

第18章 Pentium和Pentium Pro 微处理器

罗文坚
中国科大 计算机学院

<http://staff.ustc.edu.cn/~wjluo/mcps/>

1

本章内容

- Pentium微处理器简介
- Pentium的特定寄存器
- Pentium的存储管理
- Pentium的新指令
- Pentium Pro微处理器简介
- Pentium Pro的特性

2

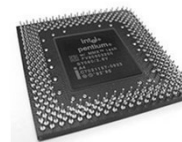
Pentium微处理器简介

- 引脚的功能
- 存储系统
- 输入/输出系统
- 系统时序
- 分支预测逻辑
- 高速缓存结构
- 超标量体系结构

3

Pentium简介

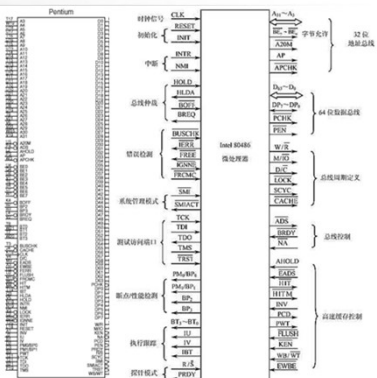
- Pentium微处理器有两个版本
 - 称为Pentium OverDrive的P24T型
 - 32位数据总线
 - 可以兼容80486机器（P24T插座）
 - 当作老旧486等级电脑的升级选项
 - 全功能型Pentium
 - 64位数据总线
 - 本章讲解



4

Pentium微处理器的外部引脚

- 237个引脚
 - 29个地址引脚
 - 64个数据引脚
 - 75个控制引脚
 - 其它, 69个
 - VCC
 - VSS
 - NC
- CLK (clock) : 时钟信号, 为CPU提供基本的定时信号。



5

地址总线 (address bus)

- A31~A3 (address) : 32位地址总线, 三态、输出, 用于定义存储器和I/O端口地址。
- BE7#~BE0# (byte enable) : 字节允许信号, 低电平有效。
- 注意: A31~A3和BE7#~BE0#构成32位地址总线, 可寻址4GB的内存空间和64KB的I/O空间。
 - 4GB的内存空间分为八个512MB的存储体, 每个存储体分别由字节允许信号选通。当BE_i (i=0~7) 有效时, 选择相应的存储体, 然后由A31~A3选择相应的字节进行读/写操作。
 - 寻址64KB的I/O空间时, 只有A15~A3和BE_i (i=0~7) 有效。

6

地址总线 (address bus)

- **A20M# (address bit 20 mask)**：第20位地址屏蔽信号，输入，低电平有效。
 - 当有效时，将屏蔽A20及以上地址，使Pentium微处理器仿真8086CPU的1MB存储器地址。
 - 该引脚用于在实模式中通知Pentium进行地址回绕，就像在8086微处理器中那样，该引脚供HIMEM.SYS驱动程序使用。
 - 只有在CPU工作在实模式下才有意义。

7

地址总线 (address bus)

- **AP (address parity)**：地址奇偶校验位，双向，高电平有效，指示地址总线A31~A3上偶检验信息。
- **APCHK# (address parity check)**：地址奇偶位检测，输出，低电平有效，指示微处理器检测到地址总线奇偶错。
 - Pentium微处理器对地址总线增加了奇偶校验功能，它输出的地址信号A31~A3会产生一个奇偶校验位，在AP信号线上输出，存储器子系统可据此对地址进行校验。

8

数据总线 (data bus)

- **D63~D0 (data lines)**：64位双向数据总线，可以传输8位、16位、32位和64位数据。
- **DP7~DP0 (data parity)**：数据奇偶校验信号，双向。DP7~DP0分别对应64位数据中字节7~字节0的校验位，即DP7对应D63~D56，DP0对应D7~D0。
- **PCHK# (parity check)**：奇偶校验状态信号，输出，低电平有效。当低电平时，表示CPU在上一个读周期采样的数据奇偶校验出错。

9

数据总线 (data bus)

- **PEN# (parity enable)**：奇偶校验允许信号，输入，低电平有效，用来规定在发生校验时，是否进行异常处理。
 - 若为低电平，且控制寄存器CR4的MCE位为1，则出现校验错误时，微处理器将会自动执行异常处理。

10

总线周期定义 (bus cycle definition)

- 读/写信号W/R#
- 存储器或I/O访问信号M/IO#
- 数据/控制信号D/C#
 - D/C#=1，表示数据传输周期；D/C#=0，表示指令代码传输周期。
- 总线锁定信号LOCK#
- **SCYC (split cycle)**：分隔周期信号
- **CACHE# (cache)**：可高速缓存信号

11

总线控制 (bus control)

- 地址选通信号ADS# (address strobe)：地址选通信号，输出，低电平有效。当为低电平时，表明地址总线上输出的地址有效。
- **BRDY# (burst ready)**：突发传送就绪信号，输入，低电平有效。指示外部系统已送出或接收有效数据。
 - 注意：对于80486，该引脚指示突发传送就绪。当突发传送时，一次数据传送只要一个时钟周期，而不是通常的两个时钟周期。非突发传送的就绪信号用RDY#指示。
- **NA# (next address)**：下一个地址信号，输入，低电平有效，用于形成流水线式总线周期。
 - 有效时，表明即使当前总线周期还没有完成，外部存储系统已经准备就绪，将下一个地址输出到总线上，用以开始一个新的总线周期。

12

高速缓存控制 (cache control)

- **AHOLD** : 地址保持请求信号
- **PWT**: 页面通写控制信号
- **PCD**: 页面高速缓存禁止信号
- **KEN#**: 高速缓存允许信号
- **FLUSH#**: 高速缓存清除信号
- **EADS**: 外部地址有效信号
- **WB/WT#**: 回写/通写方式信号
- **HIT#**: 询问周期命中信号
- **HITM#**: 命中数据cache的修改行信号
- **INV**: 无效请求信号
- **EWBE#**: 外部写缓冲器空信号

13

初始化 (initialization)

- **RESET**: 复位信号
- **INIT (initialization)**: 初始化引脚

14

中断请求 (interrupts)

- **INTR**: 可屏蔽中断请求
- **NMI**: 非屏蔽中断请求

15

总线仲裁 (bus arbitration)

- **HOLD**: 总线请求信号
- **HLDA**: 总线请求响应信号
- **BREQ**: 内部总线请求信号
- **BOFF#**: 强制CPU放弃系统总线信号

16

错误检测 (error reporting)

- **FREE# (floating-point error)**: 浮点错误报告, 输出。
- **IGNNE# (ignore numeric error)**: 忽略数字错误, 输入, 低电平有效。
- **BUSCHK# (bus check)**: 总线周期检测, 输入, 低电平有效。
- **PRCMC# (function redundancy checking)**: 功能冗余检测, 输入, 低电平有效。
- **IERR# (internal error)**: 内部出错指示, 输出, 低电平有效。如果在读周期产生奇偶错误, 系统将强制IERR#输出一个时钟周期的低电平。

17

系统管理模式 (system management mode)

- **SMI# (system management interrupt)**: 系统管理中断请求, 输入, 低电平有效。有效时, 进入系统管理模式。
- **SMIACK# (system management interrupt active)**: 启动系统管理模式工作, 输出, 低电平有效, 指示Pentium进入系统管理模式。
- 系统管理模式主要用于实现系统电源管理功能, 进入该模式后, 处理器的状态被保存起来, 并可以有单独的存储空间。执行RSM指令, 则退出系统管理模式。

18

测试访问端口（tap port）

- **TCK（testability clock）**：测试时钟，输入，为系统的边界扫描提供时钟信号。
- **TDI（test data input）**：测试数据输入引脚，输入。
- **TDO（test data output）**：测试数据输出引脚，输出。
- **TMS（test mode select）**：测试模式选择，输入。
- **TRST#（test reset）**：测试复位引脚，输入。

19

断点/性能检测（breakpoint/performance monitoring）

- **BP3~BP0（breakpoint）**：断点匹配检测，输出，高电平有效。BP3~BP0与调试寄存器DR3~DR0相对应，当调试寄存器编程设置为断点匹配测试时，若BP3~BP0输出为高电平，则表明断点匹配。
- **PM1, PM0（performance monitoring）**：性能监测，输出高电平有效，与BP1和BP0多路复用，由调试模型寄存器（debug mode control register）中的PB1和PB0位来确定它们是否用于断点匹配BP1, BP0还是性能监测PM1, PM0。

20

执行跟踪

- **BT3~BT0（branch trace）**：分支跟踪，输出。在分支跟踪的特殊周期，BT3~BT0提供分支目标的线性地址的位2~位0，BT3代表特定的操作尺度。
- **IU（instruction U-pipeline）**：U流水线指令执行完成，输出。
- **IV（instruction V-pipeline）**：V流水线指令执行完成，输出。
- **IBT（instruction branch trace）**：指令分支发生，输出。

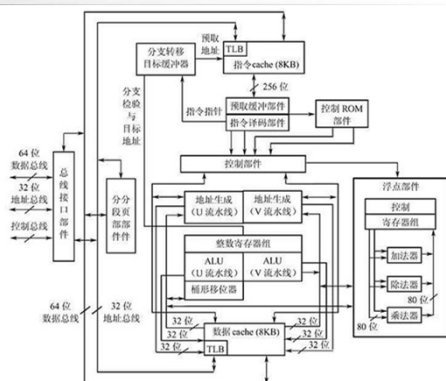
21

探针模式（probe mode）

- **R/S#（run/stop）**：异步边沿中断请求
- **PRDY（probe ready）**：探针就绪输出信号

22

Pentium微处理器内部结构



23

Pentium微处理器简介

- 引脚的功能
- 存储系统
- 输入/输出系统
- 系统时序
- 分支预测逻辑
- 高速缓存结构
- 超标量体系结构

24

存储系统

- Pentium微处理器的存储系统大小为4GB。
 - 与80386DX 和80486 微处理器的存储系统一样大小。
 - 它们之间的差别在于存储器数据总线的宽度。
- Pentium使用64 位数据总线来寻址8个存储体，每个存储体包含512MB的数据。



25

存储系统

- A31~A3与BE7#~BE0#形成32位地址

A₃₁~A₃与BE₇~BE₀形成32位地址

A ₃₁ ~A ₃					物理地址							
					BE ₇	BE ₆	BE ₅	BE ₄	BE ₃	BE ₂	BE ₁	BE ₀
A ₃₁	...	A ₃	A ₂	A ₁	A ₀	×	×	×	×	×	×	×
A ₃₁	...	A ₃	0	0	0	×	×	×	×	×	×	0
A ₃₁	...	A ₃	0	0	1	×	×	×	×	×	×	1
A ₃₁	...	A ₃	0	1	0	×	×	×	×	×	0	1
A ₃₁	...	A ₃	0	1	1	×	×	×	×	0	1	1
A ₃₁	...	A ₃	1	0	0	×	×	×	0	1	1	1
A ₃₁	...	A ₃	1	0	1	×	×	0	1	1	1	1
A ₃₁	...	A ₃	1	1	0	×	0	1	1	1	1	1
A ₃₁	...	A ₃	1	1	1	0	1	1	1	1	1	1

26

存储系统

- BE7#~BE0#与64位数据总线的对应关系

字节允许信号	数据总线信号	
BE ₀	D ₇ ~D ₀	字节 0 (最低位)
BE ₁	D ₁₅ ~D ₈	字节 1
BE ₂	D ₂₃ ~D ₁₆	字节 2
BE ₃	D ₃₁ ~D ₂₄	字节 3
BE ₄	D ₃₉ ~D ₃₂	字节 4
BE ₅	D ₄₇ ~D ₄₀	字节 5
BE ₆	D ₅₅ ~D ₄₈	字节 6
BE ₇	D ₆₃ ~D ₅₆	字节 7 (最高位)

27

存储系统

- Pentium存储系统被分为8个存储体，每个存储体都有一个检验位，使得8个存储体就可用一个字节存放校验位。
 - 与486一样，Pentium采用内部校验发生和检查逻辑来获得存储系统的数据总线信息。
 - 注意：多数Pentium系统不使用校验检查（因为ECC是可用的）。
- 64 位宽的存储器对于双精度浮点型数据是很重要的，因为双精度浮点型数据正好是64位宽。Pentium可以在一个读周期里得到浮点数据。
 - 这使得Pentium比80486的吞吐量更高。

28

存储系统

- 与早期的微处理器相似，Pentium存储系统也是以字节方式从00000000H 到FFFFFFFFH 计数的。
- 存储器选择由体允许信号（BE7# -BE0#）来完成，这些单独的存储器体使Pentium在一个存储器传送周期里可以存取单个字节、字、双字或四字的数据。
- 与早期的存储器选择逻辑一样，通常产生8个独立的写脉冲向存储器中写数据。

29

存储系统

- Pentium所添加的一个新特性是能够在特定操作中为地址总线（A31- A5）检查和产生奇偶校验。
 - AP引脚为系统提高偶校验信息。
 - APCHK引脚指示地址总线出现一个错误的奇偶校验检查。
 - 当检查到一个地址奇偶校验错误时，Pentium并不采取任何措施，此错误必须由系统获得，如果需要可由系统采取适当措施处理（例如中断）。

30

Pentium微处理器简介

- 引脚的功能
- 存储系统
- 输入/输出系统
- 系统时序
- 分支预测逻辑
- 高速缓存结构
- 超标量体系结构

31

输入/输出系统

- Pentium的I/O系统完全与早期的Intel微处理器兼容。
- I/O端口号出现在地址线A15- A3, 和使能信号(BE7#~BE0#)一起选择实际用于I/O传送的存储体。
- 从80386 微处理器开始, 当Pentium在保护模式下操作时, I/O特权信息被添加到TSS 段。
 - 注意: 这使得I/O端口可以有选择地禁止。
 - 如果一个锁定的I/O地址被访问, Pentium就产生一个13号中断来指示I/O特权冲突。

32

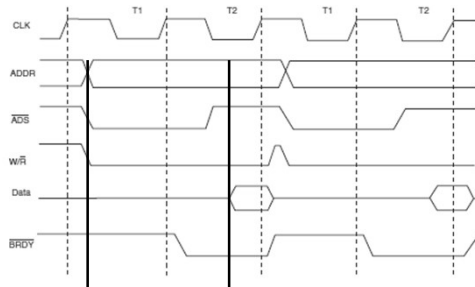
Pentium微处理器简介

- 引脚的功能
- 存储系统
- 输入/输出系统
- 系统时序
- 分支预测逻辑
- 高速缓存结构
- 超标量体系结构

33

非流水线存储周期

- 基本的Pentium非流水线存储周期包括两个时钟周期T1和T2。
- 例, 如果存储器速度足够快, 66MHz的Pentium每秒可完成3300万次存储器传送。



34

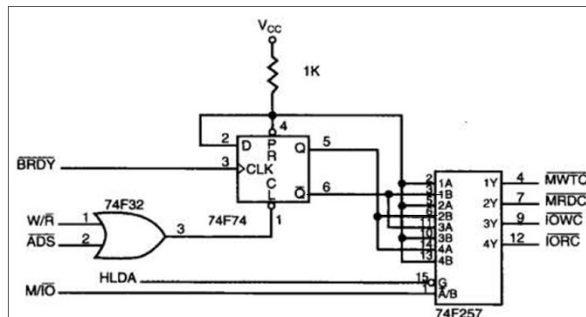
非流水线存储周期

- 如果ADS#在时钟周期的上升沿(T1末端)为逻辑0, W/R#信号有效; 必须用该时钟确定是读周期还是写周期。
- 在T1周期, 微处理器发出ADS#、W/R#、地址和M/I/O#信号。
- 为了确定W/R#信号, 并产生正确的MRDC#和MWTC#信号, 我们采用触发器来产生W/R#信号, 然后使用二选一的多路器来产生存储器和I/O控制信号。

35

非流水线存储周期

- 用于产生存储器和I/O控制信号的电路



36

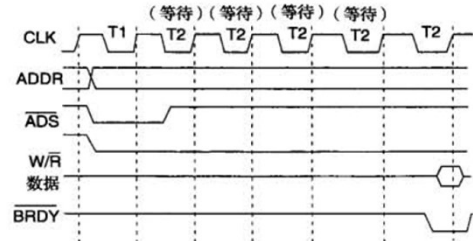
非流水线存储周期

- 在T2周期，数据总线在T2末端时钟上升沿被同步采样。
 - 时钟之前的建立时间是3.8ns。
 - 在时钟后的保持时间是2.0ns。
 - 这意味着在此时钟边缘有5.8ns的数据窗口。
- 在T1开始后最多8ns后地址信号出现。
- 这就是说，66MHz的Pentium允许的访问时间为30.3ns（两个时钟周期）减去8.0ns的地址延迟再减去3.8ns的数据准备时间。
 - 没有等待状态的存储器访问时间是 $30.3 - 8.0 - 3.8 = 18.5\text{ns}$ 。
 - 这个时间对于访问SRAM足够了。SRAM通常用在外部的二级高速缓存中。
 - 但如果不在时序中插入等待状态，这么短的时间对于任何DRAM都是不够的。

37

非流水线存储周期

- 通过控制Pentium的BRDY#输入信号可以插入等待状态。
 - 在T2结束之前，BRDY#信号必须变为逻辑0。否则，多余的T2状态就会插入到时序中。
- 例，插入4个等待状态，访问时间为79.5ns的Pentium时序图。



38

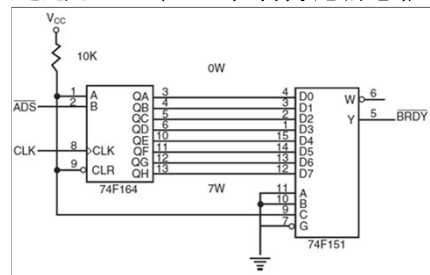
非流水线存储周期

- 向时序中插入等待状态的结果是延长了时序，以便存储器有较多的时间访问数据。
- 在所示的时序中，访问时间被延长到可以使用标准的60ns的DRAM。
 - 注意：这需要加入4个15.2ns（一个时钟周期）的等待状态，从而将访问时间延长至79.5ns。
 - 这段时间对于DRAM和译码器的工作都足够了。

39

非流水线存储周期

- BRDY#是由系统时钟产生的同步信号。
- 例，通过延迟ADS产生4个等待状态的电路。



40

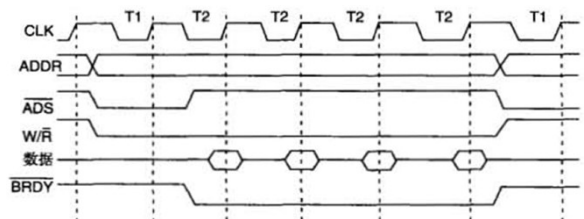
猝发周期

- 读写存储器数据的更有效方法是使用猝发（Burst）周期。
- Pentium在一个猝发周期里的5个时钟周期中可传送4个64位数。
 - 没有等待状态的猝发周期，需要存储系统每15.2ns传送一个数据。
 - 如果有二级高速缓存，获得这个速度是没有问题的，只要从高速缓存读取数据即可。
 - 如果高速缓存中没有包含所需数据，那么就必須加入等待状态，这将会降低系统的吞吐量。

41

猝发周期

- 在微处理器和存储器之间传送4个64位数据的Pentium猝发周期操作。



42

Pentium微处理器简介

- 引脚的功能
- 存储系统
- 输入/输出系统
- 系统时序
- 分支预测逻辑
- 高速缓存结构
- 超标量体系结构

43

分支预测逻辑

- 所谓分支预测是指当CPU遇到无条件或有条件转移指令、CALL调用指令、RET返回指令、INT n中断调用，以及中断返回指令IRET等跳转指令时，指令预取单元能够较准确地判断是发生转移取指，还是依据EIP指针顺序往下取指。
- 80486微处理器的指令流水线没有分支预取功能，它不能将转移指令与其他指令区别开来，预取单元只能依据IP/EIP指针顺序地取出下一条指令的代码送入预取队列，因此取出跳转指令后也总是继续读取下一条指令。

44

分支预测逻辑

- Pentium微处理器借助分支目标缓冲器BTB（branch target buffer）等逻辑部件实现了分支转移的动态预测。
 - BTB是一个具有256行的四路组合相关映射高速cache，以跳转的32位目标地址、两位历史状态及一位有效状态作为一个cache存储的内容。
 - 被预取的指令送入U和V两条流水线，同时将指令所在的EIP地址送入BTB中进行查找比较。
 - 如果在BTB中没有这个地址，就不进行预测。倘若在BTB中找到了该地址，即为命中，那么微处理器就将根据BTB中对应记录的历史状态来预测当前是否发生跳转。

45

分支预测逻辑

- Pentium微处理器借助分支目标缓冲器BTB（branch target buffer）等逻辑部件实现了分支转移的动态预测。
 -
 - 这样分支预测逻辑就能比较可靠地预测分支走向，指令预测队列就可以预取较多的指令。
 - 当执行跳转指令时，其跳转的目标地址就将用于更新BTB中相应的地址记录。
 - Pentium微处理器的分支转移动态预测功能，使得主流水线不会空闲而且大大加速了程序的执行。

46

Pentium微处理器简介

- 引脚的功能
- 存储系统
- 输入/输出系统
- 系统时序
- 分支预测逻辑
- 高速缓存结构
- 超标量体系结构

47

高速缓存结构

- 80486微处理器片内只有8KB cache，Pentium微处理器则有16KB，且将指令cache与数据cache完全分开，各为8KB，这样就完全避免了预取指令与数据两者之间的冲突。
 - 指令Cache和数据Cache分开，可以避免数据密集的程序很快占满缓存，几乎没有空间用于指令缓存。
- Pentium CPU中，指令cache与数据cache都有各自的旁路转换缓冲器TLB；存储器管理部件MMU中的分页部件就能迅速地将代码或数据的线性地址转换成物理地址。

48

Pentium微处理器简介

- 引脚的功能
- 存储系统
- 输入/输出系统
- 系统时序
- 分支预测逻辑
- 高速缓存结构
- 超标量体系结构

49

超标量体系结构

- Pentium微处理器有3个执行单元。
 - 两个（U管道、V管道）执行整型指令
 - 一个执行浮点指令
- Pentium可同时执行3条指令。
 - 例如，指令FADD ST, ST(2)，指令MOV EAX, 10H，指令MOV EBX, 12H可同时执行。
- 编写软件时，应当充分利用这个特性，对那些相互依赖而又可以分解为非依赖的指令进行修改。
 - 修改后，某些软件的执行速度可能会提高40%。
 - 注意：编译器设计！

50

本章小结

- Pentium微处理器简介
- Pentium的特定寄存器
- Pentium的存储管理
- Pentium的新指令
- Pentium Pro微处理器简介
- Pentium Pro的特性
- 了解Pentium微处理器的引脚功能、存储系统、输入/输出系统、系统时序、分支预测逻辑、高速缓存结构、超标量体系结构。

51