**CSR-FLY实验系统的硬件结构**

1.1 CSR-FLY实验系统的特点

CPU作为计算机系统的核心，是计算机组成原理和计算机系统结构实验中的重要内容。为了给实验学生最大的空间，充分发挥学生的创造力，本实验系统提出了一种新的设计思想，即希望建立一个研究型、设计型的开放式CPU实验平台。在该平台中能够尽可能消除各种限制，从CPU的指令集、指令格式、数据表示到寻址方式、存储方式等全部由学生自由设计，并且能够对各种不同类型的16位CPU进行调试和验证。学生可以在这个开放式CPU设计平台上进行计算机组成原理各部件实验，完全自由的进行CPU内核设计、测试、验证以及整个计算机系统的实验。

在目前的计算机系统设计中，掌握现代设计方法对于每一个学生是至关重要的，计算机设计只有通过电子设计自动化（EDA），即在条件允许的情况下使用流行的EDA设计工具和测试工具才可能实现。通过该系统，不仅使学生深入理解计算机组成原理，并且能够通过自己动手设计CPU来学习流行的设计方法和设计工具，提高实际设计能力。通过掌握最新的设计工具和测试工具，还可以解决实际计算机系统设计中出现的各种软硬件问题。可编程逻辑器件是计算机应用领域广受技术人员欢迎的器件，它不仅是计算机实验中的重要器件、也已经成为样机试制中的最佳选择。

CSR-FLY开放式CPU实验教学系统具有很好的通用性和灵活性，能够提供足够的硬件资源，尽量不依赖于硬件实现特性。它允许实验人员从指令系统构架确定开始，自由的进行指令集、指令格式、寻址模式、数据通路、控制通路的设计，到整个CPU的实现。可以满足不同CPU设计人员的各种设计思路和实现方法，能够允许自由的设计任何构架的16位CPU。

在CSR-FLY上首先需要用硬件描述语言设计一个“CPU”，通过软件进行模拟测试，最后下载到FPGA中在CSR-FLY实验系统中进行实际的调试运行，CSR-FLY提供调试的各种手段。在计算机组成原理部件实验中，可以设计实现计数器、移位器、ALU等功能部件；在综合实验中，即可以设计实现无流水无Cache的最基本最简单的CPU也可以设计实现既有流水又有Cache的复杂的CPU。设计完成后将CPU的文件由USB BLASTER下载线下载到FPGA芯片中，通过CSR-FLY实验系统来提供CPU的实际运行、调试和测试环境。

如果在设计CPU时没有任何的约束条件，则很难自动测试，约束条件太多，则影响设计。系统为测试CPU设置了最小的约束条件，某些通用的IO引脚设置为寄存器选择、寄存器输入输出、外部RAM存储器地址选择和存储器数据的输入输出等。这些约束条件不会影响CPU设计的灵活性，只是在设计完成以后的引脚配置时应按CSR-FLY的要求配置芯片的管脚。

1.2 CSR-FLY的功能概述

本节中，我们将从功能上对CSR-FLY进行更为具体的描述。由于CSR-FLY可以支持简单的功能部件实验，也支持复杂的CPU设计实验。以下将从从复杂CPU设计的角度进行阐述，对于功能部件的实验，其原理及操作要相对简单些。

首先，CSR-FLY支持CPU的单步调试和连续运行。CPU的运行是通过执行预先写入到外部存储器中的目标程序来进行的，即CPU从外部16位存储器的零地址开始读入指令并逐条执行。这实际上是CSR-FLY的基本功能，也是实现所有其它功能的基础。

为了支持用户对外部存储器执行预写入操作，我们提供了通讯软件，它在PC的Windows（Windows2000和WindowsXP）环境下运行，除了支持对外部存储器的读写操作之外，还支持对CPU的状态监控和对正在执行的程序调试，比如支持对程序设置断点，以及对地址总线和数据总线进行监控。

对于计算机组成原理和计算机系统结构课程的综合实验来说，CSR-FLY还具有以下功能。

1．对CPU的调试功能

上面已经提到，除了支持CPU的运行和外部程序的下载之外，CSR-FLY还支持各种常用的程序调试功能。目前能够进行调试的程序为汇编级的代码，再加上对于CPU总线和内部寄存器等数据的实时监测，将为CPU的测试提供很大方便。

2．支持各种类型的指令系统

CSR-FLY支持用户下载到CPU中的各类指令系统，只要用户能够提供关于该指令系统的相应描述，CSR-FLY就可以为用户生成针对这种指令系统的目标代码，并进行测试。这样用户在设计指令系统时就不会受到其他方面的限制，获得了很大的灵活性。从实验平台的角度来看，它能够为支持任意指令系统的CPU提供测试功能，其灵活性也得到了充分体现。

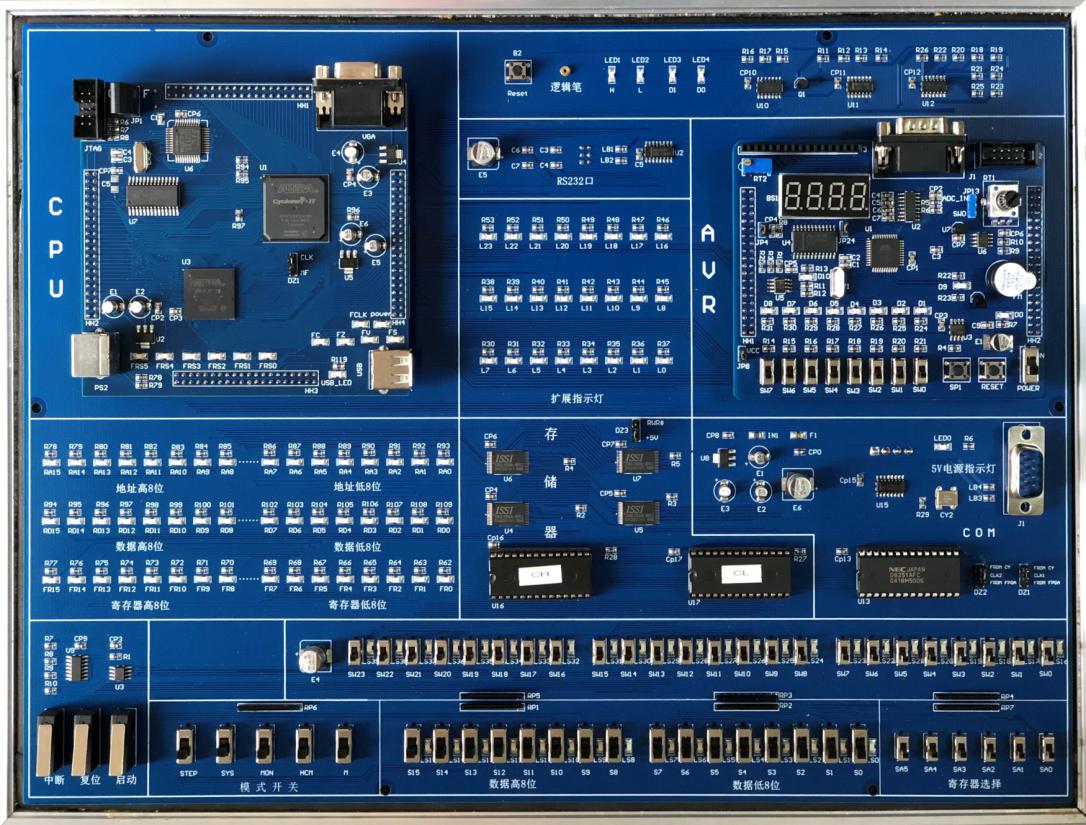
1.3 CSR-FLY实验系统的硬件结构

CSR-FLY开放式CPU实验教学系统的硬件主要分为两部分：子板和实验平台。子板插在实验平台上，是可以更换的。子板的主要部分由FPGA、CPLD和单片机器件组成。FPGA器件是现场可编程器件，通过写FPGA内部的SRAM IS62LV256对FPGA内部的逻辑电路和互连进行配置（重构），完成指定的逻辑功能。FPGA和ispCPLD不同，ispCPLD也以重新配置，它是通过写ispCPLD内部E2PROM方式进行配置，因此断电后，配置内容可以保存；FPGA由于采用写内部SRAM IS62LV256方式进行配置，因此断电后配置的内容会丢失。如果要使FPGA配置的内容断电后不丢失，必须在FPGA外部增加一片E2PROM。由于CSR-FLY主要用于CPU实验，因此子板上没有配置E2PROM，只配备了一个JTAG下载插座。采用这种主板―子板结构增加了CSR-FLY开放式CPU实验教学系统的灵活性。当需要用新型号的FPGA器件做实验时，只需设计装有新FPGA器件的子板即可。

FPGA的部分I/O通过双排座引到主板上，提供足够的通用性和灵活性。JTAG接口提供对FPGA的在线下载，即在PC机上通过QuartusII将设计好的CPU下载到子板上的FPGA芯片中去。下载时，下载电缆的一端接PC机的USB口，另一端接子板的JTAG插座。

实验平台的作用是与调试软件通讯软件一起，对用户设计的CPU进行调试。实验台上有4片SRAM IS62LV256芯片，它们与CPU构成一个16位计算机。另外实验台上设置了许多拨动开关和指示灯，用于设置和显示CPU的各种工作状态。





**逻辑笔电路**，可以测试电路中的高、低电平和时钟脉冲

**FPGA\_CPU**,做系统结构时，接收来自内存的指令，产生控制信号，控制指令执行，并将执行的结果写入寄存器组；做组成原理部件实验，接收来自开关的输入，执行后的结果回送到LED灯

16位内存地址指示灯

RA15~RA0

16位内存数据指示灯

RD15~RD0

16位寄存器内容指示灯

FR15~FR0

**CPLD,**接收来自单片机和FPGA的数据地址以及读写信号，按照模式开关设置，进入不同的CPU运行模式，并送出相应的数据、地址和内存读写信号，从内存读取指令，运行测试程序

**MCU**,完成跟上位机的软件通讯功能，可以下载测试程序；通过单步、连续、设置断点等方式调试CPU并将运算结果回送到CSR软件界面

**AVR单片机**，8位CPU及其应用电路，

**LED指示灯**，L23~L0

**串口通讯电路**，单片机通过串口通讯电路跟软件连接，进行通讯

**内存RAM/ROM**，存储测试程序，单片机将编译好的测试程序下载到指定内存单元

**寄存器选择开关**，SA5~SA0

**拨动开关**，SW23~SW0,提供部件实验需要的数据或控制信号

**16位数据开关**，SD15~SD0

**模式开关**，STEP SYS MON HCM M

复位、启动按键

1.3.1实验平台上的指示灯

实验台上的指示灯分为两大部分：FPGA用户I/O指示灯和各种总线数据指示灯。

总线指示灯主要分成三大类，位于实验台的左侧中部。

1. 存储器地址指示灯RA15~RA0

用于指示存储器的地址。跟CPLD的管脚相连，内存地址可能来自于单片机，用于单片机控制的下载和程序执行；也可能来自于FPGA，用于CPU的执行和测试，由模式开关决定工作方式和来源。

1. 存储器数据指示灯RD15～RD0

指示存储器的数据（写入或者读出）。跟CPLD的管脚相连，内存数据可能来自于单片机，用于单片机控制的下载和程序执行；也可能来自于FPGA，用于CPU的执行和测试，由模式开关决定工作方式和来源。

1. 寄存器数据指示灯FR15～FR0

指示CPU中寄存器的数据。

除了总线指示灯外，还有寄存器指示灯和标志位指示灯以及其他用户I/O指示灯，前两种在核心板上，位于板子的下方。

1. 寄存器选择指示灯FRS5~FRS0

指示寄存器选择（编址）的状态。

1. 标志位指示灯FS、FV、FZ、FC和其它指示灯/FWR、/FDCLK

S、V、Z、C分别指示CPU运算时产生的符号标志、溢出标志、零标志和进位标志。/FWR指示CPU产生的对存储器的读/写信号，高表示读，低表示写。FCLK指示实验平台产生的CPU的主时钟。

1. 用户I/O指示灯L23~L0

在实验平台中部偏上的位置，共24个用户I/O指示灯，指示FPGA芯片用户I/O的状态。这些指示灯用于FPGA芯片作为设计一般数字电路、计算机组成原理部件等实验，计算机系统结构课程实验中通常不使用。

1.3.2 实验平台上的开关

1. 模式选择开关

模式选择开关位于实验平台下部中偏左。用于选择CSR-FLY常用的工作模式。

1. **数据开关S15～S0**

数据开关SD0～SD15位于实验平台的下部中间。这16个开关的作用是给CPU的总线置数。

1. **寄存器地址开关SA5～SA0**

寄存器地址开关SA0～SA5位于实验平台的下部右边。这6个开关的的作用是指定CPU中的寄存器编址号，被指定的寄存器的内容在实验平台上的指示灯FR15－FR0显示出来。

**4．扩展备用开关SW23～SW0**

扩展备用开关位于实验平台下部，直接跟FPGA相连，提供数字电路和组成原理部件实验所需的信号。

**1.4** CSR-FLY的工作模式

为了更好地对用户设计的计算机组成部件和CPU进行调试，CSR-FLY支持三种基本调试模式： CPU独立运行模式，CPU附加外部RAM运行模式和单片机控制CPU调试运行模式。

1．CPU独立运行模式

这种模式适用于调试计算机组成部件的实验。由于计算机组成部件不是一个完整的CPU，因此无法构成计算机。在这种方式下，单片机不应同时也无法对CPU的运行进行监控，CPU只接受实验平台SW23－SW0作为输入，运行结果可通过实验平台上的指示灯L23－L0显示出来。单步调试时可使用按单脉冲按钮产生的正脉冲作为主时钟。

2．CPU附加外部RAM运行模式

在这种模式下，CPU和实验平台上的存储器（4片SRAM IS62LV256）构成一个16位的计算机。实验平台上的存储器作为CPU的存储器，运行计算机程序和存放数据。在这种模式下，单片机不参与对计算机的监控。调试之前，16位计算机的测试程序通过单片机（在PC机的通讯软件的指挥下）预先下载到外部RAM中。测试程序运行结束后，如果存储器中存放有测试程序运行的部分结果，由单片机在PC机的通讯软件的指挥下读出RAM中存放的运行结果，显示在PC机的屏幕上。如果测试程序运行结束后，存储器中没有存放有测试程序运行的部分结果，则不必从存储器中读出数据。这种模式支持单步调试，单步调试时使用单脉冲按钮产生CPU的主时钟，使用实验平台上的指示灯显示中间结果。这种模式也支持连续运行，但是如果要连续运行，CPU设计时必须设计停机指令。在调试过程中，查看CPU中寄存器（如程序计数器PC、指令寄存器Ireg和用户设计的通用寄存器）的内容时，应先在寄存器地址开关SA5－SA0上设置寄存器的地址，指定地址的寄存器的内容在寄存器内容指示灯FR15－FR0上显示出来。按CSR-FLY的规定，程序计数器PC的地址位62，指令寄存器的地址为63，通用寄存器的地址由用户自己确定。调试测试程序的过程中，各种运算的标志位FC（进位）、FZ（结果为零）、FS（符号）和FV（溢出）的值在核心板相应的指示灯上观察。

**3．单片机控制CPU调试运行模式**

这是最常用的一种模式。在这种模式下，CPU和实验平台上的存储器（4片SRAM IS62LV256）构成一个计算机，实验平台上的存储器作为CPU的存储器，运行实验计算机程序和存放数据，这是与CPU附加外部RAM运行模式相同的。与CPU附加外部RAM运行模式主要不同之处在于，在CPU附加外部RAM运行模式下，单片机不参与对实验计算机的监控；在单片机控制CPU调试运行模式下，单片机参与对实验计算机的监控。在这种模式下，CPU的运行及调试完全由单片机来控制，用户可以通过PC监控程序向由单片机组成的控制电路发出控制指令，从而实现程序断点设置、CPU内部数据监测、程序连续运行等多种调试功能。在这种模式下，CPU可以单步调试，也可已连续调试。单片机控制CPU调试运行模式与CPU附加外部RAM运行模式不同的另一点是，在单片机控制CPU调试运行模式下，CPU的时钟是由单片机提供的；在CPU附加外部RAM运行模式下，CPU的时钟是由按单脉冲按钮产生的单脉冲提供或者由实验平台上的时钟电路提供。在程序调试过程中，CPU中各寄存器（包括程序计数器、指令寄存器和标志寄存器）的值由PC机的通讯软件在屏幕上显示出来；实验平台上的指示灯也可以显示出存储器的地址和内容。在CPU测试程序运行之前，首先在PC机的通讯软件指挥下通过单片机将测试程序装入到存储器中去。测试程序运行结束后，如果有存放于存储器中的部分运行结果，则需要由单片机在PC机的通讯软件指挥下从存储器中取出来，显示在PC机屏幕上。

模式开关表：

|  |  |  |
| --- | --- | --- |
| 开关 | 代表的意义 | 备注 |
| SYS | ‘1’- 系统结构；‘0’- 下载或读RAM； |  |
| HCM | ‘1’-平台（ST/MF）；‘0’- 联PC机（MCLK）； | SYS=1 |
| STEP | ‘1’-单步（ST）；‘0’- 连续（MF）； | SYS=1 and HCM=1 |
| MON | ‘1’-带监控（MON+RAM3/4）；‘0’- 无监控（RAM1/2）； | 前提：SYS=0 |

【未提及的开关值默认为0】

模式1：SYS=1，MON=1，CPU带监控运行模式；

模式2：SYS=1，MON=0，HCM=1，STEP=1 系统结构手动单步（跟单片机无关，外部提供时钟）；

模式3：SYS=1，MON=0，HCM=1，STEP=0 系统结构手动连续（跟单片机无关，外部提供时钟）；

模式4：SYS=1，MON=0，HCM=0 系统结构CPU无监控联机模式（单片机控制运行模式）；

模式5：SYS=0，MON=0，读写（下载）RAM；

模式6：SYS=0，MON=1，读写（下载）ROM；

模式7：SYS=0，EDA实验（PS2、VGA）；

模式8：SYS=0，STEP=1，组成原理逻辑部件实验；

模式9：SYS=0，STEP=0，组成原理时序部件实验；

以上为实验箱常用的工作模式，在系统结构实验中，最常用的模式是：模式5（下载读写RAM）和模式4（通过单片机控制CPU运行），即通过软件，将测试程序下载到内存芯片中，并通过单片机和通讯软件交互，可以单步，连续，设置断点来运行程序，运行的结果可以通过软件的寄存器内容和内存地址的结果来观察，测试所设计的CPU是否正确。

1.5 CPU的一般实验步骤

本节简单介绍一下CPU的设计和实验步骤。

1．用VHDL语言编写设计方案，并将设计编译、连接、适配，形成.sof的文件

在进行CPU的实验之前，首先在Altera公司的EDA软件Quartus Ⅱ下用VHDL语言编写CPU，基于FPGA的CPU设计流程如下：用户按照自己的设计方案，用VHDL语言（或者其他EDA设计语言）对CPU的功能进行描述；在这一步完成之后，整个设计完成后在Quartus进行编译、连接和适配工作，形成.sof形式的文件。上述步骤成功以后，指定芯片引脚，重新进行编译、连接和适配工作，形成新的.sof形式的文件。

2．编写规则文件

规则文件是用户自己指定的汇编指令格式文件。由于在CSR-FLY系统中指令集是用户自己定义的，因此需要用户自己编写规则文件。规则文件以行为单位。汇编器在初始化的时候会逐行解释规则文件，生成指令表和部分符号表。如果用户进行计算机组成部件设计，由于没有测试程序，因此不需要规则文件，这一步骤可以跳过。规则文件在任何文本编辑器下生成即可。

3．编写测试程序

如果做CPU设计实验，由于CPU和实验平台上的存储器构成16位实验计算机，编写测试程序是必须的。测试程序用于检验CPU设计的正确性。必须按照规则文件指定的格式编写测试文件。如果用户进行计算机组成部件设计，由于没有测试程序，这一步骤可以跳过。

以上三个步骤是正式实验之前的准备工作。

4．将PC机和CSR-FLY连接

将PC机和CSR-FLY连接就是做两件事情。一是将下载电缆的一头插到PC机的USB口上，将下载电缆的另一头插到CSR-FLY子板上的下载插座（JTAG）上。二是将RS232通讯电缆的一头插在PC机的USB口上，将RS232通讯电缆的另一头插到CSR-FLY实验箱背面的9针插座上。

5．打开CSR-FLY实验系统的电源

打开电源后检查实验平台上的＋5V指示灯是否点亮。如果指示灯点亮，表示电源系统正常。

6．选择实验的模式

按选定的调试模式设置好模式选择开关。

7．按复位按钮，使实验平台处于初始状态

8．将CPU设计下载到CSR-FLY子板上的FPGA中

在PC机上启动EDA软件Quartus，将.sof文件形式的CPU设计下载到CSR-FLY子板上的FPGA芯片中，构成一个物理上的CPU。

9．将测试程序装到存储器中

在PC机上启动通讯软件，将测试程序装到实验平台上的存储器中。在CPU独立调试模式中，没有存储器，也没有测试程序，这一步骤跳过。

10．根据选择的调试模式调试程序。