

总线



第5章 课件

总线是一组信号线的集合,是在计算机系统各部件之间传输地址、数据和控制信息的公共通路。微型计算机系统中,总线存在于CPU、存储器各芯片内部,也存在于各模块之间,本章将介绍总线的相关概念。

5.1 总线基本概念

总线是构成计算机系统的互连机构,是多个系统功能部件之间进行数据传送的公共通路,通过总线可以传输数据信息、地址信息、各种控制命令和状态信息。

在计算机的发展历史中,早期冯·诺依曼提出的模型并不包含总线,到微型计算机以后,才正式采用总线结构。有了总线结构以后,计算机系统的组装、维护和扩展才得以方便地进行,使系统具有了支持模块化设计、开放性、通用性和灵活性等特点。



总线基本概念

5.1.1 总线的类型与总线结构

1. 总线的类型

一个系统常常包含多种类型的总线。计算机系统的总线按其所传输信号的性质分为3类:地址总线、数据总线和控制总线。地址总线 and 数据总线相对比较简单,功能也较为单一。尽管在系统的不同层面上它们的名称和性能有所不同,但地址总线 and 数据总线的功能就是传输、交换地址信息和数据信息。控制总线差异较大,这一特点决定了各种模块的不同接口和功能特点。

整个计算机系统包含许多模块,这些模块位于系统的不同层次上,整个系统按模块进行构建。同一类型的总线在不同的层面上连接不同部位上的模块,其名称、作用、数量、电气特性和形态各不相同。按总线连接的对象和所处系统的层次来分,总线有芯片级总线、系统总线、局部总线和外部总线。芯片级总线用于模块内芯片级的互连,是该芯片与外围支撑芯片的连接总线。如连接CPU及其周边的协处理器、总线控制器、总线收



发器等的总线称为 CPU 局部总线或 CPU 总线,连接存储器及其支撑芯片的总线称为存储器总线。系统总线是连接计算机内部各模块的一条主干线,是连接芯片级总线、局部总线和外部总线的纽带。系统总线符合某一总线标准,具有通用性,是计算机系统模块化的基础。由于经过缓冲器驱动,其负载能力较强。与所连接的 CPU 和外设相比,系统总线发展滞后、速度缓慢、带宽较窄,成为数据传输的瓶颈。为了打破这一瓶颈,人们将一些高速外设从系统总线上卸下,通过控制和驱动电路直接挂到 CPU 局部总线上,使高速外设能按 CPU 速度运行。这种直接连接 CPU 和高速外设的传输通道就是局部总线。局部总线一端与 CPU 连接,另一端与高速外设和系统总线连接,好像在系统总线和 CPU 总线之间又插入一级。外部总线又称设备总线或输入输出总线,是连接计算机与外设的总线。外部总线经总线控制器挂接到系统总线上。CPU 与连接到系统板上的外设打交道须经过芯片级总线、局部总线、(系统总线)和外部总线这样 3~4 级总线。

按照允许信息传送的方向来分,总线还可分为单向传输和双向传输两种。双向传输又分为半双向和全双向两种。前者允许在某一时刻只能向其中的一个方向进行数据传送,而在另一时刻可以实现反方向的数据传送。后者允许在同一时刻进行两个方向的数据传送。全双向的速度快,但造价高,结构复杂。

按照用法,总线又可以分为专用总线和非专用总线。只连接一对物理部件的总线称为专用总线。其优点是系统的流量高,多个部件可以同时发送和接收信息,几乎不争用总线;控制简单,不用指明源部件和目的部件;任何总线的失效只会影响连接该总线的两个部件不能直接通信,但它们仍可通过其他部件间接通信,因而系统可靠。专用总线的主要缺点是总线数目多;难以小型化和集成电路化,而且总线长时成本高。另外,专用总线的时间利用率往往很低,系统的模块化也较难实现。专用总线只适用于实现某个设备(部件)仅与另一个设备(部件)相连。非专用总线可以被多种功能或多个部件所分时共享,同一时间只有一对部件可以使用总线进行通信。非专用总线的主要优点是总线数少,造价低;总线接口的标准化、模块性强,易于简化和统一接口的设计;可扩充能力强,部件的增加不会使电缆、接口和驱动电路等剧增;易于采用多重总线来提高总线的带宽和可靠性,使故障弱化。缺点是系统流量小,经常会出现总线争用,使那些未获得总线使用权的部件不得不等待而降低效率。如果处理不当,总线可能成为系统速度性能的瓶颈,对单总线结构尤其如此。

2. 总线结构

总线用来连接系统内各模块,组织方法不同,总线结构也不同。一般的总线结构有单总线结构、双总线结构和三总线结构。

单总线结构是指在许多单处理器的计算机中,使用一条单一的系统总线来连接 CPU、主存和 I/O 设备。此时要求连接到总线上的逻辑部件必须高速运行,以便在某些设备需要使用总线时能迅速获得总线控制权;而当不再使用总线时,能迅速放弃总线控制权。单总线结构容易扩展成多 CPU 系统,只需要在系统总线上挂接多个 CPU 即可。

双总线结构保持了单总线系统简单、易于扩充的优点,但又在 CPU 和主存之间专门

设置了一组高速的存储总线,使 CPU 可通过专用总线与存储器交换信息,并减轻了系统总线的负担,同时主存仍可通过系统总线与外设之间实现直接存储器存取(DMA)操作,而不必经过 CPU。当然这种双总线系统以增加硬件为代价。三总线结构是在双总线系统的基础上增加 I/O 总线形成的。

5.1.2 总线的性能

总线的性能主要从以下 3 方面来衡量:总线宽度、总线频率和传输率。

1. 总线宽度

总线宽度是指一次可以同时传输的数据位数。一般来说,总线的宽度越宽,在一定时间内传输的信息量就越大。不过在一个系统中,总线的宽度不会超过 CPU 的数据宽度。

2. 总线频率

总线频率是指总线工作时每秒内能传输数据的次数。总线的频率越高,传输的速度越快。

3. 传输率

传输率是指每秒内能传输的字节数,用 MB/s 来表示。

传输率和宽度、频率之间的关系是

$$\text{传输率} = \text{宽度} / 8 \times \text{频率}$$

【例 5.1】 设总线宽度为 32b,频率为 100MHz,求传输率。

根据上述公式,即

$$\text{传输率} = 32\text{b} / 8 \times 100\text{MHz} = 400\text{MB/s}$$

又如,PCI 总线的宽度为 32 位,总线频率为 33MHz,所以,PCI 的数据传输率为 132MB/s。

总线宽度越宽,频率越高,则传输率越高。

5.1.3 总线信息的传送方式

数字计算机使用二进制数,它们或用电位的高、低来表示,或用脉冲的有、无来表示。计算机系统中,总线传输方式即总线通信方式,俗称为总线握手方式。总线信息的传送方式一般有 3 种:串行传送、并行传送和分时传送。出于速度和效率上的考虑,系统总线上传送的信息必须采用并行传送方式。

1. 串行传送

当信息以串行方式传送时,只有一条传输线,且采用脉冲传送。在串行传送时,按顺序传送来表示一个数的所有二进制位(bit)的脉冲信号,每次一位。通常以第一个脉冲信



号表示数的最低有效位,最后一个脉冲信号表示数的最高有效位。进行串行传送时,被传送的数据需要在发送部件进行并-串变换,而在接收部件又需要进行串-并变换。串行传送的主要优点是只需要一条传输线,这一点对长距离传输显得特别重要,不管传送的数据量有多少,只需要一条传输线,成本比较低廉。

2. 并行传送

用并行方式传送二进制信息时,对要传送的数的每个数据位都需要单独一条传输线。信息由多少二进制位组成,就需要有多少条传输线,从而使得二进制数 0 或 1 在不同的线上同时进行传送。

并行传送一般采用电位传送。由于所有的位同时被传送,所以并行数据传送比串行数据传送快得多。

3. 分时传送

分时传送有两种概念。一是采用总线复用方式,某个传输线上既传送地址信息,又传送数据信息。为此必须划分时间片,以便在不同的时间间隔中完成传送地址和传送数据的任务。分时传送的另一种概念是共享总线的部件分时使用总线。

5.2 32 位微处理器的外部引脚与总线时序

理解微处理器的引脚功能是微型计算机系统中微处理器与存储器和微处理器与 I/O 接口进行连接的重要基础之一,本节主要论述 32 位微处理器的引脚功能。

5.2.1 Pentium 微处理器的引脚功能

Pentium 由于增加了许多功能,使得信号数量大大增加,芯片有 168 个引脚,介绍如下。

1. 地址线及控制信号

- (1) $A_{31} \sim A_3$: 地址线。
- (2) AP: 地址的偶校验码位。
- (3) $\overline{ADS}$: 地址状态输出信号。
- (4) $\overline{A_{20}M}$: A_{20} 以上的地址线屏蔽信号。
- (5) $\overline{APCHK}$: 地址校验出错信号。

由于 Pentium 有片内 Cache,所以地址线是双向的,既能对外选择主存和 I/O 设备,也能对内选择片内 Cache 的单元。Pentium 的 32 位地址线可寻址 4GB 内存和 64KB 的 I/O 空间,32 位地址信号中、低 3 位地址 $A_2 \sim A_0$ 组合成字节允许信号 $\overline{BE_7} \sim \overline{BE_0}$,所以, $A_2 \sim A_0$ 不对外。

当 $A_{31} \sim A_3$ 有输出时,AP 上输出偶校验码,供存储器对地址进行校验。在读取

Cache 时, Pentium 会对地址进行偶校验, 如校验有错, 则地址校验信号 $\overline{\text{APCHK}}$ 输出低电平。 $\overline{\text{ADS}}$ 为地址状态信号, 它表示 CPU 已启动一个总线周期。

$\overline{\text{A}}_{20}\text{M}$ 信号是所有与 ISA 总线兼容的计算机系统中必须有的信号, 当此信号为 0 时, 将屏蔽第 20 位以上的地址, 以便在访问 Cache 和主存时可访问 1MB 存储空间。

2. 数据线及控制信号

(1) $\text{D}_{63} \sim \text{D}_0$: 数据线。

(2) $\overline{\text{BE}}_7 \sim \overline{\text{BE}}_0$: 字节允许信号。

(3) $\text{DP}_7 \sim \text{DP}_0$: 奇偶校验信号。

(4) $\overline{\text{PCHK}}$: 读校验出错。

(5) $\overline{\text{PEN}}$: 奇偶校验允许信号, 若输入为低电平, 则在读校验出错时处理器会自动进行异常处理。

Pentium 对外用 64 位数据线, 所以数据总线 $\text{D}_{63} \sim \text{D}_0$, 并增加了奇偶校验。在对存储器进行读写时, 每字节产生一个校验位, 通过 $\text{DP}_7 \sim \text{DP}_0$ 输出, 而读操作时, 则按字节进行校验。 $\overline{\text{PCHK}}$ 信号在读校验出错时为 0, 以便送外部电路告示校验出错。

$\overline{\text{BE}}_7 \sim \overline{\text{BE}}_0$ 为字节允许信号, 对应 8 字节 (即 64 位) 数据。

3. 总线周期控制信号

(1) $\text{D}/\overline{\text{C}}$: 数据/控制信号。高电平表示当前总线周期传输的是数据, 低电平表示当前总线周期传输的是指令。

(2) $\text{M}/\overline{\text{IO}}$: 存储器和 I/O 访问信号。高电平时访问存储器, 低电平时访问 I/O 端口。

(3) $\text{W}/\overline{\text{R}}$: 读写信号。高电平时表示当前总线周期进行写操作, 低电平时则为读操作。

(4) $\overline{\text{LOCK}}$: 总线封锁信号。低电平有效, 此时将总线锁定, $\overline{\text{LOCK}}$ 信号由 LOCK 指令的前缀设置, 总线被锁定时使得其他总线主设备不能获得总线控制权, 从而确保 CPU 完成当前操作。

(5) $\overline{\text{BRDY}}$: 突发就绪信号。表示结束一个突发总线传输周期, 此时外设处于准备好状态。

(6) $\overline{\text{NA}}$: 下一个地址有效信号。低电平有效, 从此端输入低电平时, CPU 会在当前总线周期完成之前就将下一个地址送到总线上, 从而开始下一个总线周期, 构成总线流水线工作方式, Pentium 允许 2 个总线周期构成总线流水线。

(7) SCYC: 分割周期信号。表示当前地址指针未对准字、双字或四字的起始字节, 因此, 要采用 2 个总线周期完成数据传输, 即对周期进行分割。

$\text{M}/\overline{\text{IO}}$ 、 $\text{D}/\overline{\text{C}}$ 和 $\text{W}/\overline{\text{R}}$ 信号与 80386 的对应信号相同。 $\overline{\text{BRDY}}$ 和 $\overline{\text{RDY}}$ 信号类似, $\overline{\text{RDY}}$ 信号有效, 表示结束一个普通传输周期; $\overline{\text{BRDY}}$ 有效, 表示结束一个突发传输周期。在 $\overline{\text{RDY}}$ 和 $\overline{\text{BRDY}}$ 均有效时, CPU 会忽略 $\overline{\text{BRDY}}$ 。

4. Cache 控制信号

(1) $\overline{\text{CACHE}}$: Cache 控制信号。在读操作时,如果此信号输出低电平,表示主存中读取的数据正在送入 Cache;写操作时,如果此信号为低电平,表示 Cache 中修改过的数据正写回到主存。

(2) $\overline{\text{EADS}}$: 外部地址有效信号。此信号为低电平时外部地址有效,此时可访问片内 Cache。

(3) $\overline{\text{KEN}}$: Cache 允许信号。确定当前总线周期传输的数据是否送到 Cache。

(4) $\overline{\text{FLUSH}}$: Cache 擦除信号。此信号有效时,CPU 强制对片内 Cache 中修改过的数据回写到主存,然后擦除 Cache。

(5) $\overline{\text{AHOLD}}$: 地址保持/请求信号。高电平有效,用以强制 CPU 浮空地址信号,为 $A_{31} \sim A_4$ 输入地址访问 Cache 做准备。

(6) $\overline{\text{PCD}}$: Cache 禁止信号。高电平时,禁止对片外 Cache 的访问。

(7) $\overline{\text{PWT}}$: 片外 Cache 的控制信号。高电平时使 Cache 为通写方式,低电平时为回写方式。

(8) $\overline{\text{WB}}/\overline{\text{WT}}$: 片内 Cache 回写/通写选择信号。此信号为 1 则为回写方式,为 0 则为通写方式。

(9) $\overline{\text{HIT}}$ 和 $\overline{\text{HITM}}$: Cache 命中信号和命中 Cache 的状态信号。 $\overline{\text{HIT}}$ 低电平时,表示 Cache 被命中。 $\overline{\text{HITM}}$ 低电平时,表示命中的 Cache 被修改过。

(10) $\overline{\text{INV}}$: 无效请求信号。此信号为高电平时,使 Cache 区域不可再使用而成为无效。

如果外部存储器子系统将 $\overline{\text{KEN}}$ 信号设置为低电平,就会在存储器读周期中将数据复制到 Cache。

$\overline{\text{PCD}}$ 和 $\overline{\text{PWT}}$ 是用来控制片外 Cache 的。 $\overline{\text{PCD}}$ 信号用来向外接 Cache 告示,当前访问的页面已在片内 Cache 中,所以,不必启用外接 Cache。 $\overline{\text{PWT}}$ 信号有效时,对外接 Cache 按通写方式操作,否则按回写方式操作。

$\overline{\text{AHOLD}}$ 和 $\overline{\text{EADS}}$ 信号用来保证 Cache 数据的一致性。这种情况发生在 DMA 传输中,主存和外设直接交换数据,当主存中某个数据被修改时,如果这两个信号有效,Pentium 会马上检查此处原来的数据是否在 Cache 中,如是,则应使 Cache 中的数据无效,以保证数据的正确性和一致性。为此,主存系统在写操作后,通过外部电路将 $\overline{\text{AHOLD}}$ 置 1,Pentium 收到此信号以后,使地址处于高阻状态即无效状态,然后,外部电路把被修改单元的地址送到地址总线,并将 $\overline{\text{EADS}}$ 置 0,使外部地址有效,此时,Cache 系统如检测到 Cache 有此地址的数据,则会做无效处理,从而保证 Cache 和主存的数据保持一致。

$\overline{\text{HIT}}$ 、 $\overline{\text{HITM}}$ 和 $\overline{\text{INV}}$ 用于一种特殊的称为询问周期的操作,在这种总线周期中,通过一个专用端口查询数据 Cache 和指令 Cache,以确定当前地址是否命中 Cache。如果命中,则 $\overline{\text{HIT}}$ 为低电平;如果不命中 Cache,而且此数据已修改过,则 $\overline{\text{HITM}}$ 也为低电平。而 $\overline{\text{INV}}$ 端输入高电平时,使 Cache 不能访问。

5. 系统控制信号

- (1) INTR：可屏蔽中断请求信号。
- (2) NMI：非屏蔽中断请求信号。
- (3) RESET：系统复位信号。
- (4) INIT：初始化信号。
- (5) CLK：系统时钟信号。

INIT 信号和 RESET 信号类似，都用于对 CPU 处理器做初始化，但两者有区别。RESET 有效时，会使处理器在两个时钟周期内终止程序，即进行复位，而 INIT 有效时，处理器先将此信号锁存，直到当前指令结束时才执行复位操作。另外，用 INIT 信号复位时，只对基本寄存器进行初始化，而 Cache 和浮点寄存器中的内容不变。但不管是用 RESET 信号还是用 INIT 信号，系统复位以后，程序均从 FFFFFFF0H 处重新开始运行。复位后微处理器内部寄存器的值如表 5.1 所示。

表 5.1 复位后微处理器内部寄存器的值

寄 存 器	初 始 值	寄 存 器	初 始 值
EAX	不定	ES	0000H
EBX	不定	CS	F000H
ECX	不定	SS	0000H
EDX	0400H+版本 ID	DS	0000H
EBP	不定	FS	0000H
ESP	不定	GS	0000H
EDI	不定	IDTR	基址=0,界限=3FFH
ESI	不定	CR0	6000~0000H
EFLAGS	0000~0002H	DR7	0000~0000H
EIP	0FFF0H	浮点寄存器	不变

6. 总线仲裁信号

- (1) HOLD：总线请求信号。这是其他总线主设备请求 CPU 让出总线控制权的信号。
- (2) HLDA：总线请求响应信号。这是对 HOLD 的回答信号，表示 CPU 已让出总线控制权。
- (3) BREQ：总线周期请求信号。此信号有效时，向其他总线主设备告示，CPU 当前已提出一个总线请求，并正在占用总线。
- (4) $\overline{\text{BOFF}}$ ：强制让出总线信号。此信号强制 CPU 让出总线控制权，CPU 接到此信号时，立即放弃总线控制权，直至此信号无效时，CPU 再启动被打断的总线周期。

$\overline{\text{BOFF}}$ 和 HOLD 有类似之处，但有两点不同：一是 $\overline{\text{BOFF}}$ 会使当前时钟周期一结束即让出总线控制权，此时总线周期并没结束，而 HOLD 则在当前总线周期结束时才让出总线控制权，所以可能还会持续一个或几个时钟周期，动作较慢；二是 $\overline{\text{BOFF}}$ 没有对应的

响应信号,而 HOLD 有对应的响应信号 HLDA。外部总线主设备可用 $\overline{\text{BOFF}}$ 信号快速获得总线控制权。

7. 检测与处理信号

- (1) $\overline{\text{BUSCHK}}$: 转入异常处理的信号。
- (2) $\overline{\text{FERR}}$: 浮点运算出错的信号。
- (3) $\overline{\text{IGNNE}}$: 忽略浮点运算出错的信号。低电平有效,此时 CPU 会忽略浮点运算错误。
- (4) $\overline{\text{FRCMC}}$: 输入此信号会使 CPU 进行冗余校验。
- (5) $\overline{\text{IERR}}$: 冗余校验出错信号。与 $\overline{\text{FRCMC}}$ 配合使用,此信号有效表示冗余校验出错。

$\overline{\text{BUSCHK}}$ 信号由外部电路输入,外部电路在检测到当前总线周期未正常结束时,将此信号置于低电平,此后,CPU 会采样此信号,如为低电平,则使当前错误总线周期结束转入异常处理。

CPU 在 RESET 信号由高到低时,对 $\overline{\text{FRCMC}}$ 已采样,如采样到低电平,则 CPU 进入冗余校验状态;如校验出错,则 $\overline{\text{IERR}}$ 输出低电平。

8. 系统管理模式信号

- (1) $\overline{\text{SMI}}$: 系统管理模式中断请求信号。这是对进入系统管理模式的中断请求。
 - (2) $\overline{\text{SMIACT}}$: 系统管理模式信号。这是对 $\overline{\text{SMI}}$ 信号的响应信号,当 $\overline{\text{SMI}}$ 中断请求有效时,CPU 输出 $\overline{\text{SMIACT}}$ 表示中断请求成功,当前已处于系统管理模式。
- $\overline{\text{SMI}}$ 用来进入系统管理模式,要退出系统管理模式时,可用 RSM 指令。

9. 测试信号

- (1) TCK: 从此端输入测试时钟信号。
- (2) TDI: 用来输入串行测试数据。
- (3) TDO: 此端获得输出的测试数据结果。
- (4) TMS: 用来选择测试方式。
- (5) $\overline{\text{TRST}}$: 测试复位,退出测试状态。

10. 跟踪和检查信号

- (1) $\text{BP}_3 \sim \text{BP}_0$ 以及 $\text{PM}_1 \sim \text{PM}_0$: $\text{BP}_3 \sim \text{BP}_0$ 是与调试寄存器 $\text{DR}_3 \sim \text{DR}_0$ 中的断点相匹配的外部输出信号, $\text{PM}_1 \sim \text{PM}_0$ 是性能监测信号。
- (2) $\text{BT}_3 \sim \text{BT}_0$: 分支地址输出信号, $\text{BT}_2 \sim \text{BT}_0$ 上输出分支地址的最低 3 位。
- (3) IU: 高电平有效,表示此时 U 流水线完成指令的执行过程。
- (4) IV: 高电平有效,表示此时 V 流水线完成指令的执行过程。
- (5) IBT: 指令发生分支。
- (6) $\text{R}/\overline{\text{S}}$: 探针信号输入端,此信号从高到低的跳变会使处理器停止执行指令进入空

闲状态。

(7) PRDY: 这是对 $R/\bar{S}$ 的响应信号, 输出高电平时表示 CPU 当前停止执行指令, 从而可以进入测试。

IU、IV、IBT 都是输出信号, 可通过对其电平的检测来跟踪指令的执行。PM₁~PM₀ 和 BP₁~BP₀ 是复用的, 由调试寄存器 DR₇ 中的 GE 和 LE 两位确定, 如两者为 1, 则为 BP₁~BP₀, 否则为 PM₁~PM₀。

5.2.2 32 位微处理器的典型总线操作时序

1. 时钟周期、总线周期和指令周期

任何计算机系统都具有时钟信号, 它为系统工作提供了时序基准。微处理器内部的部件和子部件往往是以时钟信号作为启动条件。因此, 计算机内部的时钟信号必须是一个有规律的脉冲信号。

时钟信号通常又称为节拍脉冲, 它的周期称为时钟周期(Clock Cycle)或 T 周期, 是处理器处理操作的最基本单位。时钟周期是 CPU 的时间基准, 它由计算机的主频决定, 大小等于主频的倒数。例如, 某 CPU 的主频 $f = 5\text{MHz}$, 则其时钟周期 $T = 1/f = 1/(5\text{MHz}) = 200\text{ns}$ ($1\text{ns} = 10^{-9}\text{s}$)。若主频为 100MHz , 时钟周期为 10ns 。

若干个时钟周期则可组成一个总线周期(Bus Cycle), 总线周期是指 CPU 通过外部总线对存储器或 I/O 端口进行一次读写操作所需要的时间。为了完成对存储器或者 I/O 端口的一次访问, CPU 需要先后发出存储器或者 I/O 端口地址, 发出读写操作命令, 进行数据的传输。以上的每一个操作都需要延续一个或几个时钟周期, 所以, 一个总线周期通常由多个时钟周期组成, 一个时钟周期对应一个总线状态, 状态(State)又称为 T 。所以, 一个总线周期由多个 T 状态组成。8086 CPU 的总线周期至少由 4 个时钟周期组成, 即有 4 个总线状态, 分别以 T_1 、 T_2 、 T_3 和 T_4 表示。80486 CPU 的总线周期至少由两个时钟周期组成, 分别以 T_1 和 T_2 表示。

一个总线周期完成一次数据传输, 至少要有地址传送和数据传送两个过程。在第一个时钟周期 T_1 期间由 CPU 输出地址, 在随后的 T 周期则完成数据的传输。换言之, 数据传送必须在 $T_2 \sim T_4$ 内完成, 否则, 在最后一个 T 周期结束以后, 会进入下一个总线周期。在实际应用中, 当一些慢速设备在规定的几个 T 周期内无法完成数据读写时, 那么就必须在总线周期中插入等待周期 T_w , T_w 也是以时钟周期 T 为单位, 但加入 T_w 的个数则与外部请求信号的持续时间长短有关。

CPU 每条指令的执行都由取指令、译码和执行等操作组成, CPU 读取并执行一条指令所花费的时间称为指令周期(Instruction Cycle), 一个指令周期由若干个总线周期组成。取指令需要一个或多个总线周期, 如果指令的操作数来自内存, 则需要另一个或多个总线周期取出操作数, 如果要把结果写回内存, 还要增加总线周期。因此, 不同指令的指令周期长度各不相同。指令周期一般由若干个处理器周期组成。

2. Pentium 总线周期的时序分析

不同类型的 32 位微处理器, 其总线周期也不相同。80486 微处理器的一般总线周期

占用 2 个时钟的时间,即读或写都要 2 个时钟,这称为 2-2 周期。第一个 2 对应读,第二个 2 对应写。如果在读或写中增加了等待状态,则在读写的对应位置加上等待状态数。例如,写操作需增加一个等待状态,则称为 2-3 周期。而 Pentium 微处理器的总线周期则更复杂。

1) Pentium 的总线状态

Pentium 有多种总线状态,各个状态之间是可以转换的。

(1) T_1 状态:这是总线周期的第 1 个时钟周期即第 1 个状态,此时,地址和状态信号有效, $\overline{ADS}$ 信号也有效,同时,外部电路可以将地址和状态送入锁存器。

(2) T_2 状态:此时数据出现在数据总线上,CPU 对 $\overline{BRDY}$ 信号采样,如果 $\overline{BRDY}$ 信号有效,则确定当前周期为突发式总线周期,否则为单数据传输的普通总线周期。

(3) T_{12} 状态:这是流水线式总线周期中所特有的状态,此时系统中有 2 个总线周期并行进行,第 1 个总线周期进入 T_2 状态,正在传输数据,并且 CPU 采样 $\overline{BRDY}$ 信号,第 2 个总线周期进入 T_1 状态,地址和状态信号有效,并且 $\overline{ADS}$ 信号也有效。

(4) T_{2p} 状态:这是流水线式总线周期中所特有的状态,此时系统中有 2 个总线周期,第 1 个总线周期正在传输数据,并且 CPU 对 $\overline{BRDY}$ 采样,但由于外设或存储器速度较慢,所以, $\overline{BRDY}$ 仍未有效,也因此仍未结束总线周期,第 2 个总线周期也进入第 2 个或后面的时钟周期。 T_{2p} 一般出现在外设或存储器速度较慢的情况下。

(5) T_D 状态:这是 T_{12} 状态后出现的过渡状态,一般出现在读写操作转换的情况下,此时数据总线需要一个时钟周期进行过渡,这种状态下,数据总线上的数据还未有效,CPU 还未对 $\overline{BRDY}$ 进行采样。

(6) T_i 状态:这是空闲状态,不在总线周期中, $\overline{BOFF}$ 信号或 RESET 信号会使 CPU 进入此状态。

2) Pentium 的总线周期

Pentium 支持多种数据传输方式,可以是单数据传输方式,也可以是突发式传输方式。单数据传输时,一次读写操作至少要用 2 个时钟周期,可进行 32 位数据传输,也可进行 64 位数据传输。突发式传输方式是 80486/Pentium 特有的一种新型传输方式,用这种方式传输时,在一个总线周期中可传输 256 位数据。与此相应,Pentium 的总线周期有多种类型。

按总线周期之间的组织方法来分,有流水线和非流水线类型。在流水线类型中,前一个总线周期中已为下一个总线操作进行地址传输;而在非流水线类型中,每个总线周期独立进行一次完整的读操作或写操作,与其他总线周期无关。

按总线周期本身的组织方法来分,有突发式传输和非突发式传输类型。突发式传输时,连续 4 组共 256 位数据可在 5 个时钟周期中完成传输,这样可以加快对主存的信息存取。非突发式传输时,通常用 2 个时钟周期构成一个总线周期传输单个数据,可为 8 位、16 位、32 位或 64 位。

下面对常用的非流水线式读写周期进行说明。

这种总线周期至少每个占用 2 个时钟周期,即 T_1 和 T_2 ,在外设或存储器较慢时,则