

计算机组成原理

第一章—计算机系统概论

1.1 计算机系统的简介

1. 计算机系统由**硬件**与**软件**两大部分组成
2. 将高级程序语言翻译成机器语言的程序称为翻译程序，翻译程序有两种，一种是**编译程序**，一种是**解释程序**，编译与解释的区别在于，编译程序是将高级语言程序一次性翻译为机器语言程序，而解释程序是翻译一句，执行一句。
3. 高级语言经过**编译程序**翻译为**汇编语言**，汇编语言经**汇编程序**，翻译为**机器语言程序**

1.2 计算机的基本组成

1. 1945年冯诺依曼提出了“**存储程序**”的概念，冯诺依曼机特点：

1. 计算机由存储器，运算器，控制器，输入设备与输出设备组成
2. 指令与数据以同等地位存放在存储器内，按地址寻访
3. 指令与数据均按二进制数表示
4. 指令由操作码与地址码组成，操作码用来表示操作的性质，地址码用来表示操作数在存储器中的位置
5. 指令在存储器内按顺序存放
6. 计算机以运算器为中心，输入设备与输出设备的数据传送通过运算器来完成

2. 冯诺依曼机是由**运算器**为中心的，现代计算机是以**存储器**为中心的

3. 计算机的工作过程（必考）

涉及的元器件：MAR(地址寄存器)，MDR(指令寄存器)，ALU(算数逻辑单元)，ACC(累加器)，MQ(乘商寄存器)，PC(程序计数器)，IR(指令寄存器)

（**掌握执行指令的全过程**）

4. 机器字长：机器字长是指CPU一次能处理数据的位数，通常与CPU的寄存器位数有关

5. 存储容量：

$$\text{存储容量} = \text{存储单元个数} \times \text{存储字长}$$

6. 运算速度(可能出计算)： $V_m = 1 / T_m$ 单位MIPS(百万指令每秒)

CPI（执行每条指令所需要的时钟周期）= $1 / IPC$ (CPU每一周期执行指令的条数，一旦CPU设计完成，IPC的值不会变)

第三章—系统总线

3.1 总线的基本概念

总线是连接多个部件的信息传输线，是各部件共享的介质（总线的每条传输线可以传输1位二进制代码）

3.2 总线的分类

总线按照数据传送方式可分为两类：

1. 并行传输总线
2. 串行传输总线

按连接部件的不同可以分为三类（掌握加粗部分）：

1. **片内总线(指芯片内部的总线)**
2. **系统总线**
3. 通信总线

3.2.1 片内总线

概念：片内总线是指芯片内部的总线

3.2.2 系统总线

系统总线是指CPU,I/O设备，主存各大部件的信息传输线

按照系统总线的传输信息不同，可分为三类：

1. 数据总线
2. 地址总线
3. 控制总线

1.数据总线：双向传输总线，与机器字长与存储字长有关

2.地址总线：单向传输总线，由CPU发出，主存的地址线位数与存储单元的个数有关

3.控制总线：从单个来说传输是单向的，从总体来说传输的双向的

3.2.3 通信总线（了解即可）

这类总线用于计算机系统之间或计算机系统与其他操作系统之间的通信

3.3 总线特征与性能指标

3.3.2 总线性能指标

1.总线宽度：总线宽度可以数据总线的宽度，用位来表示，例如8位，16位，32位

2.总线带宽（要求会计算，且掌握提高总线速率的方式）：总线带宽可以理解为总线的传输速率，即单位时间上的传输数据的位数，通常用每秒传输的字节数来衡量，单位Mbps(兆字节每秒)

例子：总线的频率为33Hz，总线宽度为32位，求总线带宽？

$$33 * (32 / 8) = 132 \text{Mbps}$$

3.总线复用：一条信号线上传输两种线号，例如，一条总线上即可传输地址信号，又可传输数据信号，此称之为总线复用

3.3.3 总线标准（掌握PCI,USB）

1.PCI总线：为了提升总线性能，由Intel首先提出，PCI中文名称为**外围部件互连**，其最出名的特性为即插即用，即任何扩展卡插入系统便可直接工作，现在已推出了PCI-Express

2.USB总线：通用串行总线，真正的即插即用，这里的串行指的是串行通信，即使用一条数据线，将数据1位1位的进行传输，**不可同时传输2位数据**

3.5总线控制

1.为何使用总线控制？

由于总线上连接着多个部件，什么时候由哪个部件发送信息，如何给信息传送定时，如何防止信息丢失，如何避免多个部件同时发送，如何规定接受信息的部件等一系列问题，都需要由总线控制器统一管理。他主要包括**判优控制**与**通信控制**

1. 判优控制解决的是总线使用权的问题
2. 通信控制解决的是获得使用权后如何有效通信的问题

3.5.1总线的判优控制

1.主设备与从设备的区别，主设备**有总线控制权**，从设备**只能响应从主设备发来的总线命令**，对总线没有控制权。

当有多个设备需要使用总线，需要使用**总线控制器判优**。总线判优控制可以分为**集中式**与**分布式**两种

三种仲裁方式：

1.链式查询：使用三根线用于总线控制，分别为BS（总线忙），BR（总线请求），BG（总线同意），BG线依优先次序穿过所有I/O设备

优点：结构简单

缺点：电路故障的敏感性高，优先级一经写定无法更改

2.计数器定时查询：计数器定时查询使用计数器计数，向设备发送一种地址信号，当某个地址信号与计数值一致时，获得总线使用权。

优点：总线的优先级可以修改

缺点：引入了一组设备地址线，结构变的复杂（使用 $\log_2 N$ 根线）

3.独立请求方式：每一个I/O设备都有一组BR,BG线，根据排队线路决定优先级。

优点：相应速度快，优先次序控制更加灵活

缺点：控制线路多，总线控制更加复杂（三种方式之中最复杂的一种）使用 $2n+1$ 根线

3.5.2总线通信控制（需会例题3.1）

解决的是在获得使用权后如何有效通信的问题，通常将**一次总线操作所用时长称为总线周期**，通常有四种通信方式：**同步通信**，**异步通信**，半同步通信，分离式通信

1.同步通信：通信双方由统一的时钟标准控制数据传输

优点：规定明确，统一配合简单一致

缺点：主，从模块时间配合的强制性同步，必须在规定时间内完成规定的要求

适用场景:适用于总线长度较短，各部件存取时间一致的场所

2.异步通信：克服了同步通信的特点，允许各模块的速度不一致，没有公共的时钟标准，采用**应答方式**与**握手方式**

异步通信三种类型：不互锁，半互锁，全互锁

(1) 不互锁

主模块发出请求后不必等待从模块的回答信号，而是经过一段时间确定已收到请求信号后，撤销其请求信号

(2) 半互锁

主模块在发出请求后，必须等到从模块的响应之后才能撤销其请求信号。从模块在接受到请求信号后必须回应，但不用回应主模块的撤销信号

(3) 全互锁

主模块发出请求信号，必须等待从模块回答后再撤销其请求信号，从模块发出回答信号，必须得知主模块请求信号撤离后再撤销其回答信号。

异步通信的速率：

异步串行通信的速率用波特率来衡量。波特率是单位时间内传送的二进制位数，单位用bps来表示，记做波特

比特率：单位时间内传送的**有效**二进制位数，单位用bps来表示

第四章—存储器

4.1.1 存储器的分类

1.按存储介质分类：

- 半导体存储器：体积小，功耗低，存储时间短。当电源消失后信息也会丢失，具有易失性
- 磁表面存储器：磁表面是在金属或者塑料表面涂上一层磁性材料作为记录介质，具有非易失性的特点
- 磁芯存储器：由硬磁材料做成的环形元件，有驱动线与读出线，后被半导体存储器所取代
- 光盘存储器：不多说了

2.按存取方式分类：

按照存取方式可以分为，随机存储器，只读存储器，顺序存取存储器，直接存储器

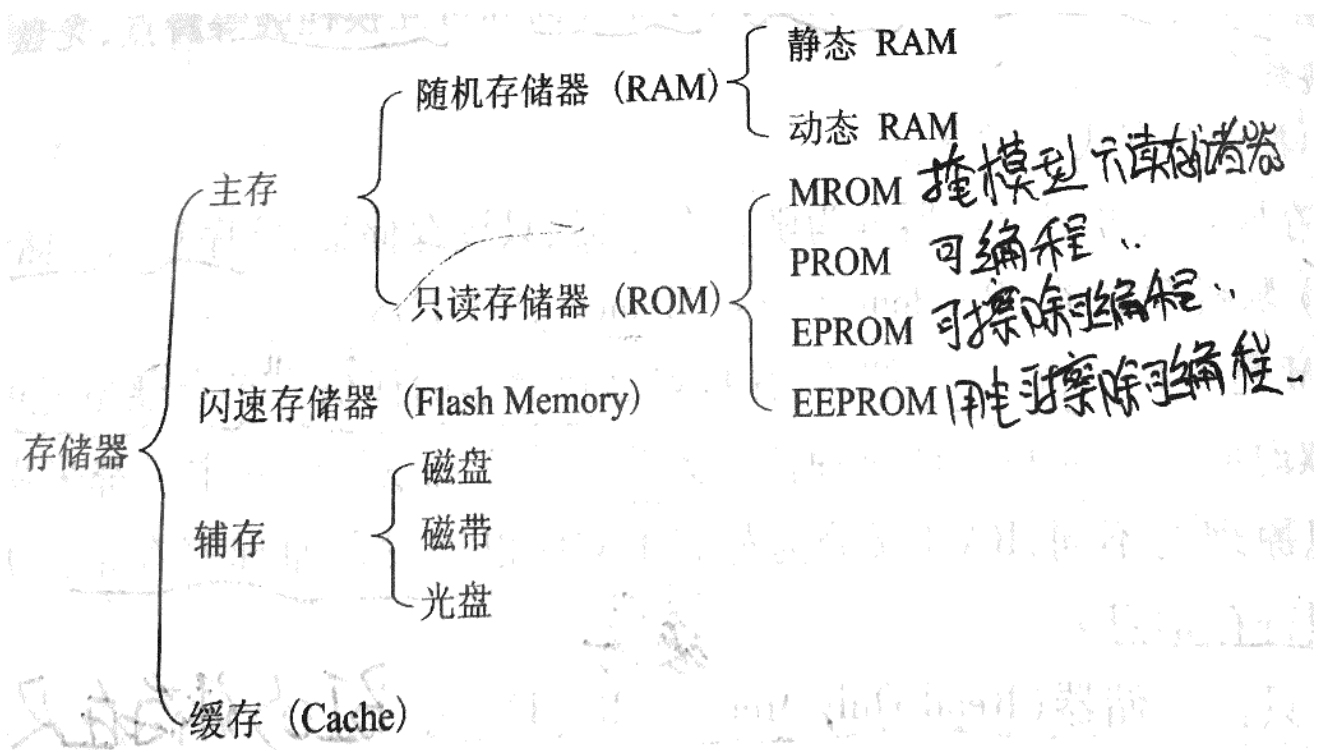
(1) 随机存储器 (RAM) :RAM是一种可读\写存储器，其特点是存储器的任何一个存储单元的内容都可以随机存取，而且**存取的时间与存储单元的物理位置无关**，分为静态RAM (以触发器原理寄存信息)，与动态RAM (以电容原理寄存信息)。

(2) 只读存储器 (ROM) :能对其存储的内容读出，而不能对其重新写入的存储器，在程序执行过程中，只能将内部信息读出，而不能随意重新写入新的信息去改变原始信息

(3) 串行访问存储器：需按照物理位置的先后顺序寻找地址，则这种存储器称为串行访问存储器，又称顺序访问存储器

3.按照在计算机中的作用分类 (如下图)

按照在计算机中的作用不同，主要分为，主存储器，辅助存储器，缓存存储器



存储系统的层次结构主要体现在，缓存—主存，辅存—辅存两个存储层次上

缓存—主存：解决的是CPU与主存速度不匹配的问题（速度）

主存—辅存：解决的是存储系统的容量问题（容量）

主存与辅存之间的数据调动是操作系统与硬件共同完成的

4.2主存储器

现代计算机的主存，都由半导体集成电路构成

1.主存中存储单元的地址分配

计算机中有两种存储方式：按字寻址与按字节寻址

使用字地址进行寻址，一次读出一个字。使用字节地址进行寻址，一次读出一个字节，字地址尽量使其为2的倍数（此处会考察选择高地址或低地址作为字地址，其还会考察根据容量求范围）

2.主存的技术指标

（1）.存储容量（指能存放二进制代码的总位数）：

$$\text{存储容量} = \text{存储单元} \times \text{存储字长}$$

也可以用字节数来表示，即

$$\text{存储容量} = \text{存储单元个数} \times \text{存储字长} / 8$$

（2）存取速度（使用存取时间或存取周期来表示）

存取时间又称存储器的访问时间，是指启动一次存储器的操作到操作完成的全部时间。

存取周期是指存储器连续两次进行独立的存储器操作所需要的最小间隔时间

(3) 存储器的带宽 (单位时间内存储器存取的信息量)

单位为 字/秒 字节/秒 位/秒

提高存储器带宽的三种方案 (important) :

1. 缩短存取周期
2. 增加存储字长, 使每个周期可以读取更多位二进制数
3. 增加存储体

4.2.2 半导体存储芯片简介

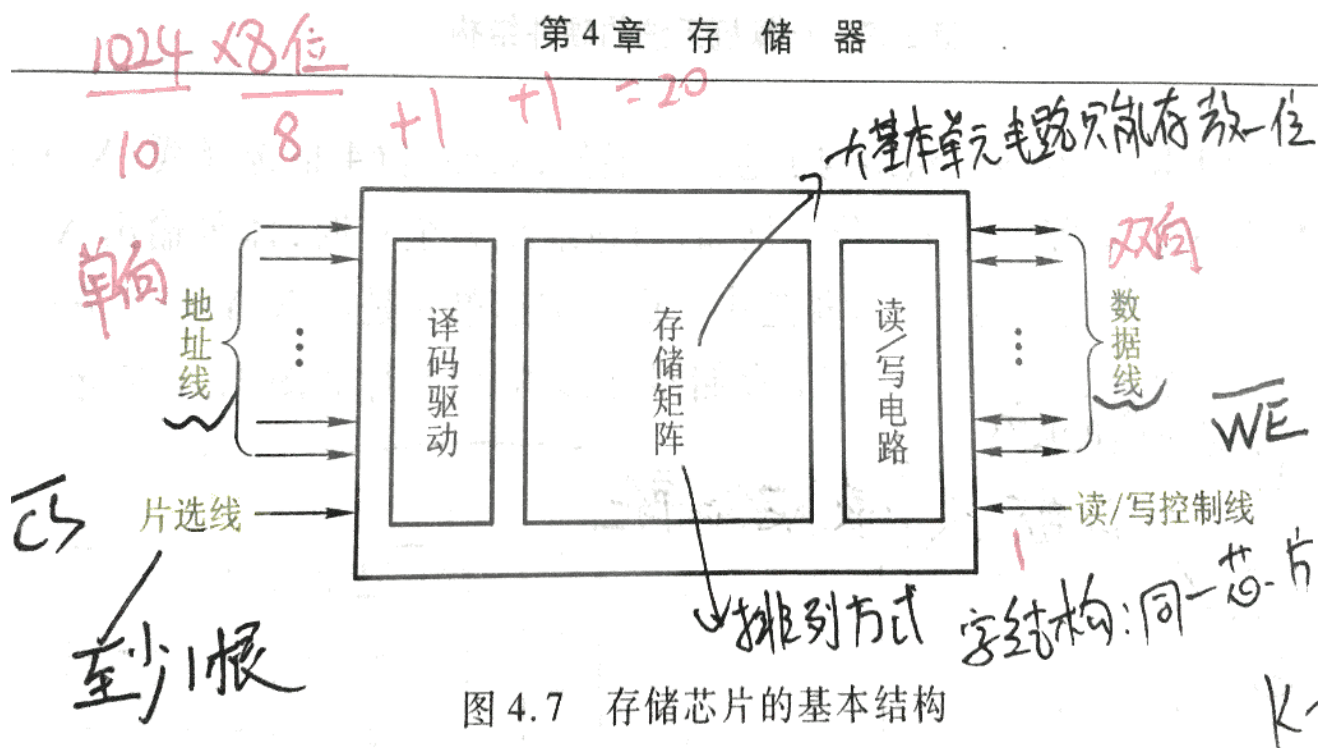


图 4.7 存储芯片的基本结构

(考察使用最少线的数量)

2. 半导体芯片的译码方式, 线选法 (又称单译码法), 重合法 (又称多译码法) (会计算字线与位线数量)

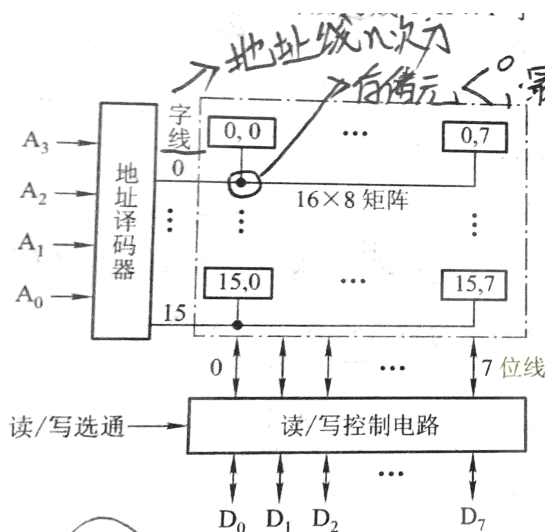


图 4.9 16×1 字节线选法结构示意图

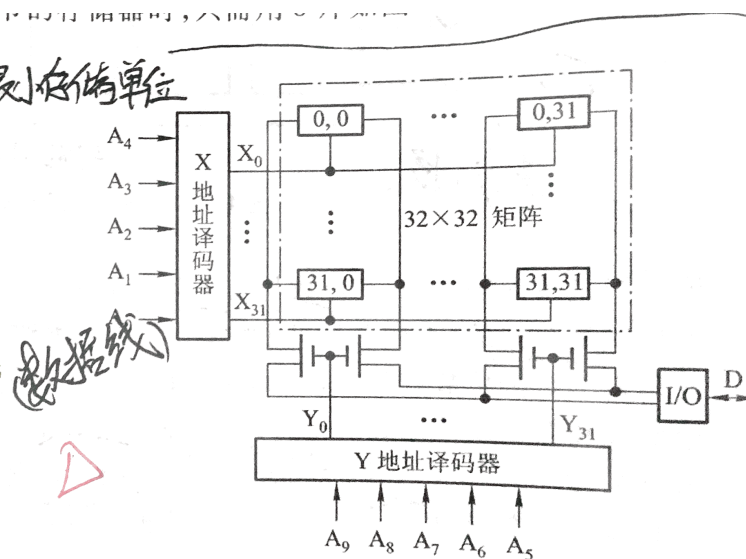


图 4.10 1K×1 位重合法结构示意图

4.2.3 随机存取存储器

静态RAM：使用触发器的工作原理存储数据，即使信息读出后，依然会保持原状态，不需要再生，但是掉电后信息丢失

动态RAM：使用电容的工作原理存储数据，由于电容的特性，即是不掉电信息也会消失，则2ms进行1次复原，称为刷新

动态RAM的刷新：因为动态RAM信息自动消失的特性，即是不访问也需要进行刷新，规定在一定时间内对动态RAM进行一次刷新，一般取**2ms**，通常有三种方式，集中刷新，分散刷新，异步刷新

下面描述均使用128*128的矩阵举例，不再赘述

(1) 集中刷新，是在一个周期内，对全部存储单元在一段时间逐行进行刷新，刷新时无法进行读写操作，此块时间称为死区，存储周期为0.5us，刷新周期为2ms，则刷新时间为0.5us * 128 = 64us

(2) 分散刷新，分散刷新是将存储单元的刷新分散到每个存取周期来完成，若机器的读写周期为0.5us，在存取周期后紧接着安排一次刷新，将存储周期提升至1s，这样128us就可以完全刷新一遍，缺点延长了存取周期，是降低了性能

(3) 异步刷新：是前两种方式的结合，将刷新操作平均分到最大时间间隔，完全刷新时间一趟固定为2ms，刷新时间间隔为 2000us/128

4.2.4 只读存储器

ROM只能读不能写，PROM是可实现一次性编程的只读存储器，(EPROM, EEPROM) 闪存，快擦性存储器

4.2.5 存储器与CPU的连接 (15分大题)

详见P94例4.1

4.2.6 存储器的校验 (考察7位以下汉明码的编码与解码)

汉明码的检测位数计算公式，n为二进制代码位数，k为检测位位数

$$2^k \geq n + k + 1$$

检测位所检测序列的规则为：

$$C_1 = 1, 3, 5, 7 \dots$$

$$C_2 = 2, 3, 6, 7 \dots$$

从下标数字开始，数 n 个数，空过 n 个数

奇校验规则，检测序列中1的个数有奇数个，若出错，则出错位在p4,p2,p1全部取非所对应的10进制位上。

偶校验规则，检测序列中1的个数有偶数个，若出错，则出错位在p4,p2,p1所对应的10进制位上

例题 参照书上p4.4

4.2.7 提高访存速度的措施

1. 寻找高速元件
2. 采用层次结构
3. 调整主存结构

1. 单体多字系统：使用一个存储体，一次读出多个字，然后按顺序送给CPU

2. 多体并行系统：采用多体模块组成的存储器，每个模块有相同的容量与存取速度，各模块有独立的MAR与MDR, 及各种电路

1. 高位地址交叉，高位为体号
2. 低位地址交叉，体号为低位

剩余部分参照 例题4.6

4.3 高速缓冲存储器

解决问题：

1. 避免CPU空等现象
2. CPU与主存之间的速度差异

工作原理：

指令与数据在主存地址不是随机的而是相对的**簇聚**，访问具有相对局部性，这就是**程序访问的局部性原理**，主存与cache以**块**为单位交换信息

Cache命中计算问题：例4.7

块长与命中率的变化关系：

随着块长的增长，命中率首先会增加，增加到一定程度后，命中率反而会降低

Cache的基本结构

主要由**Cache存储体**，**地址变换机构**，**替换机构**三部分组成

Cache存储体：Cache存储体以**块**为单位与主存交换信息

地址变换机构：Add (主存) \rightarrow Add (cache)

替换机构：当Cache满时发生替换，共有两种替换方式，分别是写直达法与写回法

写直达法：操作时数据即写入Cache又写入主存

写回法：操作时数据写入Cache不写入主存，仅当数据被替换出Cache时，写回主存

4.3.2 主存地址映射

例4.8 - 4.10

4.3.3 替换策略

先进先出：没有依据程序访问的局部性原理，故不能提高命中

近期最少使用：需要随时记录cache中存储字块的使用情况，已确定最少使用的字块，复杂，效率高

随机法：没有依据程序访问的局部性原理，故不能提高命中

第七章—指令系统

7.1 机器指令

人们习惯把每一条机器语言的语句称为机器指令，而又将全部的机器指令称为指令系统

7.1.1 指令的一般格式

指令由**操作码**与**地址码**组成

(1) 操作码：操作码用来指明指令的操作，如加减乘除等，若操作码有7位则对应128条指令

(2) 地址码：地址码用于指出该指令原操作数的地址，结果的地址以及下一条指令的地址

7.1.2 指令字长

在本科目中，约定机器字长、指令字长、存储字长的长度相同

7.3 寻址方式

7.3.1 指令寻址

指令分为单字长指令，双字长指令。PC的内容的更新会依据存储字长与指令字长的关系，若指令长度为单字长，则 $PC + 1$ ，若指令字长为双字长， $PC + 2$

7.3.2 数据寻址（考察9种情况）

形式地址与指令地址的区别，形式地址不代表真实地址，记做A，真实地址即是内存中的有效地址，记为EA

(1) 立即寻址

立即寻址的特点是操作数本身设在指令内的，**即形式地址A不是操作数的地址，而是操作数本身**

(2) 直接寻址

直接寻址方式的特点是，指令字中的**形式地址A就是操作数的真实地址EA**

(3) 隐含寻址

指令中不明显给出操作数的地址，其操作数的地址隐含在操作码的某个寄存器中，**一地址格式只给出一个操作数的地址**

(4) 间接寻址

指令中的形式地址不给出操作数所在的地址，而给出的是操作数所在的地址存储单元的地址，即为简介寻址（很像C语言中指向指针的指针）

(5) 寄存器寻址

地址码字段直接指出了寄存器编号，即 $EA=R$ ，直接去对应编号的寄存器中将数取出即可

(6) 寄存器间接寻址

地址码对应的寄存器中存放的是主存中操作数所对应的地址

(7) 基址寻址

$EA = A + (BR)$ ，BR基址寄存器的中的基址，通常由操作系统直接指定，用户无法修改，此时将指令的A + 基址中存在的地址，即可得到操作数在主存中的地址

(8) 变址寻址

与基址寻址类似， $EA = A + (IX)$ ，与基址寻址不同的是，此处的A不可变，IX可变，IX由用户指定，很类似于数组处理的过程

(9) 相对寻址

将指令中的A，加上PC + 指令所占字节数，得到操作数在主存中的位置

考察点：操作码的扩展技术，九种寻址方式对应的操作数与PC值