

1.1 名词解释

1. 位(bit)、字节(byte)、字(word)

位是指能够表示 0 或 1 值的一位二进制数字。

字节是指由 8 位二进制数字构成的位组。

字也是位组单位,指出某个特定计算机内部一次可以处理的二进制数字的位数,它通常是字节的倍数。

2. K 字节、M 字节、G 字节、T 字节

$1\text{K 字节} = 2^{10}\text{字节} = 1024\text{字节}$

$1\text{M 字节} = 2^{10}\text{K 字节} = 2^{20}\text{字节}$

$1\text{G 字节} = 2^{10}\text{M 字节} = 2^{30}\text{字节}$

$1\text{T 字节} = 2^{10}\text{G 字节} = 2^{40}\text{字节}$

3. CPU、ALU、RAM、ROM

中央处理部件(central processing unit,CPU)又称微处理器,是一个包含了算术逻辑部件;累加器和通用寄存器组;程序计数器(指令指针)、指令寄存器和译码器;时序和控制部件的单个集成电路芯片。

算术逻辑部件(arithmetic/logical unit,ALU)是 CPU 内部用于完成算术、逻辑运算的部件。

随机访问存储器(random access memory,RAM)也称读写存储器,用来临时存放程序和数据,断电后存放的程序和数据将丢失,因此被称为挥发性存储器。

只读存储器(read-only memory,ROM)用于存放对计算机操作十分重要的程序和消息,存放在 ROM 中的信息是永久的,断电后不会丢失。

4. CISC、RISC

复杂指令系统计算机(complex instruction set computer,CISC)指的是这样一种计算机系统,设计指令系统时由于考虑了每一种可能的情况,使得指令系统所包含的指令不仅多而且复杂,这也使得用于实现这样指令系统的计算机结构十分复杂。但程序员在实际

编程时通常只用到其中一部分,许多功能复杂的指令并不经常在程序中出现,导致硬件成本的浪费。

精简指令系统计算机(reduced instruction set computer, RISC)指的是指令系统所包含的指令较少且功能简单的计算机系统,这样的指令系统通常可以直接由硬件实现,因此执行速度较快,另外,由于实现精简指令系统的硬件开销较少,可以将实现精简指令系统的通用 CPU 和类似图形处理器这样的专用 CPU 集成在同一个芯片上,增强该芯片的功能。

1.2 例题解析

1. 简要回答下列问题

(1) 微处理器、微型计算机和微型计算机系统三者之间有何不同?

【答案】 微处理器通常也称 CPU,是内部集成了算术/逻辑部件和控制部件的集成电路芯片。微型计算机由 CPU、存储器、输入输出电路和系统总线构成,而微型计算机系统是以微型计算机为主体,配上系统软件和外设之后构成的。

(2) 微型计算机采用总线结构有什么优点?

【答案】 微型计算机采用总线结构后,使得系统中各功能部件之间的相互关系变为各个部件面向总线的单一关系,一个部件只要符合总线标准,就可以连接到采用这种总线标准的系统中去,使系统功能得到扩展。

(3) 数据总线和地址总线在结构上有什么不同? 如果一个系统的数据和地址合用一套总线或者合用部分总线,那么,靠什么来区分地址和数据?

【答案】 由于既可以将数据从 CPU 传送到其他部件,也可以从其他部件发送到 CPU,因此,数据总线是双向的。由于只能由 CPU 向其他部件发送地址,所以地址总线是单向的。如果数据和地址必须合用一套总线(或合用部分总线)必须在 CPU 外部对地址进行锁存,CPU 用地址锁存信号来区分总线上的信息。由于锁存地址信息需要时间,因此,采取合用或部分合用一套总线的 CPU 的总线周期一般都较长。

(4) 数据总线和地址总线宽度有什么意义?

【答案】 数据总线负责在 CPU 和其他设备之间传输信息,它的宽度决定了 CPU 在单个总线周期内(或单次读写操作)和其他设备交换的二进制位数,如 16 位数据总线意味着 CPU 和其他设备一次交换的二进制位数最多为 16,数据总线宽度是 CPU 性能指标之一,数据总线宽度越大,CPU 性能越好,通常情况下,CPU 数据总线宽度和它的字长一致。

地址总线宽度决定了 CPU 能够访问的物理存储器空间,如果存储单元为字节,则 x 位地址总线宽度决定了 CPU 允许访问的最大物理存储器空间为 2^x 字节,如地址总线宽度为 20 位,则最大物理存储器空间为 2^{20} 字节 = 1M 字节。

(5) 控制总线传输的信号有哪几种?

【答案】 控制总线传输的控制信号包括两大类,一类是 CPU 送往存储器和输入输

出接口电路的控制信号,如读信号、写信号和中断响应信号等,另一类是其他部件送给 CPU 的信号,如时钟信号、中断请求信号和准备就绪信号等。

2. 运算题

(1) 将下述数据转换成 8 位补码形式

98 -98

【解析】 根据 8 位补码计算公式:

$$[x]_{补} = \begin{cases} x, & 0 \leq x \leq 127 \\ 2^8 + x, & -128 \leq x < 0 \end{cases}$$

$[98]_{补} = [98]_B = 01100010$

$[-98]_{补} = [256 + (-98)]_B = [158]_B = 10011110$

其实求 $[-98]_{补}$ 时,可以先求出 $[98]_{补}$,然后从右边数起,所有的 0 及第一个 1 不变,其余全部求反。

$[98]_{补} = 011000 | 10$

$[-98]_{补} = 100111 | 10$

(2) 完成下述十六进制运算:

23D9H + 94BEH、759FH - 42B8H

【解析】

$$\begin{array}{r} 23D9 \\ + 94BE \\ \hline B897 \end{array}$$

十六进制加法和十进制加法非常相似,从低位开始逐位相加,相加结果超过 16 后,向上进位,每进 1,从结果中减去 16。

9H + EH = 9 + 14 = 23, 23 - 16 = 7 向上进一位,余为 7

DH + BH + 1 = 13 + 11 + 1 = 25, 25 - 16 = 9 向上进一位,余为 9

3 + 4 + 1 = 8 和没有超过 16,不产生进位

2 + 9 = 11 = BH 和没有超过 16,不产生进位

$$\begin{array}{r} 759F \\ - 42B8 \\ \hline 32E7 \end{array}$$

FH - 8 = 15 - 8 = 7 够减,不需要借位

9 + 16 - 11 = 14 = EH 不够减,向上借一位,借一位增加 16

5 - 2 - 1 = 2 够减,不需要借位

7 - 4 = 3 够减,不需要借位

2.1 知 识 要 点

2.1.1 8086 引脚和时序

在开始学习微处理器 8086 时,理解 8086 的引脚和时序是比较困难的,困难之处在于:为何要设计这些引脚? 如何理解引脚时序?

下面以 CPU 最小方式下读写存储器为例,详细讨论 8086 引脚的功能和时序。如图 2.1 所示。

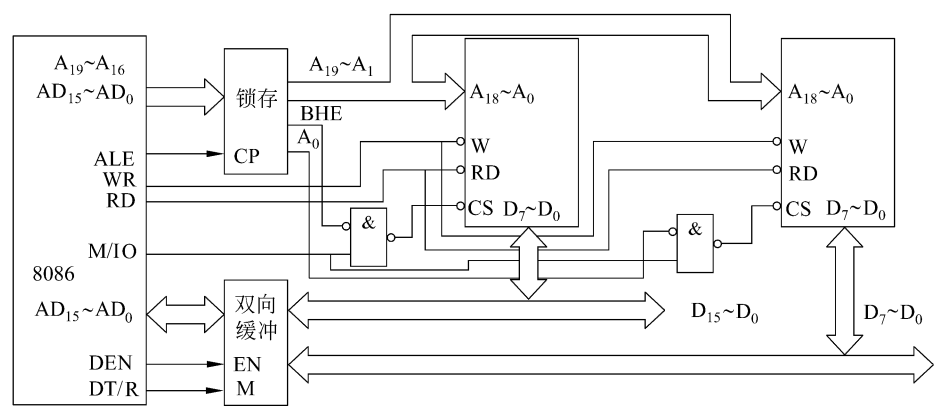


图 2.1 CPU 在最小方式下和存储器的连接图

CPU 的引脚配置主要是为了实现和存储器、I/O 接口之间的数据交换。对于存储芯片而言,要实现读写操作,外部必须对其施加信号,而且这些信号之间必须满足时间关系(时序)。图 2.2 是一个 512K×8 位的存储芯片的引脚分布和读写时序要求。

存储芯片的读时序表明:在进行读操作之前,首先必须在地址线上输入有效地址,同时使片选信号有效,然后使读信号有效,对由输入地址指定的存储单元进行读操作。从开始读操作到将指定存储单元内容(8 位数据)送到外部数据引脚上,需要一定的时间,这个时间就是存储芯片的读出时间,不同类型的存储芯片有不同的读出时间。

CPU 如何通过控制外部引脚的时序,完成对指定存储单元的读操作呢?

8086 的每一个总线周期至少包含 4 个时钟周期,为保证 CPU 能够读到数据,图 2.1

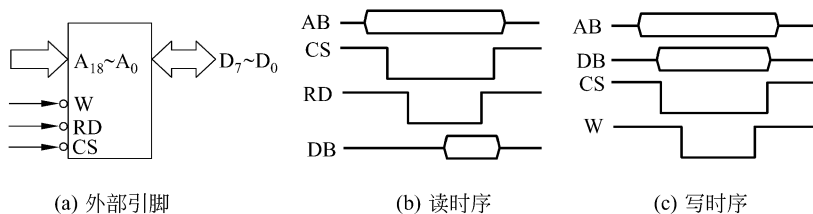


图 2.2 存储芯片引脚及时序

系统必须在第 4 个时钟周期(T4)开始前,将指定存储单元数据送到 CPU 的数据线上,从图 2.1 中可以看出:首先必须将数据从指定存储单元读出,并经过双向缓冲器(如 74LS245)送到 CPU 数据引脚($AD_{15} \sim AD_0$)上。为了能够从指定存储单元读出数据,按照图 2.2(b)所示时序,必须对存储芯片输入正确的地址信号、片选信号和读操作信号,而且保证这些信号必须一直有效,直到外部数据引脚上的数据被取走。从图 2.1 中可以看到,20 位地址信号和 BHE 均输入到锁存器(如 74LS373)输入端上,锁存器的 CP 端接 8086 的 ALE,这就要求在 ALE 发生低跳变时,锁存器的输入端必须为有效地址,而当 ALE 变成低电平后,连接锁存器输入端的信号可以作为其他用途。从图 2.3 中可以看出,ALE 发生低跳变时, $A_{19} \sim A_{16}$ 、 $AD_{15} \sim AD_0$ 、BHE 上均是有效地址信息,因此,在 ALE 变成低电平后,锁存器的输出一直维持有效的地址信息不变。假定本次读操作为 16 位数据传输,地址信息中的 $\overline{BHE} = 0$ 、 $A_0 = 0$,由于 $M/\overline{IO}$ 为 1,在锁存器输出有效地址信息后,两组存储芯片的 CS 为低电平,这样,在读操作信号有效($\overline{RD}$)后,两组存储芯片同时开始读操作。

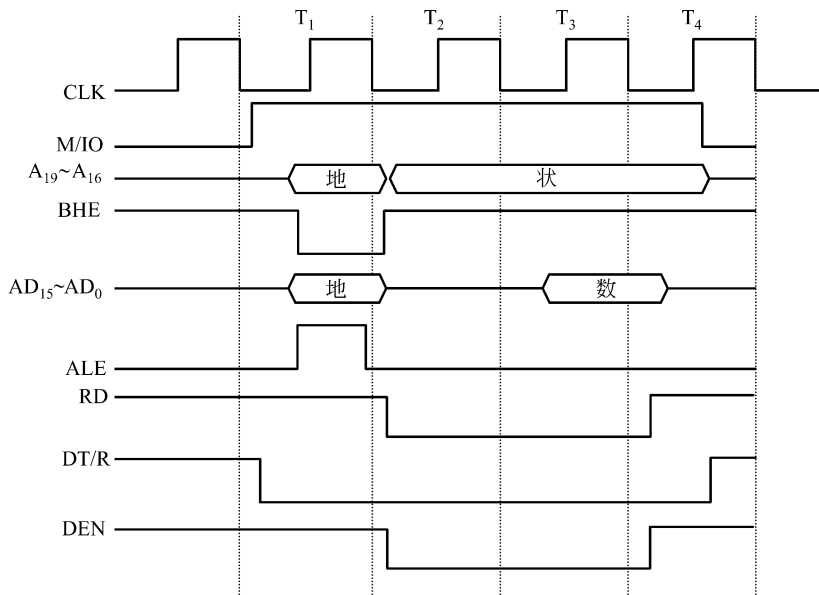


图 2.3 CPU 最小方式下读时序

当存储芯片完成读操作,指定存储单元内容送到存储芯片外部数据引脚上,为将存储

芯片数据引脚上的数据发送到 CPU 数据引脚上,必须使双向缓冲器的 EN 端为低电平, M 端为低电平,而双向缓冲器的 EN 端和 M 端连接到 8086 的 DEN 和 $\overline{DT}/\overline{R}$ 引脚,因此,必须在 T4 开始前,将 DEN 和 $\overline{DT}/\overline{R}$ 信号置成低电平。

可以粗略估计,CPU 从发出读操作有效($\overline{RD}=0$),到 T4 开始时从外部数据引脚将数据打入内部寄存器,大约有两个时钟时期的时间,存储芯片必须在这两个时钟周期内完成从指定存储单元读出数据的操作,并通过存储芯片外部数据引脚和双向缓冲器将数据发送到 CPU 数据引脚上。如果存储芯片的读出时间较长,而 CPU 的主频较高,有可能导致在 T4 开始时,图 2.1 系统无法把正确的数据发送到 CPU 外部数据引脚上,如果这样,可以在 T4 之前插入若干个等待周期 T_w ,插入等待周期由 Ready 引脚实现。

2.1.2 8086 中断处理过程

1. 中断概念的提出

用一个比喻可以说明中断的重要性:有同学甲和同学乙两人,同学甲希望和同学乙一起去看电影,但同学乙称目前必须先洗衣服,但一旦洗完衣服就径自去看电影,不会告知同学甲。同学甲为了达到和同学乙一起去看电影的目的,必须在同学乙洗衣服时守候在旁,不能利用这一段时间处理一下自己的事情。如果同学乙愿意在洗完衣服后通知同学甲一下,同学甲就可以在同学乙洗衣服的这段时间,处理其他事情,但一旦同学乙通知他,他必须立即放下目前正在处理的事情,和同学乙一齐去看电影。

在微机系统中,同学乙就是外设,同学甲就是 CPU,CPU 为了及时读取外设输入的数据或及时发送数据给外设,必须时刻检测外设状态。在外设状态变化时,及时作出处理。但如果外设状态变化时能够主动通知 CPU,CPU 就不需要时刻检测外设状态,只需在接收到外设状态改变的通知后,及时对外设进行处理即可。这种外设状态改变后主动告知 CPU 的技术就是中断,CPU 放下当前正在处理的事务,对状态改变的外设进行处理的过程就是中断响应过程。

2. 中断响应过程

中断响应过程就是在接收到外设状态改变的通知后,转到处理外设状态改变事情的程序的过程。

对于 CPU 而言,用于向它通知外设状态改变的信号线就是中断请求 INTR,CPU 在接收到中断请求信号后,向外发送两个中断响应信号,并在第 2 个中断响应信号有效期间读入发生状态改变的外设所对应的中断类型码,并根据中断类型码找到中断处理程序的入口地址,以此完成从当前程序到中断处理程序的转移。

一般情况下,需要有专用的接口芯片来向 CPU 提供中断类型码,这种专用的配套芯片就是 8259,由它在 CPU 发出第 2 个中断响应信号期间,通过外部数据引脚输出发生状态改变的外设所对应的中断类型码。图 2.4 是 CPU、8259 和外设接口之间连接图,图 2.5 是 CPU 接收到中断类型码后,完成从当前程序到中断处理程序的转移过程。

假定键盘所对应的中断类型码为 09H,键盘中断处理程序存放在内存中以地址为

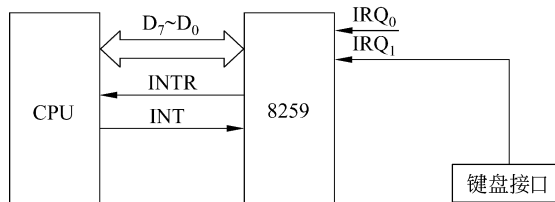


图 2.4 CPU、8259 和键盘接口连接图

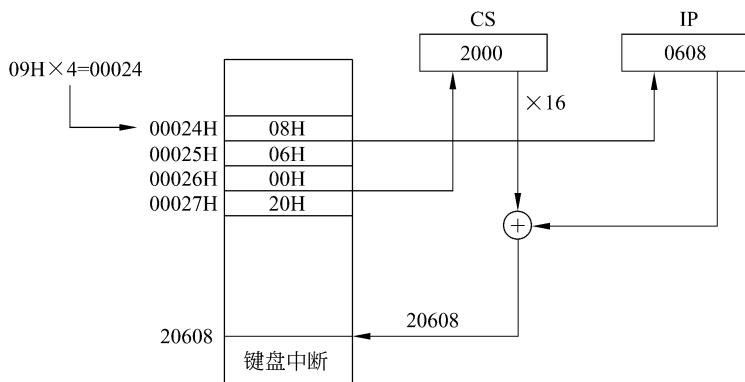


图 2.5 转移到中断处理程序过程

20608H 的存储单元开始的存储空间中，由于中断处理程序的入口地址为 2000H:0608H，必须将该入口地址送入中断向量表中地址为 00024H($4 \times 09H$)~00027H 的 4 个存储单元中，当 CPU 通过 8259 获得键盘所对应的中断类型码：09H，它将其乘 4 后作为中断向量表地址，并从该地址开始的 4 个相邻存储单元中取出 16 位偏移量和段地址，分别将偏移量和段地址送入指令指针寄存器(IP)和代码段寄存器(CS)。由于下一条指令地址由代码段寄存器内容(CS)乘 16 后加上指令指针寄存器内容(IP)获得。CPU 从此处开始执行键盘中断处理程序。

2.1.3 80386 虚拟存储机制

80386 形成的逻辑地址是 16 位段地址和 32 位偏移量。与 80286 不同，80386 可以采用段式和段页式存储管理，段式存储管理和 80286 非常相似，见图 2.6 上半部分，它用段地址(段选择子)的高 13 位作为索引去检索全局或局部描述符表，得到该段所对应的描述符，从中取出 32 位段起始地址，加上 32 位偏移量即可得到 32 位物理地址。但如果采用段页式存储管理，上述操作得到的 32 位地址不是真正用于访问存储器的物理地址，而是线性地址，必须通过检索页表才能获得真正的物理地址。将 32 位线性地址分成 3 部分，最高 10 位地址作为页组目录项索引，次 10 位地址作为页表项索引，最低 12 位地址作为页内偏移量，通过两级索引，获得线性地址指定单元所在页的页起始地址，由于页是固定长度(4KB)的存储块，页起始地址的最后 12 位地址肯定为 0，因此，页表只需给出 20 位的页起始地址，20 位页起始地址和线性地址中最低 12 位页内偏移量拼接成 32 位物理地址。

址,整个过程如图 2.6 所示。

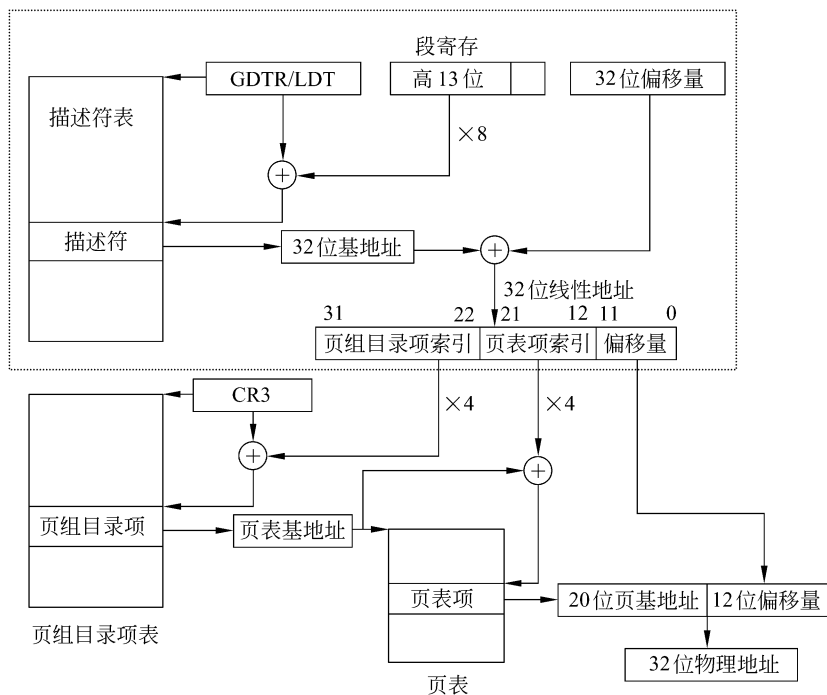


图 2.6 80386 形成 32 位物理地址的过程

2.2 例题解析

2.2.1 选择题

1. 8086 读一个以奇数地址开始的双字,最少需要_____个总线周期。

- (A) 1 (B) 2 (C) 3 (D) 4

【答案】 C

【解析】 8086 的外部总线为 16 位,最多可以一次传输 16 位数据,但前提是偶数地址开始的相邻两个字节,如果是偶数地址开始的双字,一次一字(16 位),两个总线周期就可以了,但本题是奇数地址开始的双字,需要 3 个总线周期,故选 C。

2. 根据下面提供的 PC 机内存中的数据,INT 11H 中断服务程序的入口地址是_____。

0000:0040 B3 18 8A CC 4D F8 00 F0-41 F8 00 F0 C5 18 8A CC

0000:0050 39 E7 00 F0 A0 19 8A CC-2E E8 00 F0 D2 EF 00 F0

- (A) CC8A:19A0 (B) A019:8ACC
(C) 4DF8:00F0 (D) F000:F84D

【答案】 D

【解析】 用于存放中断类型为 11H 的中断服务程序入口地址的起始存储单元地址

为 $11\text{H} \times 4 = 00044\text{H}$, 因此, 中断类型为 11H 的中断服务程序的入口地址存放在 $00044\text{H} \sim 00047\text{H}$ 这 4 个存储单元中, 将其转换成段地址为 0 的段地址: 偏移量模式, 即为 $0000:0044\text{H} \sim 0000:0047\text{H}$, 其中 $0000:0044\text{H}$ 、 $0000:0045\text{H}$ 为 IP, $0000:0046\text{H}$ 、 $0000:0047\text{H}$ 为 CS, 根据高地址为高 8 位的原则, 将 CS:IP 拼装成 16 位形式后即 为 F000:F84D , 故选 D。

3. 现行数据段位于存储器 $\text{B0000H} \sim \text{B5FFFH}$, 则段寄存器 DS 的内容及该段长度(字节数)分别为_____。

(A) B000H , B5FFFH

(B) 0000H , B5FFFH

(C) B000H , 5FFFH

(D) B000H , 6000H

【答案】 D

【解析】 该数据段第一个存储单元的物理地址为 B0000H , 故段地址为 B000H , 长度为 $\text{B5FFFH} - \text{B0000H} + 1 = 6000\text{H}$, 故选 D。

4. 下述何种引脚输入用于 CPU 内部同步_____。

(A) NMI

(B) INTR

(C) CLK

(D) RESET

【答案】 C

【解析】 用于 CPU 同步的信号为时钟信号, 故选 C。

5. 下述何种引脚输入用于终止 CPU 操作并初始化 CPU _____。

(A) NMI

(B) INTR

(C) CLK

(D) RESET

【答案】 D

【解析】 用于 CPU 初始化的信号为复位信号 RESET, 故选 D。

6. 下述何种引脚输入用于硬件中断请求(可屏蔽)_____。

(A) NMI

(B) INTR

(C) CLK

(D) RESET

【答案】 B

【解析】 用于输入可屏蔽的中断请求的引脚为 INTR, 故选 B。

7. 下述何种引脚输入用于硬件中断请求(不可屏蔽)_____。

(A) NMI

(B) INTR

(C) CLK

(D) RESET

【答案】 A

【解析】 用于输入不可屏蔽的中断请求的引脚为 NMI, 故选 A。

8. 8088 最小模式下, 哪一种信号组合用于存储器读操作_____。

(A) $\text{M}/\text{IO}=1, \text{RD}=0$

(B) $\text{M}/\text{IO}=1, \text{WR}=0$

(C) $\text{M}/\text{IO}=0, \text{RD}=0$

(D) $\text{M}/\text{IO}=0, \text{WR}=0$

【答案】 A

【解析】 在最小模式下, 当 $\text{M}/\text{IO}=1$ 时, 标志着访问存储器总线周期, 当 $\text{RD}=0$ 时, 意味着读操作, 故只有 $\text{M}/\text{IO}=1$ 且 $\text{RD}=0$ 才能对存储器进行读操作, 故选 A。

9. 8088 最大模式下, 哪一种信号组合用于存储器读操作_____。

(A) $\text{MEMR}=0, \text{MEMW}=0$

(B) $\text{MEMR}=0, \text{MEMW}=1$

(C) $\text{MEMR}=1, \text{MEMW}=0$

(D) $\text{MEMR}=1, \text{MEMW}=1$

【答案】 B

【解析】 在最大模式下,访问存储器控制信号由 8088 产生,8088 用 MEMR=0 表示进行存储器读操作,而 MEMW 为存储器写操作控制信号,两个信号不允许同时为低,因此,当 MEMR 为低电平时,MEMW 必须为高电平,故选 B。

10. 8086 和 8088 的主要差别在于_____。

- (A) 8086 是 16 位外部数据总线,8088 是 8 位外部数据总线
- (B) 8086 是 16 位计算机,8088 是 8 位计算机
- (C) 8086 可以访问 1M 字节,8088 只能访问 64K 字节
- (D) 8086 有浮点运算部件,8088 没有浮点运算部件

【答案】 A

【解析】 8088 为内部 16 位数据总线,而外部为 8 位数据总线,而 8086 内部和外部均为 16 位数据总线,故选 A。

11. 8086/8088 用双向缓冲器增强数据线信号强度,但不用单向缓冲器而是锁存器锁存地址的原因是_____。

- (A) 地址线和数据线复用
- (B) 地址线是双向的
- (C) 单向缓冲器不能增强地址线信号
- (D) 单向缓冲器有三态输出

【答案】 A

【解析】 缓冲器没有锁存作用,只能用于平衡两端的电流,由于 8086/8088 采用数据线和地址线复用相同引脚的技术,必须在这些引脚上输出有效地址时,将地址锁存到锁存器中,这样才能保证在整个总线周期都能维持有效地址不变,因此,用锁存器锁存地址的原因完全是因为地址线、数据线复用相同的引脚,故选 A。

12. 8086/8088 在 CPU 内部的逻辑地址形式为_____。

- (A) 16 位段地址和 16 位偏移量
- (B) 16 位段地址和 20 位偏移量
- (C) 20 位段地址和 20 位偏移量
- (D) 20 位线性地址

【答案】 A

【解析】 8086/8088 内部逻辑地址形式为 16 位段地址+16 位偏移量,故选 A。

13. 80386 在非保护模式(实时模式)下,CPU 内部的逻辑地址形式为_____。

- (A) 16 位段地址和 16 位偏移量
- (B) 16 位段地址和 32 位偏移量
- (C) 32 位段地址和 32 位偏移量
- (D) 32 位线性地址

【答案】 A

【解析】 80386 在非保护模式(实时模式)下,它的内部逻辑地址形式和 8086/8088 是兼容的,也是 16 位段地址+16 位偏移量,故选 A。

14. 80386 在保护模式下,CPU 内部的逻辑地址形式为_____。

- (A) 16 位段地址和 16 位偏移量
- (B) 16 位段地址和 32 位偏移量
- (C) 32 位段地址和 32 位偏移量
- (D) 32 位线性地址

【答案】 B

【解析】 80386 在保护模式下的逻辑地址形式为 16 位段地址+32 位偏移量,故选 B。但 80386 从逻辑地址到物理地址的转换操作和 8086/8088 有很大差异。

15. Pentium 的总线结构为_____。