

沈阳航空航天大学

课 程 设 计 报 告

课程设计名称： 计算机组成原理课程设计

课程设计题目： 基于加减交替法的定
点原码一位除法器的设计与实现

院（系）：计算机学院

专 业：网络工程

班 级：

学 号：

姓 名：

指导教师：孙恩岩

学术诚信声明

本人声明：所呈交的报告（含电子版及数据文件）是我个人在导师指导下独立进行设计工作及取得的研究结果。尽我所知，除了文中特别加以标注或致谢中所罗列的内容以外，报告中不包含其他人已经发表或撰写过的研究结果，也不包含其它教育机构使用过的材料。与我一同工作的同学对本研究所做的任何贡献均已在报告中做了明确的说明并表示了谢意。报告资料及实验数据若有不实之处，本人愿意接受本教学环节“不及格”和“重修或重做”的评分结论并承担相关一切后果。

本人签名：吉学金

日期； 2016 年 1 月 16 日

目 录

第 1 章	总体设计方案	1
1.1	设计原理	1
1.2	设计思路	2
1.3	设计环境	3
第 2 章	详细设计方案	5
2.1	顶层方案图的设计与实现	5
2.1.1	创建顶层图形设计文件	5
2.1.2	器件的选择与引脚锁定	6
2.1.3	编译、综合、适配	7
2.2	功能模块的设计与实现	7
2.2.1	控制器模块的设计与实现	8
2.2.2	输入寄存器的设计与实现	9
2.2.3	选择器模块的设计与实现	11
2.2.4	相反数补码模块的设计与实现	12
2.2.5	加法器模块的设计与实现	14
2.2.6	移位电路模块的设计与实现	15
2.2.7	恢复余数寄存器模块的设计与实现	17
2.2.8	商寄存器模块的设计与实现	19
2.3	仿真调试	20
第 3 章	编程下载与硬件测试	22
3.1	编程下载	22
3.2	硬件测试及结果分析	22
参考文献	26

第 1 章 总体设计方案

1.1 设计原理

本次课设要求使用加减交替法 /不恢复余数法，用三个寄存器 A，B，C 分别存放被除数 /余数，除数 Y 以及[-Y] 补，和商。 D 为除数 Y/[-Y] 补送加法器电路。 A 寄存器用被除数初始化，作为初始余数。实现余数和 D（即加减交替法的选择结果，通过加法器运算结果的最高两位即符号位确定， 如果符号为负，则选择 +Y，否则+[-Y] 补）相加是通过 A 送加法器和 D 送加法器，在加法器中完成的。加法器的输出经过移位电路向左移一位送入 A 寄存器中。 C 寄存器是用移位寄存器实现的，左移时将移入通过加法器输出结果高位符号确定的商。完成除法运算后， A 寄存器中保存余数， C 寄存器中保存商。主要步骤如下，当某一次求得的差值（余数 Ri）为负时，不恢复它，继续求下一位的商，但用加上除数（ +[-Y] 补）的办法来取代（ -Y）的操作，其他操作依然不变。即（ 1）当余数为正时，商上“ 1”，求下一位商的办法是余数左移一位，再减去除数；

（2）当余数为负数时，商上“ 0”，求下一位商的办法是余数左移一位，再加上除数。

（3）这种方法不用恢复余数，但若最后一次上商 为“ 0”，而又需要得到正确的余数，则在这最后一次仍需恢复余数。

例 1.1： X 的值为 0.1011，Y 的值为 0.1101，求 X/Y

被除数（余数）	商	操作说明
001011	00000	开始情形
+) 110011		+[-Y] 补
111110	00000	不够减，商上 0
111100	00000	左移
+) 001101	+Y	
001001	00001	够减，商上 1
010010	00010	左移

+) 110011			$+[-Y]_{补}$
000101	00011		够减，商上 1
001010	00110		左移
+) 110011			$+[-Y]_{补}$
111101	00110		不够减，商上 0
111010	01100		左移
+) 001101		$+Y$	
000111	01101		够减，商上 1
余数	商		
$X/Y=0.1101$	余数=0.0111		

1.2 设计思路

基于加减交替法定点原码一位除法器设计方法是；由一个 ALU 模块，一个被除数寄存模块，一个除数寄存模块，一个控制模块，一个加法模块，一个移位模块，一个除数选择模块，一个余数选择模块，一个移位模块、一个商寄存模块，一个恢复余数寄存模块构成，由控制器发出移位和减法命令，顶层设计采用原理图设计输入方式。定点原码一位除法器的实现框图如图 1.1 所示：

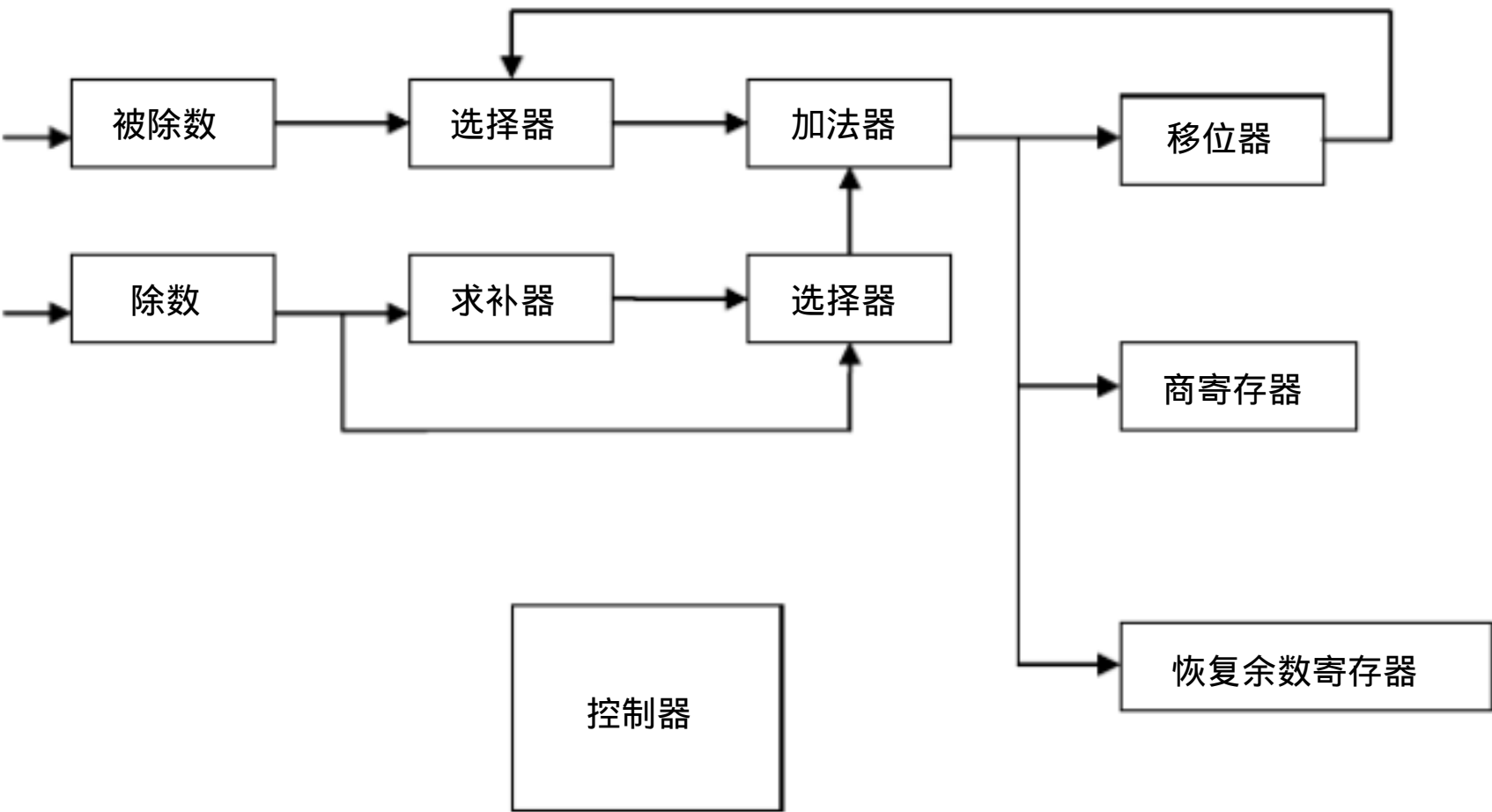


图 1.1 定点原码一位除法器原理框图

1.3 设计环境

（1）硬件环境：伟福 COP2000 型计算机组成原理实验仪

COP2000 计算机组成原理实验系统由实验平台、开关电源、软件三大部分组成。实验平台上有寄存器组 R0-R3、运算单元、累加器 A、暂存器 B、直通/左移/右移单元、地址寄存器、程序计数器、堆栈、中断源、输入/输出单元、存储器单元、微地址寄存器、指令寄存器、微程序控制器、组合逻辑控制器、扩展座、总线插孔区、微动开关/指示灯、逻辑笔、脉冲源、20 个按键、字符式 LCD、RS232 口。

（2）EDA 环境：Xilinx foundation f3.1 设计软件

该系统由设计入口工具、设计实现工具、设计验证工具三大部分组成。设计入口工具包括原理图编辑器、有限状态机编辑器、硬件描述语言（HDL）编辑器、LogiBLOX 模块生成器、Xilinx 内核生成器等软件。其功能是：接收各种图形或文字的设计输入，并最终生成网络表文件。设计实现工具包括流程引擎、限制编辑器、基片规划器、FPGA 编辑器、FPGA 写入器等软件。设计实现工具用于将网络表转化为配置比特流，并下载到器件。设计验证工具包括功能和时序仿真器、静态时序分析器等，可用于对设计中的逻辑关系及输出结果进行检验，并详尽分析各个时序限制的满足情况。如图 1.2 所示：

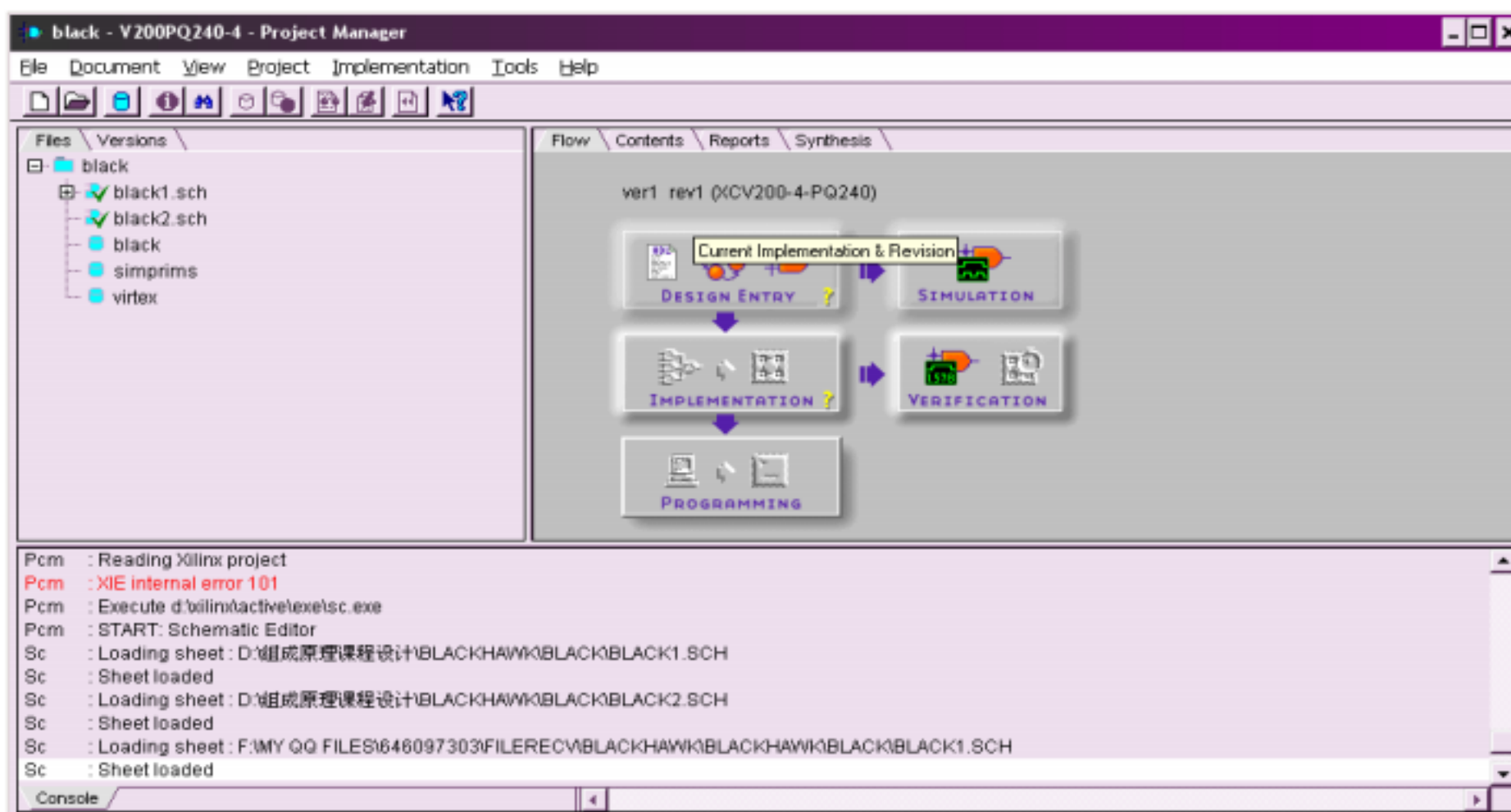


图 1.2 Xilinx foundation f3.1 设计平台

(3) COP2000 集成调试软件

COP2000 集成开发环境是为 COP2000 实验仪与 PC 机相连进行高层次实验的配套软件，它通过实验仪的串行接口和 PC 机的串行接口相连，提供汇编、反汇编、编辑、修改指令、文件传送、调试 FPGA 实验等功能，该软件在 Windows 下运行。如图 1.3 所示：

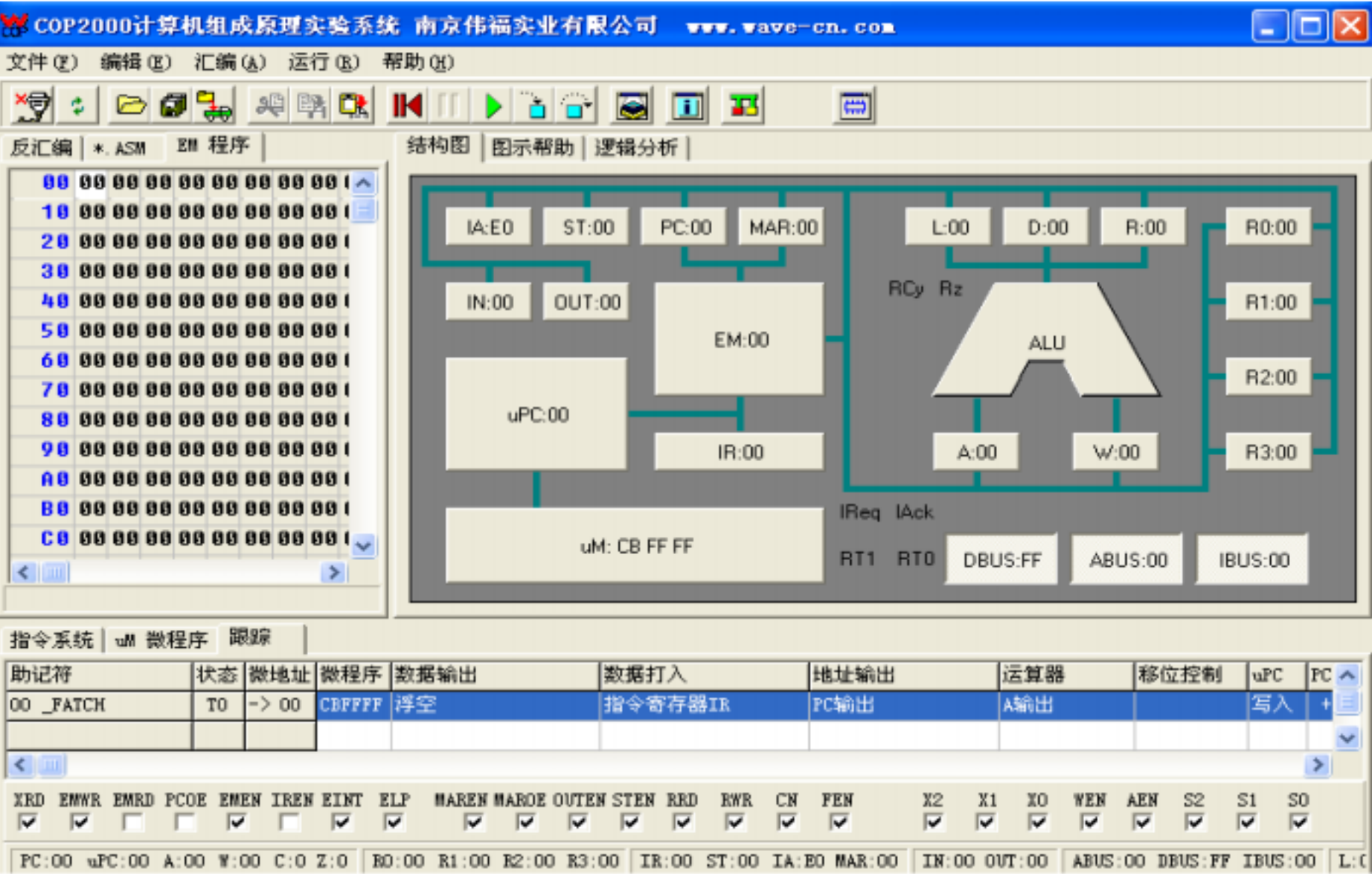


图 1.3 COP2000 计算机组成原理集成调试软件

第 2 章 详细设计方案

2.1 顶层方案图的设计与实现

顶层方案图是实现两个四位二进制数的定点原码一位除法器的逻辑功能，采用原理图设计输入方式完成，电路实现基于 XCV200 可编程逻辑芯片。在完成原理图的功能设计后，把输入和输出信号安排到 XCV200 指定的引脚上去，实现芯片的引脚锁定。除法运算作为顶层模块，可利用 **Xilinx foundation f3.1** 中的器件来实现，顶层图形文件结构如图 2.1 所示：

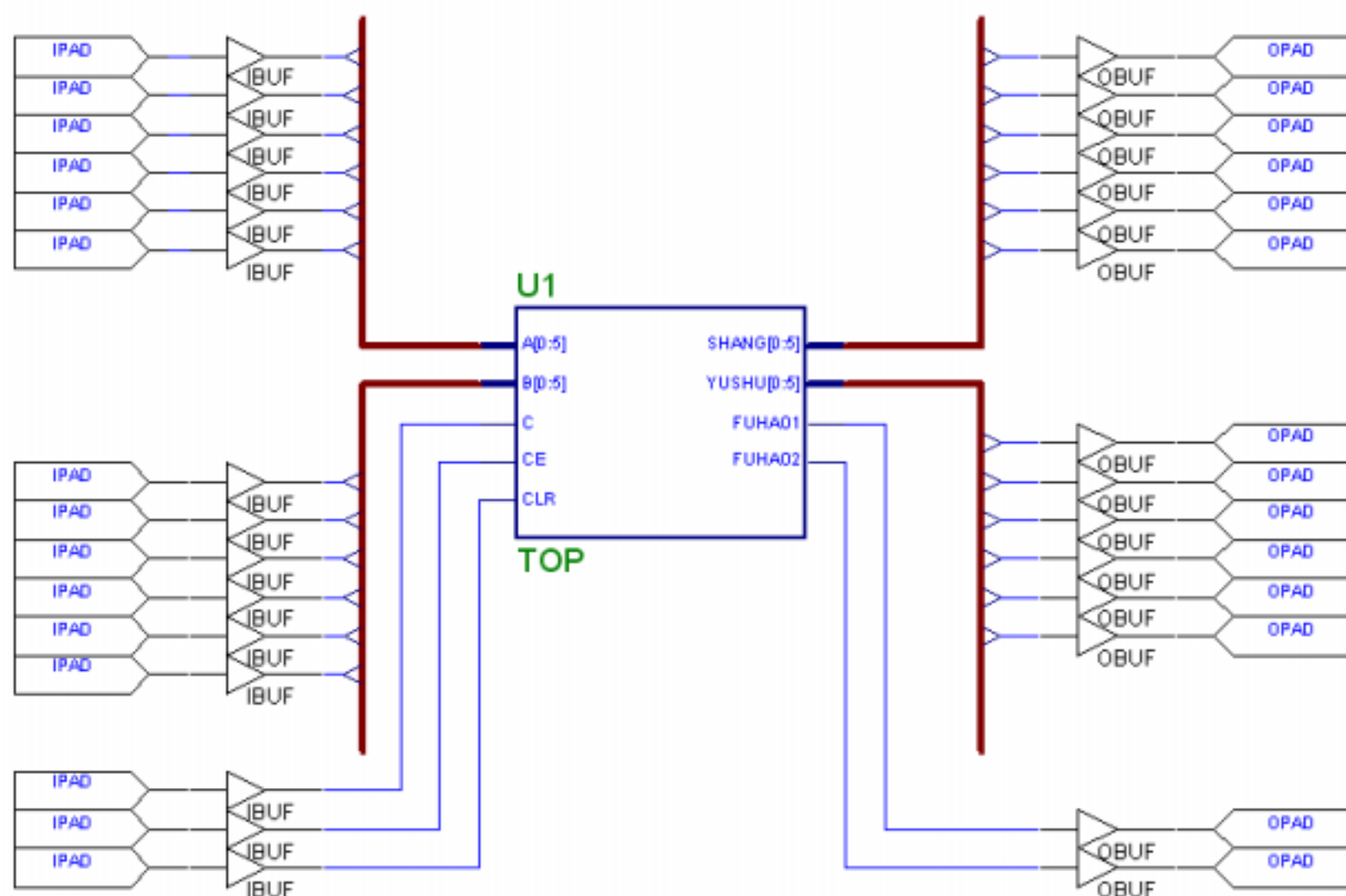


图 2.1 定点原码一位除法器顶层图形文件结构

2.1.1 创建顶层图形设计文件

顶层图形文件是由两个六位的二进制数输入端、一个功能使能端、一个数据清零端和一个脉冲控制端，一个四位商输出端，一个六位余数输出端和一个两位符号位输出端组装成的一个设计体系。除法器的设计采用自顶向下的设计思路 and 自底向上的实现思想。

2.1.2 器件的选择与引脚锁定

(1) 器件的选择

由于硬件设计环境是基于伟福 COP2000型计算机组成原理实验仪和 XCV200 实验板，故采用的目标芯片为 Xilinx XCV200 可编程逻辑芯片。

(2) 引脚锁定

被除数 A [0:5]为六位输入信号，除数 B [0:5] 为六位输入信号，时钟脉冲 CLK、电路使能端 CE 和清零端 CLR 各占一位管脚；所得的商 SHANG [0:4] 占有五位信号，余数 YUSHU [0:5] 占六位信号，以及符号 FUHAO1 和 FUHAO2 各占一位信号，把顶层图形文件中的输入和输出信号安排到 Xilinx XCV200 芯片指定的引脚上去，实现芯片的引脚锁定，各信号及 Xilinx XCV200 芯片引脚对应关系如表 2.1 所示：

表 2.1 引脚锁定对应

输入信号	
XCV200 芯片引脚信号	XCV200 实验板
P72	K2：1
P71	K2：2
P213	CLOCK
P96	K0：5
P97	K0：4
P100	K0：3
P101	K0：2
P102	K0：1
P103	K0：0
P81	K1：5
P82	K1：4
P84	K1：3
P85	K1：2
P86	K1：1

P87	K1 : 0
输出信号	
XCV200 芯片引脚信号	XCV200 实验板
P178	A5
P184	A4
P185	A3
P203	A2
P111	A1
P110	A0
P78	B7
P93	B6
P107	B4
P108	B3
P109	B2
P124	B1
P125	B0

2.1.3 编译、综合、适配

利用 Xilinx foundation f3.1 的原理图编辑器对顶层图形文件进行编译，并最终生成网络表文件，利用设计实现工具经综合、优化、适配，生成可供时序仿真的文件和器件下载编程文件。

2.2 功能模块的设计与实现

定点原码一位除法器的整体设计的主要模块为：输入寄存器模块、选择器模块、移位器模块、加法器模块、求补码模块、控制器模块、输出寄存器模块为基础而实现的。

2.2.1 控制器模块的设计与实现

该模块的输出为清零 (CLR), 使能端 (CE), 五个寄存器的控制端和两个选择器的片选端时钟控制信号。

(1) 选择器模块原理图

相反数补码器的输入端用 PATHIN[0:5] 来表示, 输出端用 PATHOUT[0:5] 表示。加法器模块原理如图 2.2 所示:

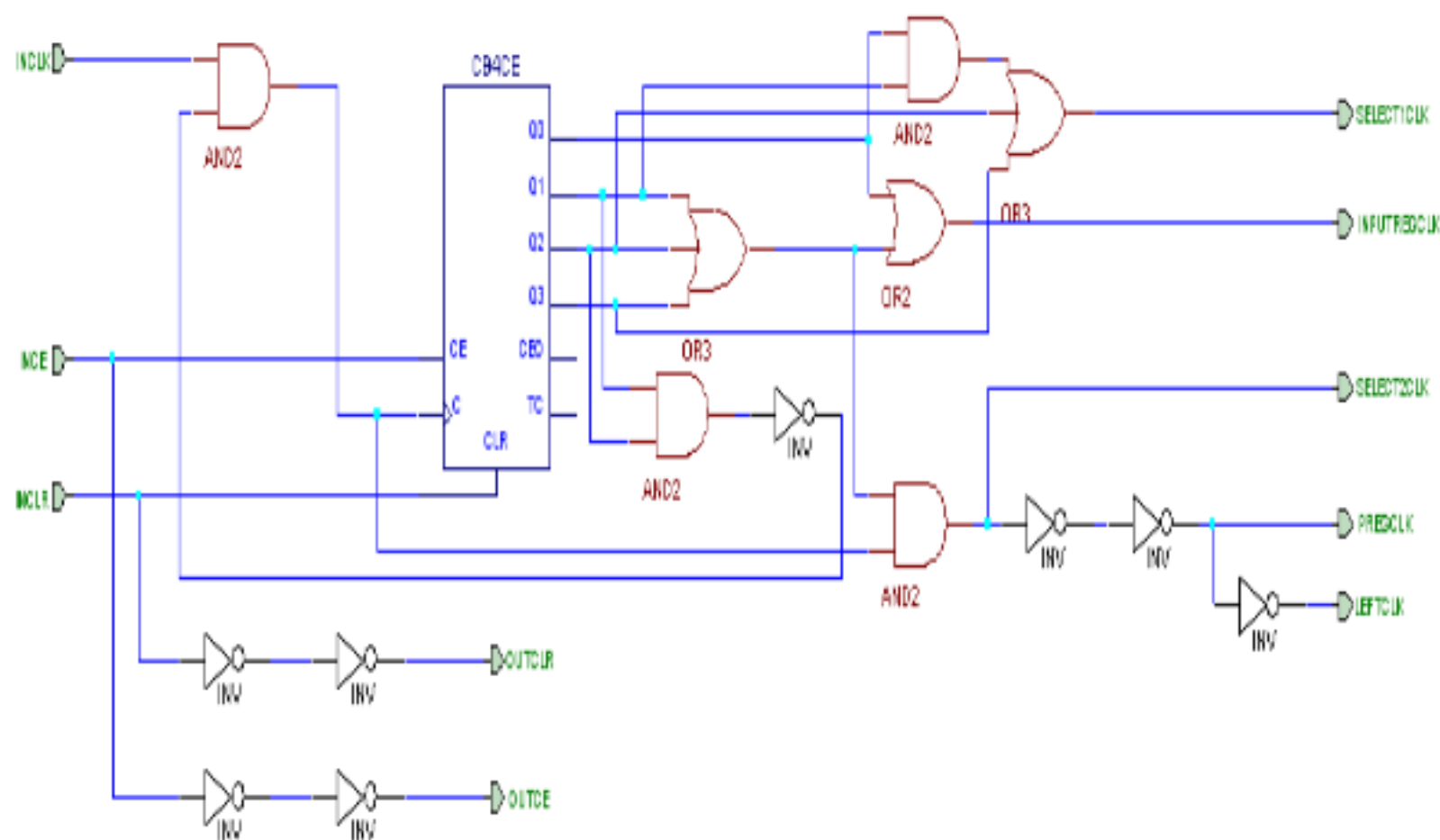


图 2.2 控制器模块逻辑电路

(2) 创建元件图形符号

为能在图形编辑器 (原理图设计输入方式) 中调用 CONTROLUN 芯片, 需要为 CONTROLUN 模块创建一个元件图形符号, 可利用 Xilinx foundation f3.1 编译器中的如下步骤实现: Tools=>Symbol Wizard=>下一步。主时钟脉冲 (INCLK)、除法器使能端 (INCE)、除法器清零端 (INCLR) 是输入信号, INPUTREGCLK、SELECT1CLK、SELECT2CLK、PRECLK、OUTCE、LEFTCLK 和 OUTCL 是数据输出信号。其元件图形符号如图 2.3 所示:

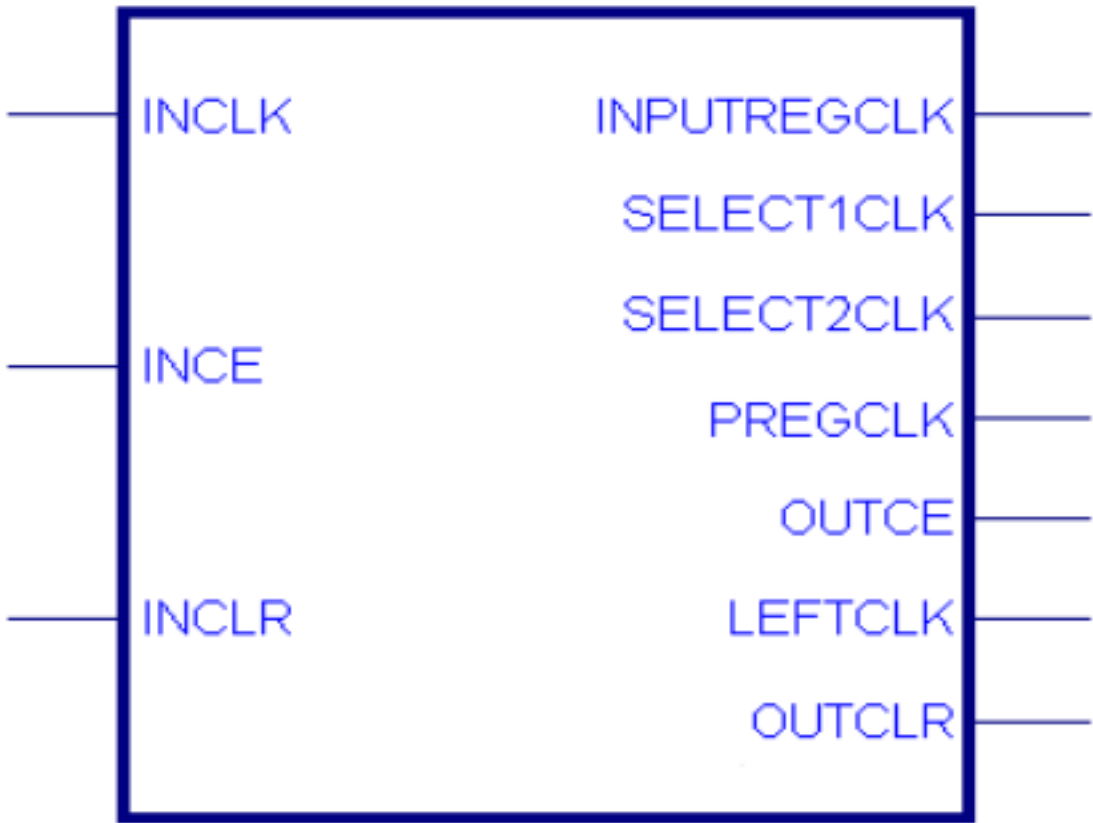


图 2.3 控制器模块元件图形符号

（3）功能仿真

对创建的控制器电路模块进行功能仿真，验证其功能的正确性，可用 Xilinx Foundation f3.1 编译器 Simulator 模块实现。仿真结果如图 2.4 所示：

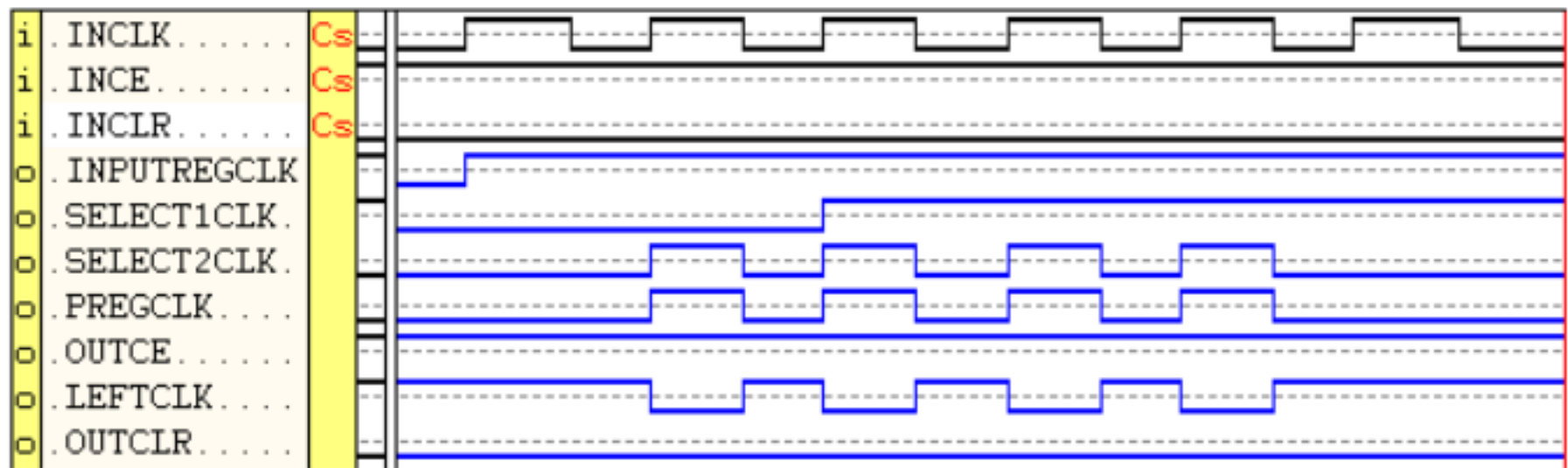


图 2.4 控制器模块仿真结果

将控制器模块的功能和仿真结果对照，可知控制器模块的仿真结果正确。

2.2.2 输入寄存器的设计与实现

输入寄存器电路需要使用两个 FD4CE 寄存器来实现，寄存器模块被应用于被除数寄存器模块和除数寄存器模块，具有使能、清零和时钟脉冲端，可以对于输入的数据进行控制，用于求出被除数与除数的绝对值以供使用。

（1）输入寄存器模块原理图

输入端用 PREGIN[0:5]，输出用 PREGOUT[0:5]表示，控制端用 CE、CLK、CLR 来表示。输入寄存器模块原理如图 2.5 所示：

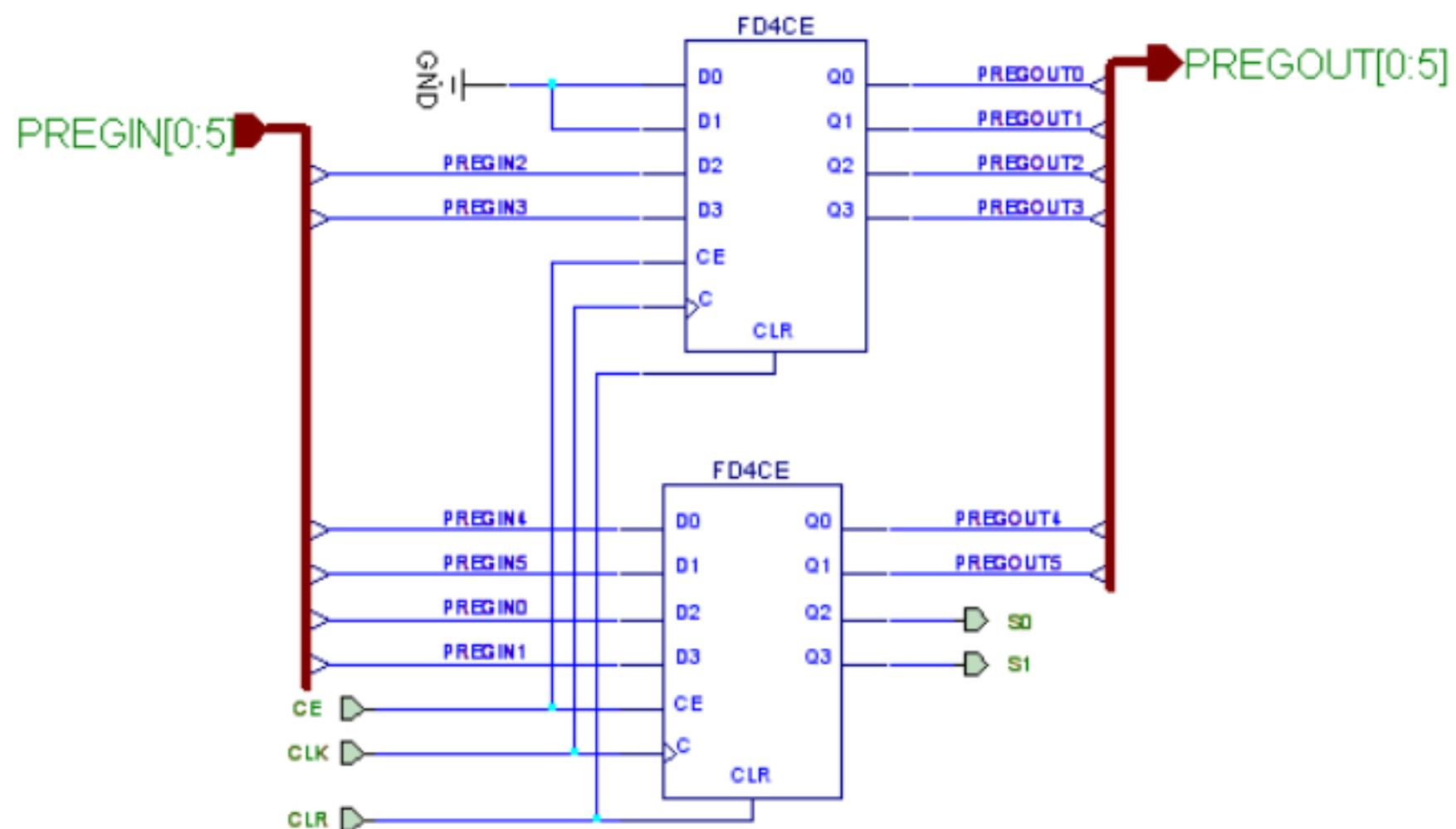


图 2.5 输入寄存器模块逻辑电路

(2) 创建元件图形符号

为能在图形编辑器（原理图设计输入方式）中调用 PINREGISTE 芯片，需要为 PINREGISTER 模块创建一个元件图形符号，可利用 Xilinx foundation f3.1 编译器中的如下步骤实现：Tools=>Symbol Wizard=>下一步。PREGIN[0:5] 是数据输入信号，CE CLR CLK 是输入信号，PREGOUT[0:5] S0 S1 是数据输出信号。其元件图形符号如图 2.6 所示：

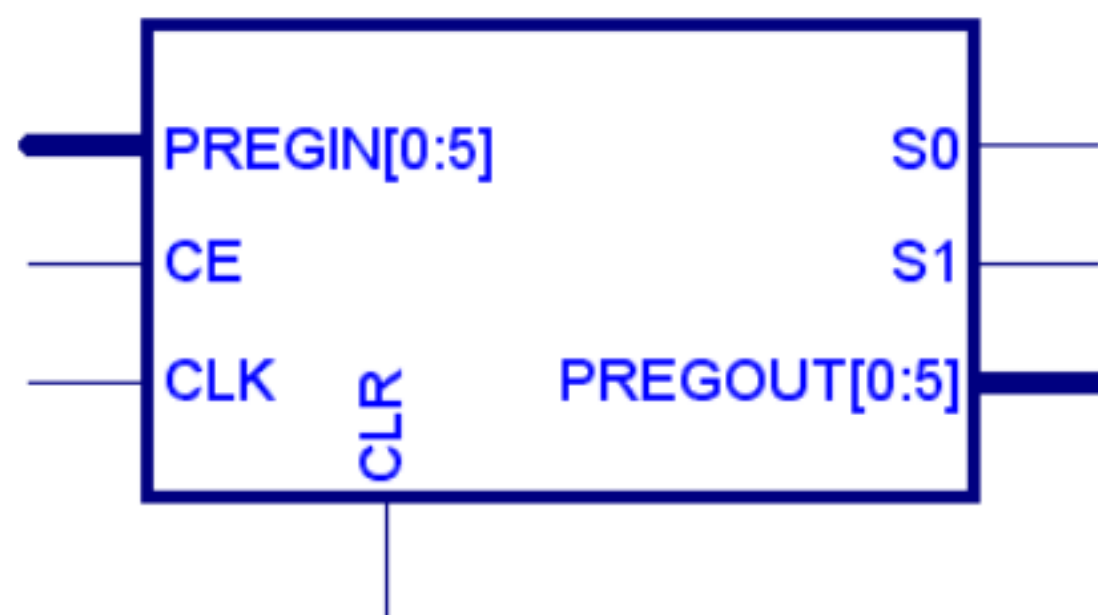


图 2.6 输入寄存器模块元件图形符号

(3) 功能仿真

对创建的输入寄存器模块功能进行仿真，验证其功能的正确性，可用 Xilinx Foundation 编译器的 Simulator 模块实现。仿真结果如图 2.7 所示：

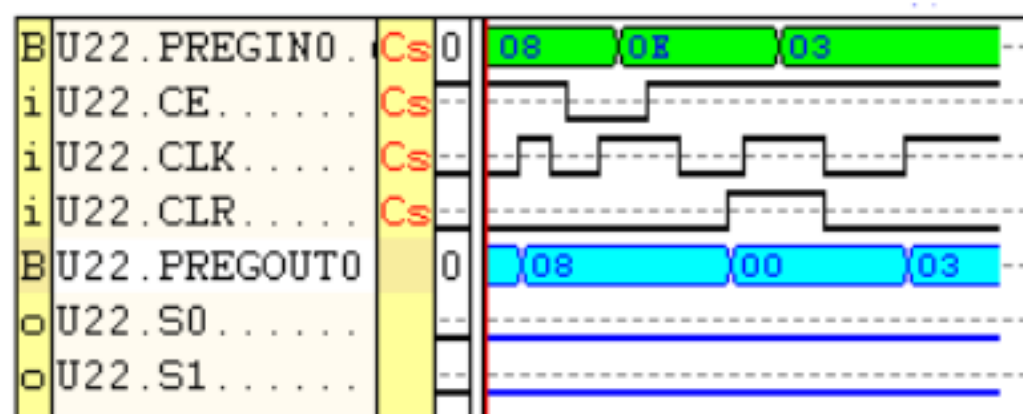


图 2.7 输入寄存器模块仿真结果

将寄存器模块的功能和仿真结果对照可知，寄存器模块的仿真结果正确。

2.2.3 选择器模块的设计与实现

选择器用于实现对数据的选择，该模块由六个 M2_1 器件构成。该选择器应用于选择符合要求的除数和余数。

(1) 选择器模块原理图

选择器的输入端用 SELECTINA[0:5] 和 SELECTINB[0:5] 来表示输入的要选择的数据，输出端用 SELECTOUT[0:5] 来表示，用 S 来判断选择哪个数据。选择器模块原理如图 2.8 所示：

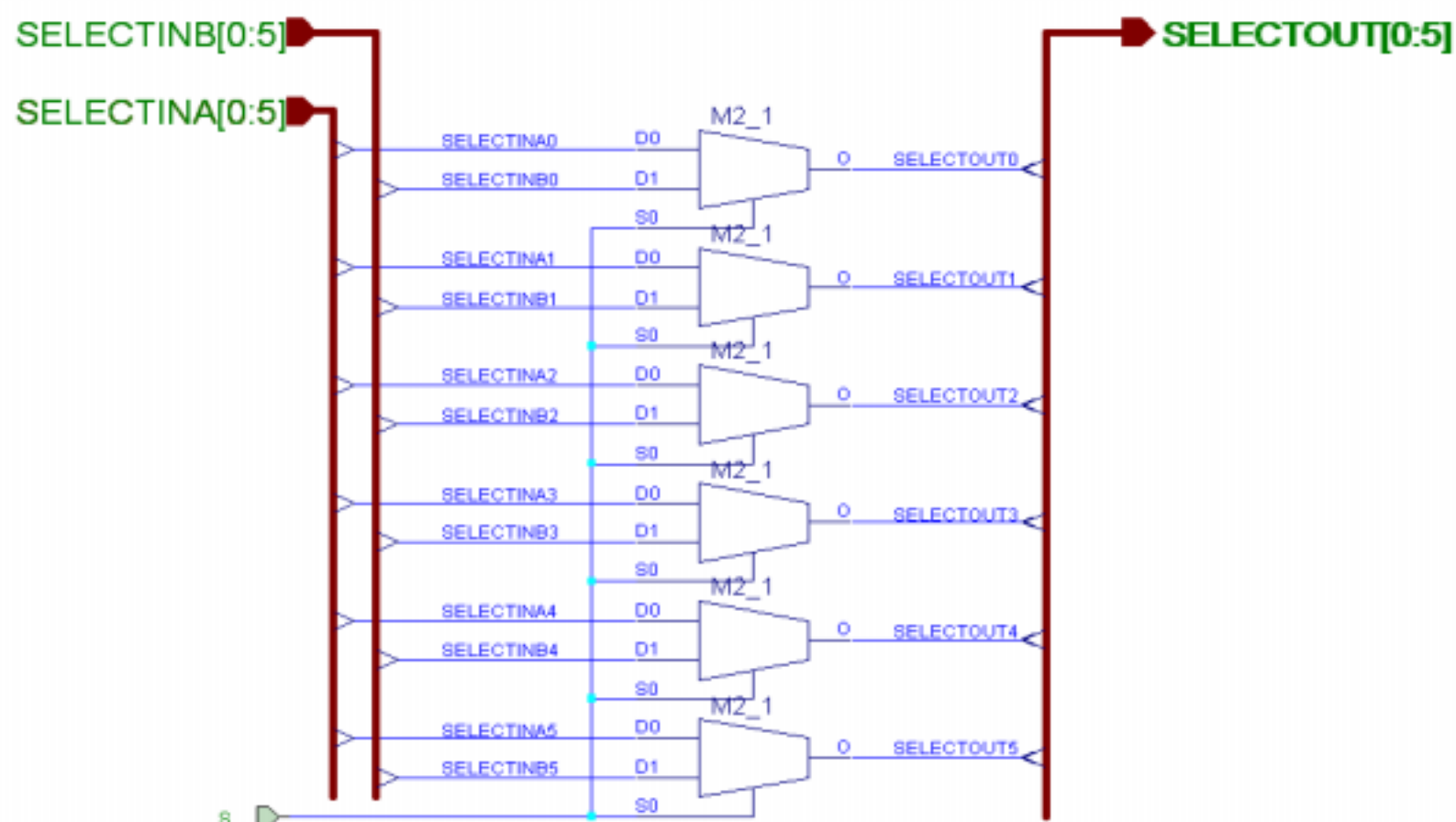


图 2.8 选择器模块逻辑电路

(2) 创建元件图形符号

为能在图形编辑器（原理图设计输入方式）中调用 SELECTUNI芯片，需要为 SELECTUNI模块创建一个元件图形符号，可利用 Xilinx foundation f3.1 编译器中的如下步骤实现：Tools=>Symbol Wizard=> 下一步。SELECTINA[0:5]、SELECTINB[0:5]和 S是输入信号，SELECTOUT[0:5]是数据输出信号。其元件图形符号如图 2.9 所示：

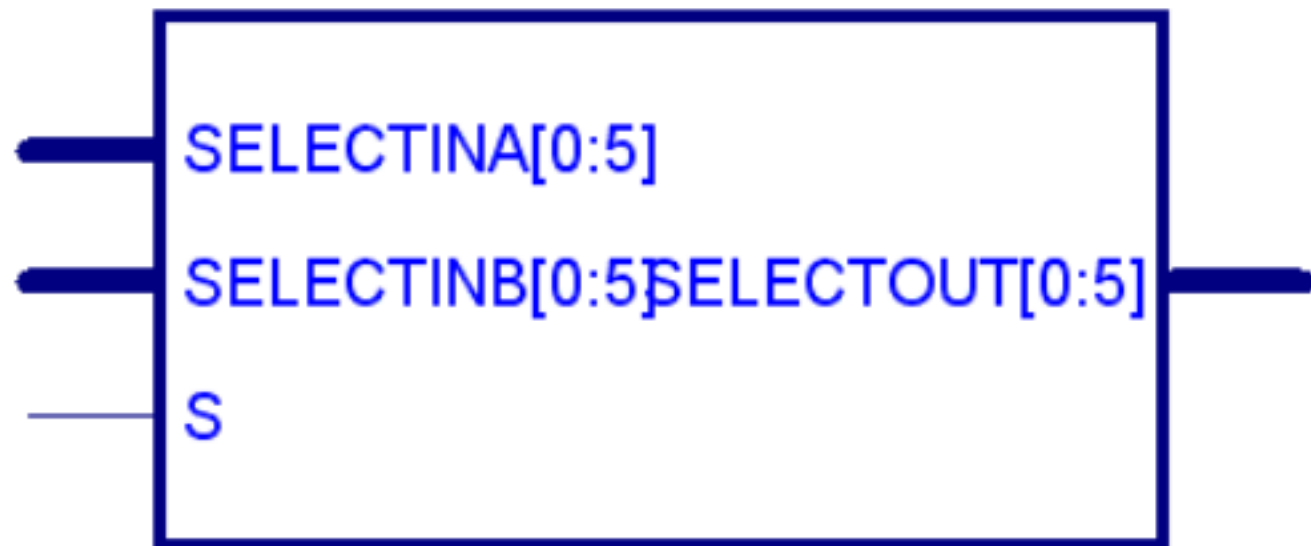


图 2.9 选择器模块元件图形符号

（3）功能仿真

对创建的选择器模块功能进行仿真，验证其功能的正确性，可用 Xilinx Foundation f3.1 编译器 Simulator 模块实现。仿真结果如图 2.10 所示：

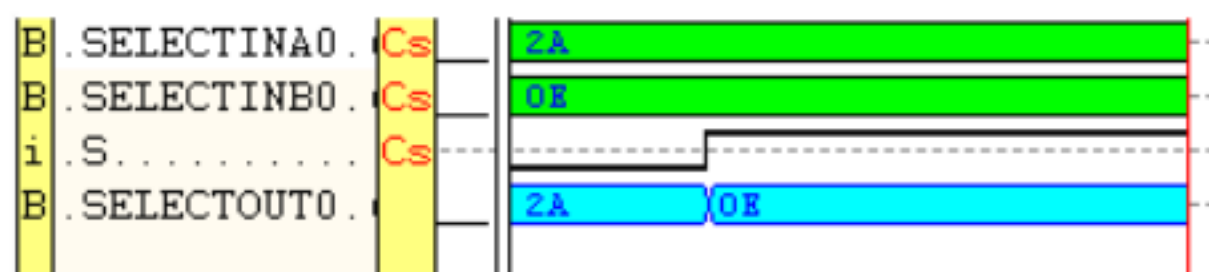


图 2.10 选择器模块仿真结果

将选择器模块的功能和仿真结果对照，可知选择器模块的仿真结果正确。

2.2.4 相反数补码模块的设计与实现

由于符号位已经单独作考虑，所以参与运算的数都是正数，即通过六个非门和高电平的处理，就可得到求补后的输出信号。

（1）选择器模块原理图

相反数补码器的输入端用 PATHIN[0:5] 来表示，输出端用 PATHOUT[0:5] 表示。加法器模块原理如图 2.11 所示：

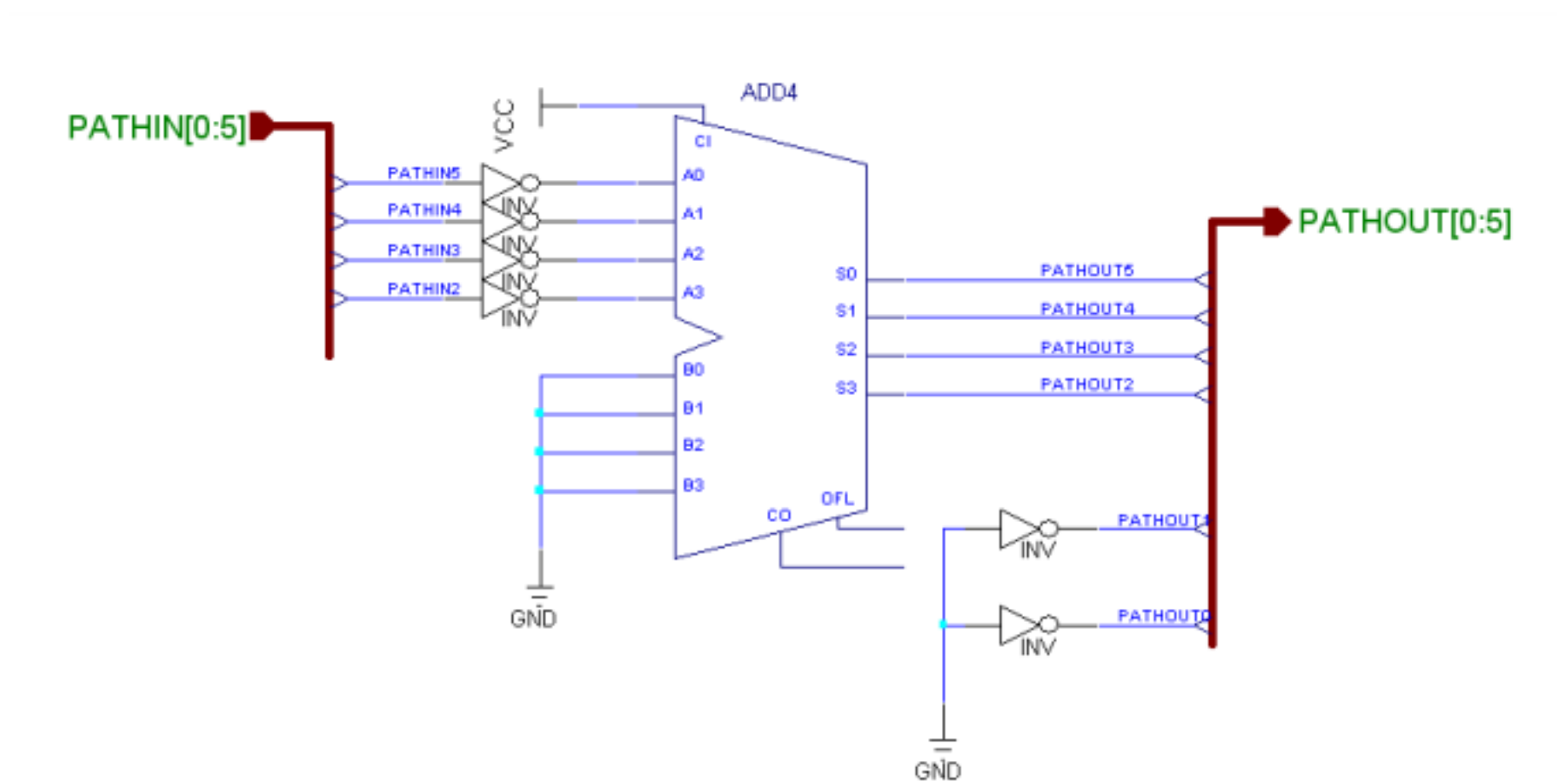


图 2.11 相反数补码模块逻辑电路

(2) 创建元件图形符号

为能在图形编辑器（原理图设计输入方式）中调用 PATHCOD 芯片，需要为 PATHCOD 模块创建一个元件图形符号，可利用 Xilinx foundation f3.1 编译器中的如下步骤实现：Tools=>Symbol Wizard=> 下一步。PATHIN[0:5] 是输入信号，PATHOUT[0:5] 是数据输出信号。其元件图形符号如图 2.12 所示：



图 2.12 相反数补码模块元件图形符号

(3) 功能仿真

对创建的加法器电路模块进行功能仿真，验证其功能的正确性，可用 Xilinx Foundation f3.1 编译器 Simulator 模块实现。仿真结果如图 2.13 所示：

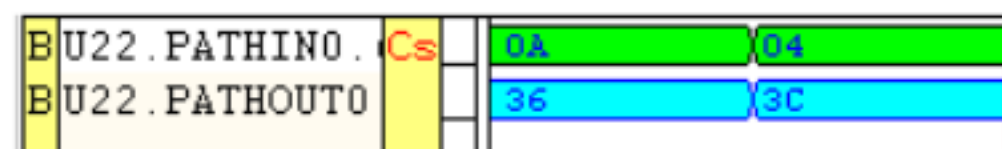


图 2.13 相反数补码模块仿真结果

将加法器模块的功能和仿真结果对照可知，加法器模块的仿真结果正确。

2.2.5 加法器模块的设计与实现

该模块要实现的是两个四位二进制数的相加，此模块的功能和一般加法器的功能一致。利用已给的加法器设计所需要的加法器

(1) 选择器模块原理图

移位寄存器的输入端用 $ADD\text{AIN}[0:5]$ 和 $ADD\text{BIN}[0:5]$ 来表示，输出端用 $ADD\text{OUT}[0:5]$ 表示。加法器模块原理如图 2.14 所示：

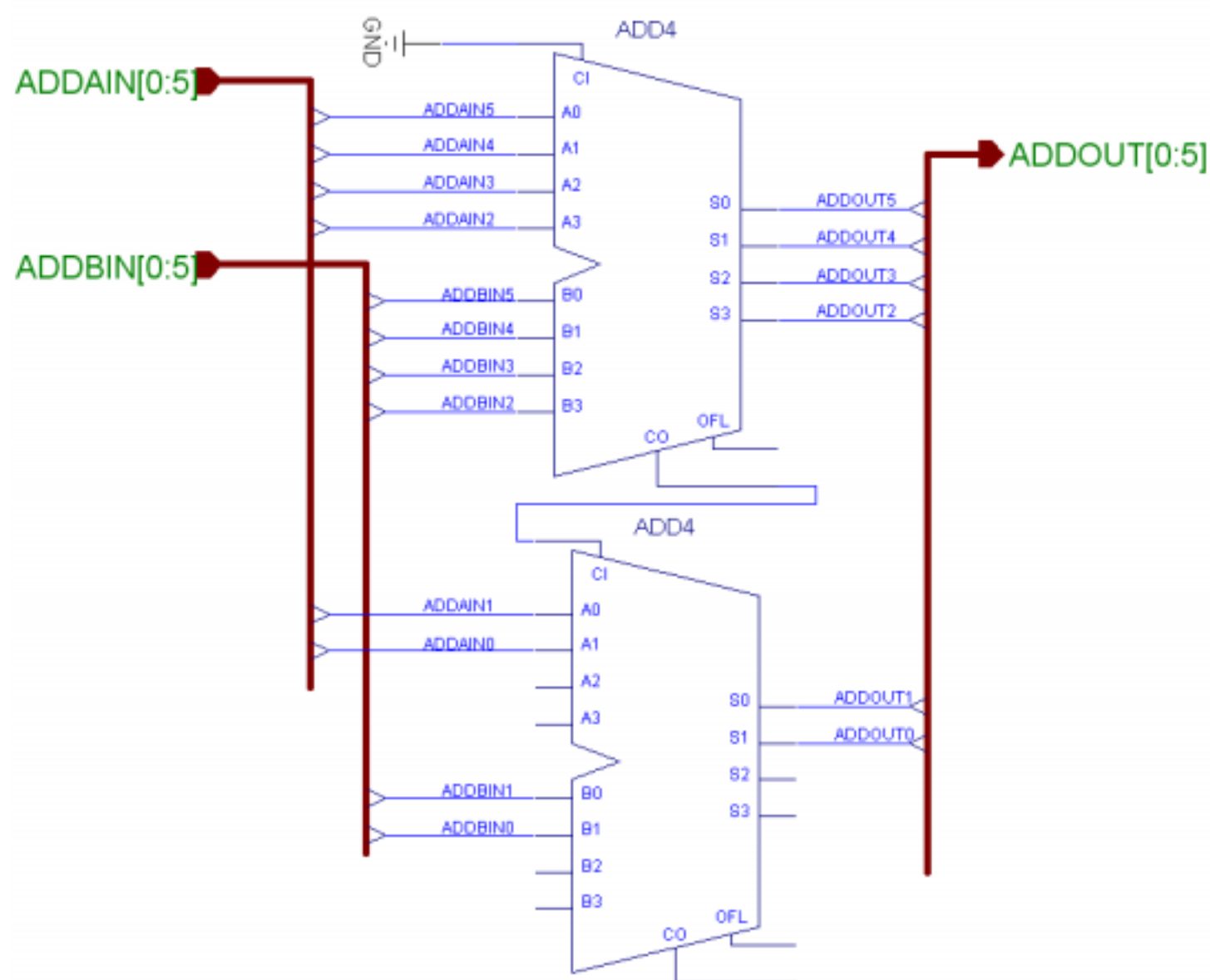


图 2.14 加法器模块逻辑电路

(2) 创建元件图形符号

为能在图形编辑器（原理图设计输入方式）中调用 ADD6 芯片，需要为 ADD6 模块创建一个元件图形符号，可利用 Xilinx foundation f3.1 编译器中的如下步

骤实现：Tools=>Symbol Wizard=>下一步。 ADDAIN[0:5] 和 ADDBIN[0:5] 是输入信号， ADDOUT[0:5]是数据输出信号。其元件图形符号如图 2.15 所示：

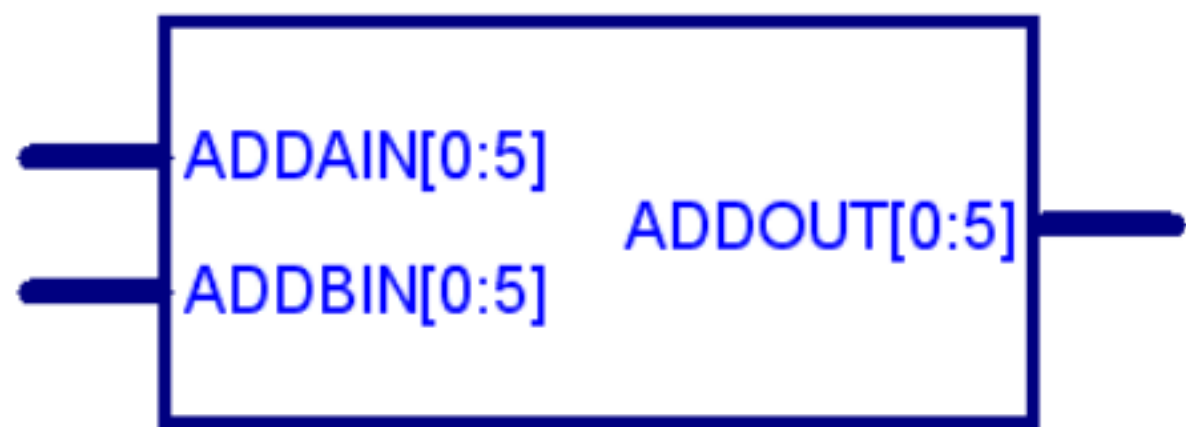


图 2.15 加法器模块元件图形符号

(3) 功能仿真

对创建的加法器电路模块功能进行仿真，验证其功能的正确性，可用 Xilinx Foundation f3.1 编译器 Simulator 模块实现。仿真结果如图 2.16 所示：



图 2.16 加法器模块仿真结果

将加法器模块的功能和仿真结果对照，可知加法器模块的仿真结果正确。

2.2.6 移位电路模块的设计与实现

移位电路需要使用两个 FD4CE 寄存器来实现，并且利用时钟脉冲、使能端口和清零端口来实现启停、清零和实现余数左移一位的功能。

(1) 移位电路模块原理图

移位寄存器的输入端用 PLEFTIN[0:5] 来表示输入的所要的数据，输出端用 PLEFTOUT[0:5] 表示，用 CE、C 和 CLR 来控制数据的输入。移位寄存器寄存器模块原理如图 2.17 所示：

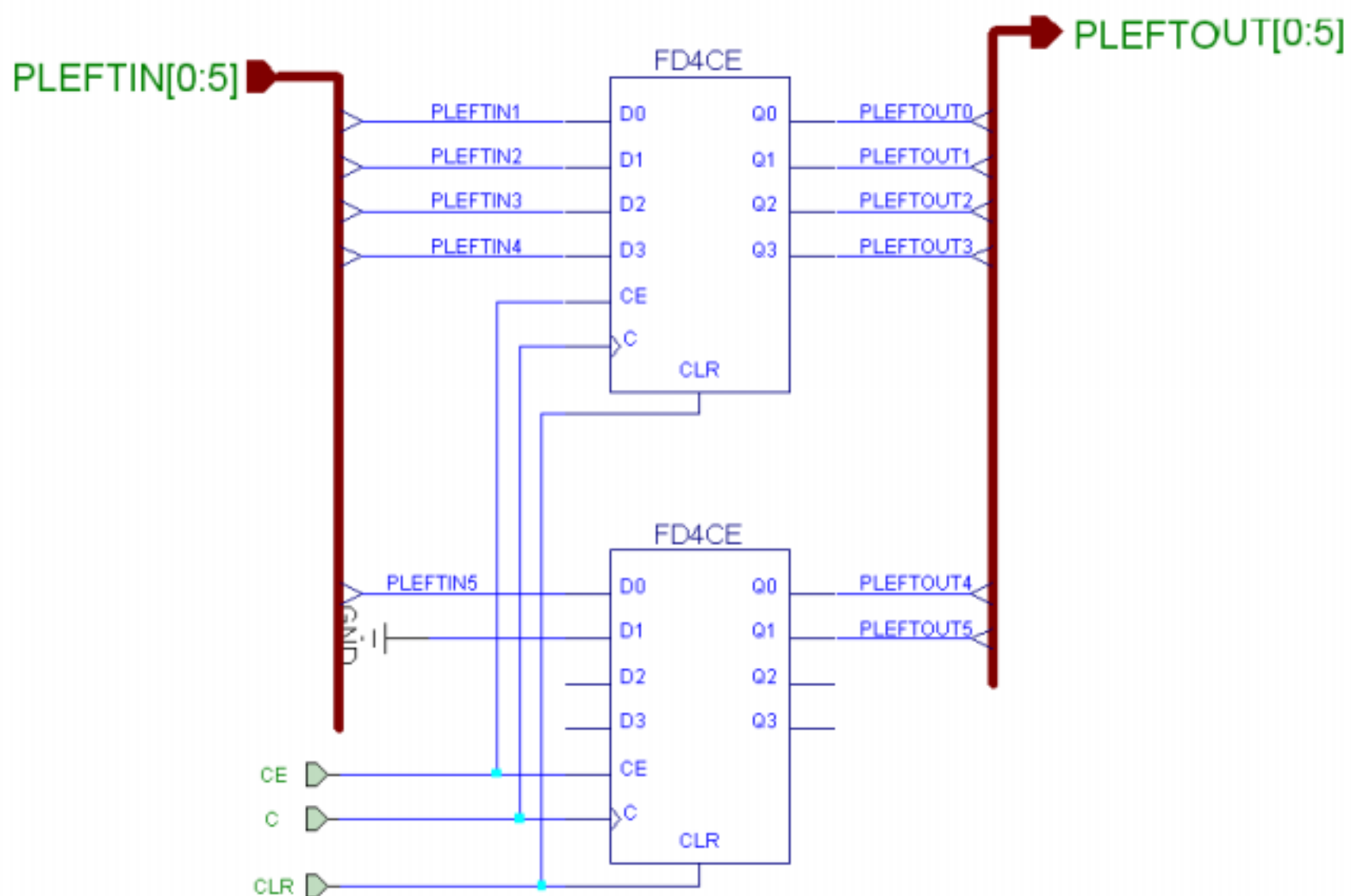


图 2.17 移位电路模块逻辑电路

(2) 创建元件图形符号

为能在图形编辑器（原理图设计输入方式）中调用 PLEFTREGISTE 芯片，需要为 PLEFTREGISTE 模块创建一个元件图形符号，可利用 Xilinx foundation f3.1 编译器中的如下步骤实现：Tools=>Symbol Wizard=>下一步。PLEFTIN[0:5]、CE、C 和 CLR 是输入信号，PLEFTOUT[0:5] 是数据输出信号。其元件图形符号如图 2.18 所示：

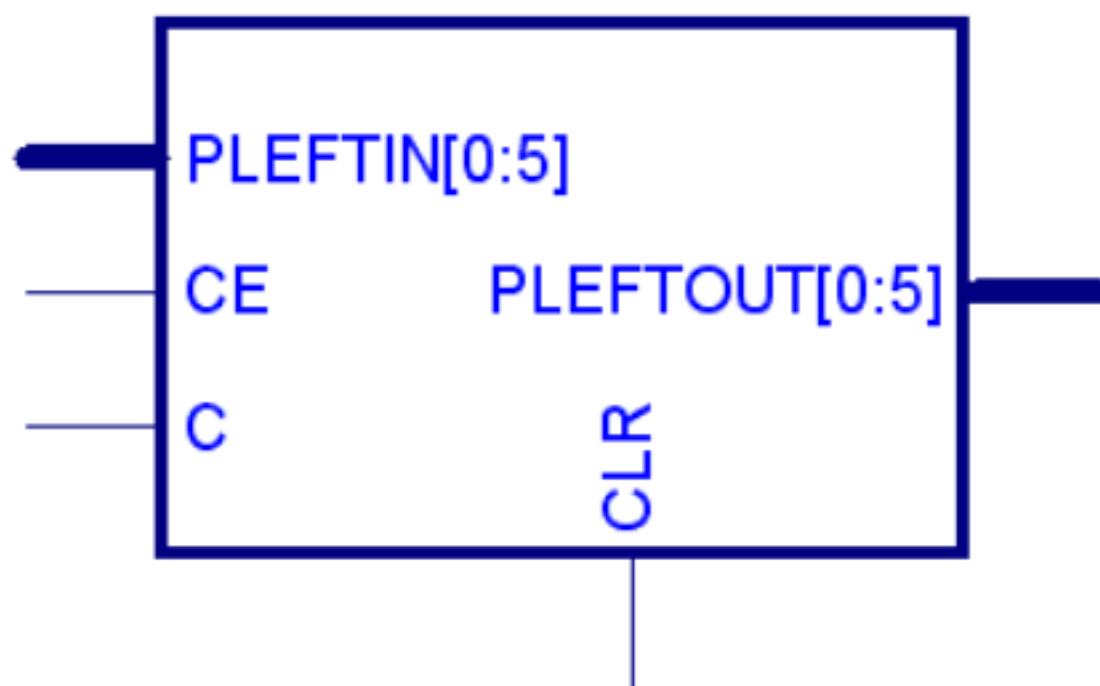


图 2.18 移位寄存器模块元件图形符号

(3) 功能仿真

对创建的移位电路模块进行功能仿真，验证其功能的正确性，可用 Xilinx Foundation f3.1 编译器 Simulator 模块实现。仿真结果如图 2.19 所示：

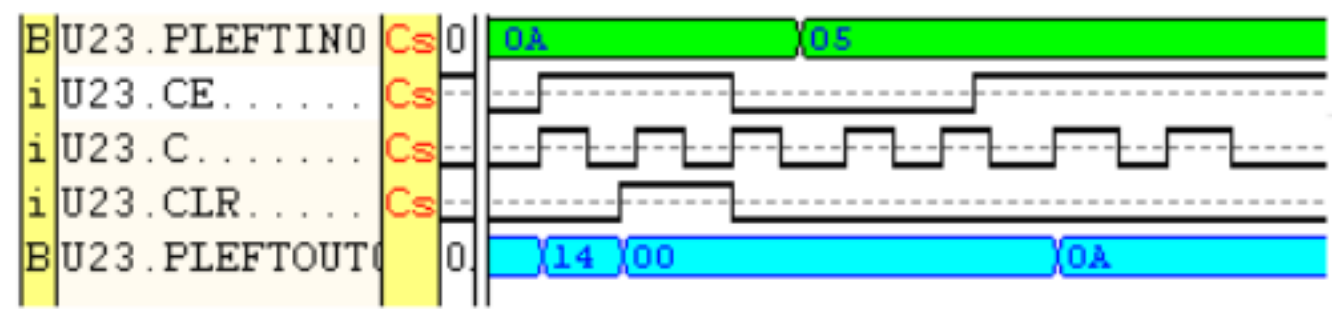


图 2.19 移位寄存器模块仿真结果

将移位器模块的功能和仿真结果对照，可知移位器模块的仿真结果正确。

2.2.7 恢复余数寄存器模块的设计与实现

由于要恢复余数，首先要判断片选端的信号是高电平还是低电平，进而做出判断是要实行加法，还是直接把数据送入到就寄存器中。在判断信号的时候，我们要利用 M2_1 器件来实现，如果是低电平的话，就要实行加法这一操作，故需要加法器；而如果是高电平的话，就是要实行数据储存的这一操作，我们要利用寄存器来存储。

(1) 恢复余数寄存器模块原理图

相反数补码器的输入端用 RESIN[0:5]、CHUSHUIN[0:5] RES\$ RESCE RESC和 RECLR来表示，输出端用 RESOUT[0:5]表示。恢复余数寄存器模块原理如图 2.20 所示：

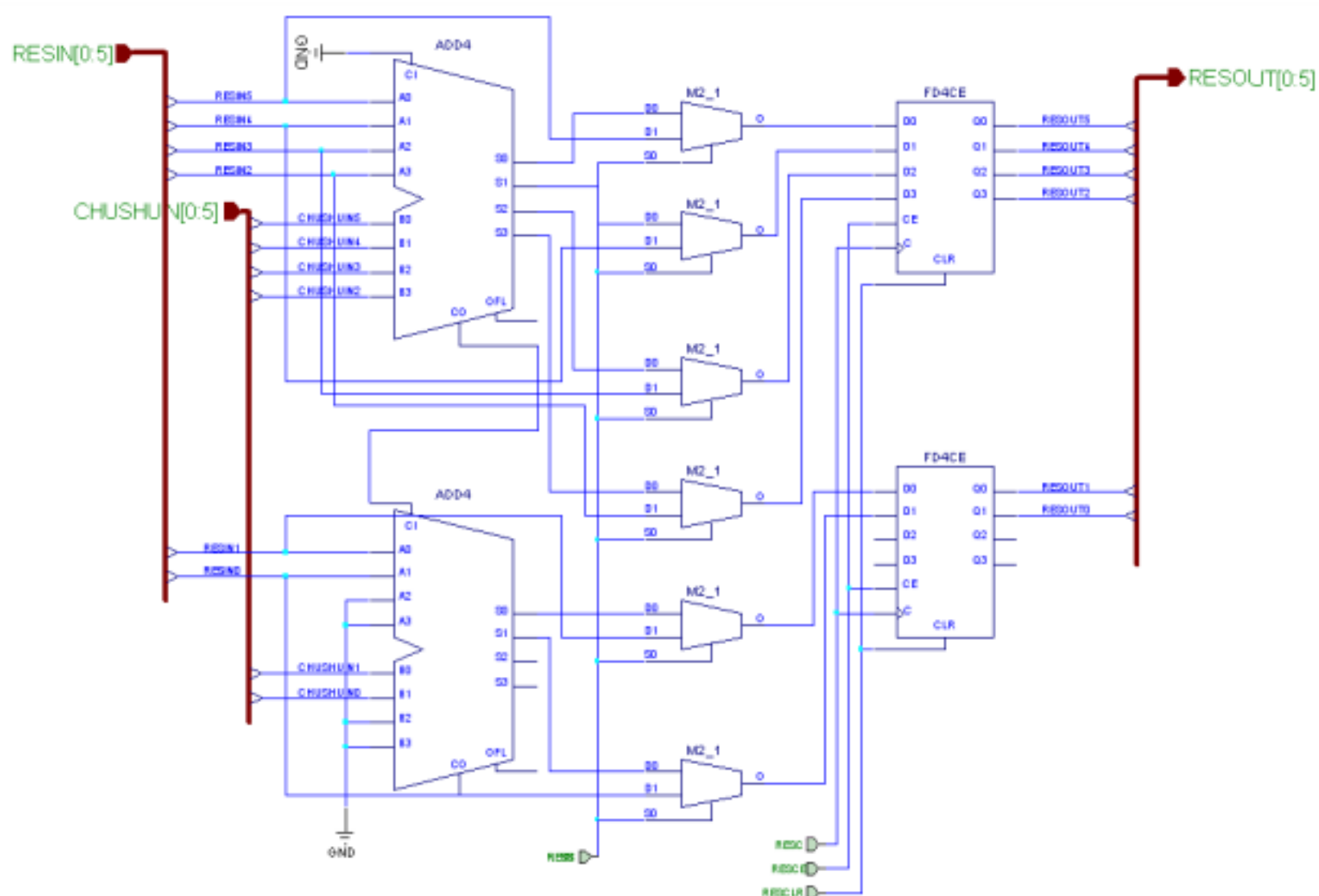


图 2.20 恢复余数寄存器模块逻辑电路

（2）创建元件图形符号

为能在图形编辑器（原理图设计输入方式）中调用 RESUMECOM 模块，需要为 RESUMECOM 模块创建一个元件图形符号，可利用 Xilinx foundation f3.1 编译器中的如下步骤实现：Tools=>Symbol Wizard=>下一步。RESIN[0:5]、CHUSHUIN[0:5]、RESS、RESCE、RESC 和 RESCLR 是输入信号，RESOUT[0:4] 是数据输出信号。其元件图形符号如图 2.21 所示：

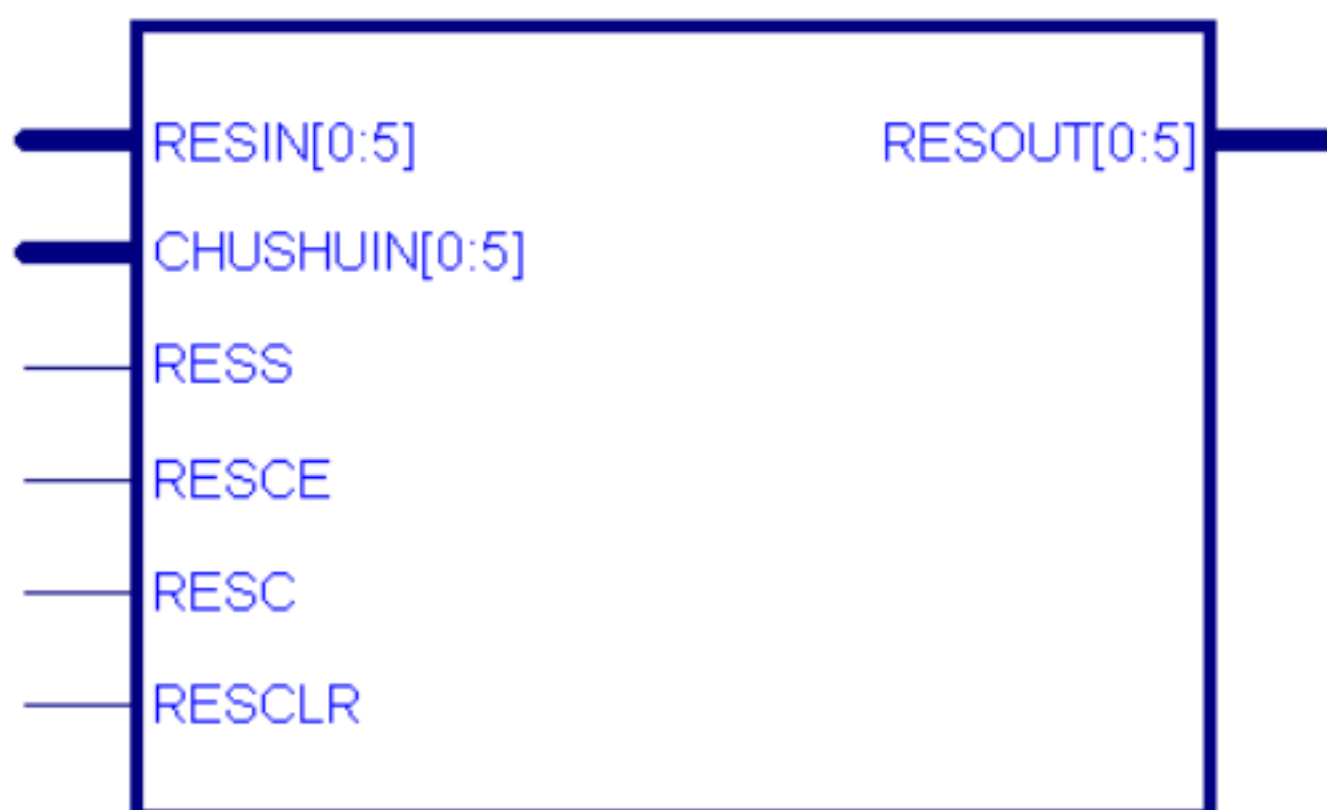


图 2.21 恢复余数寄存器模块元件图形符号

(3) 功能仿真

对创建的恢复余数寄存器模块进行功能仿真，验证其功能的正确性，可用 Xilinx Foundation f3.1 编译器 Simulator 模块实现。仿真结果如图 2.22 所示：

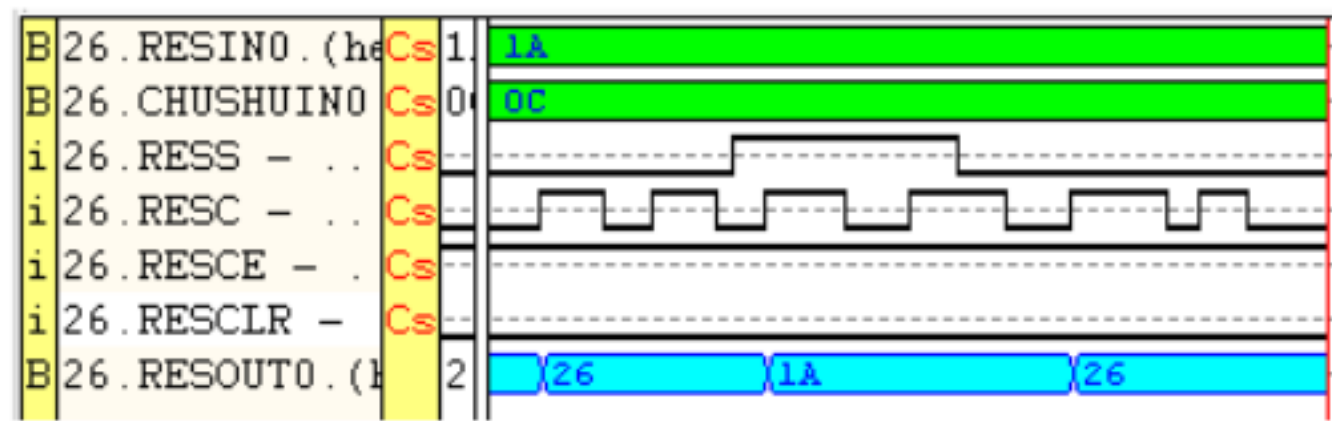


图 2.22 恢复余数寄存器模块仿真结果

将恢复余数寄存器模块的功能和仿真结果对照，可知恢复余数寄存器模块的仿真结果正确。

2.2.8 商寄存器模块的设计与实现

商寄存器数据的输入是利用移位输入，利用数据的左移来达到商的输出，并利用时钟脉冲、使能端口和清零端口来实现启停和清零功能。

(1) 商寄存器模块原理图

相反数补码器的输入端用 ONLEFTIN CE CLK CLR 来表示，输出端用 SHANGOUT[0:4]表示。加法器模块原理如图 2.23 所示：

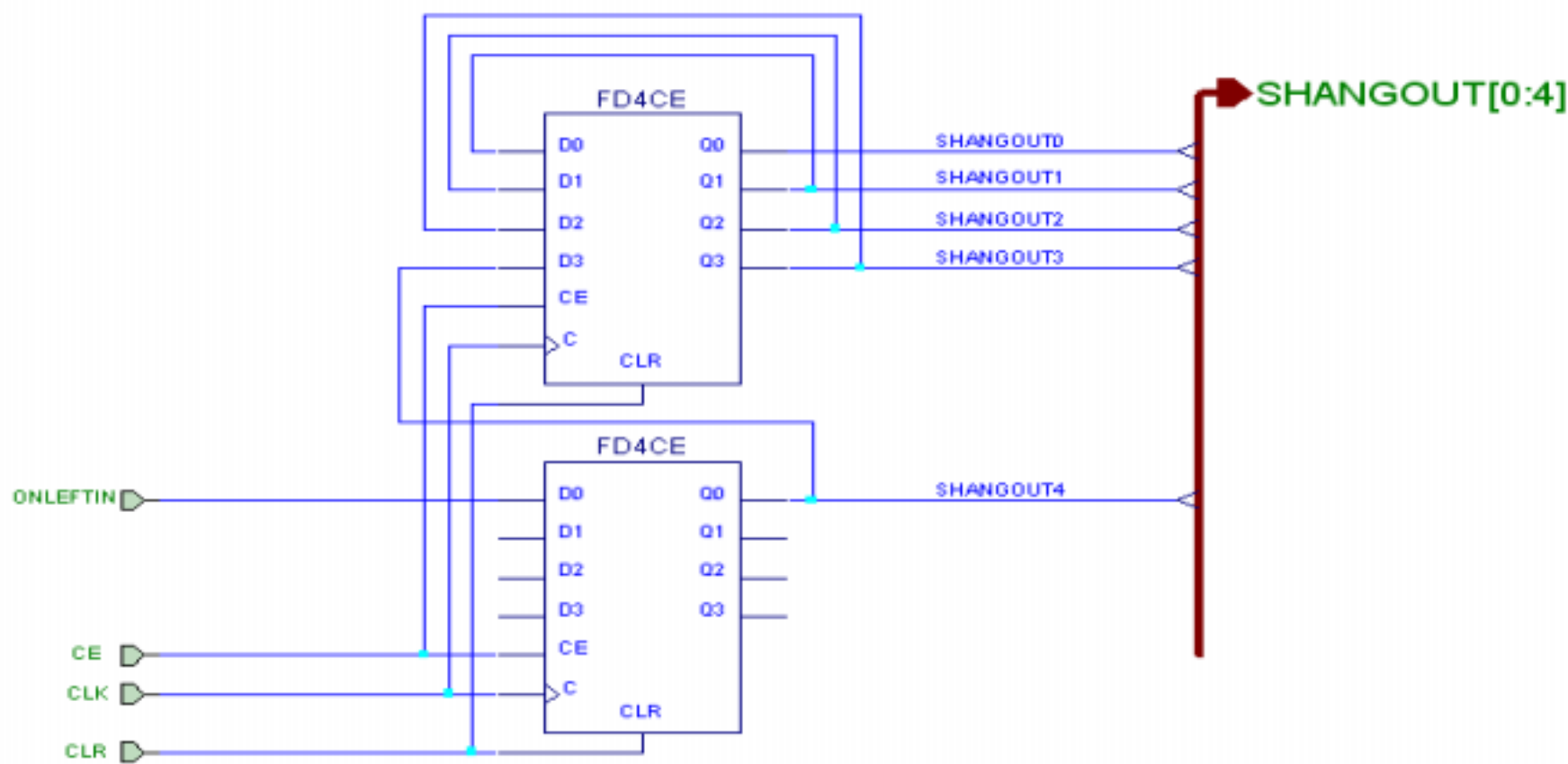


图 2.23 商寄存器模块逻辑电路

(2) 创建元件图形符号

为能在图形编辑器（原理图设计输入方式）中调用 ONELEFTREGISTER 芯片，需要为 ONELEFTREGISTER 模块创建一个元件图形符号，可利用 Xilinx foundation f3.1 编译器中的如下步骤实现：Tools=>Symbol Wizard=>下一步。ONLEFTIN CE CLK CLR是输入信号，SHANGOUT[0:4]是数据输出信号。其元件图形符号如图 2.24 所示：



图 2.24 商寄存器模块元件图形符号

(3) 功能仿真

对创建的商寄存器模块进行功能仿真，验证其功能的正确性，可用 Xilinx Foundation f3.1 编译器 Simulator 模块实现。仿真结果如图 2.25 所示：

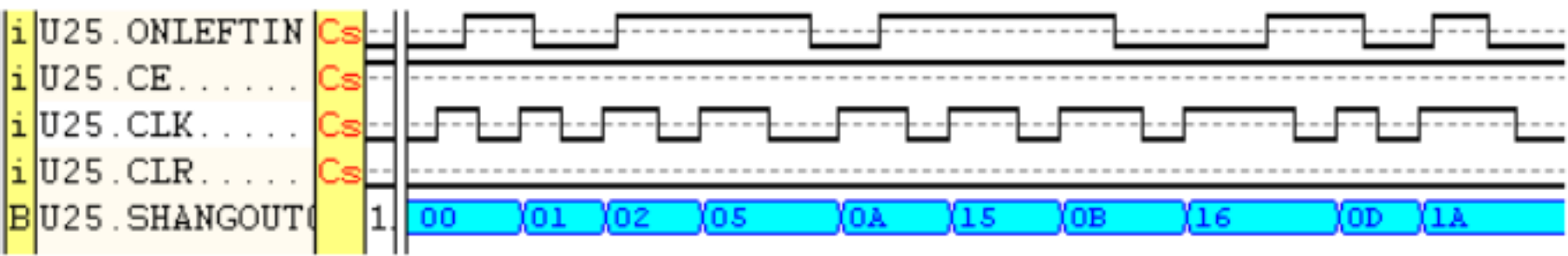


图 2.25 商寄存器模块仿真结果

将商寄存器模块的功能和仿真结果对照可知，商寄存器器模块的仿真结果正确。

2.3 仿真调试

仿真调试主要验证设计电路逻辑功能、时序的正确性，本设计中主要采用功能仿真方法对设计的电路进行仿真。

(1) 建立仿真波形文件及仿真信号选择

功能仿真时，首先建立仿真波形文件，选择仿真信号，对选定的输入信号设置参数，选定的仿真信号和设置的参数如表 2.2 所示：

表 2.2 仿真信号选择和参数设置

输入信号					输出信号		
A	B	CLR	CLK	CE	FUHAOSHANG	YUSHU	
A0-A5	B0-B5	0		1	00/11	S0-S4	Y0-Y5

(2) 功能仿真结果与分析

功能仿真波形结果如图 2.26 所示，仿真数据结果如表 2.3 所示。参照 1.1 的例子，可以看出功能仿真结果是正确的，进而说明电路设计的正确性。

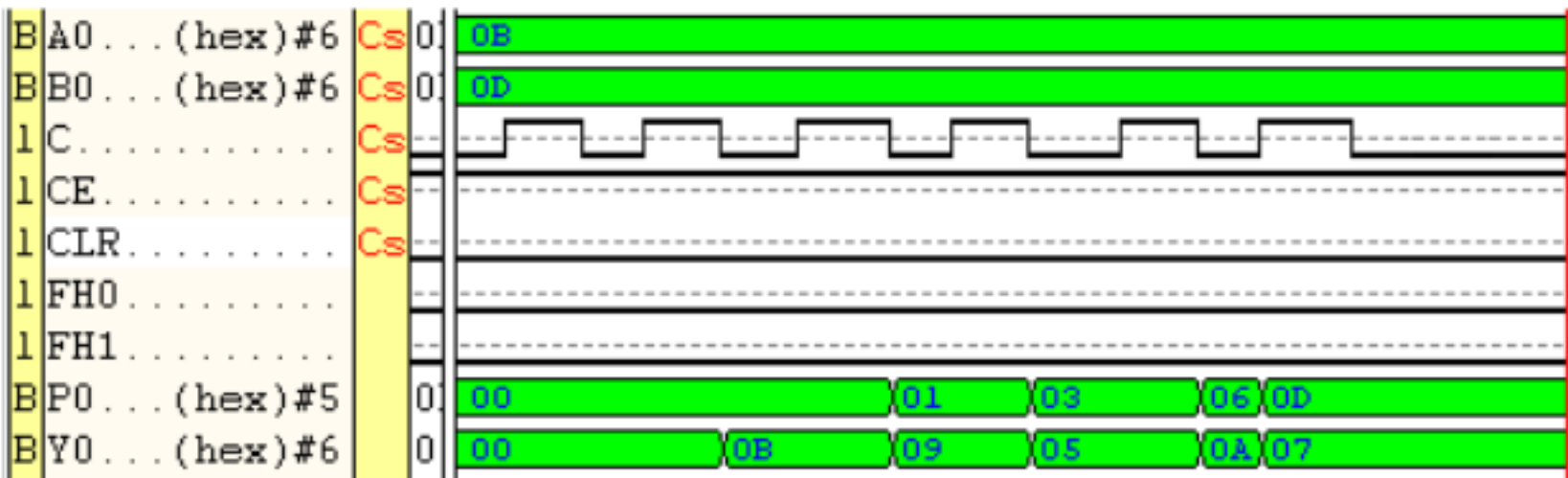


图 2.26 功能仿真波形结果

仿真的结果如下表 2.3 所示：

表 2.3 仿真结果

输入信号					输出信号		
A	B	CLR	CLK	CE	FUHAOSHANG	YUSHU	
001011	001101	0		1	00	01101	000111

第 3 章 编程下载与硬件测试

3.1 编程下载

利用 Xilinx foundation f3.1 的编程下载功能，将得到的 .bit 文件下载到 XCV200 实验板的 XCV200 可编程逻辑芯片中。

3.2 硬件测试及结果分析

利用 XCV200 实验板进行硬件功能测试。原码一位除法器的输入数据通过 XCV200 实验板的输入开关实现，输出数据通过 XCV200 实验板的 LED 指示灯实现，其对应关系如表 3.1。

表 3.1 XCV200 实验板信号对应关系

输入信号	
XCV200 芯片引脚信号	XCV200 实验板
P72	K2：1
P71	K2：2
P213	CLOCK
P96	K0：5
P97	K0：4
P100	K0：3
P101	K0：2
P102	K0：1
P103	K0：0
P81	K1：5
P82	K1：4
P84	K1：3
P85	K1：2

P86	K1 : 1
P87	K1 : 0
输出信号	
XCV200 芯片引脚信号	XCV200 实验板
P178	A5
P184	A4
P185	A3
P203	A2
P111	A1
P110	A0
P78	B7
P93	B6
P107	B4
P108	B3
P109	B2
P124	B1
P125	B0

对于原码为： X 的值为 000101 , Y 的值为 001100 , 那么运算结果如表 3.2 :

表 3.2 输入值与计算结果

输入信号					输出信号		
A	B	CLR	CLK	CE	FUHAOSHANG	YUSHU	
000101	001100	0		1	00	00110	001000

所得到的结果如下图 3.1 所示：

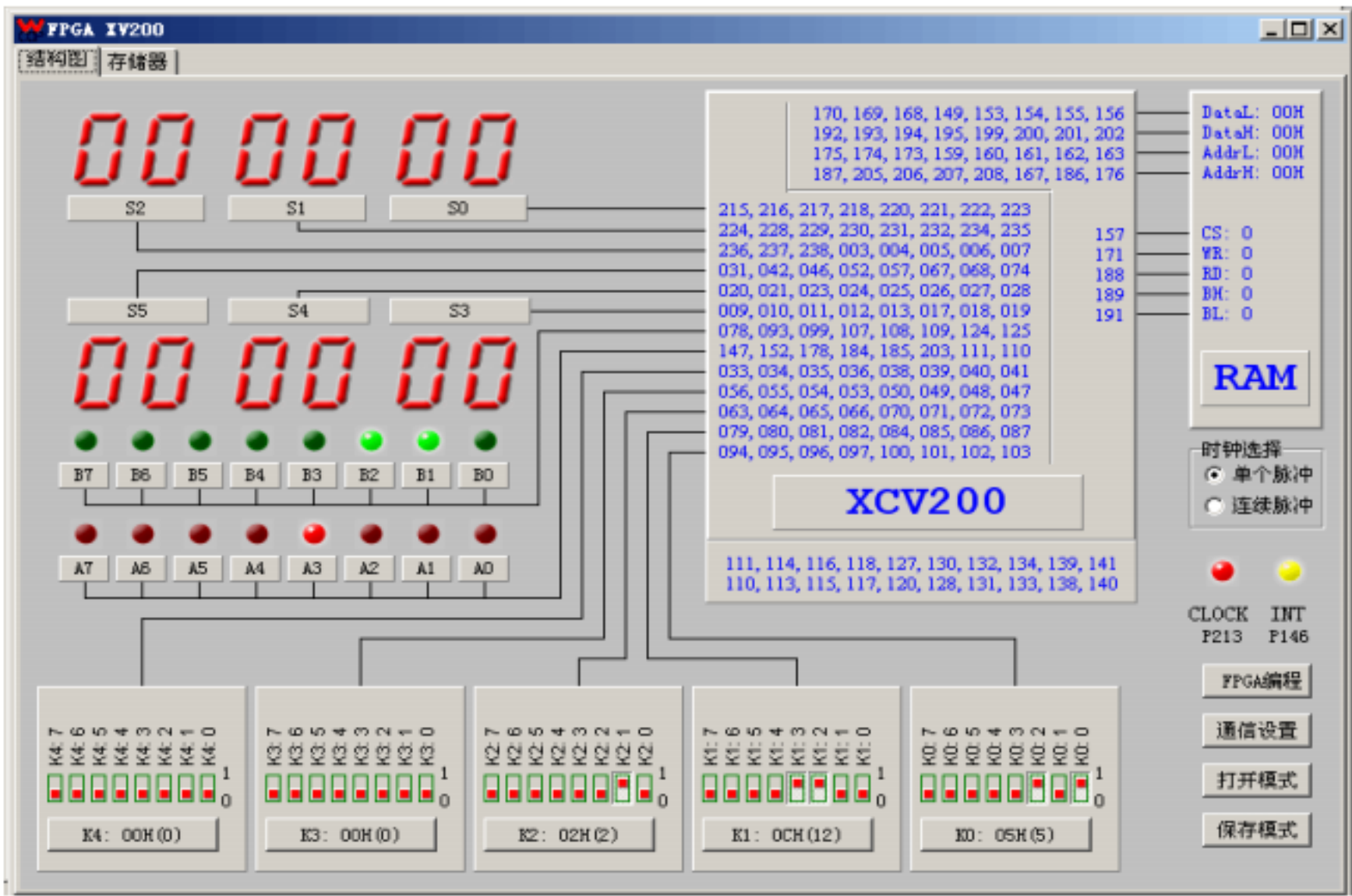


图 3.1 对表 3.2 输入数据所得到的结果

根据实验板上得到的结果和计算出来的表 3.3 的数据一致。

对于原码为：X 的值为 111011, Y 的值为 001101, 那么运算结果如表 3.3 所示：

表 3.3 输入值与计算结果

输入信号		输出信号					
A	B	CLR	CLK	CE	FUHAOSHANG	YUSHU	
111011	001101	0		1	11	01101	000111

所得到的结果如下图 3.2 所示：

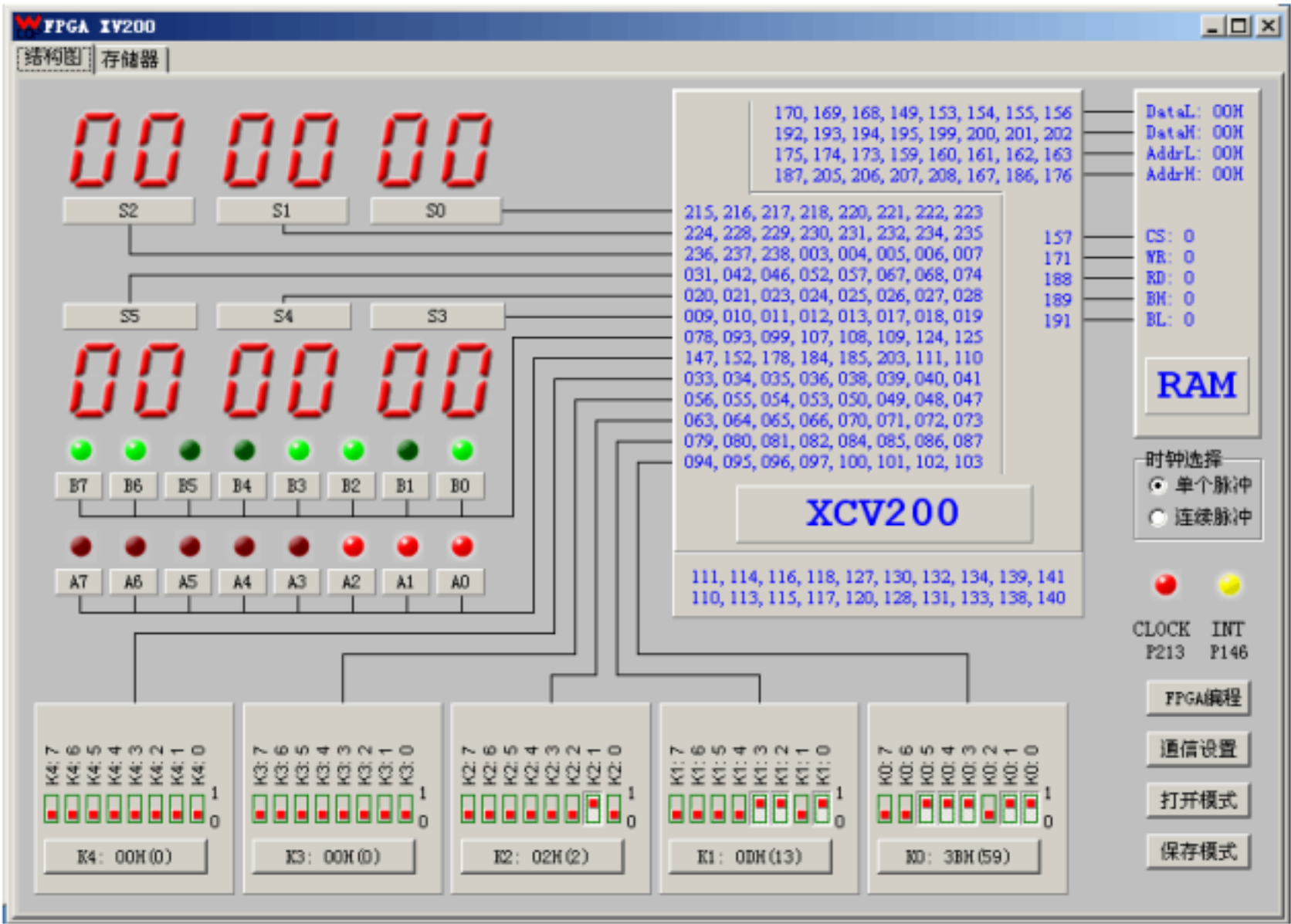


图 3.2 对表 3.3 输入数据所得到的结果

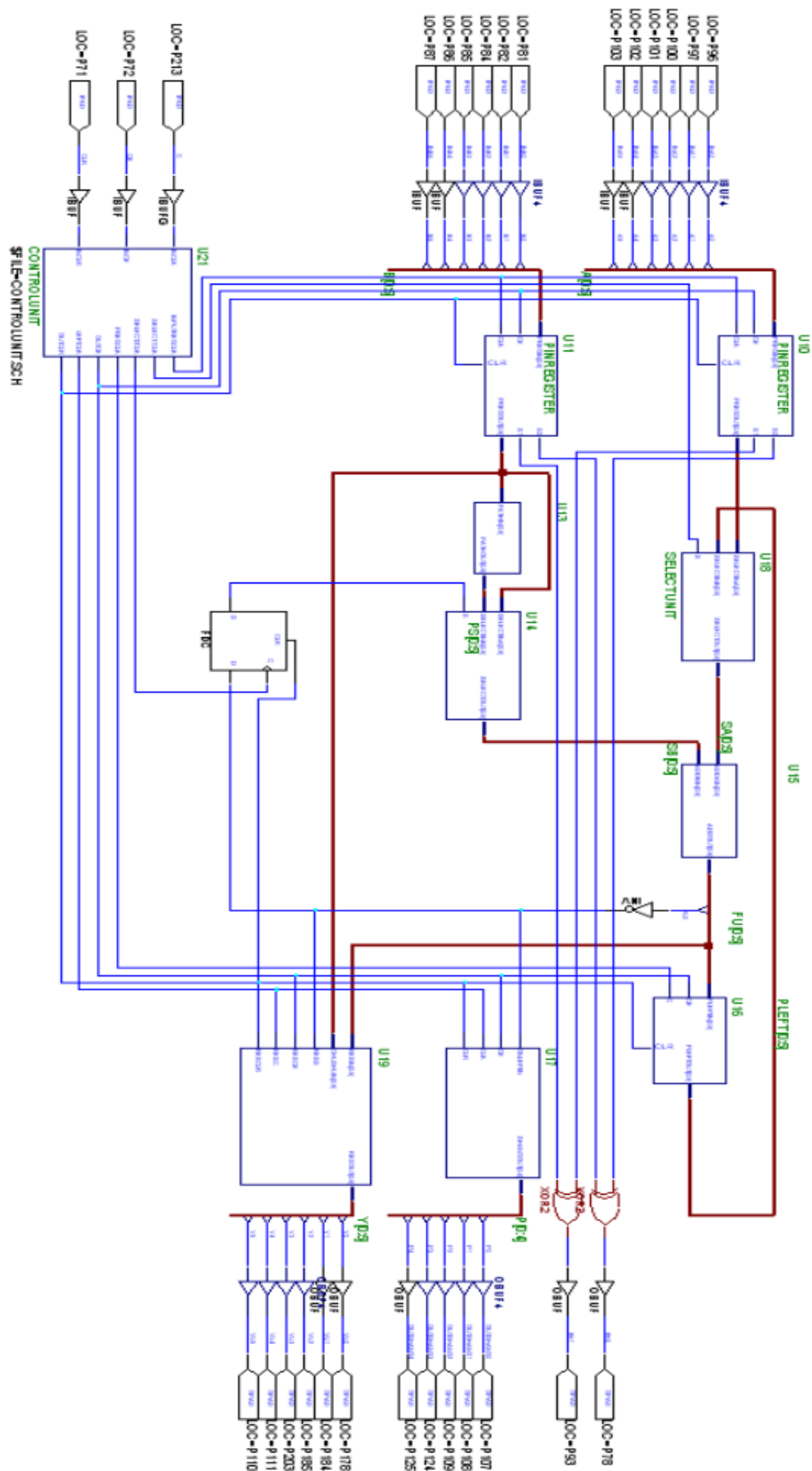
根据实验板上得到的结果和计算出来的表 3.3 的数据一致。

综合以上的实验板所得到的结果说明加减交替法定点原码一位除法器设计成功。

参考文献

- [1] 莫正坤. 计算机组成原理 [M]. 武汉 : 华中理工大学出版社 , 1996
- [2] 范延滨. 微型计算机系统原理与 EDA设计技术 [M]. 北京邮电大学出版社 2003
- [3] 百中英. 计算机组成原理 (第三版) [M]. 北京 : 科学出版社 , 2005
- [4] 李景华. 可编程逻辑器件与 EDA技术 [M]. 北京 : 东北大学出版社 , 2001
- [5] 高明伦编著 . Verilog 与 pc 机接口电路设计 [M] . 合肥安徽科技出版社 , 2002
- [6] 王爱英. 计算机组成与结构 (第 4 版)[M]. 北京 : 清华大学出版社 , 2006
- [7] 江国强 .EAD 技术习题与实验 [M]. 北京 : 电子工业出版社 , 2005 学出版社 , 2006
- [8] 柳春风 . 电子设计自动化 (EAD)教程 [M]. 北京 : 北京理工大学大学出版社 2005

附录（电路原理图）



课程设计总结：

在本次课设过程中，我感触最深的就是我们不仅要学习书本上的内容，还的加强练习动手能力，真正运用知识的地方都是实践过程，在做课设的过程中我了解到了更深层次的计算机内部的系统，包括编译，下载，等很多有趣也很有用的内容，这也是在课设过程中得到的最宝贵的东西，尽管题目很难，但是在老师的帮助下，我慢慢学会了怎么去做这个课设，所以学到了很多东西，当然也询问了很多同学一些问题，这才完成了这次课设。

这次课设对于我来说是很大的财富，我十分感谢这次课设给我带来的帮助的老师、同学也许是对实验的软硬件环境不熟悉，所以在学习工具方面花的时间比真正画原理图的时间要多一些。这使我认识到在平时的时候应该多学一些东西，而不仅仅是把考试课过了就完事，也让我明白了硬件设计和熟练应用软件进行设计的重要性，从初步构思到一步步完善，直到最后完成整个课程设计，让我更好的巩固了自己所学的知识。翻阅图书去学习一些没有接触过的知识，让我更好的了解了这门课程，增强了实践动手能力。总的来说，虽然完成了此次课程设计，但是还没有达到非常理想的地步，还有很多要学习，要实践。我知道我们才刚刚踏入计算机硬件领域，并且还有太多太多的知识有待我们去学习。

指导教师评语：

指导教师（签字）：_____年__月__日

课程设计成绩

