

简易计算机的设计与实现
西安交通大学
软件学院
数字逻辑综合实验

2017 年 6 月 9 日

组 长 易 凯 2151601053
成 员 庞建业 2151601012
屈 松 2151601067
邢贺宇 2151601070
联系邮箱 williamyi96@gmail.com
联系电话 13772103675
项目网站 <https://williamyi.tech>
提交日期 2017 年 6 月 9 日

目录

1 实验研究基本背景	4
1.1 简介	4
1.2 计算机体系结构	4
1.3 计算机各部件基本功能	5
1.4 计算机的指令系统	5
1.5 计算机的工作原理	5
1.6 计算机的设计过程	6
2 实验目的	6
3 实验原理	6
4 系统设计	6
4.1 存储器	6
4.2 寄存器, 存储器及其功能	8
5 逻辑设计	8
5.1 基本配置框图	8
5.2 程序计数器 PC 设计与实现	9
5.3 指令寄存器 IR 与指令译码电路设计与实现	9
5.4 寄存器 A,B,R 设计与实现	9
5.5 运算累加器设计与实现	9
6 实验平台	9
6.1 实验设备	9
6.2 实验器件	9
6.3 软件工具	9
7 实验内容	9
7.1	9
7.2	9
8 实验步骤及过程	10

目 录	3
9 实验总结	10
10 致谢	10
11 参考文献	10
12 附录	10
12.1 简易计算机指令集	10
12.2 内存地址及其对应指令	10
12.3 微操作表	10
12.4 集成化微操作序列	10

1 实验研究基本背景

1.1 简介

计算机作为当代社会最典型、最常用，同时也是最为复杂的数字系统，在各行各业中越来越发挥着不可替代的作用。虽然当下以布尔代数为基础的计算机系统发展相对成熟，但是仍然具有着极高的学习和研究意义。

1.2 计算机体系结构

现代计算机的结构遵循的是冯·诺依曼计算机的定义，也就是说计算机由五大部件构成，其分别为运算器 ALU(算法逻辑部件)、控制器 CU、存储器 RAM(随机存取存储器) 和 IO(输入输出设备)。

由于运算器和控制器在逻辑关系和电路结构上联系的密切性，尤其是在大规模集成电路制作工艺出现之后，往往将运算器和控制器制作在同一芯片上，称为中央处理器，简称为 CPU。

现代计算机可认为是由 CPU、存储器和 I/O 设备这三个部分构成。

如下则是一个典型的现代计算机系统的结构图：

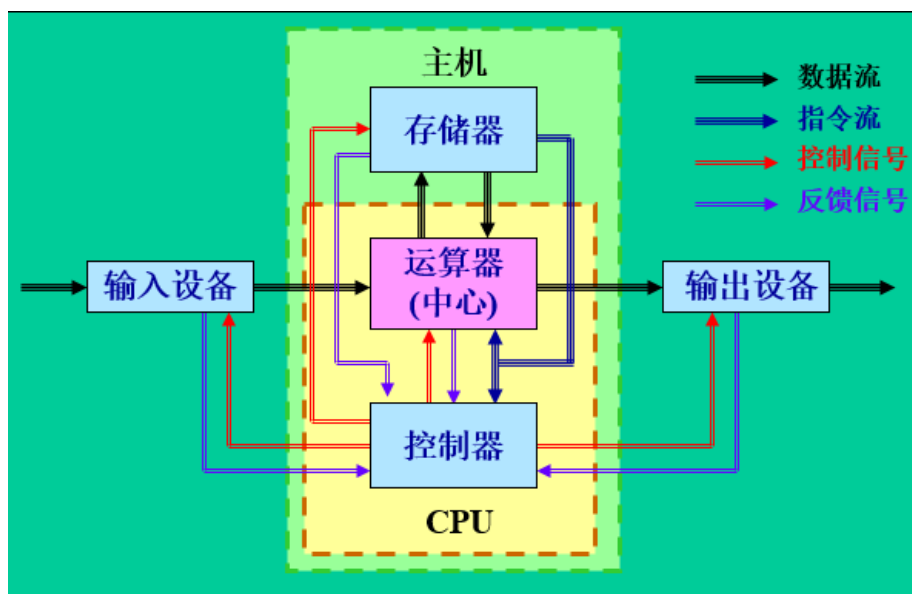


图 1: 计算机体系结构

1.3 计算机各部件基本功能

CPU 的基本功能： 根据程序执行所期望的信息处理，其中 ALU 执行算术逻辑运算，CU 根据程序从存储器中读出并执行指令和数据，以执行要求的运算和操作。

存储器的基本功能： 计算机存储指令、数据和信息的空间；

输入输出设备的基本功能： 输入设备将指令、数据和信息输入到计算机，输出设备将处理后的结构由计算机输出出来。

1.4 计算机的指令系统

指令就是用以表示计算机微操作序列的二进制代码，它决定了 CPU 应该执行什么样的具体操作。

操作系统就是一台计算机所能够执行的全部指令，在某种程度上，指令系统的完备与否以及功能强弱体现了计算机整体性能的优劣。同时，计算机设计中最基本的问题就是如何选择和设计出一个完备的、使用方便的指令系统。

由于我们的出发点是在学习的基础之上加深对计算机的认识，以制作模型机为目标，以期该计算机能够满足基本的运算操作，因此我们实际设计时不考虑指令系统的完备性，重点放在考察该系统的可用性。

1.5 计算机的工作原理

计算机的指令和数据都存储在存储器中，控制器每次从存储器中取出一条指令，它决定 CPU 应该执行什么样的操作具体操作，解释其含义，并据此产生一系列功能来执行此指令。

处理一条指令所包含的操作序列称之为一个指令周期，其中，

$$\text{指令周期} = \text{取值周期} + \text{执行周期}。$$

取指周期指的是将指令从存储器中读出的操作序列，执行周期由三部分构成，分别为指令译码、取操作数、完成操作。

另外，值得注意的是，不同指令有不同字长、寻址方式、不同的操作，则其所对应的指令周期也长短不同。

同时，每个指令周期划分为若干个机器周期，每个机器周期划分为若干个节拍，一个节拍通常对应一个时钟周期 T ，时钟周期是执行微操作的最小时间单位。

1.6 计算机的设计过程

计算机的设计可以分为两个阶段，分别为系统设计阶段和逻辑设计阶段。

系统设计： 设计系统的技术指标以及总的性能，确定设计目标、基本结构方案和指令系统。

逻辑设计： 将计算机结构的描述用逻辑电路来实现。

2 实验目的

我们希望通过我们数字逻辑的所学，以及查找相关的资料，按照典型计算机的实现原理来构建简易型的计算机，以满足基本的数据处理需要。

同时，希望在学习实践制作中级计算机的过程中掌握数字逻辑设计的基本方法以及重要思想，在广泛查阅相关资料的基础之上对于现代化的许多硬件设计以及计算机设计方法有一个基本的了解，另外加深对于计算机体系结构的认识，为以后的学习研究打下坚实的基础。

3 实验原理

4 系统设计

综合考虑实际需要，该中级计算机系统将包括 m 个存储器，3 个寄存器，分别为 A、B 和 R。另外有 p 个译码器等 (回过头来进行说明)。

4.1 存储器

每个存储器都是 256 个字，其中每个字 8 位；

该存储器具有以下寄存器：

符号	位数	名称	功能
MAR	8	存储地址寄存器	保存存储器地址
MBR	8	存储缓冲寄存器	保存存储器内容
A	8	寄存器 A	处理数据寄存器
B	8	寄存器 B	处理数据寄存器
R	8	寄存器 R	处理数据寄存器
PC	8	程序计数器	保存指令地址
IR	8	指令寄存器	保存指令操作码
T	8	时序计数器	产生时序信号

表 1: 寄存器及其功能

存储地址寄存器 MAR: 与存储器的地址总线相连, 存放着将要访问的存储单元的地址;

存储缓冲寄存器 MBR: 与存储器的数据总线相连, 存放着要写入或者刚从存储单元读出的信息。

如果读出的为: **指令 = 操作码部分 + 地址部分**, 则将操作码部分送至指令寄存器 IR, 将保留在 MBR 中的地址部分发送至 MAR。

如果读出的为: **操作数**, 则将其放在寄存器 A 和 R 中。

程序计数器 PC: 存放的是后续指令的地址, 且有计数功能。

指令寄存器 IR: 存放的是现行指令的操作码。

操作译码器: 对指令寄存器 IR 提供的每个操作码译码出一个相应的输出变量 q_i 。

时序译码器: 提供计算机时序信号 t1 到 t30 这 8 个时钟信号。

时序计数器: 为计算机提供时序信号的计数器。

以下为寄存器、存储器及其对应功能图表表示:

4.2 寄存器，存储器及其功能

5 逻辑设计

5.1 基本配置框图

为了制作一台简易计算机，总体的配置框图如下所示：

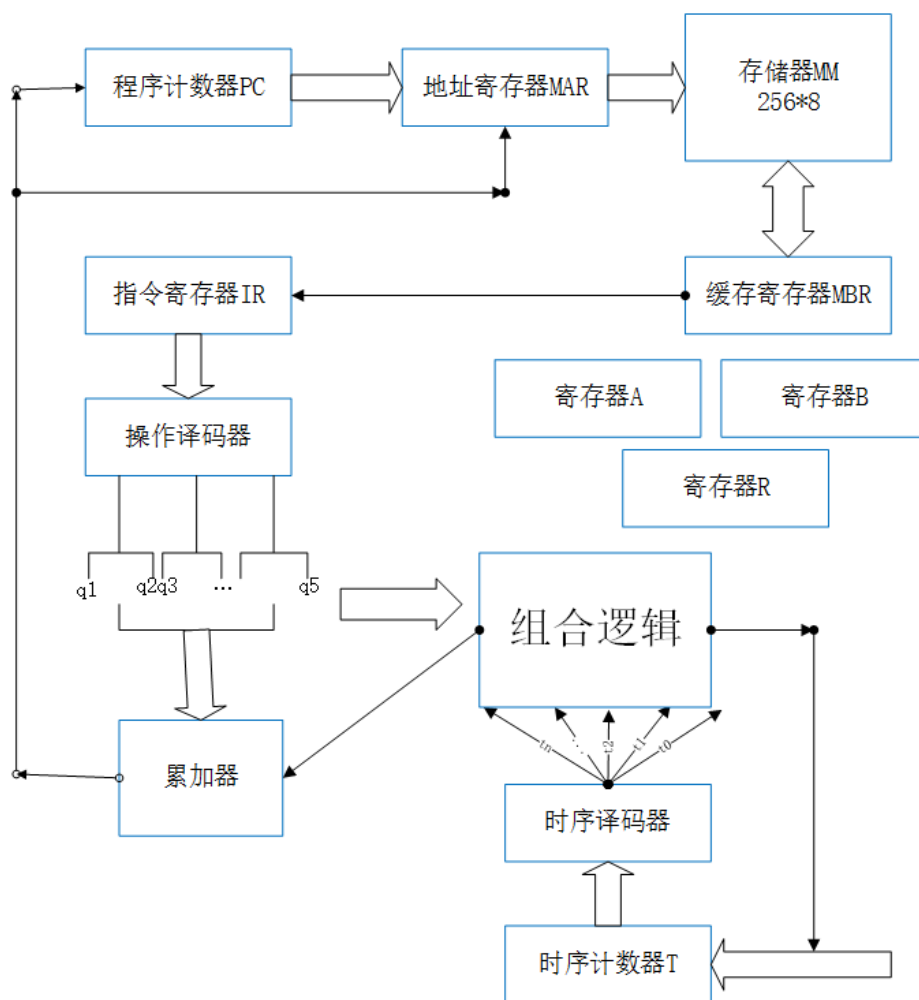


图 2: 简易计算机配置框图

5.2 程序计数器 PC 设计与实现

5.3 指令寄存器 IR 与指令译码电路设计与实现

5.4 寄存器 A,B,R 设计与实现

5.5 运算累加器设计与实现

我们累加器的实现使用 8 位 JK 触发器与组合逻辑电路进行实现，对于任何一个逻辑单元，具有如下的分析框图：

6 实验平台

6.1 实验设备

Cyclone II EP2C8T144C8 试验箱.

6.2 实验器件

相关数据转换线，Win XP 系统台式电脑，Ubuntu gnome 16.01 之下 Win 7 虚拟机.

6.3 软件工具

QUARTER II 9.0, 截图工具，win 系统画图软件，Latex texlive, Win Edt 10.2, Sublime text 3, MiKTeX.

7 实验内容

7.1

7.2

实验内容观看至第二十六页

8 实验步骤及过程

9 实验总结

10 致谢

11 参考文献

- [1] 鲍家元, 毛文林, 张琴. 《数字逻辑》(第三版). 高等教育出版社. 2011 年 11 月.
- [2] 毛文林. 《数字逻辑课件》(第八章). 数字系统设计. 2017 年 4 月.
- [3] 张琴. 《数字逻辑实验课件》. 西安交通大学. 2017 年 4 月.
- [4] Intel® 64 and IA-32 Architectures Software Developer's Manual. Volume 2 (2A, 2B, 2C and 2D): Instruction Set Reference, A-Z
- [5] 杜小智. 《嵌入式软件设计》. ARM 指令集. 西安: 西安交通大学. 2017 年 3 月.
- [6] D.G.Meyer. Design of a Simple Computer. Introduction to Digital System Design. Purdue University.

12 附录

12.1 简易计算机指令集

12.2 内存地址及其对应指令

//R 上电输入 00010101

12.3 微操作表

12.4 集成化微操作序列

操作码	助记符	执行功能
0 0 0 0 0 0 0 0	LDA addr	将地址 addr 中的内容存入 A 寄存器处
0 0 0 0 0 0 0 1	STA addr	将 A 寄存器中的内容存入地址 addr
0 0 0 0 0 0 1 0	ADD addr	将地址 addr 的内容加上 A 寄存器内容，存入 A 寄存器
0 0 0 0 0 0 1 1	SUB addr	将 A 中内容减去 addr 内容，并将结果存入 A 寄存器
0 0 0 0 0 1 0 0	AND addr	将 addr 中内容与 A 寄存器中的内容进行与操作
0 0 0 0 0 1 0 1	OR addr	将 addr 中的内容与 A 寄存器中的内容进行或操作
0 0 0 0 0 1 1 0	MOV A, R	将 R 中的内容存入 A 寄存器
0 0 0 0 0 1 1 1	MOC B, R	将 R 中的内容存入 B 寄存器
0 0 0 0 1 0 0 0	LDI A, OPRD	将操作数存入 A 寄存器
0 0 0 0 1 0 0 1	LDI B, OPRD	将操作数存入 B 寄存器

表 2: 简易计算机指令集

地址	内容	说明
0000 0000	0000 0110	将 R 中内容 (00010101) 存入 A 寄存器
0000 0001	0000 1001	将下一个地址的操作数存入 B 寄存器
0000 0010	0001 1101	操作数 00011101
0000 0011	0000 0010	将寄存器 A,B 内容相加存入 A 寄存器
0000 0100	0000 0000	进行 LDA addr 操作
0000 0101	1000 0001	在地址 1000 0001 取操作数
0000 0110	0000 0011	将 A,B 寄存器相减的内容存入 A 寄存器
0000 0111	0000 1001	将下一个地址的操作数存入 B 寄存器
0000 1000	0011 1100	操作数
0000 1001	0000 0100	将 A, B 寄存器中内容相与存入 A 寄存器
0000 1010	0000 1000	将下一个地址操作数存入 A 寄存器
0000 1011	0000 0101	操作数
0000 1100	0000 0101	将 A, B 寄存器内容或运算, 结果存入 A 寄存器
0000 1101	0000 1000	将下一个地址操作数存入 A 寄存器
0000 1110	1111 1111	操作数
0000 1111	0000 1001	将下一个地址操作数存入 B 寄存器
0001 0000	0101 0101	操作数
0001 0001	0000 0011	将 A, B 寄存器内容相减, 结果存入 A 寄存器
— — — —	— — — —	— — — —
1000 0001	0000 0111	操作数 0000 0111
— — — —	— — — —	— — — —

表 3: 内存地址及其对应指令

时间序列	助记符号	注释
标准取指令周期		
t_0	$MAR \leftarrow PC$	传送指令地址
t_1	Empty	
t_2	$MBR \leftarrow M[MAR], /WREN, PC \leftarrow PC + 1$	读出操作码, 且 PC+1
t_3	Empty	
t_4	Empty	
t_5	$IR \leftarrow MBR$	读出操作码, 将其传给 IR
t_6	Empty	
—	—	—
MOV A, R	将 R 中的内容存入 A	使用 q1 进行控制
$q_1 t_7$	$A \leftarrow R, T \leftarrow 0$	R 内容存入 A, 同时 T 清零
$q_1 t_8$	Empty	
—	—	—
MOV B, R	将 R 中的内容存入 B 寄存器	使用 q2 进行控制
$q_2 t_7$	$B \leftarrow R, T \leftarrow 0$	R 内容存入 B, 同时 T 清零
$q_2 t_8$	Empty	
—	—	—

表 4: 微操作表 1

LDI A, OPRD q_3t_7 q_3t_8 q_3t_9 q_3t_{10} q_3t_{11} q_3t_{12} q_3t_{13} —	将下一地址操作数存入 A $MAR \Leftarrow PC$ Empty $MBR \Leftarrow M[MAR], /WREN, PC \Leftarrow PC + 1$ Empty Empty $A \Leftarrow MBR, T \Leftarrow 0$ Empty —	使用 q3 进行控制 操作数紧跟下一单元 读操作数, PC+1 传送操作数, 同时 T 清零 —
LDI B, OPRD q_4t_7 q_4t_8 q_4t_9 q_4t_{10} q_4t_{11} q_4t_{12} q_4t_{13} —	将下一个地址操作数存入 B $MAR \Leftarrow PC$ Empty $MBR \Leftarrow M[MAR], /WREN, PC \Leftarrow PC + 1$ Empty Empty $B \Leftarrow MBR, T \Leftarrow 0$ Empty —	使用 q4 进行控制 操作数紧跟下一单元 读操作数, PC+1 传送操作数, 同时 T 清零 —

表 5: 微操作表 2

AND A, B q_5t_7 q_5t_8 q_5t_9 q_5t_{10} —	将 A, B 内容与存入 A $A \leftarrow A \text{ and } B, T \leftarrow 0$ Empty Empty Empty —	使用 q5 进行控制 —
OR A, B q_6t_7 q_6t_8 q_6t_9 q_6t_{10} —	将 A, B 内容或存入 A $A \leftarrow A \text{ or } B, T \leftarrow 0$ Empty Empty Empty —	使用 q6 进行控制 —
ADD A, B q_7t_7 q_7t_8 q_7t_9 q_7t_{10} —	将 A, B 内容相加存入 A $A \leftarrow A + B, T \leftarrow 0$ Empty Empty Empty —	使用 q7 进行控制 —
SUB A, B q_8t_7 q_8t_8 q_8t_9 q_8t_{10} q_8t_{11} q_8t_{12} q_8t_{13} q_8t_{14} —	将 A, B 内容相减存入 A $B \leftarrow -B$ Empty $B \leftarrow B + 1$ Empty $A \leftarrow A + B, T \leftarrow 0$ Empty Empty Empty —	使用 q8 进行控制 将 B 取反 将 B 取补 —

表 6: 微操作表 3

LDA ADDR	将地址 addr 中的内容存入 A	使用 q9 进行控制
q_9t_7	$MAR \leftarrow PC$	将下一地址存入 A 中
q_9t_8	Empty	
q_9t_9	$MBR \leftarrow M[MAR], PC = PC + 1$	读操作数, PC+1
q_9t_{10}	Empty	
q_9t_{11}	Empty	
q_9t_{12}	$MAR \leftarrow MBR$	传递操作数地址给 MAR
q_9t_{13}	Empty	
q_9t_{14}	$MBR \leftarrow M[MAR]$	读取操作数
q_9t_{15}	Empty	
q_9t_{16}	Empty	
q_9t_{17}	$A \leftarrow MBR, T \leftarrow 0$	传递操作数, 转入 T0 取址
q_9t_{18}	Empty	
—	—	—
STA ADDR	将 A 内容存入地址 addr	使用 q10 进行控制
$q_{10}t_7$	$MBR \leftarrow A$	将 A 寄存器内容存入 MBR
$q_{10}t_8$	Empty	
$q_{10}t_9$	$MAR \leftarrow PC$	将 PC 地址传给 MAR
$q_{10}t_{10}$	Empty	
$q_{10}t_{11}$	$M[MAR] \leftarrow MBR$	将 MBR 中内容写入 MAR 地址处
$q_{10}t_{12}$	Empty	
$q_{10}t_{13}$	Empty	

表 7: 微操作表 4

Xi	微操作序列
$x_1 = t_0 + q_3t_7 + q_4t_7 + q_9t_7 + q_{10}t_9$	$MAR \Leftarrow PC$
$x_2 = q_9t_{12}$	$MAR \Leftarrow MBR$
$x_3 = t_2 + q_3t_9 + q_4t_9 + q_9t_9$	$PC \Leftarrow PC + 1$
$x_4 = t_2 + q_3t_9 + q_4t_9 + q_9t_9 + q_9t_{14}$	$MBR \Leftarrow M[MAR]$
$x_5 = q_3t_{12} + q_9t_{17}$	$A \Leftarrow MBR$
$x_6 = q_1t_7$	$A \Leftarrow R$
$x_7 = q_1t_7 + q_2t_7 + q_3t_{12} + q_4t_{12} + q_5t_7 + q_6t_7 + q_7t_7 + q_8t_{11} + q_9t_{17}$	$T \Leftarrow 0$
$x_8 = t_5$	$IR \Leftarrow MBR$
$x_9 = q_{10}t_{11}$	$M[MAR] \Leftarrow MBR$
$x_{10} = t_2 + q_3t_9 + q_4t_9$	$WREN$
$x_{11} = q_2t_7$	$B \Leftarrow R$
$x_{12} = q_4t_{12}$	$B \Leftarrow MBR$
$x_{13} = q_5t_7$	$A \Leftarrow AandB$
$x_{14} = q_6t_7$	$A \Leftarrow AorB$
$x_{15} = q_8t_7$	$B \Leftarrow B$
$x_{16} = q_8t_9$	$B \Leftarrow B + 1$
$x_{17} = q_8t_{11}$	$A \Leftarrow A + B$
$x_{18} = q_{10}t_7$	$MBR \Leftarrow A$

表 8: 集成化微操作序列