



河南大學
Henan University

汇编语言与接口技术

—— 第 6 章 I/O 接口及数据传输控制方式

主讲教师：舒高峰

电子邮箱：gaofeng.shu@henu.edu.cn

联系电话：13161693313

00

上节回顾-子程序设计

简答题

- 子程序调用指令与转移指令最主要的区别是什么?
- 子程序与主程序参数传递有几种方式?

目录

01 接口概述

02 CPU 与外设数据传输控制方式

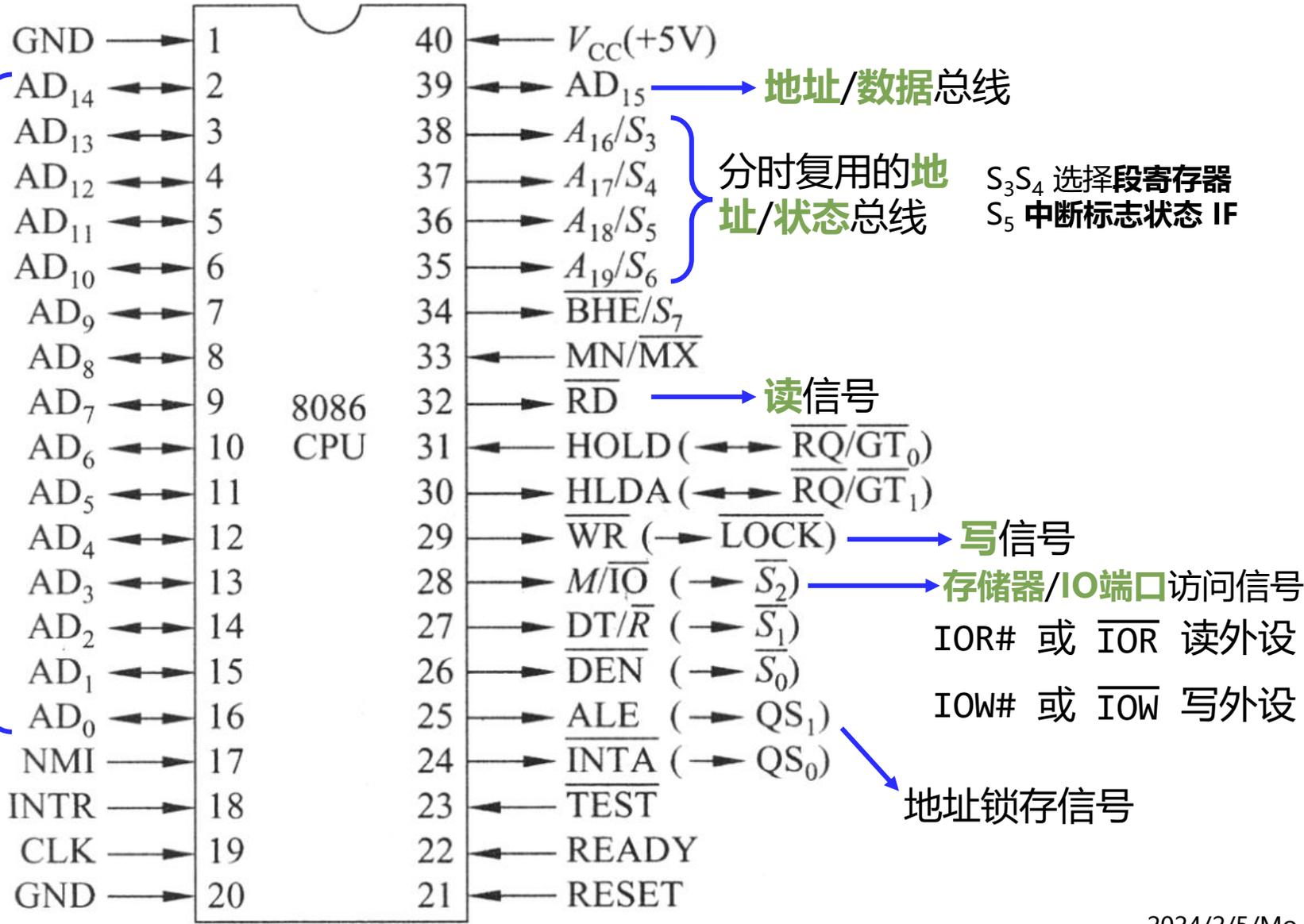
01 | 接口概述-子目录

接口概述

- 8086 CPU 管脚
- 接口的概念、功能和组成
- I/O 端口及其编址方式
- I/O 端口地址的译码
- 输入/输出指令

01 | 接口概述 - 8086管脚

低 15 位
地址/数
据总线



01 | 接口概述-子目录

接口概述

- 8086 CPU 管脚
- 接口的概念、功能和组成
- I/O 端口及其编址方式
- I/O 端口地址的译码
- 输入/输出指令

01 | 接口概述-接口的概念

(硬件) 接口 (Interface)

- CPU 与 “**外界**” 的**连接电路**，是 CPU 与外界进行信息交换的中转站
 - **外界**：除 CPU 以外的所有设备和电路
 - **连接电路**：硬件设备

- 人类与程序的接口称为**用户界面** (User Interface, UI)
- 计算机软件组件间的接口称为**软件接口** (Application Programming Interface, API)

设置接口的原因

- 速度不匹配：CPU 与外设的工作速度不一致
- 时序不匹配：CPU 与外设的时序电路不同
- 信息格式不匹配：CPU 与外设处理的信息格式不同
- 信息类型不匹配：不同外设采用的信号类型不同
 - 信息类型：数据信息、状态信息、控制信息

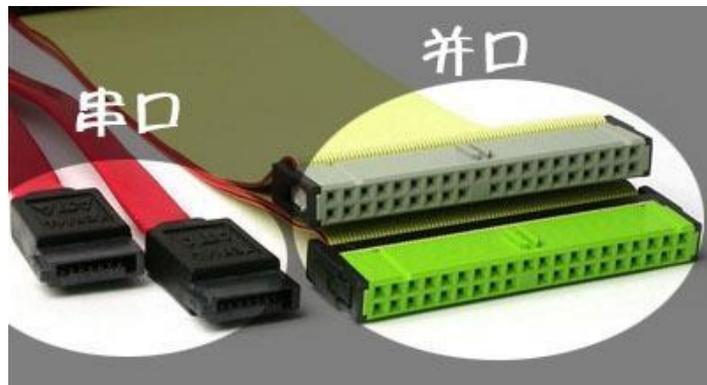
01 | 接口概述-接口技术

接口技术

- 接口技术是**硬件**和**软件**的综合技术
 - 使 CPU 与外界达到最佳连接
 - 减轻 CPU 的负担，**提高 CPU 的工作效率**
 - 实现 CPU 与外界**高效、可靠**地信息交换

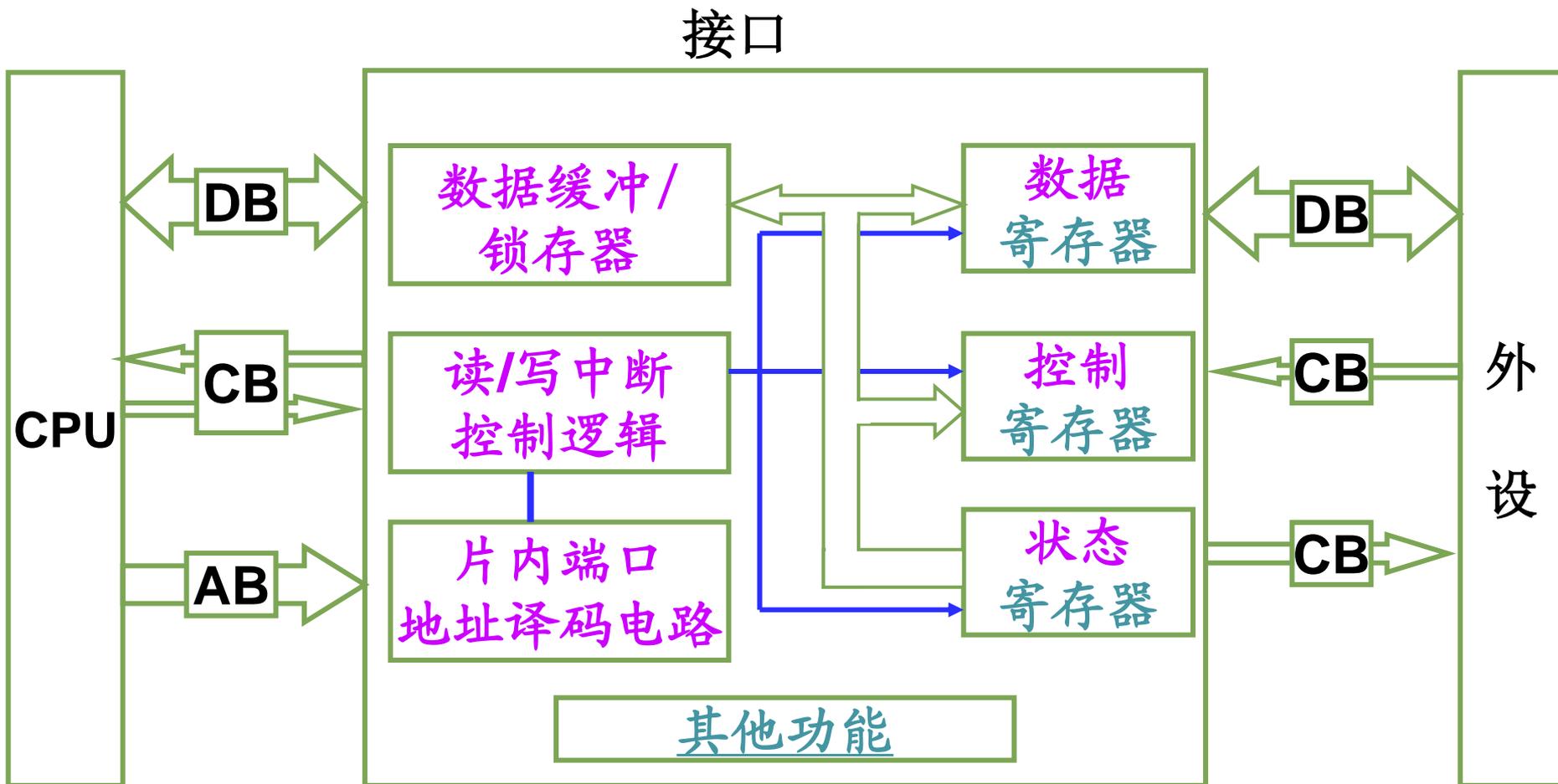
接口技术的应用

- 计算机所涉及的所有领域



01 | 接口概述-接口的组成

接口的组成



01 | 接口概述 - 接口的功能1 - 数据缓冲

功能执行

- 作为 CPU 与外设之间**数据中转站**
 - 用于 CPU 与**慢速外设**之间，可**协调**二者的速度差异
- 数据中保存在接口的**数据寄存器**中

数据缓冲的方式

- **数据缓冲器**暂存数据
 - 完成本次数据传送后，其中数据就会发生变化
- **数据锁存器**暂存数据
 - 在有新的数据写入之前，其中数据不会更改

01 | 接口概述-接口的功能2-执行CPU命令

功能执行

- CPU 按照控制字格式准备好控制字
- 接口初始化时，由 CPU 将控制字写入接口内部寄存器
- 接口工作时，译码控制字，进行规定的操作

控制字

- 控制字 (命令字): 一串二进制数据
- 每一位具有不同的含义，表示 CPU 对外设或接口的控制信息
- 不同的接口芯片，控制字的数目和格式也不同

01 | 接口概述 - 接口的功能3 - 接收外设状态

功能执行

- 在外设工作期间，接口接收其工作状态信息，形成**状态字**，保存于**状态寄存器**中
 - 状态字随时根据外设状态的变化而改变
- CPU 需要了解外设工作状态时，从接口中直接读取该状态字

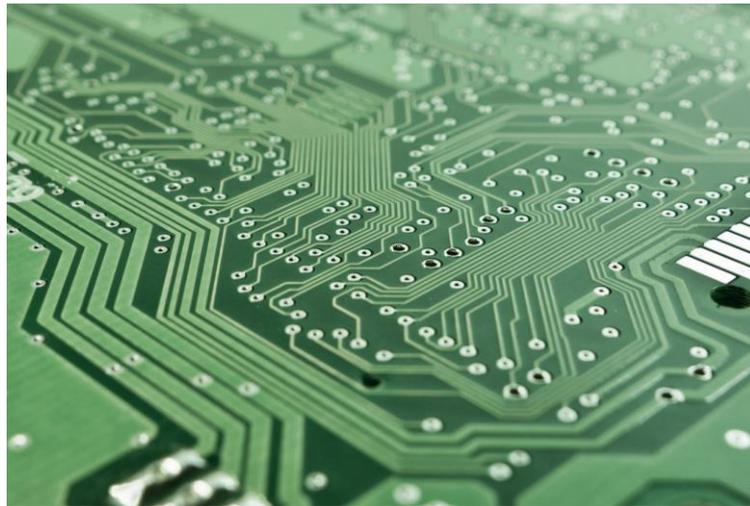
状态字

- 状态字：**一串二进制数据**
- 每一位具有不同的含义，反映当前外设的某种**状态**

01 | 接口概述-接口的功能4-设备寻址

功能执行

- 接口内部的数据寄存器、控制寄存器、状态寄存器都是CPU 可访问的
- CPU 需要对当前接口外部或内部的操作部件进行选择(寻址)
- 选择方式
 - 根据 CPU 发出的地址进行译码，从而确定当前操作部件



01 | 接口概述 - 接口的功能5 - 其他功能

功能

- 完成 CPU 与外设之间传送信号的类型转换
 - CPU 与外设的**逻辑**关系、**电平**高低和**时序**不兼容

转换类型

- 数据**宽度**转换
 - 串行接口中需要的串-并、并-串转换
- 数据**格式**转换
 - 工程接口中常用的数-模、模-数转换
- **时序**或**电平**转换
 - 时序配合

01 | 接口概述-子目录

接口概述

- 8086 CPU 管脚
- 接口的概念、功能和组成
- I/O 端口及其编址方式
- I/O 端口地址的译码
- 输入/输出指令

01 | 接口概述-I/O端口

I/O 端口 (Port)

- 接口电路中能被 CPU 直接访问的寄存器的地址

端口的类型

- 数据端口 (CPU 可读写)
 - 状态端口 (CPU 只读)
 - 命令端口 (CPU 只写)
- 每个端口可接收多种格式的同类信息

端口与接口的关系

- 一个接口电路由若干个端口及其他控制电路组成
 - 不同接口中端口的数目不定，但类型只有三种

01 | 接口概述-I/O端口编址

I/O 端口地址

- 为了区分不同的端口，需为每个端口分配一个**地址编号**，称为 **I/O 端口地址**

编址方式

- 由于 CPU 地址总线既连接到存储器又连接到 I/O 端口，因此，需要一种机制来**区分**和**寻址**操作的内存单元或 I/O 端口，这种机制称为**编址方式**
- 常用的两种 I/O 编址方式：
 - 存储器映像编址方式（统一编址方式）
 - 独立编址方式

01 | 接口概述 - I/O端口的编址方式1

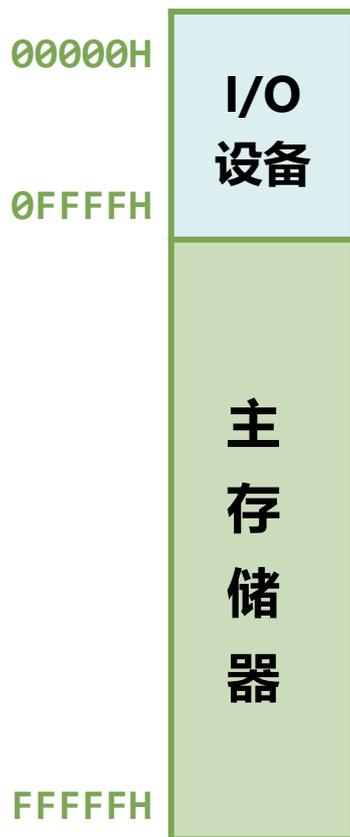
存储器映像编址方式

以 20 位主存地址
16 位 I/O 地址为例

- 将 I/O 端口地址置于存储器空间，和存储单元统一编址

优缺点

- ✓ 对端口的访问方式非常灵活
- ✓ 端口与 CPU 的连接方式和存储器与 CPU 的连接方式类似
- ✗ 占用了一部分存储空间
- ✗ 端口地址长，访问速度较慢



01 | 接口概述 - I/O端口的编址方式2

独立编址方式

- 将 I/O 设备与主存储器**分别**编址

以 20 位主存地址
16 位 I/O 地址为例

优缺点

- ✓ I/O 端口**不占用**存储空间
- ✓ 使用专门的 I/O 指令，指令短，**执行速度快**
- ✗ 需设置专门的指令和控制信号，**增加了系统的开销**



01 | 接口概述-子目录

接口概述

- 8086 CPU 管脚
- 接口的概念、功能和组成
- I/O 端口及其编址方式
- I/O 端口地址的译码
- 输入/输出指令

01 | 接口概述-I/O端口地址译码

为什么要译码电路

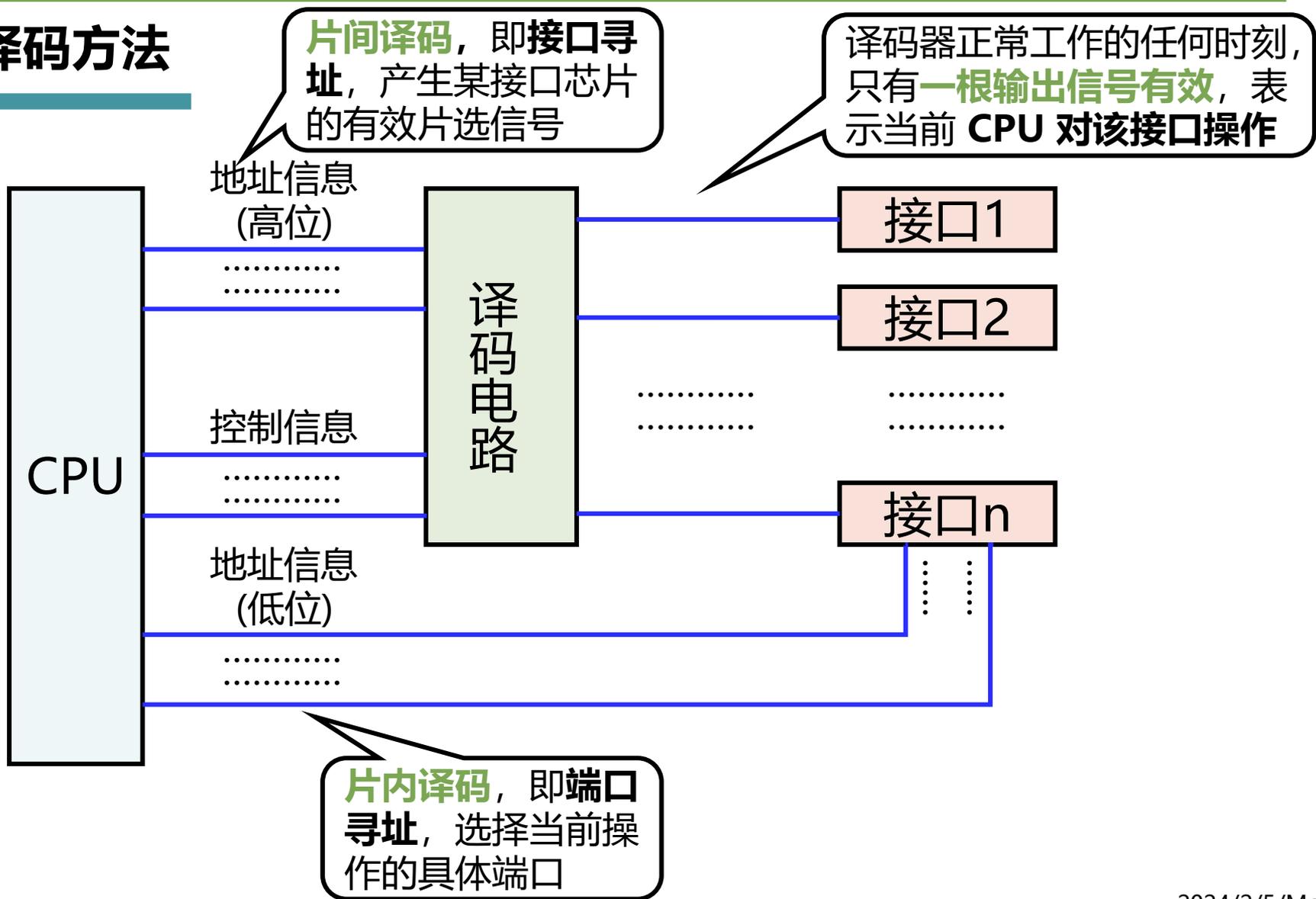
- CPU 地址总线给出地址信息，这个地址信息要通过地址译码电路翻译成所需要访问的 I/O 端口
- 将 CPU 发出的**地址信号**转换成对应端口的**选通信号**的部件就是**端口地址译码电路**

外部信号

- 输入信号
 - **地址信号** $A_0 \sim A_9$
 - **读写控制信号** IOR#, IOW#
 - DMA **地址允许信号** AEN
- 输出信号
 - 芯片或端口的**选通信号**，低电平有效

01 | 接口概述-I/O端口地址译码方法

译码方法



01 | 接口概述 - 译码电路

I/O 端口地址译码电路的形式

- I/O 端口地址译码电路的形式随设计任务的复杂度而变化

I/O 端口地址译码电路

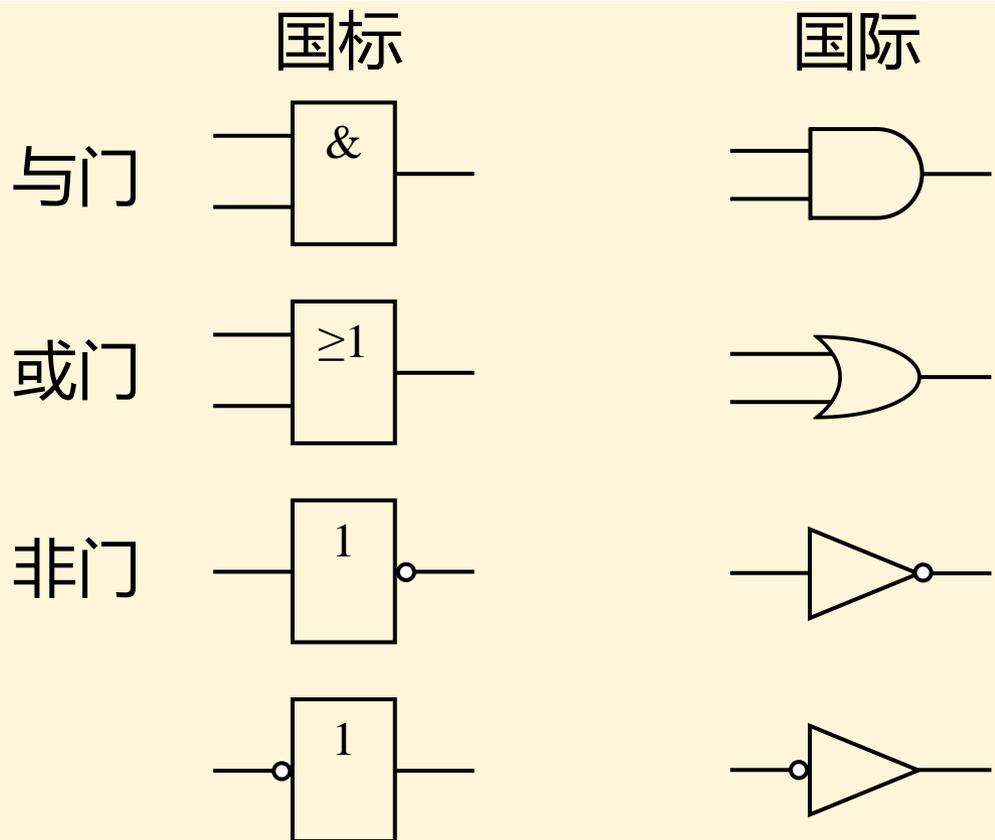
- 固定式单端口地址译码电路 → 由门电路构成
- 固定式多端口地址译码电路 → 由译码器构成
- 可选式多端口地址译码电路 → 由地址开关和比较器构成

01 | 接口概述 - 固定式单端口译码

固定式单端口译码特点

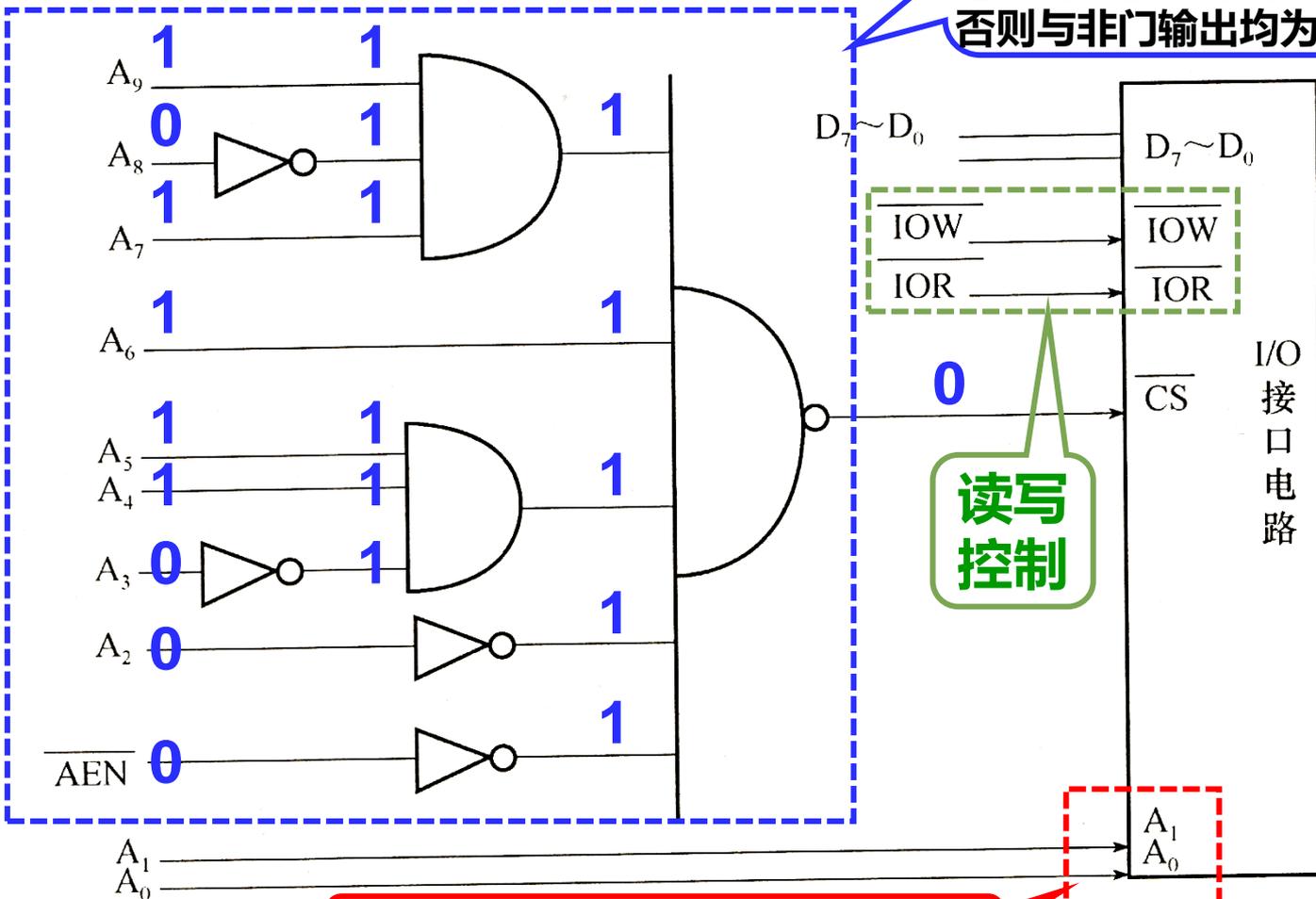
- 接口中的端口**地址固定**，不能修改
- 形式比较简单，大多采用与、或、非门等基本**门电路**来实现

门电路



01 | 接口概述 - 固定式单端口译码举例

固定式单端口译码电路



接口寻址: 当且仅当 $A_9 \sim A_2 = 10111100$ 时, 与非门的输出为0; 否则与非门输出均为1

Address Enable
Chip Select

该I/O 接口电路端口地址范围

10 1111 0000
10 1111 0011

所以, 端口地址范围为 2F0H~2F3H

该接口电路内部共4个端口

$A_1 \sim A_0$ 为片内译码的地址

01 | 接口概述-译码器译码

译码器译码

- 多端口时 (有**多个接口芯片**, 或接口芯片有**多个端口**) 常采用专用**译码器**译码

常用的译码器

- **74LS138**: 3-8 译码器
- 74LS139: 双 2-4 译码器
- 74LS154: 4-16 译码器

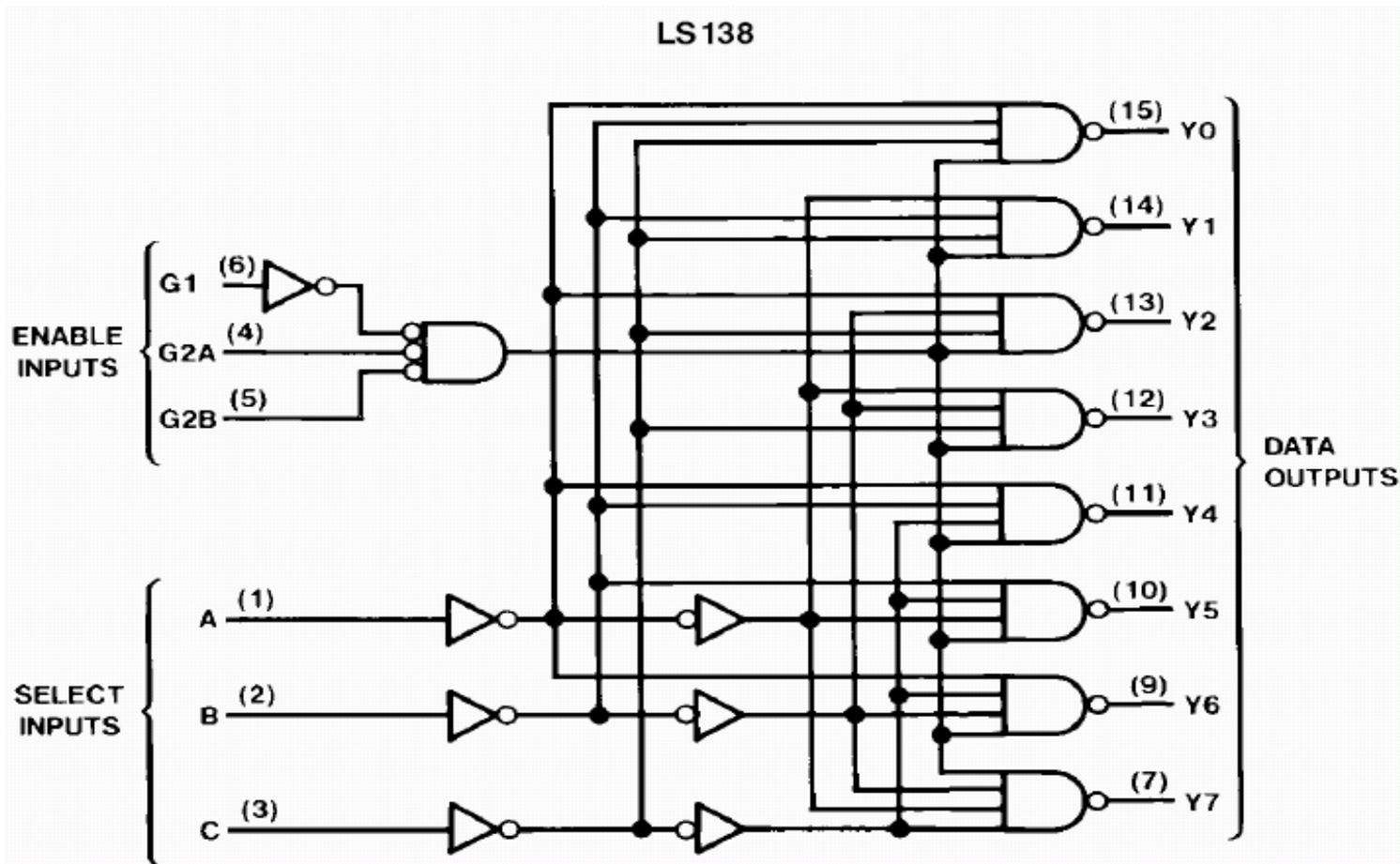


01 | 接口概述 - 74LS138译码器内部结构图

8 个固定式单端口译码电路的叠加

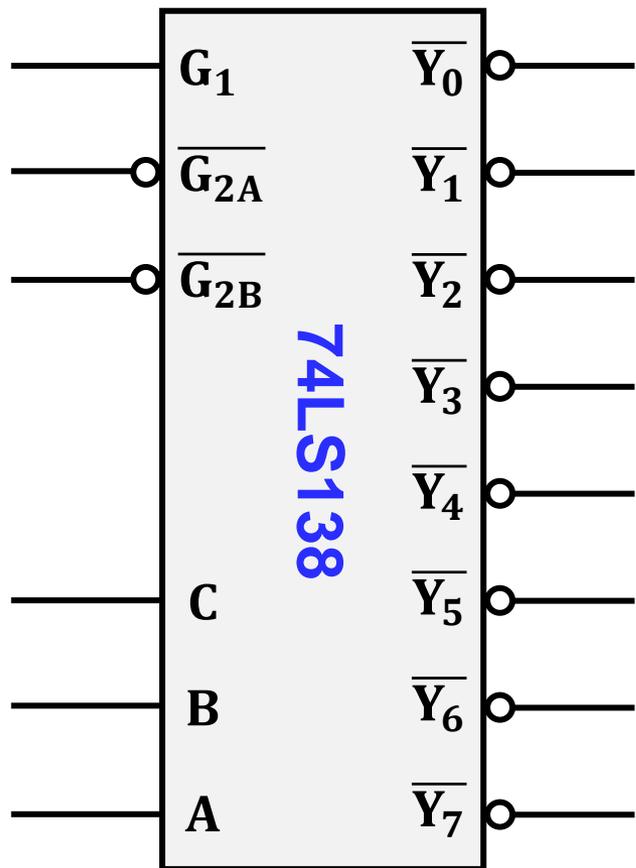
输入使能信号
电源
地址使能信号
选通信号

端口选择信号
 $CBA=000 \rightarrow Y_0$
 $CBA=001 \rightarrow Y_1$
.....
 $CBA=111 \rightarrow Y_7$



01 | 接口概述 - 74LS138译码器工作原理

74LS138译码器工作原理



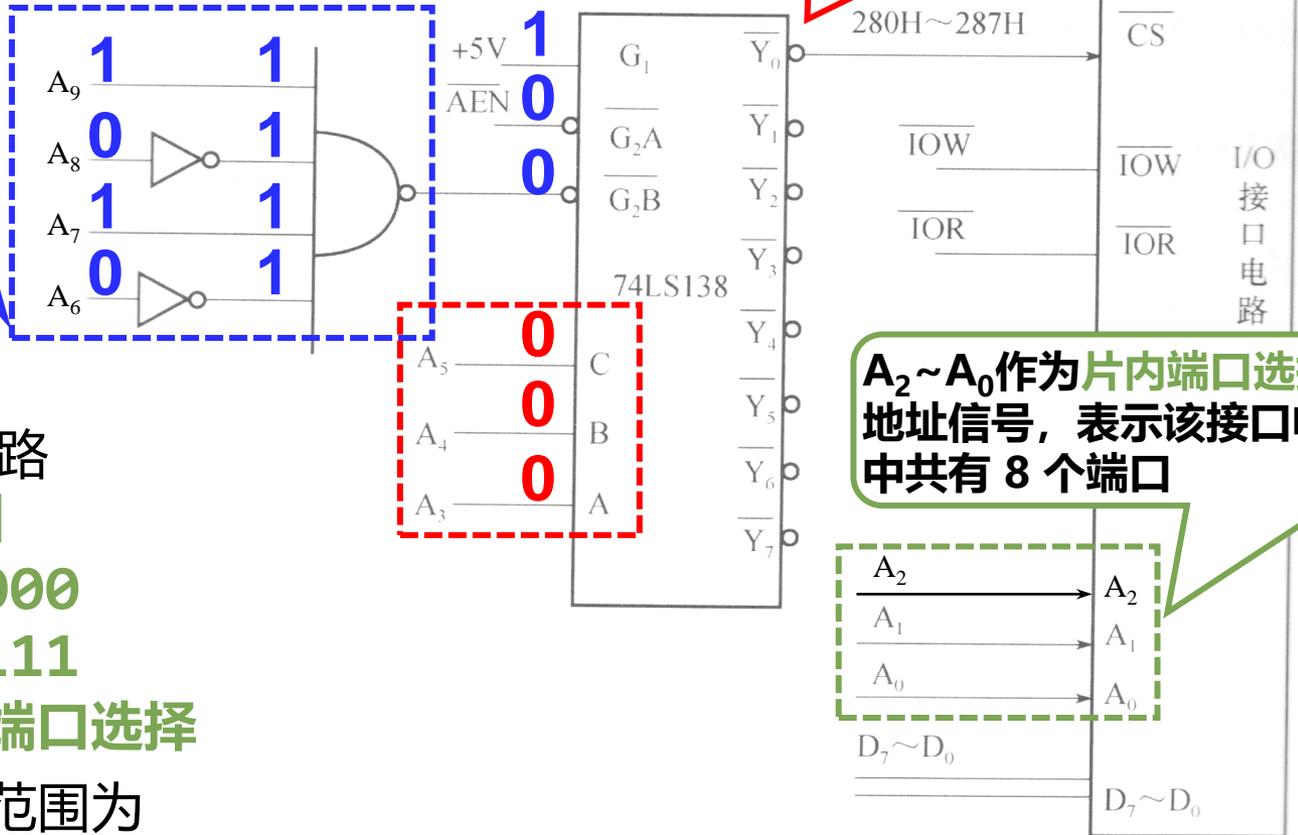
使能输入			选择输入			$\overline{Y_0} \sim \overline{Y_7}$ 输出
G_1	$\overline{G_{2A}}$	$\overline{G_{2B}}$	C	B	A	
1	0	0	0	0	0	$\overline{Y_0} = 0$, 其余为 1
1	0	0	0	0	1	$\overline{Y_1} = 0$, 其余为 1
1	0	0	0	1	0	$\overline{Y_2} = 0$, 其余为 1
1	0	0	0	1	1	$\overline{Y_3} = 0$, 其余为 1
1	0	0	1	0	0	$\overline{Y_4} = 0$, 其余为 1
1	0	0	1	0	1	$\overline{Y_5} = 0$, 其余为 1
1	0	0	1	1	0	$\overline{Y_6} = 0$, 其余为 1
1	0	0	1	1	1	$\overline{Y_7} = 0$, 其余为 1
其他			x	x	x	全为 1

01 | 接口概述 - 固定式多端口译码举例

固定式多端口译码

选通信号： 当且仅当 $A_9 \sim A_6 = 1010$ 时，与非门的输出为0；否则与非门输出均为1

$Y_0\#$ 作为接口芯片的片选，则输入的地址 $A_5 \sim A_3 = 000$ 时，该接口电路被选中



$A_2 \sim A_0$ 作为片内端口选择的地址信号，表示该接口电路中共有 8 个端口

该I/O 接口电路
端口地址范围

10 1000 0000

10 1000 0111

选通 片选 端口选择

所以，端口地址范围为
280H~287H

01 | 接口概述 - 可选式多端口译码

可选式多端口译码

- 接口中的端口地址具有一定的**可变性**。可根据要求改变端口地址，而**无需修改硬件电路**
- 当用户要求接口的端口地址能够**适应不同的地址**场合，或为系统以后**扩充**留有余地时，可用此方式

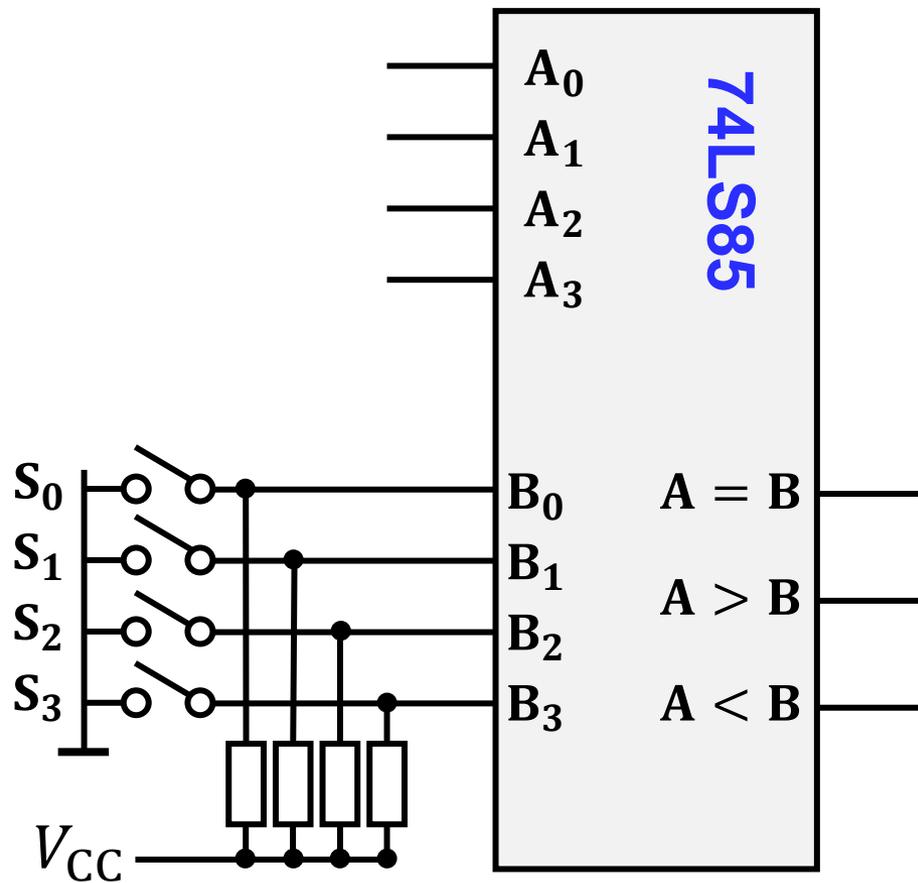
常用的实现方法

- 比较器：**74LS85** (4 位比较器)、74LS688 (8 位比较器)
- 地址开关
- 跳线器

01 | 接口概述 - 74LS85比较器

比较器译码原理

- $A_3 \sim A_0$ 输入端连**系统地址总线**, $B_3 \sim B_0$ 输入端与选择开关相连
- 当 $A_3A_2A_1A_0 = B_3B_2B_1B_0$ 时, 输出端 $A=B$ 输出逻辑 1
- 当 $A_3A_2A_1A_0 > B_3B_2B_1B_0$ 时, 输出端 $A > B$ 输出逻辑 1
- 当 $A_3A_2A_1A_0 < B_3B_2B_1B_0$ 时, 输出端 $A < B$ 输出逻辑 1



01 | 接口概述 - 可选式多端口译码举例

可选式多端口译码

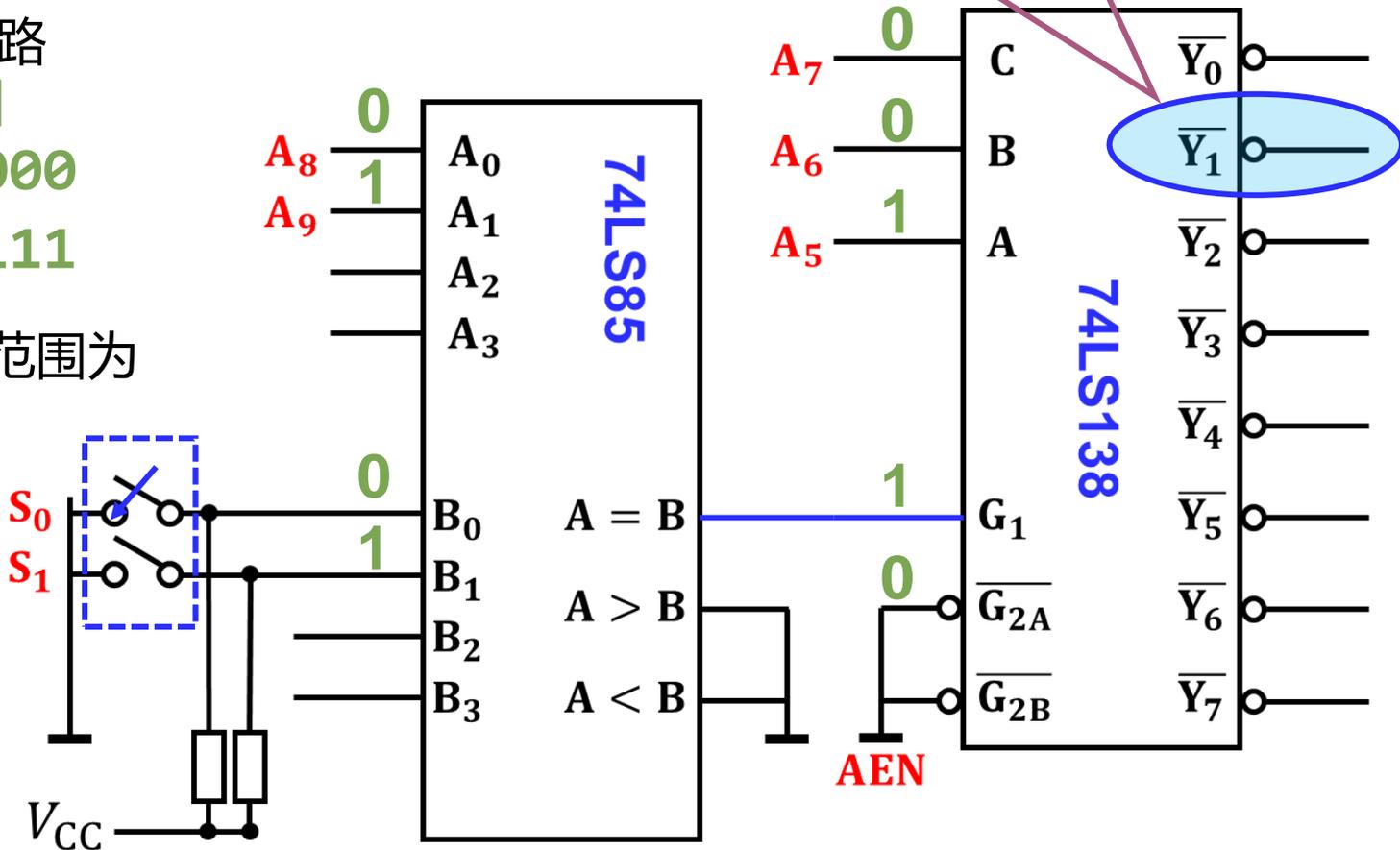
译码地址范围：
220H ~ 23FH

该I/O 接口电路
端口地址范围

10 0010 0000

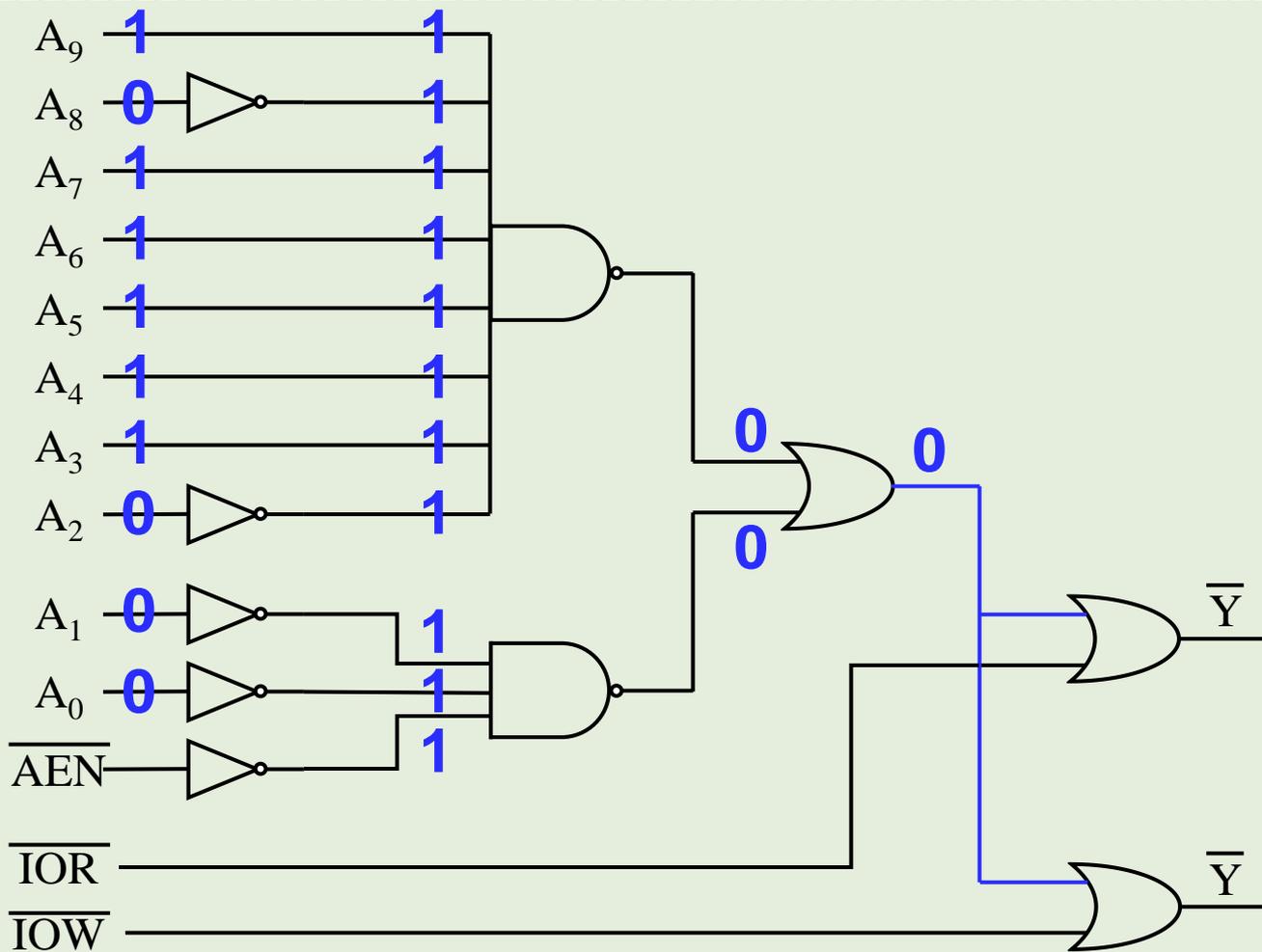
10 0011 1111

所以，端口地址范围为
220H ~ 23FH



01 | 接口概述 - 译码电路示例1

示例1：分析该译码电路的功能



- 端口地址为 **2F8H** 的译码电路
- IOR 有效时从端口读数据
- IOW 有效时向端口写数据

01 | 接口概述 - 译码电路示例2

示例2：设计译码电路

- 已知并行接口芯片 8255A 有 4 个端口，片选信号 CS 低电平有效。试用 74LS138 设计一个译码电路，使该芯片的 4 个端口地址为 2F0H~2F3H。

分析

- 4 个端口，可用低位地址线 A_1A_0 进行片内端口译码，而高位地址 $A_9\sim A_2$ 和 AEN 则用于片选，高位地址线构成 2F0H，如下表所示

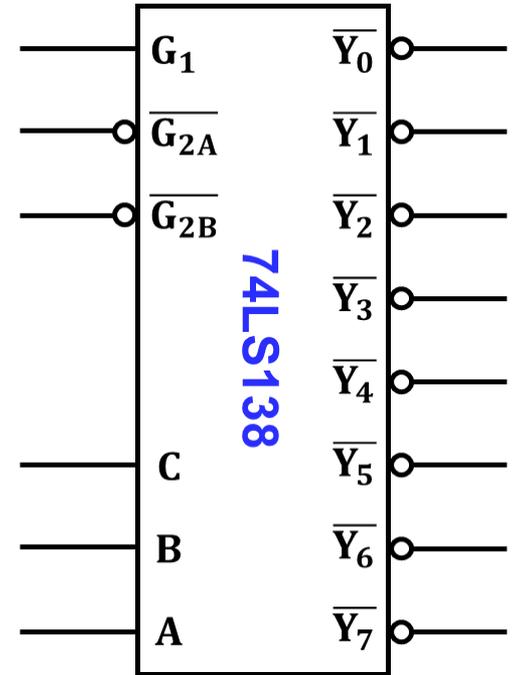
地址	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	AEN
2F0H	1	0	1	1	1	1	0	0	0	0	0
2F3H	1	0	1	1	1	1	0	0	1	1	0

01 | 接口概述 - 译码电路示例2

示例2：设计译码电路

- 已知并行接口芯片 8255A 有 4 个端口，片选信号 CS 低电平有效。试用 74LS138 设计一个译码电路，使该芯片的 4 个端口地址为 2F0H~2F3H。

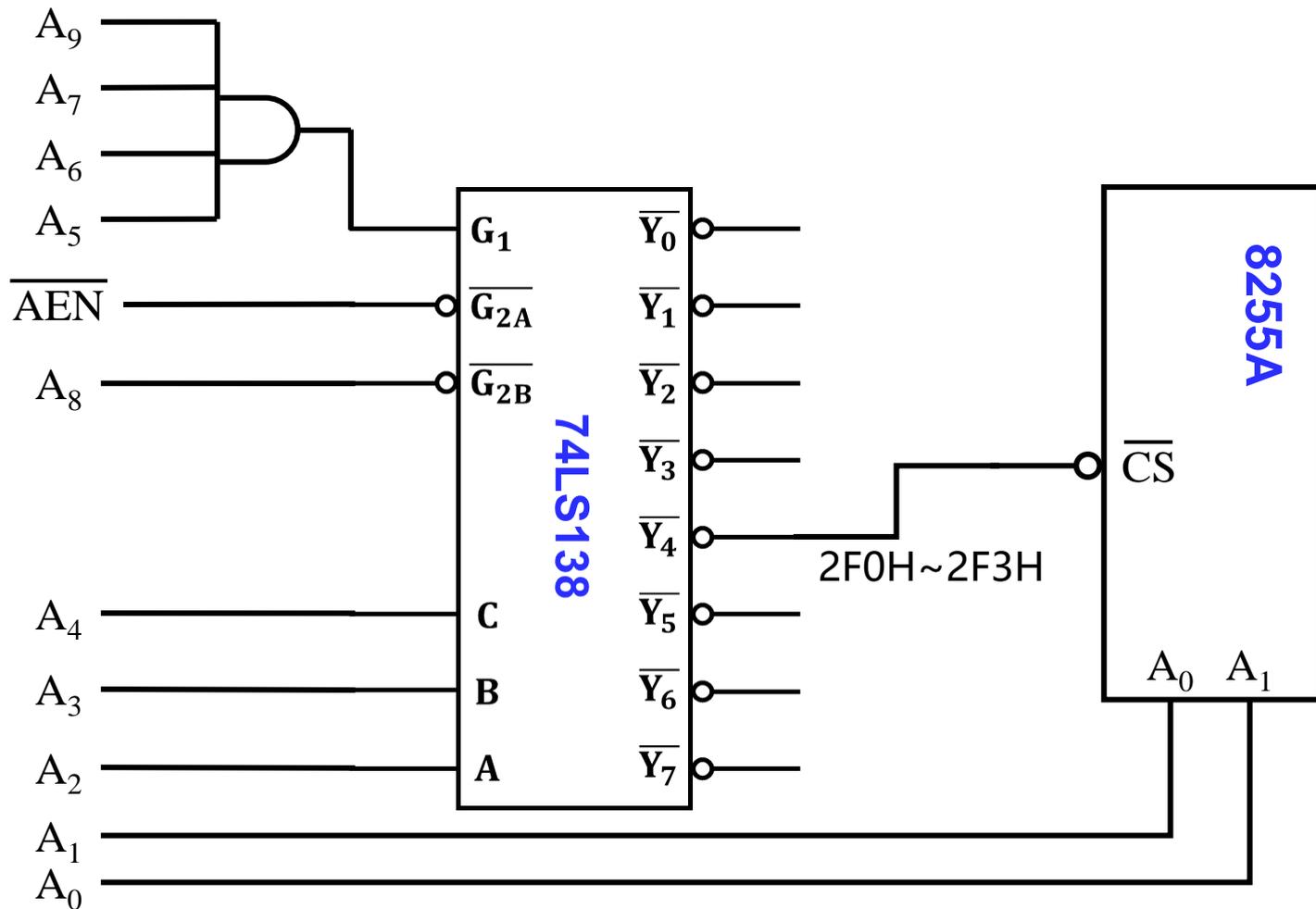
- 74LS138 有 3 个使能信号 (G 、 G_{2A} 、 G_{2B})，3 个端口选择信号 (C 、 B 、 A)



地址	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	AEN	
2F0H	1	0	1	1	1	1	0	0	0	0	0	
2F3H	1	0	1	1	1	1	0	0	1	1	0	
	G	G_{2B}	G			C	B	A				G_{2A}

01 | 接口概述 - 译码电路示例2

示例2：设计译码电路



01 | 接口概述-子目录

接口概述

- 8086 CPU 管脚
- 接口的概念、功能和组成
- I/O 端口及其编址方式
- I/O 端口地址的译码
- 输入/输出指令

01 | 接口概述-输入/输出指令

输入/输出指令 IN/OUT 格式

- 输入指令: **IN** AL/AX, PORT/DX
- 输出指令: **OUT** PORT/DX, AL/AX
 - 累加器 AL/AX 与指定 I/O 端口之间进行数据交换
 - 数据只能在端口和累加器之间进行传递

注意事项

- PORT/DX: 根据**地址宽度**决定表示方法
 - **直接寻址**: 8 位端口地址 (0~0FFH), 可直接在指令中出现
 - **间接寻址**: 16 位端口地址 (100H~3FFH), 必须由 **DX** 表示
- I/O 指令在**书写格式上**两操作数的位数不一定一致
 - 一个操作数为**数据**, 另一个操作数为**地址**

01 | 接口概述-输入/输出指令举例

从 218H 端口读入一个字数据

- `IN AX, 218H`

 ✘ 端口**直接**寻址范围只能是 00~0FFH, 应改

```
MOV DX, 218H
```

```
IN AX, DX
```

将一个字节数据写入到 254 端口

- `OUT 254H, AX`

 ✘ 首先要写入端口的数据是**字节数据**, AX 应改为 AL, 其次输入端口 254 是**十进制数**, 对应十六进制数为 0FEH, 应改为

```
OUT 254, AL
```

或

```
OUT 0FEH, AL
```

目录

01 接口概述

02 CPU 与外设数据传输控制方式

02 | 外设数据传送-子目录

I/O设备数据传送方式

- 微型机 I/O 端口的布局
- CPU 与外设数据的传输控制方式
 - 无条件传送方式
 - 条件传送方式
 - 中断控制方式
 - DMA 传送方式

02 | 外设数据传送-I/O端口布局

I/O 接口硬件分类

- **系统板**上的 I/O 芯片
 - 可编程的大规模集成电路（定时器、计数器、中断控制器等）
- I/O **扩展槽**上的接口控制卡
 - 若干集成电路按一定的逻辑功能组成的部件（串行接口、并行接口、键盘接口、磁盘接口等）

I/O 端口的地址分配

- IBM-PC 机的 I/O 端口地址范围为 0~3FFH，共 1024 个
- 地址分配：

{	0~0FFH : 主板芯片
	100H~3FFH : 扩展槽

02 | 外设数据传送 - 常用接口芯片的地址分配

常用主板芯片地址分配

接口芯片	可用地址范围	单片使用的实际地址范围
8237A (DMAC)	000 ~ 01FH	000 ~ 00FH
8259A (中断控制器)	020 ~ 03FH	020 ~ 021H
8253A (定时计数器)	040 ~ 05FH	040 ~ 043H
8255A (并行接口芯片)	060 ~ 07FH	060 ~ 063H

02 | 外设数据传送 - 常用接口芯片的地址分配

常用外设端口地址分配

IO名称	端口地址
游戏控制卡	200 ~ 20FH
并行口控制卡	270 ~ 27FH
串行口控制卡	2F0 ~ 2FFH
用户可用	300 ~ 31FH
同步通行卡	3A0 ~ 3AFH
显卡	3B0 ~ 3BFH
软驱控制卡	3F0 ~ 3F7H
PC 网卡	360 ~ 36FH

02 | 外设数据传送-子目录

I/O设备数据传送方式

- 微型机 I/O 端口的布局
- CPU 与外设数据的传输控制方式
 - 无条件传送方式
 - 条件传送方式
 - 中断控制方式
 - DMA 传送方式

02 | 外设数据传送-无条件传送方式

工作过程

- CPU 需要传送数据时，直接将数据送至指定外设

接口要求

- 接口电路中只需要有**数据寄存器**即可

程序设计要点

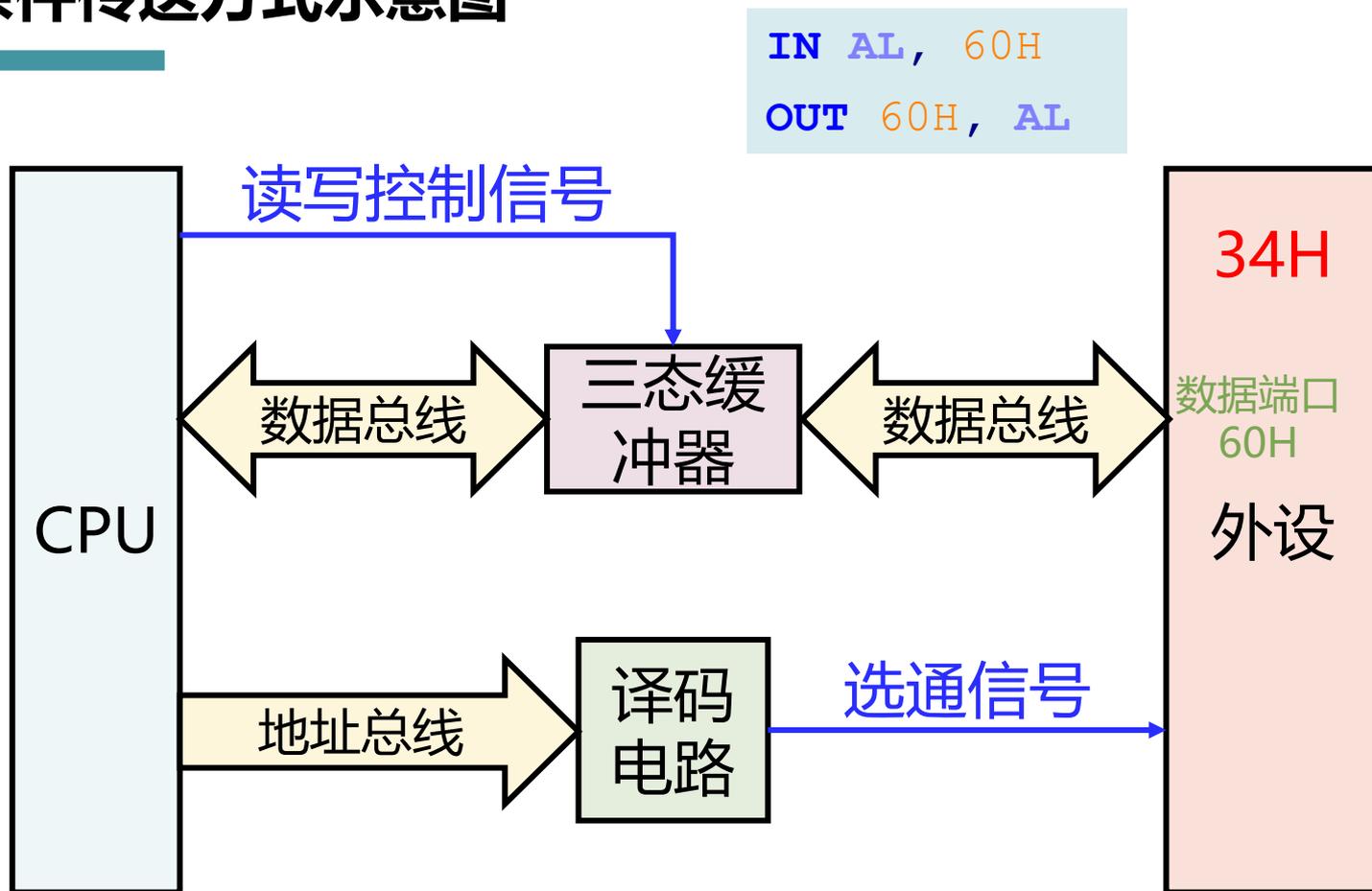
- **使用输入输出指令**，CPU 主动与指定端口数据传送

适用场合

- 与简单外设的数据传送 (**外设始终就绪**)

02 | 外设数据传送-无条件传送方式

无条件传送方式示意图



02 | 外设数据传送-条件传送方式

工作过程

程序查询传送方式

- CPU 传送数据之前先**检查外设的状态**，若没有准备好，则继续查询等待，直至外设就绪即进行数据传送

接口要求

- 接口电路中需要**数据寄存器**和**状态寄存器**

程序设计要点

- CPU 主动查询，**程序控制**数据传送过程，简单易行
 - 每次查询之后只能使用 IO 指令传送一个数据
 - **数据传输率不高**，CPU 时间浪费较多

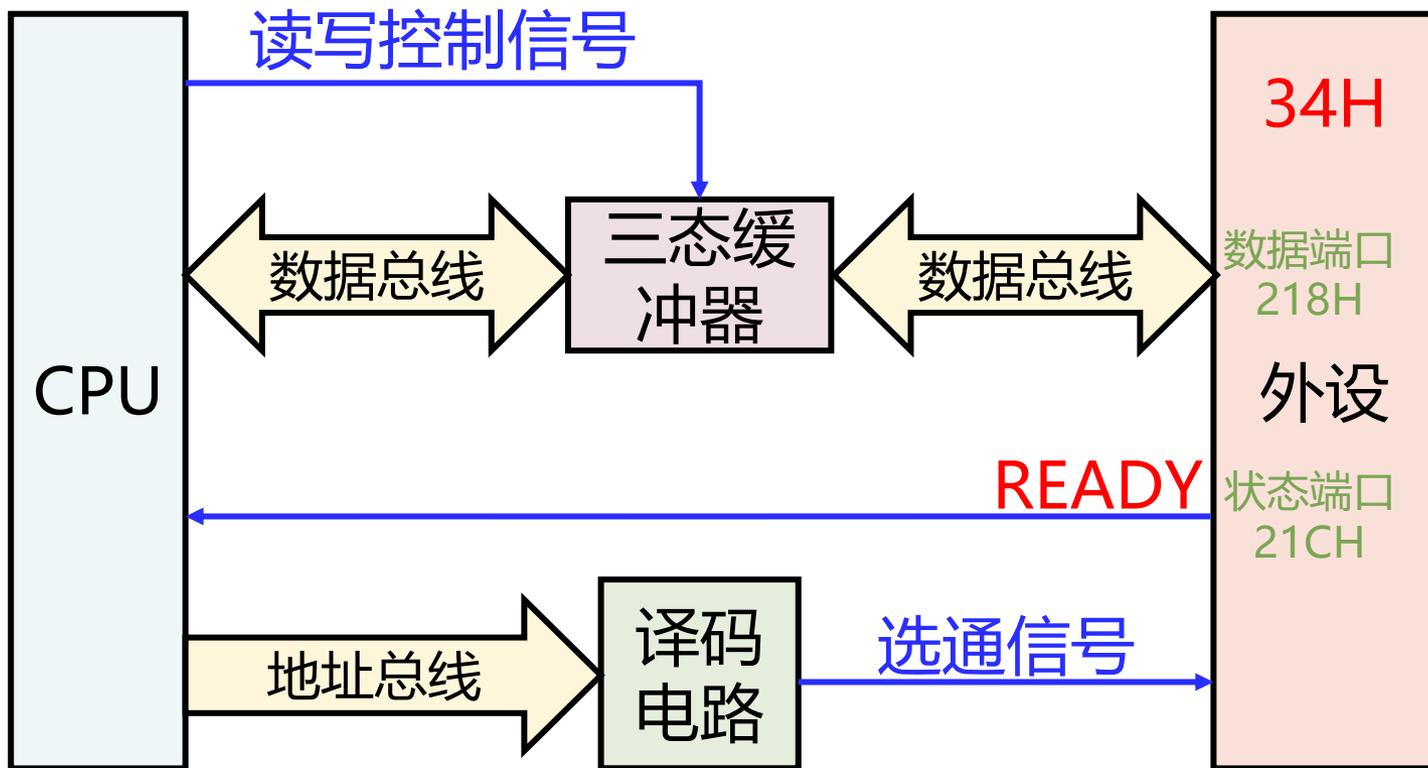
适用场合

- CPU 不太忙且对传送速度要求不高的系统

02 | 外设数据传送-条件传送方式

条件传送方式示意图

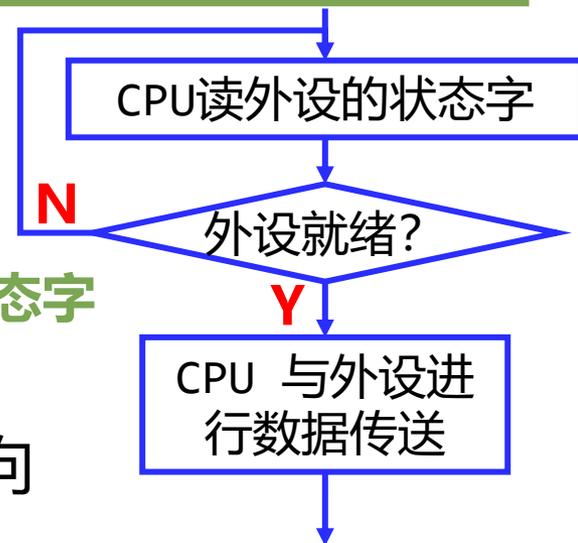
```
MOV DX, 218H  
IN AL, DX  
OUT DX, AL
```



02 | 外设数据传送-条件传送方式

程序设计流程

- CPU 获取外设的**状态**
 - 使用 **IN** 指令，读接口电路的状态端口中的**状态字**
 - 状态字随外设的工作情况而更新
- 判断状态字中的对应位，以决定程序的流向
- 若外设就绪，则进行数据传送
 - CPU 使用 **IN/OUT** 指令读写接口电路的数据端口



```
1.      .....
2. NEXT: MOV DX, 21CH
3. ASK:  IN AL, DX      ; 从状态端口读入状态信息
4.      TEST AL, 10H    ; 10H=0001 0000B 检测 D4 位
5.      JZ ASK          ; D4=0, 继续查询
6.      MOV DX, 218H
7.      IN AL, DX      ; 从数据端口读入数据
8.      .....

```

02 | 外设数据传送-中断传送方式

工作过程

- CPU 有传送要求时，启动外设后可处理其他事件，当外设准备就绪后，通过**中断的方式**和 CPU 完成数据传送工作

接口要求

- 需要附加的**中断控制电路**

软硬件综合控制方式

- 一定程度上实现了 CPU 和外设的**并行**工作
- 中断操作过程增加了软硬件的开销
- 每次数据传送只能传送一个数据，数据传送效率低

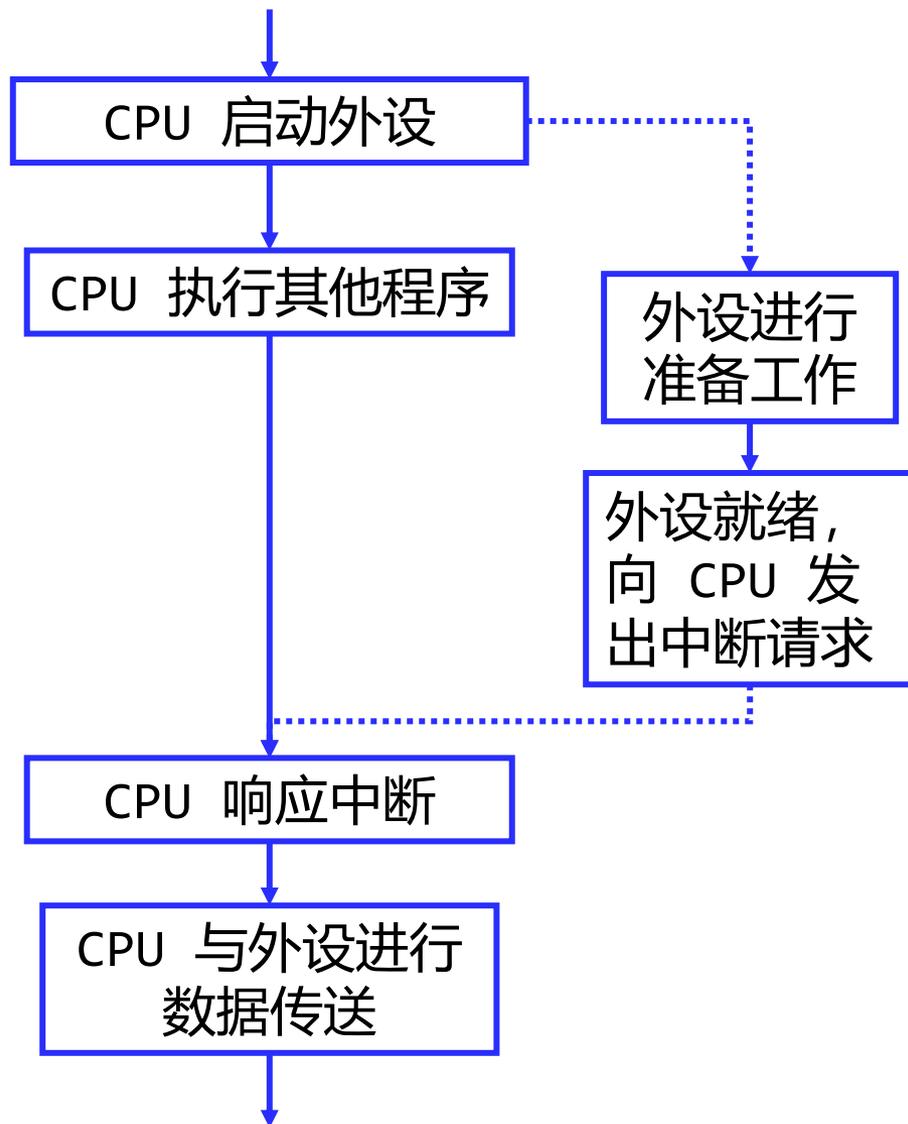
适用场合

- CPU 与慢速外设之间的数据传送

02 | 外设数据传送-中断传送方式

程序设计流程

- CPU 对接口进行初始化, 以启动外设
- 在外设进行数据传送准备的同时, **CPU 可以执行其他的程序**
- 外设就绪后, 向 CPU 发出硬件的**中断**请求信号
- CPU 接受中断请求后进行数据传送



02 | 外设数据传送-DMA方式

工作过程

DMA (Direct Memory Access, 直接存储器存取) 方式

- 在硬件 **DMA 控制器** (DMAC) 的控制下, 存储器与高速外设之间进行大量的数据传送

实现方法

- 由专用接口芯片 DMAC 控制传送过程
- 当外设需传送数据时, 通过 DMAC **向 CPU 发出总线请求**
- CPU 发出总线响应信号, **释放总线**
- **DMAC 接管总线**, 控制外设、内存之间直接数据传送

02 | 外设数据传送-DMA方式

程序设计方式

- CPU 完成 DMAC 的初始化设置后，由硬件或软件的 DMA 请求，触发 DMAC 控制的数据传送

特点

- 数据传送不需要经过 CPU，直接由硬件控制
 - 但需要 CPU 分配总线的使用权
- 数据传送量大，**传送效率高**

适用场合

- 数据传输率要求较高的系统中

本章小结

- 掌握接口的基本**功能**
- 掌握端口的概念、端口的**编址方式**
- 会**求解**译码电路的寻址范围
- 掌握 CPU 与外设之间的**信息传送方式**

- 数据缓冲
- 执行CPU命令
- 接收外设状态
- 设备寻址
- 信号的类型转换等

- 统一编址方式
- 独立编址方式

- 无条件传送方式
- 条件传送方式
- 中断控制方式
- DMA 传送方式

