

嵌入式微处理器分支预测的设计与实现

陈海民, 李 峥, 王瑞蛟

(信息工程大学 电子技术学院, 郑州 450004)

(chenhaimin1986@126.com)

摘 要:针对五级流水线嵌入式微处理器的特定应用环境,对分支预测技术进行了深入研究,提出了一种新的分支预测方案。该方案兼容带缓存设计,通过扩展指令总线,在取指段提前对分支指令跳转方向和目标地址进行预测,保存可能执行而未执行的指令和地址指针以备分支预测失效时得以恢复,减少了预测失效的代价,同时保证了指令流的正确执行。研究表明,该方案硬件开销小,预测效率高,预测失效代价低。

关键词:嵌入式微处理器;流水线;ARM指令集;分支预测;失效代价

中图分类号:TP302 **文献标志码:**A

Design and realization of branch prediction for embedded microprocessor

CHEN Hai-min, LI Zheng, WANG Rui-jiao

(College of Electronic Technology, Information Engineering University, Zhengzhou Henan 450004, China)

Abstract: Concerning the specific application environment of embedded microprocessor, the branch prediction technology was researched in this paper, and a new scheme of branch prediction was proposed. Compatible with cache design, jump direction and destination address of branch prediction happened on extended instruction bus. The unexecuted instruction and address pointer were saved for possible recovery after misprediction, which reduced misprediction penalty, simultaneously guaranteed the instruction flow to execute correctly. The study shows this scheme is of little hardware spending, high prediction efficiency and low misprediction penalty.

Key words: embedded microprocessor; pipeline; ARM instruction set; branch prediction; misprediction penalty

0 引言

现代嵌入式微处理器为了提高指令执行的并行度,一般都采用了流水线设计^[1],但程序中分支指令的存在使得指令流的方向经常发生改变,造成流水线吞吐率的下降^[2]。这种控制相关问题直接制约了流水线执行的效率,而且随着流水线级数的增加,这种矛盾越发突出。

流水线微处理器中,分支预测技术对于处理器性能的提升起着重要的作用^[3]。分支预测的本质就是克服控制相关,减少流水线断流的几率和代价,进而提高处理器性能,分支预测失效将浪费大量的时钟周期^[4]。因此提高分支预测效率主要有两个方向:一是提高分支预测准确率;二是减少预测失效的代价。目前对分支预测的研究多是如何提高分支预测的准确率为目标,不针对任何处理器模型。而对于减少预测失效的代价而言,不同的处理器模型,不同的流水线级数,需采用不同的策略。一种分支预测机制可能在某些应用中可以提高执行效率,但在另一些应用中可能效果就不明显,所以需要不同的应用采用不同的分支预测解决方案^[5]。

嵌入式微处理器硬件资源有限,因此需要一种不但硬件开销小而且执行效率高的分支预测,以达到较高的性价比。五级流水线在多种嵌入式微处理器中广泛使用,被认为是经典的处理器设计方式。本文基于 ARMv4T 架构的 32 位 ARM 指令集^[6],以 ARM9 系列嵌入式微处理器为应用原型,提出了一种适用于五级流水线的分支预测方案并得以硬件实现。

该方案不以提高分支预测准确率为目的,而是以最少的硬件开销着眼于减小预测失效的代价,具有优越的预测机制。

1 ARM9 嵌入式微处理器模型

1.1 ARM9 流水线结构

ARM9 具有五级流水线^[7],分别为取指(IF)、译码(ID)、执行(EX)、访存(ME)和写回(WB)。其中,IF段作为流水线的第一阶段承担了两大任务:一是计算下一条指令的取指地址;二是接收程序存储器传送的指令。流水线时序结构如图1所示,其中XX_enbar信号是互锁控制模块控制流水线各流水段是否接收上一流水段传递数据的暂停信号。

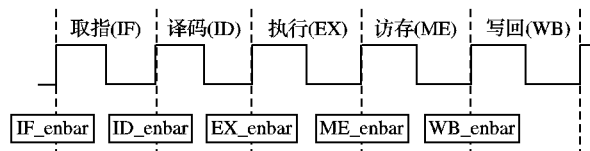


图1 流水线时序结构

1.2 ARM 指令集分支指令分析

32 位 ARM 指令集的分支指令共有两条,分别是跳转指令 B 和跳转连接指令 BL,其二进制代码如图2所示。其中 cond 表示条件执行码,offset 表示分支偏移地址。

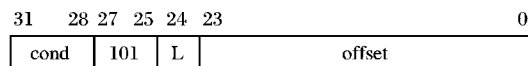


图2 分支指令二进制代码

收稿日期:2011-01-25;**修回日期:**2011-03-04。**基金项目:**国家自然科学基金资助项目(61072047);郑州市创新型科技人才队伍建设工程项目(096SYJH21099);现代通信国家重点实验室基金资助项目(9140C1106021006)。

作者简介:陈海民(1986-),男,四川遂宁人,硕士研究生,主要研究方向:安全芯片;李峥(1971-),男,河南开封人,副教授,博士,主要研究方向:密码工程、信息安全;王瑞蛟(1985-),男,湖南邵阳人,硕士研究生,主要研究方向:安全芯片。

2 传统的分支预测设计

根据各个流水段的功能,分支指令需要在ID段才能被识别出来,判断其是否跳转的条件状态需要在EX段才能计算出来,这也就造成了预测失效的可能,导致无效的指令进入流水线。

在流水线执行过程中,当分支指令位于ID段时,分支指令被识别,进行分支预测。如果预测分支指令跳转(taken),则由于IF段已经预取了一条指令,则必须将其冲刷掉,按照预测的分支地址重新取指令进入流水线;对于ID段而言,其也不能接收位于IF段的指令,而应接收所预测的分支地址处的指令。如果预测分支指令不跳转(not-taken),则指令按照流水线顺序执行。当分支指令位于EX段时,分支预测是否有效的条件状态已经计算出来,此时判断其是否有效。一旦预测失效,则必须冲刷掉IF段和ID段中预取的两条指令,计算新的正确指令的地址,且在下一个时钟接收新的指令进入流水线。

可以看出,传统的分支预测设计中,分支指令在ID段译码且进行分支预测,其分支地址需要在EX段才能计算出来,滞后判断导致了分支失效代价很大,大大降低了流水线有效指令的吞吐率。

3 新的分支预测方案

思考1 对流水线各流水段的功能进行深入分析发现,IF段没有对指令进行任何实质性的操作,其任务仅仅是从存储器选取正确的指令为下一个流水段做准备。因此,可以在其取出正确指令进入IF段的同时,将其地址顺延的下一条指令予以保存。如果分支预测失效,则直接将保存的指令送入ID段,不用再经过IF段的流水,这样就减少了气泡的产生。

思考2 分析ARM分支指令的二进制代码及执行过程可以看出,其译码过程和分支地址计算非常简单,完全可以前移到IF段来执行。由于分支指令使用的是PC相对地址,因此只需将指令代码中的分支地址加上当前的PC地址即为分支指令跳转处的分支地址。

思考3 分支指令不对数据存储器进行任何操作,对内部通用寄存器而言,也只是对R15有影响,其任务仅仅是控制程序的跳转方向和跳转地址,不一定必须单独占用一个流水段,在分支预测之后直接将预测地址处的指令送入ID段即可,如此将提高流水线执行效率。

本文按照上述的三点思考设计出一种新的分支预测方案。

3.1 指令总线扩展

传统的32位嵌入式微处理器的指令总线位宽为32位,即一次只接收一条指令。而为了便于分支预测时保存其地址顺延的下一条指令,我们扩展指令总线位宽,一次性取三条指令,这样就可以对前两条指令进行判断预测。如果分支预测跳转,就保存其地址顺延的下一条指令,当预测失效时,只需恢复其保存的指令送入ID段即可。对于非分支指令而言,指令总线上只有第一条指令有效,和传统处理器执行指令一样。如此设计将具有许多优越的性质。

由于本文的分支预测方案兼容带缓存设计,CPU取指面向指令缓存(icache)。而指令缓存中是以指令块的形式存在,每个指令块一般含有8条指令,指令块之间的地址很可能不是连续的,于是会出现连续取指的三条指令不位于同一指

令块的情况,而后面两条指令有不被执行的可能。如此一来,不被执行的指令所在的指令块如果不在指令缓存中,则将从主存中将该指令块调入指令缓存,此过程降低了程序执行的空间局部性^[8],大大降低处理器的执行效率。为了解决此问题,我们对取指令做了如下处理,如图3所示。其中i1~i8为指令块中连续地址的8条指令,“Nop”表示空指令,“Nop”进入流水线不进行任何操作。当PC地址指针指向指令i7时,取指令{Nop,i8,i7}进入流水线;当PC地址指针指向指令i8时,取指令{Nop,Nop,i8}进入流水线。如此即可将所取的三条指令都限制在同一个指令块之中,不会出现越块访问的情况。

Nop	Nop	i8	i7	i6	i5	i4	i3	i2	i1
-----	-----	----	----	----	----	----	----	----	----

图3 指令块处理示意图

3.2 分支预测算法的选取

分支预测有动态预测和静态预测之分^[9],动态预测准确率一般高于静态预测,但是却要占用处理器内部大量存储资源,而且越是经典的预测算法其复杂度越高,在嵌入式微处理器上的可实现性越低^[10]。由于本文不以提高预测准确率为目的,考虑到嵌入式微处理器的特定应用环境,从减小硬件开销、追求执行效率的角度出发,本文选择了一种简单的静态预测算法——BTFN预测(Back taken, Foreword not taken)^[5]。分支地址为正增加,则预测不跳转(not-taken);分支地址负增加,则预测跳转。BTFN预测对于程序循环非常有效,例如IBM360/370使用的就是该预测算法^[11]。

3.3 分支预测机制

决定流水线如何执行存储器的指令有四个因素:一是IF段计算出来的下一条指令的取指地址;二是此时已经进入IF段待处理的指令;三是ID段选取哪一条指令进行下一步操作;四是互锁控制模块如何控制流水线各个流水段的暂停信号,以达到冲刷掉进入流水线的错误指令的目的。为了达到分支预测指令良序执行的目的,本文设计了如表1所示的流水线执行方案。

假设连续地址的四条指令{iplus4, iplus3, iplus2, iplus1}中,{iplus3, iplus2, iplus1}位于指令总线上,如果上一时钟IF段计算出来的取指地址为PC,则此时其对应的地址分别为{PC+8, PC+4, PC}。表1描述了流水线中不存在需要判断的分支预测的情况下IF段中可能的分支预测情况。inst_if表示即将送入ID段的指令(由于前提假设是流水线中不存在需要判断的分支预测指令,因此进入ID段的指令选择对象不存在可恢复的保存指令,否则,如果存在可恢复的保存指令,当分支预测失效时,应该优先保证可恢复的保存指令进入ID段),irstore表示所保存的因为分支预测而没有执行但分支预测失效后可能送入ID段执行的指令,pestore表示所保存的因为分支预测而没有执行但分支预测失效后需重新取指送入IF段的地址指针,pred_addr表示按照分支预测方向预测的指令取指地址。

本文对前两条指令进行分支预测,预测分支指令是否跳转可能的条件有:(a)无条件分支指令;(b)有条件分支指令;(c)分支地址正增加;(d)分支地址负增加。则根据BTFN预测原则,预测iplus1指令跳转的信号ptaken1成立的条件是:iplus1满足(a)∨(d)。预测iplus1指令不跳转的信号puntaken1成立的条件是:iplus1满足(b)&(c)。预测iplus2指令跳转的信号ptaken2成立的条件是:1)没有对指令iplus1进行预测,即!(ptaken1∨puntaken1);2)iplus2满足(a)∨(d)。

表 1 分支预测新方案

分支预测	含义	Inst_if	irstore	pcstore	pred_addr
ptaken1	iplus1 为分支指令且预测跳转	iplus1	iplus2	iplus3 的地址	iplus1 的分支地址
puntaken1	iplus1 为分支指令且预测不跳转	iplus2	Nop	iplus1 的分支地址	iplus3 的地址
ptaken2	!(ptaken1 puntaken1), 且 iplus2 为分支指令且预测跳转	iplus1	iplus3	iplus4 的地址	iplus2 的分支地址

3.3.1 取指地址分析

IF 段承担的两大任务之一就是根据其他流水段反馈的控制信号计算下一条指令的取指地址 inst_addr。其来源包括:1)异常向量地址;2)EX 段修改 PC 指针;3)ME 段修改 PC 指针;4)预测失效时恢复保存的 PC 地址指针 pcstore;5)分支预测时预测的指令地址 pred_addr;6)顺序取指的递增时 PC 地址指针。其中 4)、5)与分支预测有关。

对于 pcstore 而言,分支预测时保存的 PC 地址指针有三个数据来源:1)如果 puntaken1,pcstore 保存 iplus1 的分支地址,其在 IF 段就可以直接计算出来;2)如果 ptaken1,pcstore 保存 iplus3 的指令地址;3)如果 ptaken2,pcstore 保存 iplus4 的指令地址。根据 ARM9 五级流水线分支指令的执行情况,对于连续的分支指令而言,最多连续的 3 条分支指令对流水线控制相关问题同时有影响,因此本文通过一个容量为三个字的 FIFO(先进先出队列)指针缓冲寄存器来保存相应的 pcstore,其通过一个状态机来控制。

对于 pred_addr 而言,当分支指令预测跳转时,分支地址处的指令还没有从指令缓存中取出来,而应该计算出分支地址,从分支地址处重新取指令送入 IF 段;当分支指令预测不跳转时,需要将分支指令顺延的下一条指令送入 ID 段,同时将下一条指令的地址赋值给 pred_addr,以保证流水线的正确顺序执行。因此分支预测时预测的指令地址 pred_addr 有 3 个数据来源:1)如果 ptaken1,则 pred_addr 赋值 iplus1 的分支地址;2)如果 ptaken2,则 pred_addr 赋值 iplus2 的分支地址;3)如果 puntaken1,则 pred_addr 赋值 iplus3 的指令地址。

3.3.2 IF 段待处理的指令

首先指令总线上有三条指令待处理,即 { iplus3, iplus2, iplus1 }, 其次还有之前分支指令分支预测时保存的未执行的指令 irstore,其有可能会在分支预测失效时得以恢复。分支预测时,如果 ptaken1,则 irstore 保存 iplus2;如果 puntaken1,则 irstore 保存 Nop;如果 ptaken2,则 irstore 保存 iplus3。根据 ARM9 五级流水线分支指令的执行情况,最多连续的 3 条分支指令对流水线控制相关问题同时有影响,因此本文通过一个容量为三条指令的 FIFO(先进先出队列)指令缓冲寄存器来保存相应的 irstore,其通过一个状态机来控制。

3.3.3 ID 段待接收的指令

设位于 IF 段即将传送给 ID 段的指令为 inst_if,其赋值共有三个来源:1)分支预测失效后,可恢复的所保存的指令 irstore;2)如果 puntaken1,指令总线上的指令 iplus2;3)其他情况下,指令总线上的指令 iplus1。

分支预测失效且 irstore“可恢复”时,选择相应的 irstore 传送给 ID 段。这里的“可恢复”与 3.1 节指令总线扩展有关。当地址指针指向指令 i7 时(如图 3),取指令 { Nop, i8, i7 } 进入 IF 段,如果 ptaken2,此时 iplus3 = Nop,irstore 将其保存是没有意义的,我们称此情况为不“可恢复”;当地址指针指向指令 i8 时,取指令 { Nop, Nop, i8 } 进入 IF 段,如果 ptaken1,此时 iplus2 = Nop,irstore 将其保存也是没有意义的,我们同样称此种情况不“可恢复”。除此两种情况外,其他取指情况进行分支预测称为“可恢复”的。一旦判断“可恢复”的分支预测此时失效,则下个时钟直接将相应的 irstore 里保存的指令送入 ID 段。如果此时所判断的分支预测不“可恢复”,则需要 ID

段互锁,按相应的 pcstore 所保存的地址指针重新取指令进入流水线。

3.3.4 互锁控制模块对流水线暂停信号的控制

影响互锁模块控制流水线的暂停信号有很多因素,本文只分析与分支预测相关的信号。本文定义了如下的状态机转移图,如图 4 所示。其中, M0 为常态; M1 表示分支预测失效且不“可恢复”时所恢复的地址指针处重新取得的新指令已经进入 IF 段; M2 表示分支预测失效时恢复的指令已经进入 ID 段; M3 表示分支预测失效时恢复的指令已经进入 EX 段。其中,信号 mispredicted 表示分支预测失效,信号 mis_rec 表示分支预测失效且“可恢复”。

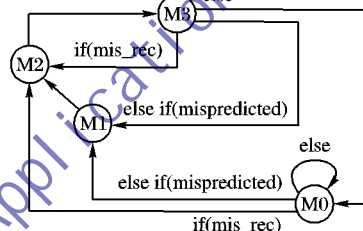


图 4 状态机转移图

1) 分支预测对 IF 段暂停信号 IF_enbar 的影响。

无论分支预测情况如何、分支预测是否失效,IF 段都要计算新的取指地址,且从指令缓存接收新的指令,因此分支预测对于 IF_enbar 没有影响。

2) 分支预测对 ID 段暂停信号 ID_enbar 的影响。

当 ptaken1 成立时,ID 段需要接收分支指令 iplus1 分支地址处的指令,而此时该指令还在指令缓存中未取出,还没有进入 IF 段,故得互锁 ID 段暂停一个时钟,拒绝接收此时 IF 段送来的指令(即表 1 中 ptaken1 时的 iplus1 并不进入 ID 段)。

当分支预测失效信号 mispredicted 成立时,如果“可恢复”,则不用互锁 ID 段,此时 ID 段直接接收相应 irstore 里保存的指令即可,状态由 M0 转移至 M2;如果不“可恢复”,则需要将 pcstore 里保存的地址指针传送给指令缓存,下一个时钟取出地址指针处的指令送入 IF 段,因此需要互锁 ID 段暂停一个时钟,状态由 M0 转移至 M1。

3) 分支预测对 EX 段暂停信号 EX_enbar 的影响。

当分支预测失效信号 mispredicted 成立时,由于处于 ID 段的指令为错误指令,必须将其冲刷掉,故需要互锁 EX 段暂停一个时钟,拒绝接收此时 ID 段送来的指令。

当处于状态 M1 时,意味着所恢复的地址指针处重新取得的新指令处于 IF 段,此时 ID 段的指令仍然为错误指令,需要将其冲刷掉,需要互锁 EX 段。

4) 分支预测对 ME 段暂停信号 ME_enbar 的影响。

当分支预测失效信号 mispredicted 成立时,由于分支指令不占用流水段,此时 EX 段的指令仍为错误指令,需要将其冲刷掉,因此必须互锁 ME 段。同理得出,当处于状态 M1,或处于状态 M2 时,也必须互锁 ME 段。

5) 分支预测对 WB 段的暂停信号 WB_enbar 的影响。

同理得出,当处于状态 M1,或处于状态 M2,或处于状态 M3 时,需要互锁 WB 段。需要说明的是,当分支预测失效信号 mispredicted 成立时,由于 ME 段的指令为正确指令,故此

时不能互锁 WB 段。

4 性能分析及功能验证

按照本文的设计方案,当 *ptaken1* 或 *ptaken2* 成立时,如果此时流水线中分支预测失效,则其保存的 *irstore* 直接进入 ID 段,不用重新取指令进入 IF 段,大大减小了分支预测失效的代价。当 *putaken1* 成立时,*iplus2* 直接进入 ID 段,*iplus1* 没有单独占用流水段时钟,提高了执行效率。整个设计使得分支预测指令良序执行。

需要指出的是,分支预测失效时,如果 *irstore* 存储是分支指令,当 *mis_rec* 成立时,则该分支指令直接进入了 ID 段,而没有进行分支预测。本文的处理方式是将其视为 EX 段修改 PC 指针的方式加以对待。因为该指令本来就没有在 IF 段占用流水线,是否对其进行分支预测代价都比较小,故此设计是合理的。

本文的分支预测思想在课题组设计的兼容 ARMv4T 架构 32 位 ARM 指令集嵌入式微处理器 ETI_ARM 上得到了完美的验证。仿真验证时,针对分支预测硬件设计时的各种特殊情况,本文设计了相应的大量分支指令组合进行验证,均达到了分支指令良序执行的目的,验证了设计的正确性。

5 结语

分支预测技术研究一般都针对深度流水线和宽发射机制的通用微处理器^[12],对于预测准确率要求很高。而嵌入式微处理器硬件资源有限,无法为高预测准确率的预测算法提供足够的硬件资源和功能消耗。本文针对嵌入式微处理器的特定应用环境,着眼于减小分支预测失效的代价,提出了一种新的分支预测方案。以较低的硬件开销实现了兼容 ARMv4T 架构 32 位 ARM 指令集的分支预测硬件设计。理论分析和仿真

验证均证明了设计的有效性和正确性,为嵌入式微处理器的分支预测设计提供了新的思路。

参考文献:

- [1] 陈智勇. 流水线微处理器的设计与实现[J]. 桂林电子科技大学学报, 2007, 27(3): 219-223.
- [2] 喻明艳, 张祥建, 杨兵. 基于跳跃访问控制的低功耗分支目标缓冲器设计[J]. 计算机辅助设计与图形学学报, 2010, 22(4): 695-702.
- [3] EVERS M, YEH T Y. Understanding branches and designing branch predictors for high-performance microprocessors [J]. Proceedings of the IEEE, 2001, 89(11): 1610-1620.
- [4] 陈跃跃, 周兴铭. 一种精确的分支预测微处理器模型[J]. 计算机研究与发展, 2003, 40(5): 741-745.
- [5] 冯子军, 肖俊华, 章隆兵. 处理器分支预测研究的历史和现状[R]. 北京: 中国科学院计算技术研究所微处理器研究中心, 2008.
- [6] ARM DDI 0210C. ARM 7 TDMI technical reference manual [R/OL]. [2010-08-20]. <http://www.arm.com>.
- [7] ARM DDI 0180A. ARM 9 TDMI technical reference manual [R/OL]. [2010-08-20]. <http://www.arm.com>.
- [8] 杜春雷. ARM 体系结构与编程[M]. 北京: 清华大学出版社, 2003: 195.
- [9] 汪永威, 樊晓桢, 黄小平. 32 位 RISC 微处理器中分支预测器的硬件实现[J]. 计算机应用研究, 2009, 26(2): 419-421.
- [10] 黄伟, 王玉艳, 章建雄. 嵌入式处理器动态分支预测机制研究与设计[J]. 计算机工程, 2008, 34(21): 163-165.
- [11] ANDERSON D W, SPARACIO F J, TOMASULO R M. The IBM System/360 model 91: Machine philosophy and instruction handling [M]. San Francisco: Morgan Kaufmann Publishers, 2000.
- [12] 姜瑞凯, 汪永威. 一种静态和动态分支预测的组合方法研究[J]. 科学技术与工程, 2009, 9(8): 2085-2089.

(上接第 2003 页)

淹没区域及居民所在区域,对淹没区及影响区内被淹没及受影响居民的基本信息进行统计分析,如图 7 所示。

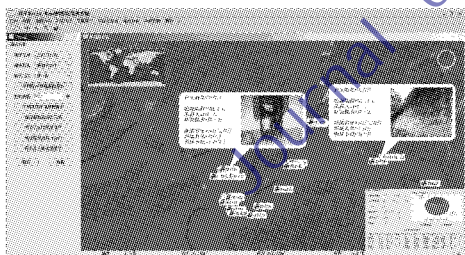


图 7 淹没与居民信息分析

4 结语

本文基于三维地理信息系统 World Wind 设计了移民信息系统,采用瓦片金字塔对地形及影像数据进行划分与调度,实现了海量地形及影像数据高效组织,并与移民数据有机融合。系统集成移民数据的管理、集成与应用为一体,为移民工程实时分析与后期扶持提供便利,随着应用功能模块的不断完善,具有较好的应用前景。目前系统以移民数据的管理与展示为主,兼具决策支持功能,随着信息技术在移民工程中的不断发展以及移民后扶持工作的不断深入,系统后续工作将集中在决策支持功能模块,以其形成一个以移民决策支持为主、移民数据管理为辅的移民信息系统。

参考文献:

- [1] 朱一成. 水利工程移民遗留问题管理信息系统开发[J]. 计算机工程, 2004, 30(23): 196-197.

- [2] 叶芳毅, 花向红. 南水北调中线工程丹江口水库移民信息系统[J]. 测绘信息与工程, 2007, 32(5): 3-5.
- [3] 冯久成, 王桂寅, 王鲜苹, 等. 小浪底水库移民信息系统[J]. 人民黄河, 2000, 22(10): 31-32.
- [4] 詹小国, 刘莉芳, 杨鹏, 等. 全国水库移民后期扶持管理信息系统设计综述[J]. 人民长江, 2009, 40(4): 39-42.
- [5] 张尚弘, 易雨君. World Wind 在汶川地震应急系统中的应用研究[J]. 系统仿真学报, 2010, 22(5): 1296-1300.
- [6] NOURIE D. Using NASA's World Wind component in your Java technology applications [EB/OL]. [2010-10-16]. <http://www.oracle.com/technetwork/articles/javase/worldwind-136570.html>.
- [7] 姜仁贵, 解建仓, 李建勋, 等. 基于数字地球的洪水淹没分析及仿真研究[J]. 计算机工程与应用, 2011, 47(13): 219-222.
- [8] Web map service implementation specification [EB/OL]. [2010-10-16]. <http://www.opengis.com/standards/wms>.
- [9] BELL D G, KUEHNEL F, MAXWELL C, et al. NASA World Wind: Opensource GIS for mission operations[C]// Proceedings of 2007 IEEE Aerospace Conference on Big Sky. Washington, DC: IEEE Computer Society, 2007: 1-9.
- [10] 姜仁贵, 解建仓, 李建勋, 等. 基于数字地球的 WebGIS 开发及其应用[J]. 计算机工程, 2010, 37(6): 225-227.
- [11] 张志强. “引汉济渭”工程信息化系统建设的思考[J]. 陕西水利, 2010(2): 75-76.
- [12] GAMETT J. GeoServer [EB/OL]. [2010-10-21]. <http://geoserver.org/display/GEOS/Welcom>.