个存储模块,该模块可存储来自同一发送源的同址信号数据,经过数据加载,存储器的容量为原始容量的 2 倍,那么存储速率也为原始数据速率速度的 2 倍。

1.1.2 分时加载存储技术

根据上述存储器中的并行总线方式,拓宽数据总线宽度,并采用分时加载存储技术对数据加载和编程进行拟合,具体拟合过程如下所示:

同时加载第一组的每一个芯片上的数据,使得经过加载后的数据能够同时进入到编程阶段;再对第二组芯片上的数据同时加载,使得第二组经过加载后的数据也能够同时进入到编程阶段;之后再对第三组芯片上的数据进行加载并自动编程,如此循环,直至完成最后一组的数据加载^[8]。当最后一组数据完成加载时,需及时判断第一组编程是否结束,如果没有结束,那么存储过程需持续进行;如果结束,需立即再次加载并自动编程,形成分时加载存储的拟合状态。存储模块数量就是分时加载存储的级数,对于该级数的确定需先充分考虑存储器的连续加载方式,当最后一个模块完成数据加载时,第一组的编程也呈现结束等待状态,那么对于第一组数据的再次加载并自动编程就无需等待直接进行^[9]。如果第一组芯片上的数据加载与编程的总时间超过最后一组数据加载时间,那么第一组直接进行编程等待阶段,不能实现连续性的分时加载,因此,为了实现连续性的分时加载存储数据的拟合,需规定分时加载存储时间:若要满足连续性的分时加载存储拟合要求,分时加载等级所耗费的时间应大于各个模块的编程时间,只有这样才能满足连续存储要求^[10]。以 8 级分时为例,根据上述内容设计连续性的分时加载数据拟合流程,如图 1 所示。

由图 1 可知:采用分时加载存储技术节省了大量编程时间,并极大的提高了存储速度,有效实现了分时加载存储数据的拟合,符合存储数据的函数关系,方便控制指令的下达。

1.2 控制指令设置

针对控制指令的设置需先对 MCU 存储器控制性能进行规定:

配置突发数据长度成 2 的倍数增加,并对物质数字识

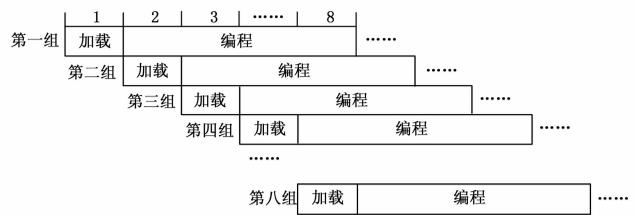


图 1 分时加载数据拟合流程

别号码等待时间按照双倍速率同步动态随机存储要求进行设置。模式寄存器可用 LOAD_MR 来表示;自动刷新可用 AUTO_REFRESH 来表示;在接口处采用 100 MHz 的双倍数据存储速率,提供物质数字识别号码之间的零时钟偏移信息^[11]。

根据控制性能,对双倍存储速率控制下达指令:

1) 按照磁共振波普学原理设置数据寄存模式,方便确定存储器的地址延迟和突发长度等模式,激活脚本程序代码指令所对应的地址,同时将行地址输入进去,再通过一一对应选择规则将列地址输入进去。将行、列地址数据进行读出或写入,待读写操作完成之后,使用预编译头文件命令终止操作。

2) 设置空操作指令为: $u_and=00001$;

3) 根据空存储操作指令,设置数据寄存指令: $u_and=00001$; $u_addr [2: 0] =$ 数据突发长度 ($01 \cdots > 2. 01 \cdots > 4. 01 \cdots > 8$), $u_addr [3: 2] =$ 地址延迟 ($01 \cdots > 2. 01 \cdots > 2. 50 \cdots > 3$), $u_addr [0: 0] = 0$;

4) 程序将自动刷新计数器在管理队列中的地址,按照 14.61 微秒的速度进行刷新;

5) 设置自动预充写: $u_and=00101$, 突发长度停止增长: $u_and=10000$, 列地址为: $S8=1$;

6) 设置自动预充读: $u_and=00102$, 突发长度停止增长: $u_and=10000$, 列地址为: $S8=2$ 。

根据该控制指令分别对分时存储和空存储操作进行精确校正。

1.3 精确校正

为了降低分时存储和空存储操作在存储速率上所带来的误差,需在 Xscale 处理器上对存储速率进行精确校正。

1.3.1 分时存储操作校正

分时存储操作是以存储器中的并行总线方式拓宽数据总线宽度的,并以此为基础实现数据加载和编程的拟合,若要满足连续性的分时加载存储拟合要求,分时加载等级所耗费的时间应大于各个模块的编程时间,只有这样才能满足连续存储要求。但是该过程存在网络包较长,容易造成数据存储速率控制结果出现误差,为了降低该部分的误差,需对每个网络包传递之间产生的时延误差进行校正。

选择一个分时存储操作循环作为数据包与数据包之间的传输时延,设 Total+1 表示存储速率降低、Total-1 表示存储速率上升。以 8 个分时存储操作循环为例,设定存

储速率误差为 S , 速率上升或降低情况如下所示:

- 当 $S \geq 8$ 时, 存储速率表现为 $Total+8$;
- 当 $1 \leq S < 8$ 时, 存储速率表现为 $Total+1$;
- 当 $-1 \leq S \leq 0$ 时, 存储速率表现为 $Total-1$;
- 当 $S \leq -8$ 时, 存储速率表现为 $Total-8$;

根据存储速率表现形式, 对最后一个模块完成数据加载以及第一组数据再次加载并自动编程过程进行校正, 可缩小该过程中数据包与数据包之间的传输时延, 从而降低数据存储速率出现的误差, 从一定程度上弥补了分时存储操作存在误差的缺点。

1.3.2 空存储操作校正

空操作指的是执行程序无法找到请求操作指令时, 需定位到控制指令中来执行, 利用该指令有效控制存储速率。由于在执行空操作时受到干扰信号影响, 导致数据存储速率控制结果出现误差, 为了降低该部分的误差, 需对干扰环境中产生的误差进行精确校正。

采用二分法实现在干扰信号影响下的数据存储双倍速率的精确校正, 具体流程如下所示:

设 H_1 为空操作最大循环次数; H_2 为空操作最小循环次数; $V_{\#}$ 为数据存储速率, 单位为 bps; S 为实际操作允许的误差数值, 统计单位时间内的并行微引擎的数据包存储数量, 即为存储器实际存储的速率。空操作平均循环次数为:

$$H = \frac{(H_1 + H_2)}{2} \quad (2)$$

实际存储速率测试函数为:

$$f = V_{\#} - t(H) \quad (3)$$

计算空操作平均循环次数 H 大小, 将 H 代入公式 (3) 中, 判断函数数值大小。如果函数数值大于实际操作允许的误差 S , 那么空操作最小循环次数 H_2 与平均循环次数 H 大小一致; 如果函数数值小于等于实际操作允许的误差 S , 那么空操作最大循环次数 H_1 与平均循环次数 H 大小一致。根据该操作调整实际存储速率大小, 能够减小干扰信号带来的误差, 及时校正空存储操作对存储速率控制所产生的误差问题。

计算存储速度, 获取双倍存储速率。实行并行总线方式, 拓宽数据总线宽度, 方便海量数据存储。采用分时加载存储技术对数据加载和编程进行拟合, 有效实现了分时加载存储数据的拟合, 符合存储数据的函数关系, 方便控制指令的下达。设置 MCU 存储器控制性能, 并对双倍存储速率控制下达指令, 根据该控制指令分别对分时存储和空存储操作进行精确校正, 进而实现 MCU 存储器双倍速率的有效控制。

2 实验

针对基于拟合的存储双倍速率精确控制技术有效性设计了实验, 实验环境如表 1 所示。

在 Xscale 处理器上设定预期存储速率, 通过测试实际存储速率来分析控制精度大小。将控制指令混乱、传输延迟和信号干扰情况作为指标对 MCU 存储器双倍速率精确控制进行实验验证。

表 1 实验环境设置

参量	指标
处理器	IXP2400 型号的处理器
CPU	Intel R 2.55Hz
内存	512M
硬盘	Pentium R IV LGA 775
散热	散热风扇
温度	0—50 摄氏度
湿度	10—80%
测试软件	Windows 网络测试系统

2.1 控制指令混乱

控制指令也被称为转移指令, 程序接收到某个执行指令时, 通常会出现几种不同的结果, 此时程序必须选择一条指令执行, 根据不同指令要求改变程序执行结果。一旦在 MCU 存储器中出现控制指令混乱现象, 那么存储速率的控制也将受到影响, 为了使验证结果更加可靠, 将传统控制技术与基于拟合的控制技术进行对比, 结果如图 2 所示。

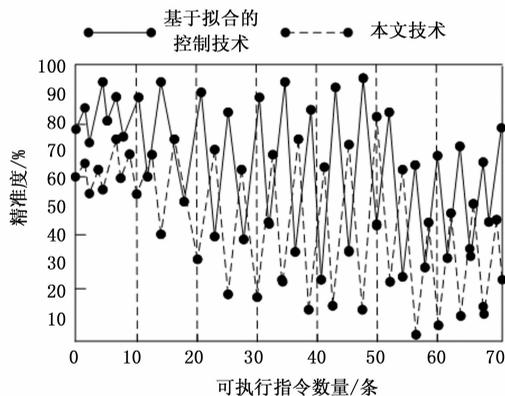


图 2 两种技术在控制指令混乱情况下精度对比结果

由图 2 可知: 当可执行指令数量为 10 时, 传统控制技术精度为 56%, 当可执行指令数量为 56 时, 传统控制技术精度为 2%; 当可执行指令数量为 48 时, 基于拟合的控制技术精度达到最高为 98%, 当可执行指令数量为 41 时, 基于拟合的控制技术精度达到最低为 25%。由此可知, 控制指令混乱对存储双倍速率控制准确性具有严重影响, 尤其是对传统技术影响最大。由于基于拟合控制技术中含有空操作和分时存储操作, 能够改善指令混乱问题, 因此, 控制指令混乱对基于拟合的控制技术影响效果较小。

2.2 传输延迟

在存储过程中出现传输延迟, 容易导致信号响应慢、数据接收不及时。一旦在 MCU 存储器中出现传输延迟现象, 那么存储速率的控制也将受到影响, 为了使验证结果

更加可靠, 将传统控制技术与基于拟合控制技术在传输出现延迟情况下, 对控制精准度对比结果进行分析, 如表 2 所示。

表 2 两种控制技术在传输延迟下精准度对比结果

实验次数	传统控制技术				基于拟合控制技术			
	信号响应	数据接收	存储过程	精准度%	信号响应	数据接收	存储过程	精准度%
1	—	正常	正常	59	—	正常	正常	69
2	正常	—	正常	86	正常	—	正常	94
3	正常	正常	—	70	正常	正常	—	91
4	—	—	正常	55	—	—	正常	79
5	—	正常	—	60	—	正常	—	71
6	正常	—	—	40	正常	—	—	52

表 2 中“—”代表出现延迟, 对比两种控制技术发现, 当信号响应出现延迟时, 传统控制技术比基于拟合控制技术精准度要低 10%; 当数据接收出现延迟时控制精准度最小, 基于拟合控制技术比传统控制技术评估精准度高 8%; 当存储过程出现延迟时控制精准度最大, 传统控制技术比基于拟合控制技术精准度要低 21%。由此可知, 传输延迟对传统控制技术影响效果较大, 使用拟合控制技术精准度较高。

2.3 信号干扰

在实际生活中, 出现信号干扰问题严重影响存储器的数据存储, 导致存储速率受影响, 针对信号干扰对技术准确性影响需将传统控制技术与基于拟合控制技术进行对比, 结果如图 3 所示。

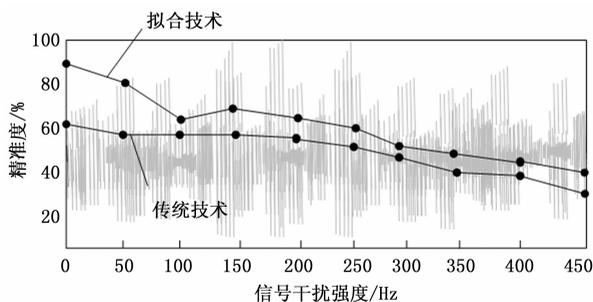


图 3 两种技术在信号干扰情况下控制精准度对比结果

由图 3 可知: 初始两种控制技术精准度分别为 60% 和 90%, 当信号干扰强度增加到 50 Hz 时, 传统技术控制精准度与拟合技术精准度持续下降, 分别达到 55% 和 78%。随着信号干扰强度增加, 两种控制技术精准度呈折线形式下降, 最终降为 38% 和 45%。但在实际存储过程中, 存储器最多能抵抗 300 Hz 强度的信号干扰, 为此, 在该条件下, 采用拟合技术能够使控制精准度最低达到 60%。由此可知, 在实际生活中, 信号干扰对传统控制技术影响效果较大, 使用拟合控制技术精准度较高。

根据上述实验内容, 可得出实验结论:

1) 针对控制指令混乱问题, 采用拟合控制技术, 最高与最低控制精准度分别为 98% 和 25%; 而采用传统控制技术, 最高与最低控制精准度分别为 82% 和 2%。

2) 针对传输延迟问题, 采用拟合技术比传统技术在信号响应、数据接收与存储延迟过程中控制精准度要高。

3) 针对信号干扰问题, 两种控制技术精准度呈折线形式持续下降, 但采用拟合技术控制精准度依然比传统控制精准度高。

综合上述内容可知, 基于拟合的存储双倍速率精确控制技术使用效果较好, 精准度较高。

3 结束语

针对 MCU 存储器存储双倍速率控制问题, 提出了基于拟合的存储双倍速率精确控制技术, 根据对该控制技术的深入研究, 解决了控制指令混乱、传输延迟和信号干扰等问题。使用该技术的创新点是采用分时加载存储方法对数据加载和编程进行拟合, 方便控制指令的下达, 并根据 MCU 存储器控制性能, 完成控制指令精确校正, 从而实现 MCU 存储器双倍速率的有效控制。虽然该技术在实验室中得到了验证, 但是相对外界恶劣条件下依然可以保持这样性能还有待考察, 因此, 在下一步工作中应进行外界现场实验, 保证该技术的稳定性能。

参考文献:

- [1] 江 杰, 岂伟楠. 四旋翼飞行器建模与 PID 控制器设计 [J]. 电子设计工程, 2017, 21 (23): 147-150.
- [2] 蔡伯峰, 王宜怀. 基于 ARM Cortex-M0+ KL 系列 MCU 的汇编构件设计 [J]. 现代电子技术, 2018, 21 (1): 38-42.
- [3] 郭晓丹, 卢 欣, 孙 哲, 等. 基于双核 MCU 并行算法的高频双 Buck 逆变器 [J]. 电力电子技术, 2017, 18 (12): 33-35.
- [4] 韩存武, 常舒瑞, 刁 奇, 等. 多时滞无线通信网络的功率和速率最优跟踪 [J]. 高技术通讯, 2017, 27 (4): 303-309.
- [5] 付建民, 张伯伦, 陈国明, 等. 液位控制系统干预下小孔全过程泄漏速率模型 [J]. 化工学报, 2017, 68 (1): 473-480.
- [6] 张 亮. 基于 MCU 和 FPGA 双处理器的感应加热电源设计 [J]. 电力电子技术, 2017, 15 (6): 92-94.
- [7] Goyanes A, 王 盈. 采用增材制造技术制备 3D 打印片, 实现药物释放速率的控制 [J]. 中国医药工业杂志, 2017, 16 (8): 1154-1154.
- [8] 张海龙, 许 江, 大久保诚介, 等. 基于应力归还控制的岩石荷载速率依存性研究 [J]. 岩石力学与工程学报, 2017, 36 (1): 93-106.
- [9] 宁 欣, 韩邦成, 房建成. 基于干扰观测器的双框架变速率控制力矩陀螺解耦控制 [J]. 机械工程学报, 2017, 53 (10): 52-59.
- [10] 张文倩, 刘玉岭, 王辰伟, 等. 无氧化剂条件下铜钴 CMP 去除速率的控制 [J]. 微纳电子技术, 2018, 23 (1): 57-62.
- [11] 王兰豪, 贾 瑶, 柴天佑. 再磨过程的泵池液位和给矿压力双速率区间控制 [J]. 自动化学报, 2017, 43 (6): 993-1006.