**数字逻辑基础**

现代数字电子计算机是由各种逻辑部件组成的。

’ 数字逻辑电路是研究具有两个离散状态(“ “0” ” 和“ “1 ”) 的开

关器件所构成的电路，描述电路输入与输出之间的关系。

’ 数字逻辑的分类： 组合逻辑 和 时序逻辑 。

’ 组合逻辑：逻辑系统的输出结果仅取决于当前各输入值；

’ 时序逻辑：逻辑系统的输出结果既由当前各输入值，又由过去的输入

值来决定。

’ 组合逻辑不包含存储元件，时序逻辑至少包含一个存储元件。

逻辑门电路 是指能够实现基本逻辑运算的单元电路。

’ 门电路是数字逻辑电路的基本组成单位

’ 门电路具有一个或多个输入端，通常只有一个输出端。

’ 门电路的各输入端所施加的逻辑信号( 脉冲或电平) ，只有

满足一定的条件时，才会产生相应的信号输出，相当于

“门”被打开了 。

**译码器与选择器**

译码器(decoder) 是一种具有“翻译”功能的 多输入多输

出 的组合逻辑电路器件。

’ 译码器的功能：将每一组编码序列信号转换为一个特定

的输出信号

’ 译码器的输入： 一组编码序列信号

’ 译码器的输出： 一条特定的译码信号 （ 与每组输入信号对应 ）

’ 译码器的工作原理：当某组编码进入输入端时，相应的

译码线输出为 低电平 ，与此同时，其他所有译码线输出

保持为 高电平 。

’ 通常，译码器的输出端与输入端的数量关系为（2^n ）

数据选择器是一个多输入、单输出的 组合逻辑电路 。

’ 基本功能：在 选择信号 的控制下，从多路输入数据中选

择其中的一路数据作为 输出 。

’ 选择信号是一组编码序列，也称为地址编码 信号 。

’ 用数据选择器可以实现数据的多路分时传送.

**数码显示管与半加器**

把十进制数的每一位分别写成二进制形式的编码

’ 在计算机中使用BCD 格式可以保存数值的精确度，又可免去使计算机

作浮点运算时所耗费的时间，也用于简化对使用十进制数字的设备

（比如时钟和计时器）的处理。

’ 8421 编码是最常用的一种BCD 码，是一种 有权码

’ 使用四位二进制数表示一位十进制数，从左到右每一位对应的权分别

是2 3 、2 2 、2 1 、2 0 （8421 ）

’ 例：1975 （D ）=0001 1001 0111 0101 （BCD ）

’ 用四位二进制表示一位十进制，会多出6 种状态（1010~1111 ），一般被

称为 非法码

**触发器实验**

从电路的组成上来看，时序逻辑电路一定包含有

触发器。

’ 时序逻辑电路的基本单元是 触发器

’ 组合逻辑电路的基本单元是门电路

触发器是一种具有存储和记忆功能的单元电路，

可用于接收、存储、输出二进制代码0 和1

’ 从时序关系来看，触发器的 次态 不仅与输入信号

状态有关，而且与触发器的 现态 有关

’ 触发器接受输入信号之前的状态叫做现态

’ 触发器接受输入信号之后的状态叫做次态

’ 现态和次态是两个相邻离散时间里触发器的输出端的状态

’ 现态和次态是一组相对的关系

触发器分为双稳态、 单稳态和无稳态触发器（多

谐振荡器）

’ 双稳态触发器输出有两个稳定状态0 和1

’ 本实验所涉及的几种触发器都是 双稳态触发器

’ 基本R-S 触发器

’ JK 触发器

’ D触发器

计数器实验

计数器是用来累计电路输入 脉冲个数 的 时序电路 。

’ 在计数功能的基础上，计数器还可以实现计时、定时、

分频等多种功能

’ 计数器的电路结构中包含有触发器（常用JK 类型）。

’ 计数器按照脉冲的输入方式可分为 同步计数器 和 异步

计数器 。

同步计数器的 时钟脉冲 接到所有的触发器CP 端，

各个触发器的翻转是同时进行的

异步计数器的 时钟脉冲 不是同时接到所有的触发器

CP 端，触发器的翻转不是同时进行的，一部分触

发器CP

利用芯片的复位端和门电路逻辑，跳越M-N 个状态，

从而获得N 进制计数器

’ 计数器的清零方式分为异步和同步两种

**计算机系统认识**

计算机系统组成（五大部件）

◦ 运算器、控制器、存储器

◦ 输入设备、输出设备

’ 计算机内部信息流

◦ 数据信息流：原始数据、中间结果、程序、地址等

◦ 控制信息流：控制各个设备部件的动作

计算机系统操作过程概述

’ 通过输入设备接收程序和数据信息，传送到

存储器进行存放

’ 通过控制器分析存放在存储器中的程序，并

将其中的数据信息读取到运算器进行处理

’ 将处理结果送到计算机的输出设备或再次返

回到存储器

’ 控制器是核心部件，负责指挥计算机内部所

有部件的活动

时序发生器组成：

◦ 脉冲源：基准时钟信号

◦ 脉冲分配器：产生节拍电位和脉冲信号

◦ 启停控制电路

’ 时序发生器工作原理：

◦ 由时序单元φ 提供脉冲源

◦ 控制信号CLR、STOP、START

◦ 通过循环移位寄存器输出T1~T4环形脉冲

**静态随机存储器**

RD (Read)：RD 信号用于读取操作。当该信号被激活时，存储器会把指定地址的数据发送到数据总线上，以便于其他组件（如处理器）读取。

WR (Write)：WR 信号用于写入操作。当该信号被激活时，数据总线上的数据会被写入到存储器的指定地址中。

IOM (Input/Output Memory)：IOM 信号用于区分存储器操作是对内存的操作还是对输入/输出设备的操作。根据IOM的状态，系统可以确定是访问内存还是进行I/O操作。

IOR (Input/Output Read, IN-B)：IOR或IN-B信号通常用于输入操作，特别是在与外部设备交互时。它指示系统从外部设备读取数据。

LDAR (Load Address Register)：LDAR 信号用于加载地址寄存器。当此信号激活时，它指示系统将地址总线上的地址值加载到地址寄存器中，以便于后续操作（如读取或写入）定位到正确的存储位置。

**基本运算器**

算术、逻辑和移位三个运算部件中，只有一个部件的结果作为ALU的输出，由多路选择开关控制（数据选择器）

’ 控制信号S3…S0和CN决定对操作数进行何种方式的运算

’ T4 由时序单元的TS4 提供

74LS245三态门，控制输入输出操作

‘ ALU-B置低电平：运算器单元的运算结果进入总线

‘ IN-B置低电平：输入单元的数据进入总线

’ 74LS273实现两个操作数的数据锁存

‘ LDA/LDB置高电平

‘ 脉冲触发-T4

’ 总线灯（LED）显示总线上的数据

**微程序控制器**

控制器是计算机的核心部件，用来实现对计算机各部件

的有效控制，完成计算机程序规定的一系列操作

’ 根据指令操作码和时序信号，产生各种控制信号，完成

取指令和执行指令的过程

’ 早期计算机采用的是组合逻辑控制器，运行速度快但设

计负责，可维护性、可扩展性差

’ 目前广泛采用的是微程序控制器

微程序控制的基本思想：

◦ 把操作控制信号编成一条“微指令”，存放在一个只

读存储器中。

◦ 当计算机运行时，按照一定的顺序逐条读出这些微指

令，从而产生各种操作控制信号，使相应部件执行所

规定的操作。

’ 微程序控制器基本任务：

◦ 完成当前指令的翻译与执行

◦ 将当前指令功能转换为硬件逻辑部件工作的控制信号

◦ 完成数据的传送、处理和存储操作

微命令（微信号）

◦ 直接作用于硬件控制电路的控制命令（信号）

’ 微操作

◦ 由微命令控制实现的一些最基本的操作

‘ 如：取指令地址、读取指令、取操作数地址、取操作数、

基本运算、写回结果等都是微操作

’ 一条微命令对应一个微操作

微指令

◦ 一个微周期所需的微命令的集合，用二进制编码表示

◦ 可以同时发出的多个控制信号，执行一组微操作

◦ 一个典型的加法指令的执行可分为：取指令、计算地址、取操

作数、执行加法运算、返回计算结果等步骤，每一步都由一条

微指令实现。

’ 微周期

◦ 读取一条微指令并完成相应微操作所用的时间，通常为一个机

器周期

’ 微程序

◦ 若干条微指令的有序序列

◦ 每条机器指令对应一段微程序

◦ 一条机器指令需要由多个微周期来完成。

一段计算机程序由一系列机器指令组成

◦ 比如：in、add、out、jump。。。

’ 每一条机器指令由一段微程序实现

’ 每一段微程序由一组微指令表示

’ 每一条微指令由一系列微操作组成

’ 一个微操作对应一条微命令

微程序控制器的组成：

◦ 控制存储器CM、微指令寄存器μIR和地址转移逻辑三大

部分。

微程序控制器组成

’ 控制存储器CM 24位

◦ 用于存放微程序

◦ 每个单元存放一条微指令代码，只读

’ 微指令寄存器μIR

◦ 存放由控制存储器读出的一条微指令信息

◦ 包括 操作控制字段 和 判别测试字段

’ 微地址寄存器μAR 6位

◦ 用于存放下一条微指令的地址

’ 微指令存放在控制存储器CM里

’ 微指令格式：24位字长

’ μA5~μA0为6位的后继地址.

’ A,B,C为三个译码字段



’ C 字段中的P<1>为测试字位。

’ 其功能是根据机器指令及相应微代码进行译码，使微程序转

入相应的微地址入口，从而实现对机器指令的识别，并实现

微程序的分支

在一般情况下，微指令由控制存储器读出后直接给出

下一条微指令的地址，即微地址，存放在微地址寄存

器中。（T2）

’ 如果微程序不出现分支，那么下一条微指令的地址就

直接由微地址寄存器给出。

’ 如果微程序出现分支，意味着微程序出现条件转移。

在这种情况下，通过判别测试字段P和执行部件的状态

条件信息，去修改微地址寄存器的内容，并按修改后

的微地址读取下一条微指令，从而实现地址转移。

工作过程

机器周期开始，执行公共取指操作

’ 读机器指令，送入指令寄存器（同时修改PC）

’ 进行指令译码，取出操作码产生对应的微程序入口地

址，送入微地址寄存器

’ 取出对应的一条微指令，送入微指令寄存器

’ 微指令操作字段经译码产生一组微命令，送往相应的

执行部件

’ 在时序控制下完成微操作（T2/T4）

’ 产生后续微地址，读取下一条微指令执行完一段微程

序后，开始新的机器周期

**总线实验**

总线是计算机各部件之间进行数据传输的公共通路，

是一组导线和相关的控制、驱动电路的集合

◦ 数据总线、地址总线、控制总线

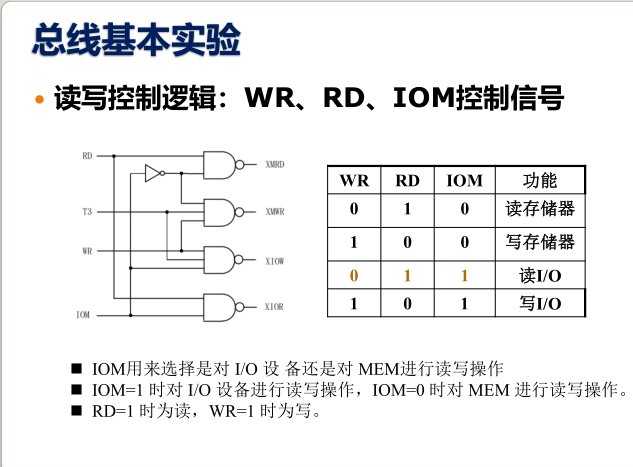
’ 计算机各部件挂接在外部总线上，通过三态门控制

’ 外部总线和 CPU 内总线之间通过三态门连接，实现

了内外总线的分离和数据流向的控制。

’ 同一时刻只能有一个部件占用总线发送信息，但可以

有多个部件通过总线接收信息



关于R0-B、LDR0、LDAR控制信号

◦ R0-B（接CON单元K7）

‘ 置1时，R0寄存器输出关闭

‘ 置0时，R0寄存器输出打开

◦ LDR0（接CON单元K6）

‘ 置1时，R0寄存器输入打开

‘ 置0时，R0寄存器输入关闭

◦ LDAR

‘ 置1时，允许访问地址寄存器

‘ 置0时，不允许访问地址寄存器

**简单模型机实验**

CPU的组成

◦ 运算器ALU：完成数据计算或处理

‘ 暂存器A、B

◦ 微程序控制器MC：控制机器指令的分析和译码

◦ 寄存器：用于暂存数据和指令

‘ 通用寄存器R0：用于临时存放中间数据

‘ 程序计数器PC：用于存放下一条指令或数据的地址

‘ 地址寄存器AR：用于存放当前指令或数据的地址

‘ 指令寄存器IR：用于存放当前正在执行的机器指令

CPU基本功能：读取并执行指令

’ CPU要完成的工作：

◦ 取指令：读主存，装入寄存器（公共）

◦ 分析指令：指令译码，决定动作（公共）

◦ 执行指令：

‘ 取数据： 从主存或I/O读取操作数（如有）

‘ 处理数据：对操作数进行算术或逻辑运算

‘ 写数据：将执行结果写到主存或I/O

之前的部件实验过程中，各部件单元的控制信

号基本上都是人为模拟产生的，比如：

◦ 运算器实验中的运算控制信号S3-S0

◦ 存储器实验中的读写控制信号WR、RD、IOM0

◦ 对输入、输出设备的控制WR、RD、IOM1

◦ 控制器实验中对微程序强制分支的控制信号P1

’ 本实验中，各部件单元的控制信号和计算机数

据通路的控制都是在微程序控制下自动产生的，

实现特定指令的功能。