



计算机组成原理大实验

——组合逻辑的控制单元的实现

班级：1403106

导师：张展

学号：1140310607

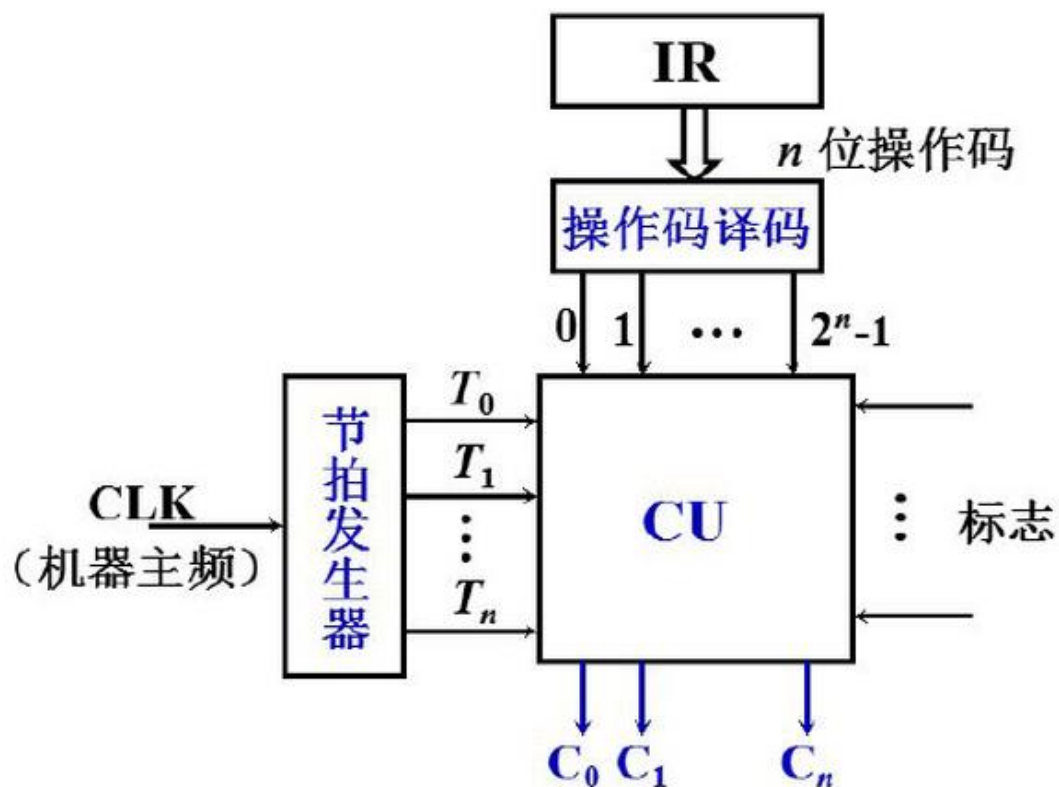
姓名：金卓林

一、 摘要

按照题目要求用组合逻辑控制法设计一个如图所示的简单模型机的控制单元 CU（微操作信号产生电路），决定外部的端口（名称、有效电平）和内部各元件的连接，设计仿真数据，用 VHDL 编程和仿真。

从控制单元的外特性图中可以看出，指令的操作码是决定控制单元发出不同控制信号的关键。为了简化控制单元的逻辑，将存放在 IR 的 n 位操作码经过一个译码电路产生 2^n 个输出，这样，每对应一种操作码便有一个输出送至 CU。当然，若指令的操作码长度可变，指令译码线路将更复杂。

控制单元的时钟输入实际上是一个脉冲序列，其频率即为机器的主频，它使 CU 能按一定的节拍 T 发出各种控制信号。节拍的宽度应满足数据信息通过数据总线从源到目的所需的时间。以时钟为计数脉冲，通过一个计数器，又称节拍发生器，便可产生一个与时钟周期等宽的节拍序列。如果将指令译码和节拍发生器从 CU 中分离出来，便可得简化的控制单元框图，如下图所示



（图一：结构图）

二、 实验目的与意义

通过学习控制单元的结构，掌握控制单元的实验原理。学会使用 VHDL 进行编程和仿真和操作。深入了解控制单元的设计与工作原理。

三、 实验设计与分析

主要元件设计：

1. 指令译码器

功能要求：4-10 译码器。

代码如下：

```
architecture Behavioral of decode is
begin
  process(S)
  begin
    if S='1' then
      op(0)<=not A(3) and (not A(2)) and (not A(1)) and (not A(0));
      op(1)<=not A(3) and (not A(2)) and (not A(1)) and A(0);
      op(2)<=not A(3) and (not A(2)) and A(1) and (not A(0));
      op(3)<=not A(3) and (not A(2)) and A(1) and A(0);
      op(4)<=not A(3) and A(2) and (not A(1)) and (not A(0));
      op(5)<=not A(3) and A(2) and (not A(1)) and A(0);
      op(6)<=not A(3) and A(2) and A(1) and (not A(0));
      op(7)<=not A(3) and A(2) and A(1) and A(0);
      op(8)<=A(3) and not A(2) and (not A(1)) and (not A(0));
      op(9)<=A(3) and (not A(2)) and (not A(1)) and A(0);

      else op<="0000000000";
    end if;
  end process;
end Behavioral;
```

(图二：译码器关键代码)

2. 控制单元

功能要求：指令系统有 10 条不同类型的指令。包括：清除累加器指令 CLA，累加器取反指令 COM，算术右移一位指令 SHR，循环左移一位指令 CSL，停机指令 STP，加法指令 ADDX，存数指令 STAX，取数指令 LDAX，无条件转移指令 JMPX，有条件转移（负则转）指令 BANX 等。根据每条指令的功能和时序，分析其执行过程中需要在各个阶段产生的全部微操作，导出产生这些微操作控制信号的逻辑表达式（用积之和式表示）。并且能够正确产生 10 条不同指令在执行中（每个机器周期、每个节拍）发出的全部微操作。

设计思路：

组合逻辑设计控制单元时，首先根据 10 条指令微操作的节拍安排，列出微操作命令的操作时间表，然后写出每一个微操作命令的逻辑表达式，最后根据逻辑表达式画出相应的组合逻辑电路图。

设计过程：

将每条指令的实现分成取指令、分析指令、执行指令三个步骤，每一步由一个机器周期实现，一条指令的实现需要三个机器周期，即 FE（取指周期）、INT（分析周期）、EX（执行周期）。

假设机器采用同步控制，每个机器周期包含 3 个节拍，而且 CPU 内部结构为非总线结构，其中 MAR 和 MDR 分别直接和地址总线 and 数据总线相连，并假设 IR 的地址码部分与 MAR 之间有通路。

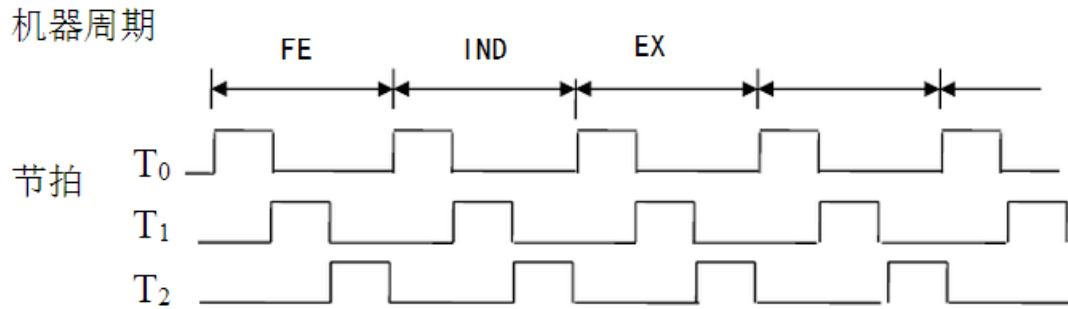
安排微操作节拍时应注意三点：

第一，有些微操作的次序是不容改变的，故安排微操作节拍时必须注意微操作的先后顺序。

第二，第二，凡是被控制对象不同的微操作，若能在一个节拍内执行，应尽可能安排在同一个节拍内，以节省时间。

第三，如果有些微操作所占的时间不长，应该将它们安排在一个节拍内完成，并且允许这些微操作有先后次序

每个机器周期由三个节拍组成，即 T₀、T₁、T₂，如下图：



(图三：周期)

非访存指令：

- 1、清除累加器指令 CLA；
- 2、累加器取反指令 COM；
- 3、算术右移一位指令 SHR；
- 4、循环左移一位指令 CSL；
- 5、停机指令 STP；

访存指令：

- 6、加法指令 ADDX；
- 7、存数指令 STAX；
- 8、取数指令 LDAX；
- 9、无条件转移指令 JMPX；
- 10、有条件转移（负则转）指令 BANX。

操作表如下图：

工作周期标记	节拍	状态条件	微操作命令信号	CLA	COM	SHR	CSL	STP	ADD	STA	LDA	JMP	BAN
FE (取指周期) MO	T0		PC→MAR	1	1	1	1	1	1	1	1	1	1
			1→R	1	1	1	1	1	1	1	1	1	1
	T1		M(MAR)→MDR	1	1	1	1	1	1	1	1	1	1
			(PC+1)→PC	1	1	1	1	1	1	1	1	1	1
	T2		MDR→IR	1	1	1	1	1	1	1	1	1	1
			OP(IR)→ID	1	1	1	1	1	1	1	1	1	1
		I	I→IND						1	1	1	1	1
		-I	I→EX	1	1	1	1	1	1	1	1	1	1
IND (间址周期) M1	T1		Ad(IR)→MAR						1	1	1	1	1
			1→R						1	1	1	1	1
	T2		M(MAR)→MDR						1	1	1	1	1
	T3	-IND	MDR→Ad(IR)						1	1	1	1	1
			1→EX						1	1	1	1	1
EX (执行周期) M2	T1		Ad(IR)→MAR						1	1	1		
			1→R						1		1		
			1→W							1			
	T2		M(MAR)→MDR						1		1		
			AC→MDR							1			
	T3		(AC)+(MDR) →AC						1				
			MDR→M(MAR)							1			
			MDR→AC								1		
			0→AC	1									
			-AC→AC		1								
			L(AC)→R(AC) , ACo 不变			1							
			P-1(AC)				1						
			Ad(IR)→PC									1	
	Ao		Ad(IR)→PC										1
			0→G					1					

(图四：操作表)

每个微操作对应的逻辑表达式：

0. PC→MAR=T0*FE

1. 1→R=T0*(FE+IND(S5+S6+S7+S8+S9)+EX(S5+S7))

2. M(MAR)→MDR=T1*(FE+IND(S5+S6+S7+S8+S9)+EX(S5+S7))

3. (PC)+1→PC=FE*T1

4. MDR→IR=T2*(FE+IND(S5+S6+S7+S8+S9))

5. OP(IR)→ID=FE*T2

6. I→IND=I*FE*T2*(S5+S6+S7+S8+S9)

7. I→EX=T2*(-I*FE+IND(S5+S6+S7+S8+S9)*(-IND))

8. Ad(IR)→MAR=T0*(IND(S5+S6+S7+S8+S9)+EX(S5+S6+S7))

9. 1→W=EX*T0*S6

10. $AC \rightarrow MDR = EX * T1 * S5$
11. $(AC) + MDR \rightarrow AC = EX * T2 * S5$
12. $MDR \rightarrow M(MAR) = EX * T2 * S6$
13. $MDR \rightarrow AC = EX * T2 * S7$
14. $0 \rightarrow AC = EX * T2 * S0$
15. $\neg AC \rightarrow AC = EX * T2 * S1$
16. $L(AC) \rightarrow R(AC), AC_0$ 不变 $= EX * T2 * S2$
17. $P-1(AC) = EX * T2 * S3$
18. $Ad(IR) \rightarrow PC = EX * T2 * (S8 + A0 * S9)$
19. $0 \rightarrow G = EX * T2 * S4$

注：S0-9 表示表格内十条指令（从左到右）； $\neg X$ 表示 X 取反。

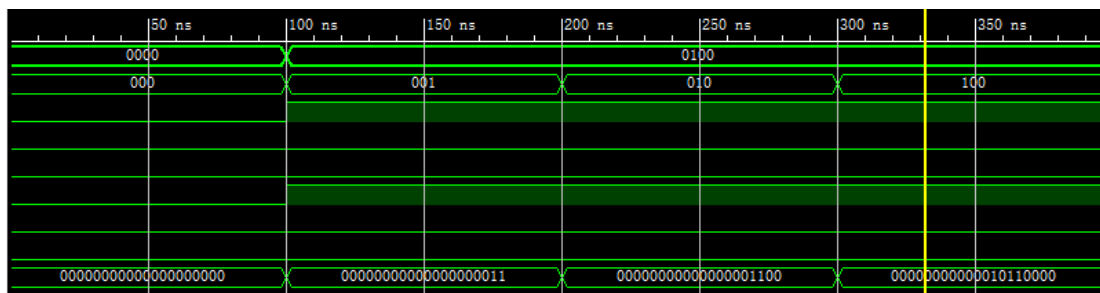
代码如下：

```
control(0) <= FE and t(0);
control(1) <= t(0) and (FE or (IND and (s(5) or s(6) or s(7) or s(8) or s(9))) or (EX and (s(5) or s(7))));
control(2) <= t(1) and (FE or (IND and (s(5) or s(6) or s(7) or s(8) or s(9))) or (EX and (s(5) or s(7))));
control(3) <= FE and t(1);
control(4) <= t(2) and (FE or (IND and (s(5) or s(6) or s(7) or s(8) or s(9))));
control(5) <= FE and t(2);
control(6) <= I and FE and t(2) and (s(5) or s(6) or s(7) or s(8) or s(9));
control(7) <= (not I and FE and t(2)) or (IND and t(1) and (s(5) or s(6) or s(7) or s(8) or s(9)) and not IND);
control(8) <= t(0) and ((IND and (s(5) or s(6) or s(7) or s(8) or s(9))) or (EX and (s(5) or s(6) or s(7))));
control(9) <= EX and t(0) and s(6);
control(10) <= EX and t(1) and s(6);
control(11) <= EX and t(2) and s(5);
control(12) <= EX and t(2) and s(6);
control(13) <= EX and t(2) and s(7);
control(14) <= EX and t(2) and s(0);
control(15) <= EX and t(2) and s(1);
control(16) <= EX and t(2) and s(2);
control(17) <= EX and t(2) and s(3);
control(18) <= (EX and t(2) and s(8)) or (A and EX and t(2) and s(9));
control(19) <= EX and t(2) and s(4);
```

（图五：控制器关键代码）

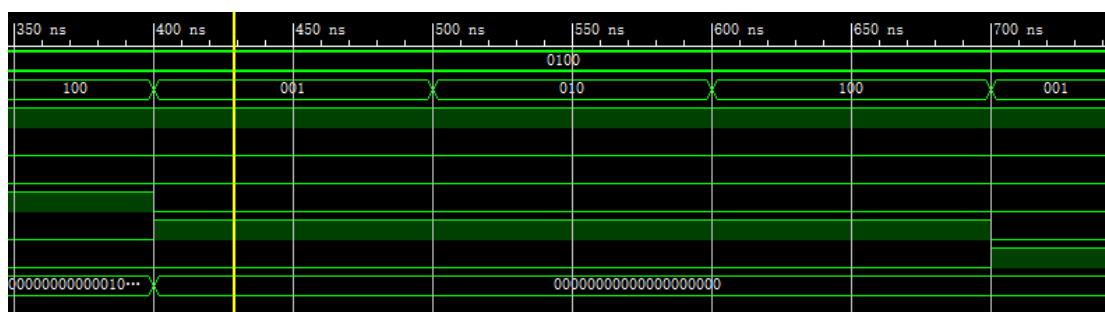
四、实验仿真与分析

我以 STP 以及 ADD 作为本次实验仿真的结果进行分析。



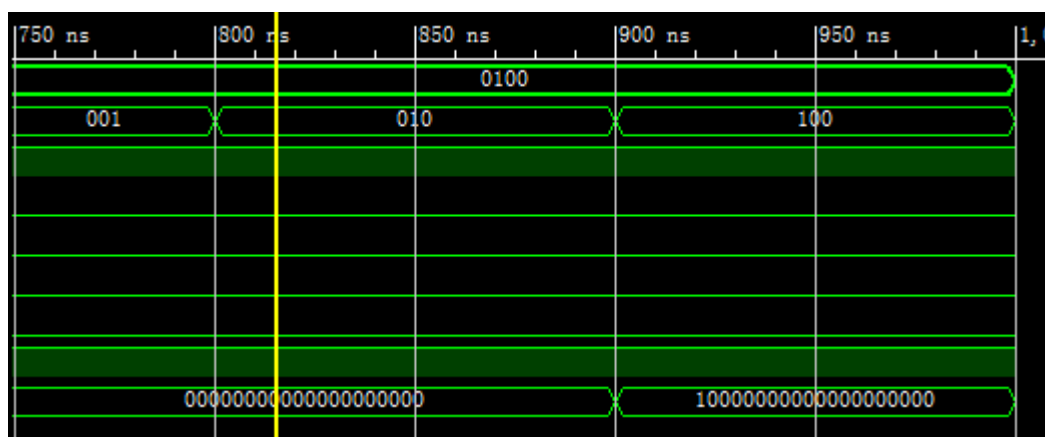
（图六-1 STP）

前 50ns 全部置位为 0。首先是取指节点，三个时钟分别问 T0, T1, T2 时刻时输出不同的微操作，以下仿真结果类似。



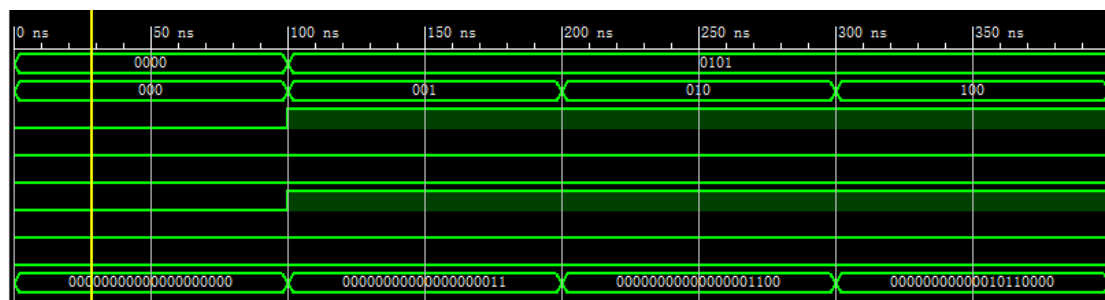
(图六-2 STP)

间址周期。

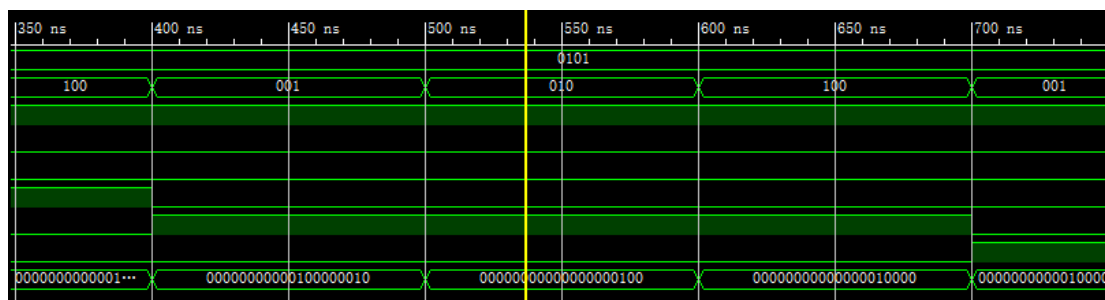


(图六-3 STP)

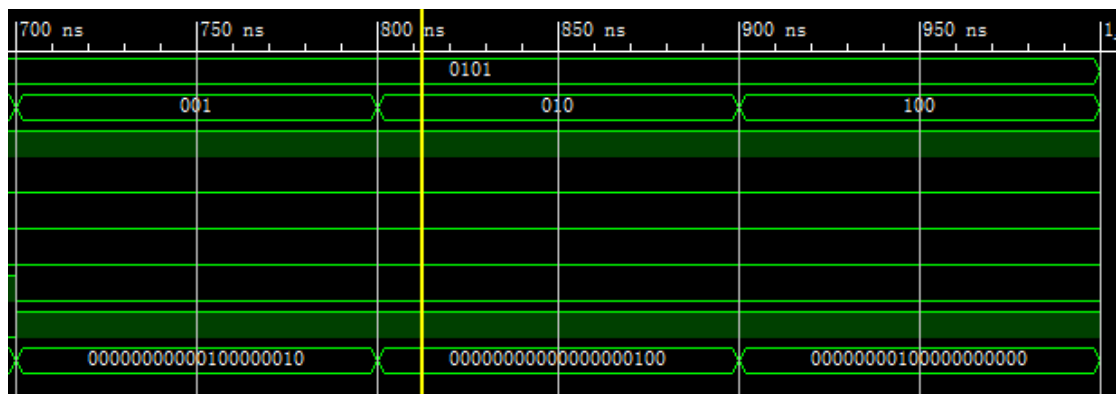
执行周期。



(图七-1 ADD)



(图七-2 ADD)



(图七-3 ADD)

由上面六张图结合操作表可以看到两条指令对应的微操作是正确的。

五、 实验拓展与调研

采用组合逻辑设计方法设计控制单元，思路清晰，简单明了，但因为每一个微操作命令都对应一个逻辑电路，因此一旦设计完毕便会发现，这种控制单元的线路结构十分庞大，也不规范，犹如一颗大树，到处都是不规整的枝杈。而且，指令系统功能越全，微操作命令就越多，线路就越复杂，调试也就越困难。为了克服这些缺点，可采用微程序设计方案。