

一种缩短共享存储访问时延的优化仲裁技术*

关娜, 李康, 马佩军, 武颖奇

(西安电子科技大学 微电子学院 宽禁带半导体材料与器件重点实验室, 西安 710071)

摘要: 提出一种提高访问性能的优先级仲裁策略, 按照不同类型的内存访问优先级进行分层仲裁, 并通过隐藏 bank 预充电时延提高了内存访问效率。本方法应用于网络处理器 (XD-NP) 的可配置 SDRAM 控制器的设计中, 并在 FPGA 平台上进行了验证, 结果表明, 采用延时隐藏策略的 SDRAM 控制器性能提升最大可达 40% 以上, 改善明显。

关键词: 多处理器片上系统; 优先级仲裁; 内存访问; 时延隐藏

中图分类号: TP391 文献标志码: A 文章编号: 1001-3695(2010)04-1391-03

doi: 10.3969/j.issn.1001-3695.2010.04.050

Optimization arbiter technique of reducing latency in sharing memory access SOC

GUAN Na, LI Kang, MA Pei-jun, WU Ying-qi

(State Key Laboratory of Wide Band-Gap Semiconductor Materials & Devices, School of Microelectronics, Xidian University, Xi'an 710071, China)

Abstract: This paper presented an optimization stratified arbitration policy to enhance the performance of SOC, accomplished arbitration according to different kinds of priority level, and improved memory access efficiency by using time-hiding techniques. This method was applied to configurable SDRAM in network processor and FPGA platform. The validation results show the speed and performance of memory access can be improved significantly up to 40%.

Key words: multi-processor SOC; priority arbitration; memory access; time-hiding

因特网的迅速发展和推广应用使人们对它提出不断增长带宽和复杂服务的需求。为适应这种不断发展的网络技术, 出现了网络处理器这种新的微处理器。网络处理器是一种专用于网络系统的微处理器, 它使得网络系统能够具备高性能和灵活性, 网络处理器的出现为网络系统构建了一个硬件平台, 能够通过软件的升级以适合不断增长的功能需求, 为网络技术的发展提供了开放的舞台, 因而具有十分重要的意义。

网络处理器系统是多 IP 核系统, 每个 IP 核负责一个或几个接口的分组处理, 对大量实时数据读取及对路由表不断的访问涉及到多 IP 核共享内存问题, 在尽可能正确及时处理数据的要求下, 必须通过仲裁器进行内存访问, 同时仲裁器的仲裁效率将直接影响网络数据包的处理速度, 从而影响整个网络处理系统的工作效率。

目前存在的内存访问控制器 (SDRAM 控制器) 采用了状态机分解、分割传输方式使内部总线、数据写缓存方式、双通道预取指令缓冲器、命令寄存器、时间隐藏等技术来提高内存访问速度。

本文着重对多处理器片上系统 (MPSoC) 的共享存储控制仲裁方法进行了研究, 分析多个处理器中不同的内存访问请求优先级需求, 提出一种分层仲裁策略。该仲裁策略能通过对奇 bank 访问与偶 bank 访问分别进行仲裁, 从而隐藏存储器 bank 访问时的预充电延时, 达到提高内存总线使用效率的目的。

1 总体框架介绍

因特网是由路由器构成的网络, 路由器系统是一种网络系统。基本的路由器主要实现数据的分组转发功能和路由信息的交换以及更新功能。在路由器应用中, 需要经常进行分组队列的管理, 分组的实际数据放在 SDRAM 中, 同时分组的描述符——路由表也存储在 SDRAM 中, 所以主处理器 (main-processor)、协处理器 (co-processors) 要不断访问 SDRAM 存储单元以便实现分组队列管理。

通过将内核处理器、各协处理器及 PCI 对 SDRAM 存储单元的访问相互隔离, 可有效减少甚至消除访问存储器冲突, 使它们维持在最高性能工作点上。统计结果显示主处理器、各协处理器及 PCI 对 SDRAM 的访问几率: $p_{(\text{main-processor})} < 10\%$, $p_{(\text{co-processors})} > 90\%$, $p_{(\text{PCI})} < 10\%$ 。因此, 在此体系架构中, 需要仲裁机制以正确地对每个模块的访问作出仲裁, 同时系统采用了硬件可支持的控制状态寄存器 (CSR) 对内部硬件进行配置。

主处理器、协处理器、PCI 单元对 SDRAM 存储单元的访问是通过 SDRAM 控制器实现的, 控制器的实现考虑到了其必须用于异构多核处理系统当中, 所以要求对于不同的访问单元, 控制器能够发出正确的时序命令。SDRAM 控制器的实现结构如图 1 所示。

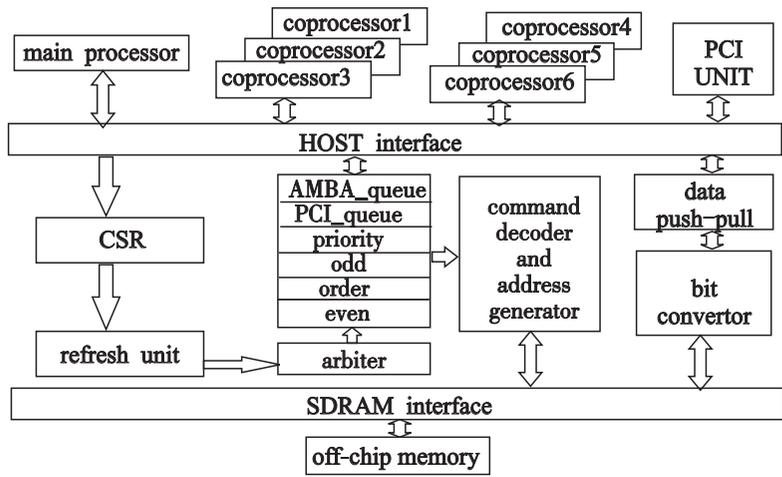


图1 SDRAM控制器控制通路结构框图

图 1 中包含了 SDRAM 控制器的主要功能单元, 即配置寄存器单元, 其中包括三个配置寄存器, 这三个寄存器在初始化过程中由主处理器进行配置, 主要的配置参数包括初始化周期长度、SDRAM 的自刷新周期、突发式长度、行/列地址宽度、tRCD、tRAS、tRP 等信息^[1]。刷新单元, 按照存储单元行数量进行定时刷新以保持数据有效。仲裁器单元, 整个 SDRAM 控制器的核心单元, 按照一定的仲裁策略对来自不同访问单元的命令作出正确有效的响应。接下来是指令存储单元, 由六个队列组成, 来自不同访问单元的命令按照命令标志位存入不同的命令队列。后经过命令译码以及数据通路单元向片外 SDRAM 存储单元发出操作命令以及相应的数据信息。

1.1 多层次并行系统中 SDRAM 控制器中的仲裁器实现

基于上述结构框图, 仲裁单元的设计必须保证以下原则:

- a) 针对每个指令队列的请求分配一个绝对的优先级, 在同等情况下, 绝对优先级高的指令队列请求优先被响应。
- b) 根据各个模块(指令队列)上次被访问的情况, 判断授权标志, 按轮询机制本次授权给上次访问后的下一个模块, 从而使得各模块得到均等的访问机会。
- c) 根据各个模块能够容忍的最大响应时间设置紧急状态标志(chain_ref), 当出现紧急状态时, 对应的请求能得到优先响应。

通过以上机制, 可以实现数据带宽的有效分配, 并且能够保证不会出现某个模块因得不到响应而导致系统崩溃的情况。

仲裁器的具体实现采用固定优先级与 round-robin 机制相结合的仲裁策略。固定优先级策略是指将多个仲裁申请者的优先级按一定顺序固定下来, 仲裁逻辑按照这个固定的优先级顺序来分配某一时刻的总线使用权。由于控制器所处的 SoC 体系中需要进行大批量实时数据的传输, 控制器内部仲裁采用固定优先级策略^[2]。同时通过引入 round-robin 机制, 使得低优先级的各个设备(指令队列)有均等的机会获得内部总线的使用权, 消除了仅采用固定优先级仲裁策略所带来的弊端。

仲裁器的状态机实现如图 2 所示。系统复位后首先进入空操作状态(i_NOP), 接着完成初始化过程, 包括全局预充电(i_PRE)、自动刷新(i_AR)、装载寄存器(i_MRS)三个状态, 在 i_MRS 下, 通过对行列地址的配置包括硬件可配置、软件可编程实现 SDRAM 控制器的通用。初始化过程完成之后进入 i_ready 状态, 根据自动刷新请求及上一条指令的完成情况分别进入刷新状态或者轮询授权标志判断状态, 按照上一次的访问标志将指令总线的使用权交给轮询到的下一个访问单元。轮询的顺序: AMBA queue PCI queue priority queue AMBA queue...。在有链式请求的情况下接着读出上次访问队列中的下一条指令, AMBA queue、PCI queue、priority queue 指令队列

中的指令都被清空时, 状态切换到 grant_token2 状态, 进行对 order queue 队列的读取, 最终回到 i_ready 状态。

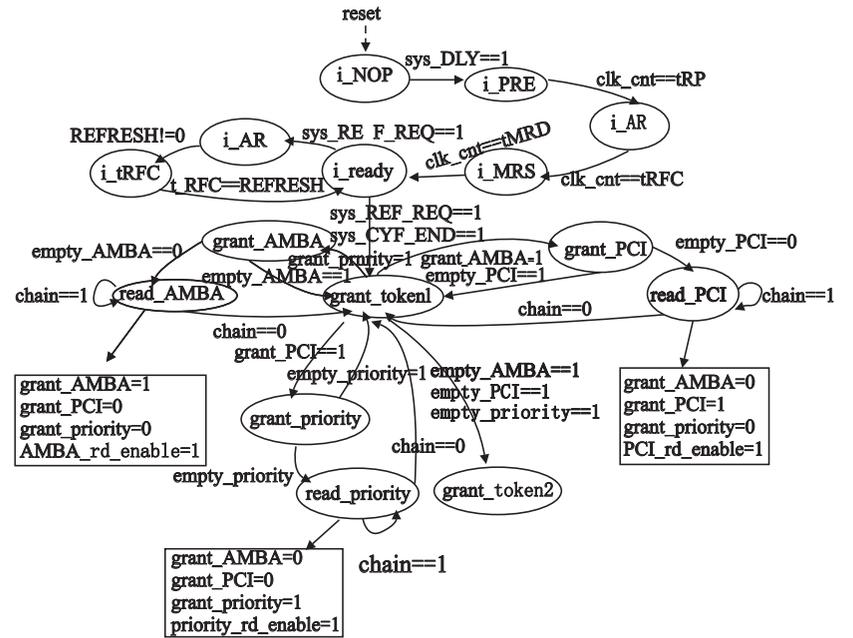


图2 仲裁器状态机实现

1.2 基于仲裁器的 SDRAM 主动优化技术实现

控制器经过优化存储位设定后可以对指令进行分类存储。指令分类基于使某一存储组对另一存储组隐藏预充电, 实现原理如图 3 所示。具体来说, 若存储器系统组织成奇数组和偶数组, 控制器工作于奇数组时, 控制器可以启动对偶数存储组的预充电, 控制器工作于偶数组时, 控制器可以启动对奇数存储组的预充电。若控制器可以在奇数和偶数组之间交替, 就能进行交替预充电, 以提高内存效率^[3]。

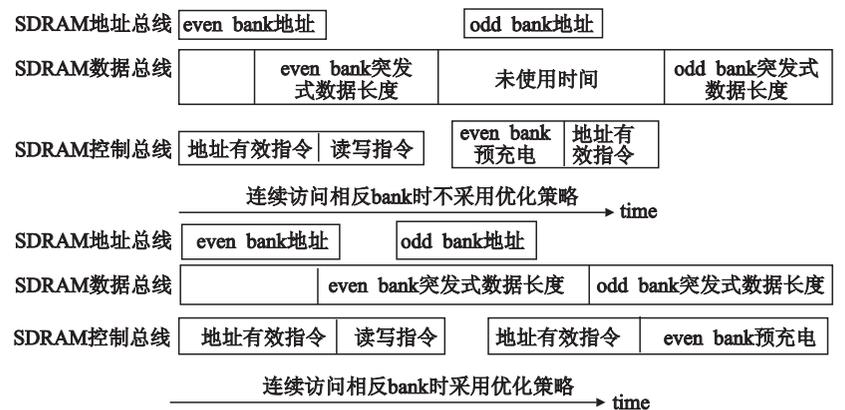


图3 奇偶优化原理图

SDRAM 控制器按照突发式长度访问 SDRAM 存储单元时, 执行完一次完整的写入操作需要的时钟周期数为

$$T_{write} = (T_{act_enable}) + tRCD + (T_{wr_enable}) + tBL + (T_{pre_enable}) + tRP \quad (1)$$

其中: T_{write} 代表完成一次完整的写入所需要的时钟周期数; tRCD 表示行地址采样直至采样到列地址之间的延时; tBL 表示按照突发式长度访问时所消耗的时钟周期数; tRP 表示预充电操作所占用的时钟周期数。

执行一次完整的读出操作占用的时钟周期数为

$$T_{read} = (T_{act_enable}) + tRCD + (T_{rd_enable}) + tBL + tCL + (T_{pre_enable}) + tRP \quad (2)$$

其中: T_{read} 代表完成一次完整的读出所需要的时钟周期数, 其他参数意义与写入操作相同。

经过奇偶队列的设置, 如果交替从奇偶队列中读出 $2n$ ($n > 1$) 指令, 则优化后从首条指令执行到第 $n-1$ 条指令, 每条指令被响应所需要的时钟周期分别缩短到:

$$T_{write(n-1)} = (T_{act_enable}) + tRCD + (T_{wr_enable}) + (T_{pre_enable(n-2)}) + tBL \quad (3)$$

$$T_{read(n-1)} = (T_{act_enable}) + tRCD + (T_{rd_enable}) +$$

$$(T_{pre_enable(n-2)}) + tBL + tCL \quad (4)$$

其中: $T_{write(n-1)} / T_{read(n-1)}$ 分别表示经过优化后前 $n-1$ 条指令每条指令执行所需要的时钟周期数。可以看出,前 $n-1$ 条指令在执行的的过程中都不包含预充电时间,体现了预充电时间的隐藏特性。

最后一条指令的执行时间分别与执行一次完整的写入、读出操作的时钟周期数相等。

$$T_{write(n)} = (T_{act_enable}) + tRCD + (T_{wr_enable}) + (T_{pre_enable(n-1)}) + tBL + (T_{pre_enable(n)}) + tRP \quad (5)$$

$$T_{read(n)} = (T_{act_enable}) + tRCD + (T_{rd_enable}) + (T_{pre_enable(n-1)}) + tBL + tCL + (T_{pre_enable(n)}) + tRP \quad (6)$$

1.3 采用主动优化技术后的仲裁器改进

指令队列中奇偶队列的设计就是为了更好地完成对 SDRAM 控制器的时序优化。按照仲裁器的仲裁策略,一旦奇偶队列都非空,则读出一条奇队列中的指令后,下一条一定读出的是偶队列中的指令,而奇队列中的 bank 地址一定对应奇 bank 地址(1 或 3);相应地,偶队列中存放的指令对应 bank 的地址一定是偶 bank 地址(0 或 2)。所以当连续访问不同 bank 时,SDRAM 控制器仲裁器可以交替的顺序分别访问 odd 和 even 指令队列,优化的实现可以通过执行完一个奇队列的指令时,不进行预充电,而在执行下一条偶队列指令的 tBL 的间隔内进行上一次的预充电操作的方式实现。

考虑到使用奇偶指令队列来提高 SDRAM 控制器的效率,对图 2 的仲裁器作出了如图 4 的改进,在图 2 的状态机基础上添加了奇偶队列仲裁机制。

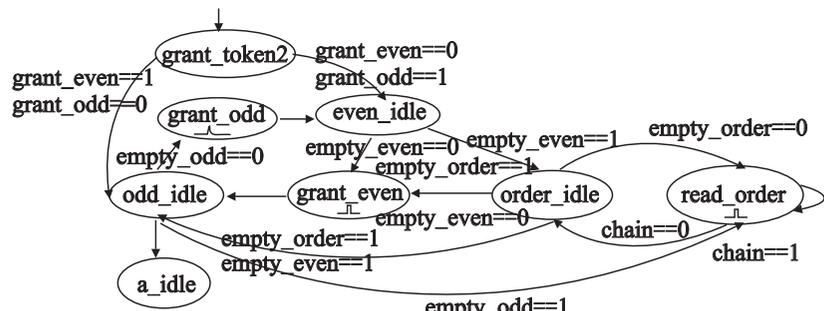


图4 改进的仲裁器状态机

当仲裁器将 AMBA queue、PCI queue、priority queue 指令队列都清空之后进入 grant_token2 状态,在 grant_token2 状态下进行奇偶授权标志位的判断,若上次授权给 even 队列,此次访问 odd 队列,反之亦然。若奇偶队列都非空,则在奇偶指令队列之间交替进行读出,读出指令使能信号分别在 grant_odd、grant_even 状态下产生,当且仅当奇偶指令队列中有一个空或者两个都空的时候才能将 order 指令队列中的指令读出。

2 并行优化控制器的 RTL 级到 FPGA 的实现

在主处理器的寄存器配置阶段,通过对行、列地址位宽的配置,控制器可支持容量为 8/16/32/64/128/256/512 MB 的外部存储设备,支持的内部逻辑存储单元可以为 2 个或者 4 个 bank,支持的外部 bank 的数目可以是 2 个、4 个或者 8 个。在仿真过程中,外部的 SDRAM 采用了 Micron 公司提供的 1M × 16、2M × 32、4M × 32 等多种仿真模型,结果正确。

以下是采用 Micron 公司 2M × 32 模型的并行优化 Modelsim 仿真结果。

图 5 显示了并行优化仲裁器交替读出奇偶指令队列中的指令后,SDRAM 控制器工作过程的 modelsim 仿真结果。分析波形可看出,第一条指令来自奇指令队列,经过译码这条指令

是对 SDRAM 存储单元进行写入操作,激活状态下采样到 bank 地址为 1,行地址为 0,经过 tRCD 的延迟,发出写入操作命令同时采样到列地址 4,接着发出对上一条指令操作行的预充电操作,因为上一条指令是对 bank0 的第 0 行的操作,所以在发出预充电命令同时采样 bank0 第 0 行地址,即对 bank0 第 0 行进行预充电操作;接着按照奇偶交替,下一条指令来自偶指令队列,激活状态下采样到 bank2 第 0 行地址,指令译码结果为写入操作,在发出写入命令后接着发出对 bank1 第 0 行的预充电命令;接着对 bank3 第 0 行进行写入操作,对 bank2 第 0 行进行预充电;接下来对 bank2 第 0 行进行读出操作,对 bank3 第 0 行进行预充电;最后对 bank1 第 0 行进行读出操作,对 bank2 第 0 行进行预充电,对于相同的 bank 地址、相同行地址,写入数据与读出数据完全吻合。

对不同的 BL 长度,采用奇偶 bank 交替技术优化后的效率如表 1 所示。

表 1 不同 BL 值优化后消耗的时钟周期数比较

BL	写入过程消耗的时钟周期数			读出过程消耗的时钟周期数		
	优化前	优化后	节省时钟周期	优化前	优化后	节省时钟周期
1	7n	4n+3	3n-3	9n	6n+3	3n-3
2	8n	5n+3	3n-3	10n	7n+3	3n-3
4	10n	7n+3	3n-3	12n	9n+3	3n-3
8	14n	11n+3	3n-3	16n	13n+3	3n-3

注:表格中 tRCD = 3, tRP = 3, tCL = 2; n 表示交替读出的指令数目。

对于相同的交替访问次数(横坐标表示交替访问次数),相同的 BL 值,如图 6 所示经过优化后,写入操作效率较读操作的效率提高明显(图 6 中短线表示写入操作效率的提高,点线表示读出操作效率的提高)。交替访问次数越多,优化效率越高;BL 的值越小,优化效果越显著;效率提高最大可达 40% 以上。



图5 Modelsim 仿真结果

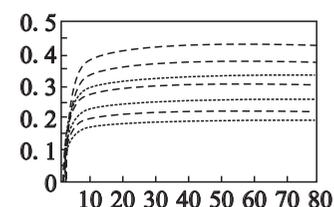


图6 不同BL值优化效率比较图

本文论述的基于并行优化 SDRAM 控制器的 Verilog 语言实现代码,已在 XILINX 公司的 Virtex2P/XC2VP30FPGA 中通过了 Xilinx ISE 的功能仿真、综合及布局布线后仿真^[4];在硬件电路上工作良好,能够使用 8/16/32/64/128/256/512 MB 的 SDRAM 来作为扩展存储单元。本设计综合编译后如表 2、3 给出详细周期、频率及资源占用情况。

表 2 综合后读写时钟频率

starting clock	estimated frequency/ns	clock period/MHz
clk_rd	2.435	410.602
clk_wr	1.563	639.898

表 3 综合后资源占用情况

device utilization summary			
selected device	2vp30fg676-7		
number of slices	52	13 696	0%
number of slice flip flops	23	27 392	0%
number of 4 input LUTs	59	27 392	0%
number of Ios	299		
number of bonded IOBs	146	416	35%
IOB flip flops	65		
number of GCLKs	2	16	12%

$$a_i = \frac{q_i l_i}{c_i l_i}$$

其中: a_i 为线路负荷度; q_i 为公交车流量 (辆 h); l_i 为路段 i 长度 (m); c_i 为线路 i 的通行能力 (辆 h)。

换乘率通常以最小换乘数表示 (次)。换乘数指由起始点到目标点所乘公交车辆的次数。公交换乘问题的实质就是确定起始点、目标点后, 找到一条快捷方便的线路, 首先求解起始点与目标站点间有没有直达的公交线路; 如没有直达线路, 就需找路径最短、换乘次数最少的线路。

3.3 西安市公交线路质量评价

以西安市公交几何网络为例, 选四个不同样区, 用上述评价指标, 计算样区内各指标值, 如表 1 所示。其中, 路网密度最高的是太白小区—省政务大厅段, 为 0.58%; 路网连接指数最高是省肿瘤医院—省体育场, 为 4; 非直线系数最高是省肿瘤医院—省体育场, 为 1.403; 公交线路通行能力最强的为劳动南路—西斜七路, 为 168.58; 公交线路负荷度最大为 0.018。各实验区详细实验数据如表 1 所示。

表 1 公交网络质量评价数据表

模型	样区			
	省肿瘤医院 - 省体育场	劳动南路 - 西斜七路	太白小区 - 省政务大厅	丁白村 - 小寨
路网密度 / %	0.38	0.51	0.58	0.44
路网连接指数	4	2.5	2.4	2.11
非直线系数	1.403	1.3858	1.2267	1.11
公交线路通行能力	$C = 114.815$ $Q = 6888.897$	$C = 168.58$ $Q = 6743.2$	$C = 109.573$ $Q = 7122.245$	$C = 148.084$ $Q = 4442.52$
线路负荷度	0.018	0.015	0.0078	0.02
换乘率	0	0	0	1

此外, 以省肿瘤医院—省体育场为例, 分别以公交线路的通行能力、公交线路负荷度、换乘率和距离最短为阻抗最小单因子最佳路径分析, 求出最佳选择路线, 结果如图 3 所示。其中: (a) 是通行能力最强的路径分析图; (b) 是线路负荷度最小的路径分析图; (c) 是换乘率最小的路径分析图; (d) 是距离最短的最佳路径分析图。

4 结束语

以 GIS、空间数据库技术为支撑, 建立西安市公交几何网络数据库, 不仅可以实现公共交通场站如站点、火车站、汽车站等信息可视化, 而且可以实现所有公交线路的统筹、可视化、信息化的管理。此外, 将公交状况评价模型与空间数据库结合在一起, 建立基于多种参数因子 (如公交线路通行能力、公交线路负荷度、换乘率、道路长度、道路的空间分布结构、运

行成本等) 的公交线路评价模型, 以网络分析为理论基础, 为道路选线、站点选址、城市快速道路选址提供科学依据, 也为城市公交线路管理提供新的途径和智能化、可视化的方法。研究区虽然以西安市为例, 但方法和评价模型非常适宜推广应用于城市的公交网络信息统筹管理中, 为公交线网扩展、站场建设, 客流规模估计、公交出行比重、公交服务质量提高、公交道路选线、站点选址和成本核算提供决策依据, 为我国城市公交信息的规范管理提供依据和新模式。

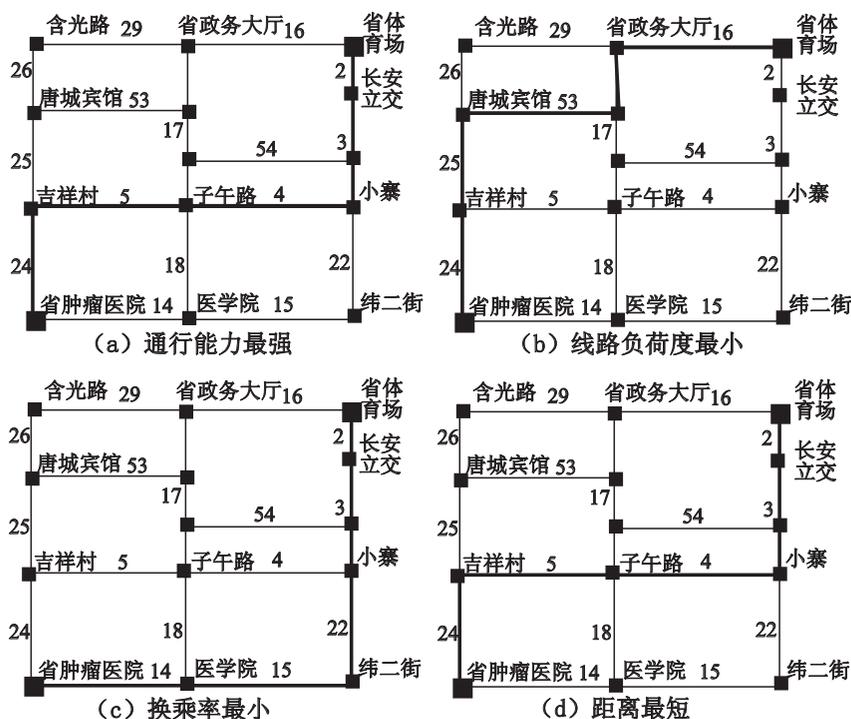


图 3 不同阻抗因子的最佳路径分析图

参考文献:

- [1] 祝微, 关敬文. 城市交通网络事故预测方法分析综述 [J]. 交通标准化, 2006, 157(9): 175-179.
- [2] 刘学军, 徐鹏. 交通地理信息系统 [M]. 北京: 科学出版社, 2003: 94-131.
- [3] 雷海燕, 张佳, 邱文静. 西安城市交通引导城市空间发展的措施研究 [J]. 交通标准化, 2006, 157(9): 183-185.
- [4] 朱大友, 李薛. 面向可持续发展的城市交通系统综合评价 [J]. 交通标准化, 2005, 147(11): 76-79.
- [5] 刘永红, 张省. 城市交通拥挤的经济学分析 [J]. 当代经济 (下半月), 2006(8): 19.
- [6] 蓝万炼, 张连波, 曲春梅. 城市交通可持续发展与地下快速干道 [J]. 长沙交通学院学报, 2006, 22(3): 44-48.
- [7] 张东. 城市公共交通线路的通行能力研究 [J]. 山西建筑, 2007, 33(27): 303-304.
- [8] 汪新凡. 城市公共交通发展水平的属性数学识别模型 [J]. 交通运输工程学报, 2007, 7(5): 118-122.
- [9] 党安荣, 贾海峰, 易善桢, 等. ARCGIS 8 Desktop 地理信息系统应用指南 [M]. 北京: 清华大学出版社, 2003: 661-662.

(上接第 1393 页)

3 结束语

本文着重设计多处理器共享内存访问控制的仲裁算法。该算法采用分层优先级策略, 并对内存的奇 bank 与偶 bank 访问分别仲裁, 能够隐藏 bank 的预充电延时, 从而能够显著地提高内存总线的利用效率。在网络处理器上的应用表明, 采用该方法的性能改善明显。该仲裁结构适用于多核处理器的共享存储访问结构, 并且能够用于其他高速存储控制器结构中。

参考文献:

- [1] 李文, 唐志敏. 一种减少内存访问延时的方法 [J]. 计算机工程, 2006, 32(3): 242-244.
- [2] PARK S, YI Y, PARK I C. High performance memory mode control for HDTV decoders [J]. IEEE Trans on Consumer Electronics, 2003, 49(4): 1348-1353.
- [3] GRIES M. The impact of recent DRAM architectures on embedded systems performance [C] // Proc of Symposium on Digital Systems Design. 2000: 282-289.
- [4] PARK J. Synthesis of multi-mode memory interfaces for FPGA reconfigurable computing machine [J]. International Journal of Computer Science and Network Security, 2007, 7(9): 39-44.