北京邮电大学课程设计报告

<table>
<tr><td>课程设计名称</td><td colspan="2">硬布线控制器的常规 CPU 设计</td><td>学 院</td><td>计算机学院</td><td>指导教师</td><td></td></tr>
<tr><td>班 级</td><td>班内序号</td><td colspan="2">学 号</td><td>学生姓名</td><td colspan="2">成绩</td></tr>
<tr><td>2013211301</td><td>21</td><td colspan="2">180</td><td>李积朝</td><td colspan="2"></td></tr>
<tr><td>2013211301</td><td>19</td><td colspan="2">1178</td><td>彭涛</td><td colspan="2"></td></tr>
<tr><td>2013211301</td><td>20</td><td colspan="2">1179</td><td>范泽天</td><td colspan="2"></td></tr>
<tr><td>2013211301</td><td>22</td><td colspan="2">1181</td><td>杨胜楠</td><td colspan="2"></td></tr>
<tr><td>课程设计内容</td><td colspan="6">目的：融会贯通计算机组成原理课程各章教学内容，通过知识的综合运用，加深对 CPU 各模块工作原理及相互联系的认识：掌握硬连线控制器的设计方法：学习运用当代的 EDA 设计工具，掌握运用 EDA 设计大规模复杂逻辑电路的方法：培养科学研究能力，去的设计和调试的实践经验
方法：通读实验要求，复习相关内容，编写 VHDL 代码然后下载到试验台并进行调试。
小组分工：李积朝（40%）：VHDL 代码；彭涛（20%）：VHDL；
范泽天（20%）：日志记载和报告；杨胜楠（20%）：调试。</td></tr>
<tr><td>学生课程设计报告
(附页)</td><td colspan="6">附页</td></tr>
<tr><td>课程设计成绩评定</td><td colspan="6">成绩:

指导教师签名:
年 月 日</td></tr>
</table>

注：评语要体现每个学生的工作情况，可以加页。

计算机组成原理课程综合设计

--硬连线控制器的常规 CPU 设计

1. 教学目的：
   （1） 融会贯通计算机组成原理或计算机组成与系统结构课程各章教学内容，通过知识的综合运用，加深对 CPU 各模块工作原理及相互联系的认识。
   （2） 掌握硬连线控制器的设计方法。
   （3） 学习运用当代的 EDA 设计工具，掌握运用 EDA 设计大规模复杂逻辑电路的方法。
   （4） 培养科学研究能力，去的设计和调试的实践经验。
2. 实验设备：

TEC-8 实验系统	一台
PC 计算机	一台
双踪示波器	一台
直流万用表	一个
逻辑笔	一支

3. 设计与调试任务：
   （1） 设计一个硬连线控制器，和 TEC-8 模型计算机的数据通路结合在一起，构成一个完整的 CPU，该 CPU 要求：
      ① 能够完成控制台操作：启动程序运行，度存储器，写存储器读寄存器和写寄存器。
      ② 能够执行新涉及到 CPU 指令系统中的指令，完成规定的指令功能。
   （2） 在 Quarts 2 下对硬布线控制器设计方案进行编程和编译。
   （3） 将编译后的硬布线控制器下载到 TEC-8 试验台上的 ISP 器件 EPM7128 中去，使 EPM7128 成为一个硬布线控制器。
   （4） 根据指令系统，编写硬布线控制器正确性的测试程序，并用测试程序对硬布线控制器在单拍方式下进行调试，直到成功。
   （5） 在调试成功的条件下，整理出设计文件，包括：
      ① 硬连线控制器的逻辑模块图；
      ② 硬连线控制器指令周期流程图；
      ③ 硬连线控制器的 VHDL 源程序；
      ④ 测试程序；
      ⑤ 设计说明书；
      ⑥ 调试总结；
4. 设计准备：
   （1） 硬连线控制器的基本原理

   微操作信号 S 是一系列输入量的逻辑函数，即用组合逻辑来实现

   S=f(Im, Mi Tk, Bj)

   其中 Im 是机器指令操作码译码器的输出信号，Mi 是节拍电位信号，Tk 是节拍脉冲信号，Bj 使状态条件信号。

   在 TEC-8 实验系统中，节拍脉冲信号 Tk 已经直接输送给数据通路，4 位指令操作码 IR4~IR7 直接成为 Im 的一部分；SWC,SWB,SWA 可以看做 Im 的另一部分。是时序发生器的节拍信号 W1~W3;Bj 包括 ALU 产生的仅为信号 C，结果为 0 信号 Z 等。

（2） 组合逻辑译码表

			组合逻辑译码表							
指令	WRF1	WRF2	WRF3	WRF4	RRF12	RRF34	RM_A	RM_D	WM_A	WM_D
SBUS	W1	W2	W1	W2			W1		W1	
MBUS								W1		W1
SEL3			W1	W2		W1				
SEL2		W2		W2						
SEL1	W1			W2		W1				
SEL0	W1		W1		W1	W1				
SELCTL	W1	W2	W1	W2	W1	W1	W1	W1	W1	W1
DRW	W1	W2	W1	W2						
STOP	W1	W2	W1	W2	W1	W1	W1	W1	W1	W1
SST0		W2					W1		W1	
LAR							W1		W1	
SHORT							W1	W1	W1	W1
MEMW										W1
ARINC								W1		W1
ST0			W1	W2				W1		
	ADD	SUB	AND	INC	LD	ST	JC(C=1)	JZ(Z=1)	JMP	STP
LIR	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1
PCINC	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1
M			W2		W2	W2,W3			W2	
S3	W2		W2		W2	W2,W3			W2	
S2		W2				W2			W2	
S1		W2	W2		W2	W2,W3			W2	
S0	W2		W2			W2			W2	
CIN	W2									
ABUS	W2	W2	W2	W2	W2	W2,W3			W2	
MBUS					W3					
DRW	W2	W2	W2	W2	W3					
LDZ	W2	W2	W2	W2						
LDC	W2	W2		W2						
LAR					W2	W2				
LONG					W2	W2				
MEMW						W3				
PCADD							W2	W2		
LPC									W2	
STOP										W2
C							W2			
Z								W2		

(3)EPM7128 器件的引脚

信号	方向	引脚号	信号	方向	引脚号
CLR	INPUT	1	C	INPUT	2
T3	INPUT	83	Z	INPUT	84
SWA	INPUT	4	DRW	OUTPUT	20
SWB	INPUT	5	PCINC	OUTPUT	21
SWC	INPUT	6	LPC	OUTPUT	22
IR4	INPUT	8	LAR	OUTPUT	25
IR5	INPUT	9	PCADD	OUTPUT	18
IR6	INPUT	10	ARINC	OUTPUT	24
IR7	INPUT	11	SELCTL	OUTPUT	52
W1	INPUT	12	MEMW	OUTPUT	27
W2	INPUT	15	STOP	OUTPUT	28
W3	INPUT	16	LIR	OUTPUT	29
LDZ	OUTPUT	30	SBUS	OUTPUT	41
LDC	OUTPUT	31	MBUS	OUTPUT	44
CIN	OUTPUT	33	SHORT	OUTPUT	45
S0	OUTPUT	34	LONG	OUTPUT	46
S1	OUTPUT	35	SEL0	OUTPUT	48
S2	OUTPUT	36	SEL1	OUTPUT	49
S3	OUTPUT	37	SEL2	OUTPUT	50
M	OUTPUT	39	SEL3	OUTPUT	51
ABUS	OUTPUT	40			

（4）硬布线控制器逻辑模块

（5）硬布线控制器指令周期图

5.实验文档

①VHDL源程序

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

entity yinbuxian is
    PORT(
        CLR,
        W1,W2,W3,
        SWC,SWB,SWA,
        C,
        Z,
        T3,                                    --CPU time signal
        IR7,IR6,IR5,IR4:IN STD_LOGIC;

        ABUS,SBUS,MBUS,                        --data and data process
        SELCTL,
        SHORT,LONG,
        DRW,
        STOP,
        LAR,
        MEMW,
        ARINC,
        LIR,
        PCINC,
        M,
        S3,S2,S1,S0,
        CIN,
        SEL3,SEL2,SEL1,SEL0,
        LDC,LDZ,

        PCADD,
        LPC: OUT STD_LOGIC);
END yinbuxian;

ARCHITECTURE BEHAVE OF yinbuxian IS

SIGNAL STO:   STD_LOGIC;
SIGNAL SSTO:  STD_LOGIC;
BEGIN
    process(STO,SSTO,CLR,W1,W2,W3,SWC,SWB,SWA,T3)
    begin
                                                --CLR
```

```
LIR<='0';
PCINC<='0';
ABUS<='0';
SBUS<='0';
MBUS<='0';
SELCTL<='0';
DRW<='0';
STOP<='0';
LAR<='0';
SHORT<='0';
LONG<='0';
MEMW<='0';
ARINC<='0';
M<='0';
S3<='0';
S2<='0';
S1<='0';
S0<='0';
SEL3<='0';
SEL2<='0';
SEL1<='0';
SEL0<='0';
CIN<='0';
LDC<='0';
LDZ<='0';
PCADD<='0';
LPC<='0';

IF (CLR = '0') THEN                --response to ClR signal
    STO <= '0';
    SSTO <= '0';
ELSIF (SSTO = '1' AND (T3'EVENT AND T3 = '0')) THEN
    STO <= '1';
END IF;

IF(STO='0' AND SWC='1'AND SWB='0' AND SWA='0')THEN
    IF (W1='1' AND W2='0')THEN
--WRF12
    SBUS<=W1;
    SELCTL<=W1;
    DRW<=W1;
    STOP<=W1;

    SEL1<= W1;
```

```
    SEL0<= W1;
    ELSIF(W1='0' AND W2='1')THEN
    SBUS<=W2;
    SELCTL<=W2;
    DRW<=W2;
    STOP<=W2;

    SEL2<= W2;
    SSTO<=W2;
    END IF;
ELSIF(STO='1'AND SWC='1'AND SWB='0' AND SWA='0')THEN
    IF (W1='1' AND W2='0')THEN
--WRF12
    SBUS<=W1;
    SELCTL<=W1;
    DRW<=W1;
    STOP<=W1;

    SEL3<= W1;
    SEL0<= W1;
    ELSIF(W1='0' AND W2='1')THEN

    SBUS<=W2;
    SELCTL<=W2;
    DRW<=W2;
    STOP<=W2;

    SEL3<= W2;
    SEL2<= W2;
    SEL1<= W2;
    END IF;
ELSIF(SWC='0'AND SWB='1' AND SWA='1')THEN            --RRF
    SEL0<=W1 OR W2;
    SELCTL<=W1 OR W2;
    STOP<=W1 OR W2;
    SEL3<=W2;
    SEL1<=W2;

ELSIF(STO='0' AND SWC='0'AND SWB='1' AND SWA='0')THEN
    SBUS<=W1;
    LAR<=W1;
    STOP<=W1;
    SSTO<=W1;
    SHORT<=W1;
```

```
        SELCTL<=W1;
    ELSIF(STO='1'AND SWC='0'AND SWB='1' AND SWA='0')THEN
        MBUS<=W1;
        ARINC<=W1;
        STOP<=W1;
        SHORT<=W1;
        SELCTL<=W1;
    ELSIF(STO='0'AND SWC='0'AND SWB='0' AND SWA='1')THEN
        SBUS<=W1;
        LAR<=W1;
        STOP<=W1;
        SSTO<=W1;
        SHORT<=W1;
        SELCTL<=W1;
    ELSIF(STO='1'AND SWC='0'AND SWB='0' AND SWA='1')THEN
        SBUS<=W1;
        MEMW<=W1;
        ARINC<=W1;
        STOP<=W1;
        SHORT<=W1;
        SELCTL<=W1;
    END IF;
IF(SWC='0'AND SWB='0' AND SWA='0')THEN    --FETCH INSTRUCTION
    LIR<=W1;
    PCINC<=W1;                      --IR7~IR4 is selection signal
    IF(IR7='0' AND IR6='0' AND IR5='0' AND IR4='1')THEN --ADD
        S3<=W2;
        S0<=W2;
        CIN<=W2;
        ABUS<=W2;
        DRW<=W2;
        LDZ<=W2;
        LDC<=W2;
    ELSIF(IR7='0' AND IR6='0' AND IR5='1' AND IR4='0')THEN    --SUB

        S2<=W2;
        S1<=W2;
        ABUS<=W2;
        MBUS<=W2;
        DRW<=W2;
        LDZ<=W2;
        LDC<=W2;
    ELSIF(IR7='0' AND IR6='0' AND IR5='1' AND IR4='1')THEN  --AND
        M<=W2;
```

```
            S3<=W2;
            S1<=W2;
            S0<=W2;
            ABUS<=W2;
            DRW<=W2;
            LDZ<=W2;
        ELSIF(IR7='0' AND IR6='1' AND IR5='0' AND IR4='0')THEN  --INC
            ABUS<=W2;
            DRW<=W2;
            LDZ<=W2;
            LDC<=W2;
        ELSIF(IR7='1'   AND   IR6='0'   AND   IR5='1'   AND   IR4='0')THEN
--NOT*****NEW
            M<=W2;
            ABUS<=W2;
            DRW<=W2;
            LDZ<=W2;
            LDC<=W2;
        ELSIF(IR7='1'   AND   IR6='0'   AND   IR5='1'   AND   IR4='1')THEN
--NAND*****NEW
            M<=W2;
            S2<=W2;
            ABUS<=W2;
            DRW<=W2;
            LDZ<=W2;
            LDC<=W2;
        ELSIF(IR7='1'   AND   IR6='1'   AND   IR5='0'   AND   IR4='0')THEN
--XOR****NEW
            M<=W2;
            S2<=W2;
            S1<=W2;
            ABUS<=W2;
            DRW<=W2;
            LDZ<=W2;
            LDC<=W2;
        ELSIF(IR7='1'   AND   IR6='1'   AND   IR5='0'   AND   IR4='1')THEN
--DEC****NEW
            CIN<=W2;
            S3<=W2;
            S2<=W2;
            S1<=W2;
            S0<=W2;
            ABUS<=W2;
            DRW<=W2;
```

```
    LDZ<=W2;
    LDC<=W2;
ELSIF(IR7='0' AND IR6='1' AND IR5='0' AND IR4='1')THEN  --LD
    IF(W2='1' AND W3='0')THEN
        M<=W2;
        S3<=W2;
        S1<=W2;
        ABUS<=W2;
        LAR<=W2;
        LONG<=W2;
    ELSIF(W2='0' AND W3='1')THEN
        DRW<=W3;
        MBUS<=W3;
    END IF;
ELSIF(IR7='0' AND IR6='1' AND IR5='1' AND IR4='0')THEN  --ST
    IF(W2='1' AND W3='0')THEN
        M<=W2;
        S3<=W2;
        S2<=W2;
        S1<=W2;
        S0<=W2;
        ABUS<=W2;
        LAR<=W2;
        LONG<=W2;
    ELSIF(W2='0' AND W3='1')THEN
        M<=W3;
        S3<=W3;
        S1<=W3;
        ABUS<=W3;
        MEMW<=W3;
    END IF;
ELSIF(IR7='0' AND IR6='1' AND IR5='1' AND IR4='1')THEN  --JC
    IF(C='1')THEN
        PCADD<=W2;
    END IF;
ELSIF(IR7='1' AND IR6='0' AND IR5='0' AND IR4='0')THEN  --JZ
    IF(Z='1')THEN
        PCADD<=W2;
    END IF;
ELSIF(IR7='1' AND IR6='0' AND IR5='0' AND IR4='1')THEN  --JMP
    M<=W2;
    S3<=W2;
    S2<=W2;
    S1<=W2;
```

```
            SO<=W2;
            ABUS<=W2;
            LPC<=W2;
        ELSIF(IR7='1' AND IR6='1' AND IR5='1' AND IR4='0')THEN  --STP
            STOP<=W2;
        END IF;
    END IF;

END PROCESS;
END BEHAVE;
```

②调试日志：

1：VHDL 书写时发现的错误及不规范：

（1）：对输入量的赋值是无效的。

（2）在同一条件下对输出量的重复赋值是逻辑错误的。

引发的深思：每一种编程语言都有其书写规则，应当注意区分；逻辑上的错误应该避免，因为它在试验台之外会造成巨大的浪费，消耗。

2：CPU 方面的理解错误：

写寄存器时,R1 的数据被 R3 的数据冲掉了,结果发现时因为 SSTO 为 1 马上导致 STO 为 1，又因为在 W2 时序，所以进行 R3 的步骤。

引发的深思：CPU 是个时序严明的结构，没有时序，就不会实现各种功能。

3：第一次调试：寄存器赋值：00H ,00H,60H，FDH;

一遍运行之后：寄存器值为：E4H,49H,C0H,B6H;(答案不正确)

检查之后是因为编程序时有一个值赋错了。

第二次调试，经过简单的指令逐条验证后发现所有指令功能能正常实现，再验证测试程序，答案正确。

②测试程序

00H	LD R0，[R2]	52H(0101,0010)	0F	SUB R0，R2	22H(0010,0010)
01	INC R2	48H(0100,1000)	10	LD R2，[R0]	58H(0101,1000)
02	LD R1，[R2]	56H(0101,0110)	11	ADD R3，R2	1EH(0001,1110)
03	ADD R0,R1	11H(0001,0001)	12	LD R3，[R3]	5FH(0101,1111)
04	JC 06H	71H(0111,0001)	13	NOT R0	0A0H(1010,0000)
05	AND R1，R0	34H(0011,0100)	14	STP	0E0H(1110,0000)
06	SUB R0，R2	22H(0010,0010)	R2(register)		60H(0110,0000)
07	INC R1	44H(0100,0100)	R3(register)		0FDH(1111,1101)
08	ST R0，[R1]	64H(0110,0100)	60H		67H(0110,0111)
09	INC R3	4CH(0100,1100)	61H		80H(1000,0000)
0A	JZ 00H	82H(1000,0010)	62H		0FDH(1111,1101)
0B	LD R2，[R3]	5BH(0101,1011)	80H		60H(0110,0000)
0C	JMP [R2]	98H(1001,1000)	0FEH		03H(0000,0011)
0D	INC R3	4CH(0100	0FFH		03H(0000,

		,1100)			0011)
0E	INC R3	4CH(0100 ,1100)			

运行之后的结果为：

R0,R1,R2,R3 分别为：7FH,83H,60H,0FDH；

实验结果正确。

③设计说明书

这次课程设计主要由 VHDL 源程序和 TEC—8 实验平台的操作两部分组成。

源程序按照设计的硬布线控制器的逻辑模块和指令周期书写，逻辑模块阐述 了 控制器的逻辑输入与输出；指令周期则说明了输入与输出的对应关系及在 这种对应下所形成的操作指令。

④操作说明书

在设计硬布线控制器的时候，除了基本的十条指令，我们小组还扩展了额外的四条指令，使得该控制器的功能更加强大；控制台的操作主要分两部分：对数据的操作和多指令的操作。

对数据的操作：

1.对寄存器存数：SWC,SWB,SWA 控制信号为 100；

2.对寄存器读数：SWC,SWB,SWA 控制信号为 011；

3.对存储器存数：SWC,SWB,SWA 控制信号为 010；

4.对存储器读数：SWC,SWB,SWA 控制信号为 001；

对指令的操作（SWC,SWB,SWA 控制信号为 000）;：

1.ADD：指令中 IR7-IR4 为 0001；

2.SUB：指令中 IR7-IR4 为 0010；

3.AND：指令中 IR7-IR4 为 0011；

4.INC：指令中 IR7-IR4 为 0100；

5.LD：指令中 IR7-IR4 为 0101；

6.ST：指令中 IR7-IR4 为 0110；

7.JC：指令中 IR7-IR4 为 0111；

8.JZ：指令中 IR7-IR4 为 1000；

9.JMP：指令中 IR7-IR4 为 1001；

10.STP：指令中 IR7-IR4 为 1110；

11.NOT（新加）：指令中 IR7-IR4 为 1010；

12.XAND（新加）：指令中 IR7-IR4 为 1011；

13.XOR（新加）：指令中 IR7-IR4 为 1100；

14.DEC（新加）：指令中 IR7-IR4 为 1101；

CPU 指令系统：

名称	汇编语言	功能	指令格式		
逻辑非(new)	NOT Rd	RD<-NOT Rd	1010	Rd	XX
异与(new)	XAND Rd,Rs	RD<-Rd XAND Rs	1011	Rd	Rs
异或(new)	XOR Rd,Rs	Rd<-Rd XOR Rs	1100	Rd	Rs
减 1(new)	DEC Rd	Rd<-Rd-1	1101	Rd	XX
加法	ADD RD,RS	Rd<-Rd+Rs	0001	Rd	Rs
减法	SUB RD,RS	Rd<-Rd-Rs	0010	Rd	Rs
逻辑与	AND RD,RS	Rd<-Rd and Rs	0011	Rd	Rs
加 1	INC RC	Rd<-Rd+1	0100	Rd	XX
取数	LD RD,[RS]	Rd<-[Rs]	0101	Rd	Rs
存数	ST RS,[RD]	Rs<-[Rd]	0110	Rd	Rs
c 条件转移	JC ADDR	C=1 则 PC<-a+offset	0111	offset	
z 条件转移	JZ ADDR	Z=1 则 PC<-a+offset	1000	offset	
无条件转移	JMP [RD]	PC<-Rd	1001	Rd	XX
停机	STOP	暂停运行	1110	XX	XX

⑤实验总结

李积朝（2013211180）：

两周的课程实验设计，让我成长了很多，被大家选为组长，即意味着责任又意味着信任。通过对以往知识的复习和坚持不懈地努力，我们终于成功做完了这次课设。成功后的喜悦也是这次艰苦的奋斗的一次回报。软硬件的结合使这次作业也为我们之后的学习生活打开了新的大门，使我们对计算机组成原理有了更加深刻的认识。一个自己设计的简易硬布线计算机的成功运转也让我们对计算机更加感兴趣了。

对计算机的其他方面的好奇以及这种努力实践后的收获感将会是我之后学习其他科目的最大动力。这就是这次试验给我的收获。

彭涛（2013211178）：

经过将近半个月的计算机组成原理课程设计，让我们体会到了苦尽甘来的滋味，这次课程设计使我对上学期老师所教的计算机组成原理的知识得到了巩固和提高。这次的计算机组成原理课程设计使我的能力得到了很大的提高，还使我对上学期所学的计算机组成原理的知识得到了提高，加深了对计算机工作原理的认识。我也体会到了作为一个大学生，要想学有所得，就得学习主动，不要什么都希望别人亲自传授，面对问题要自己去努力解决，多问问身边的同学，多动手查查，多上网找找，所以要想成功就得事事做到细心，耐心，恒心。

在这次课程设计中，我主要承担任务是一位 VHDL 程序的设计与编写。在组长的总体设计下，我们的工作进行的非常顺利，使我们的课程设计达到了事半功倍的效

果，在设计过程中，通过整体设计方案，根据课程任务设计书的要求，把我们组的课程设计报告认真的完成。

这次的课程设计使我懂得了理论与实际相结合是很非常重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从理论中得出结论，从而提高自己的实际动手能力和独立思考的能力。在整个设计过程中，构思是很花费时间的。但是在组长的帮助下，我们在电路中遇到的关于编辑、编译、调试、中的问题都一一解决了。当然，有时用错了方法，总是实现不了。同时在设计的过程中发现了自己的不足之处，对以前所学过的知识理解得不够深刻，掌握得不够牢固。而做课程设计同时也是对课本知识的巩固和加强，平时看课本时，有些问题就不是很能理解，做完课程设计，那些问题就迎刃而解了。所以这个期末测试之后的课程设计对我们的作用是非常大的。

范泽天（2013211179）：

通过几天的课程设计实验，我学到了好多，也有好多感想。首先，说下实验中我们遇到的一些问题。开始其实我对这个不是很清楚，通过看书然后慢慢的整理思路，还有和同学一起讨论，有了大体的认识。开始做的时候，感觉好多以前的知识又都忘了，于是又重新学习。遇到每一个问题，都要去查书学习，总之每一步都和艰辛。这让我深刻体会到团队合作的重要性。也让我再次熟悉了 VHDL 语言。做完实验，感觉对 TEC-8 试验台的认识比之前有了很多的提高。之前说实话对实验台不是特别的熟悉，这下基本上是大都懂了。我也对硬连线控制器有了更清晰的认识，懂得了硬连线控制器和微程序控制器的一些区别。

杨胜楠（2013211181）：

这次我们做的是硬连线控制器的常规 CPU 设计，这是一个软件与硬件结合的实验，在实验中我们要通过电脑编写程序并下载使用，很好地锻炼了我们的自学、协作和动手能力以及耐心，并且这次试验让我们对计算机组成原理有了更深的了解，对 QuartusⅡ这个软件以及对在系统编程有了更深的了解，更加意识到实验要求的重要性，只有理解透实验的具体要求和所需达到的效果，才能少走弯路，提高效率。在这个实验中我们学到了很多，包括团队协作，动手能力，自学以及创新能力，过程虽然有些辛苦，但是小小的辛苦让我们获得了收获的喜悦。