基于FPGA的CPU设计论文

白一泽 龚雪 彭槐 刘丽 曾旭东

**摘要：**本文介绍了一种利用cyclone ep1c3t144c8n fpga为核心，用VHDL语言编写的一个简易的CPU，该CPU主要包含4部分：运算器，控制器，数据通路结构，与外部的连接。还有指令系统，该指令系统包含了大部分的指令，能实现各种运算以及简单的外部控制（按键控制led流水灯等，各种运算结果有LCD1602和数码管显示）。利用状态机来尽可能地减弱和消除时序逻辑电路的竞争和冒险（毛刺现象）。

**关键词:**FPGA，VHDL，指令系统，状态机。

1、设计背景

1.1研究背景和发展现状

随着电子技术、计算机技术、智能控制技术的飞速发展，产品的智能化和小型化越来越成为人们关注的热点。各种FPGA的产品逐渐的出现，因为要想更好地研究和吃透底层，FPGA是无可厚非的地位，但囊括所有内容的项目CPU最为适合。因此，CPU的研究是非常有意义的，具有很大潜在市场价值的。

CPU，全名中央处理器（CPU，Central Processing Unit）是一块超大规模的集成电路，是一台计算机的运算核心（Core）和控制核心（ Control Unit）。它的功能主要是解释计算机指令以及处理计算机软件中的数据。

1.2 研究目的和意义

随着人们物质文化生活水平的不断提高，越来越多的开始意识到硬件市场的发展前景，因为FPGA上至航天技术下至智能玩具，FPGA牵涉内容很多，CPU可以囊括其所有内容。因此，CPU的研究不仅具有很大的现实意义，还具有极为广阔的应用前景和市场价值。

1.3 研究内容

本系统设计的CPU具有传统8086CPU的基本功能，比如基本运算和简单操作外部设备的功能。

整体设计可以分为如下几个模块，运算部件、寄存器组、微命令

产生部件、时序系统等主要部件，这些部件由CPU内部的总线连接起来，实现它们之间的信号交换。其中运算部件和一部分寄存器属于传统运算器部分，另一部分寄存器、微命令产生部件、时序系统等属于传统控制器部分。实验测试结果满足要求。

2、总体方案设计与各模块设计

2.1 总体方案设计

本cpu是8位的（每个设计实体是16位的），cpu的含义是取出指令并执行指令的部件。其主要功能有：数据加工 ----ALU. 算术/逻辑运算；指令控制：指令执行的顺序控制;操作控制: 产生各种操作信号;时间控制: 控制操作信号的发生时间;其组成是控制器和运算器以及一些寄存器。寄存器主要有：PC(Program Counter)----程序计数器AR(Address Register)---地址寄存器DR(Data Register)----数据缓冲寄存器IR(Instruction Register)-----指令寄存器AC(Accumulate Count)---累加寄存器PSW (Program Status Word)程序状态字。控制器主要有：PC (Program Counter)----程序计数器IR (Instruction Register)-----指令寄存ID (Instruction Decoder)---指令译码器OC (Operate Controller)---操作控制器TG (Timer Generator) ---时序发生器。运算器当然是ALU。该cpu能够实现基本的运算和对外围设备的控制。

根据题目的要求，确定如下方案：先将个模块也就是各实体写好，然后再在顶层实体里面统一的合并起来，也就是例化。并结合相关的状态机，达到最终的效果。

。系统整体方框图如图2-1所示。

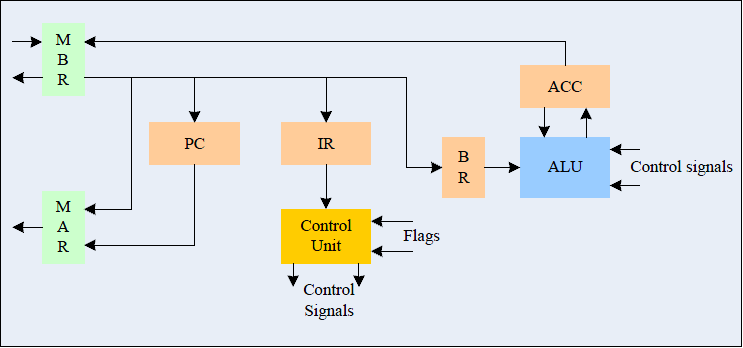
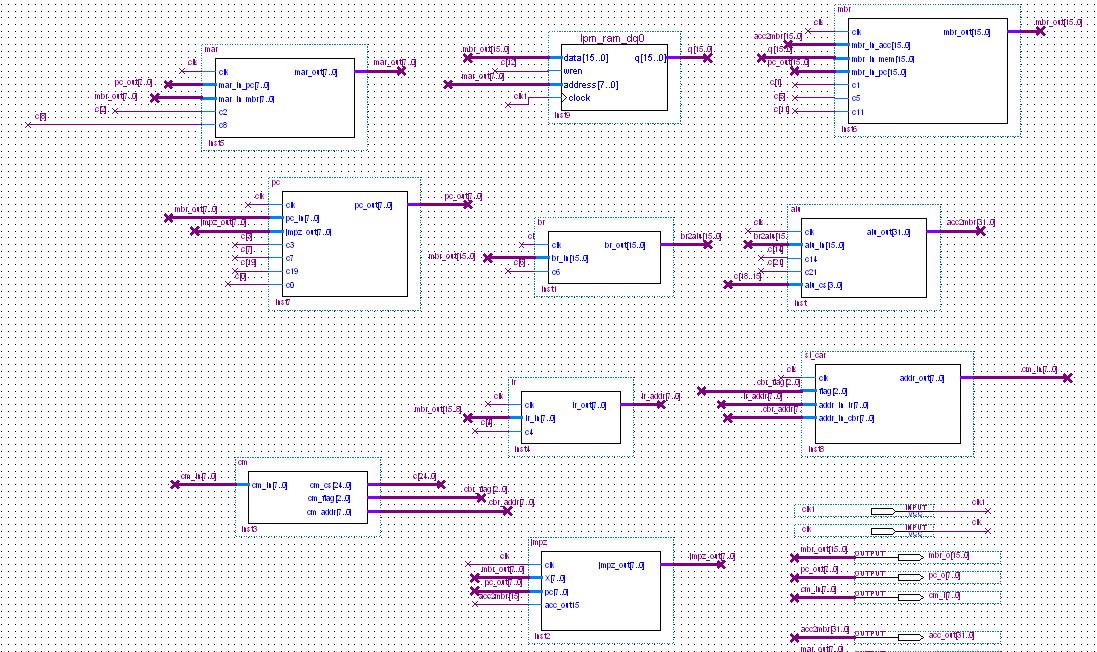


图2-1 系统总体设计框图

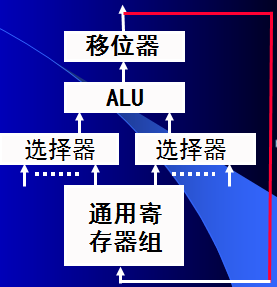
2.2 各模块设计（设计实体）

将下图中的各个实体一一地分析和设计。

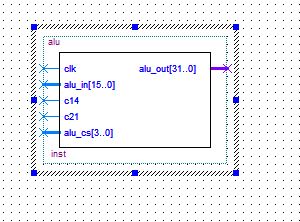


2.2.1 运算部件（ALU）

对数据进行加工处理。

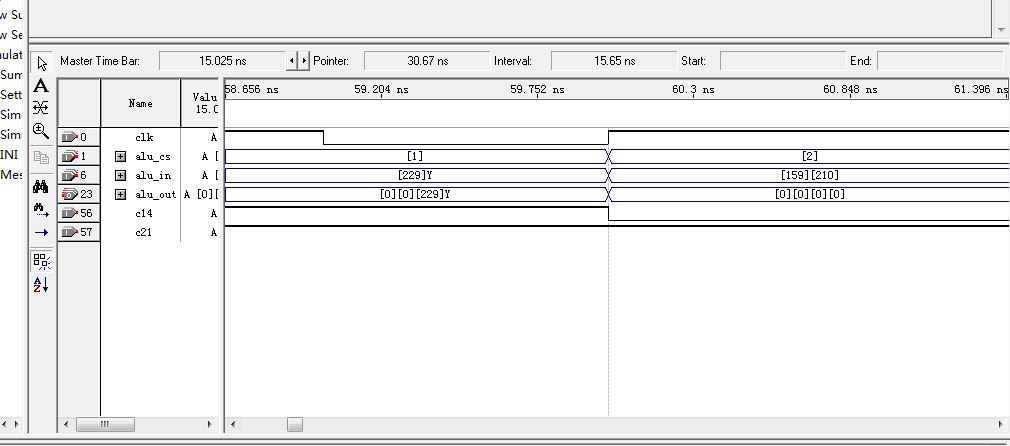


其原理图如下：



Cs用来控制其做什么运算，c14，c21用来控制做的是直接不做运算过去还是做运算的使能端。Alu-in是接收数据输入端。

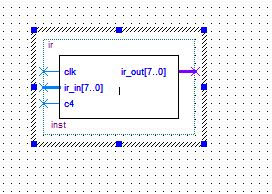
仿真结果图如下：



2.2.2 指令寄存器IR

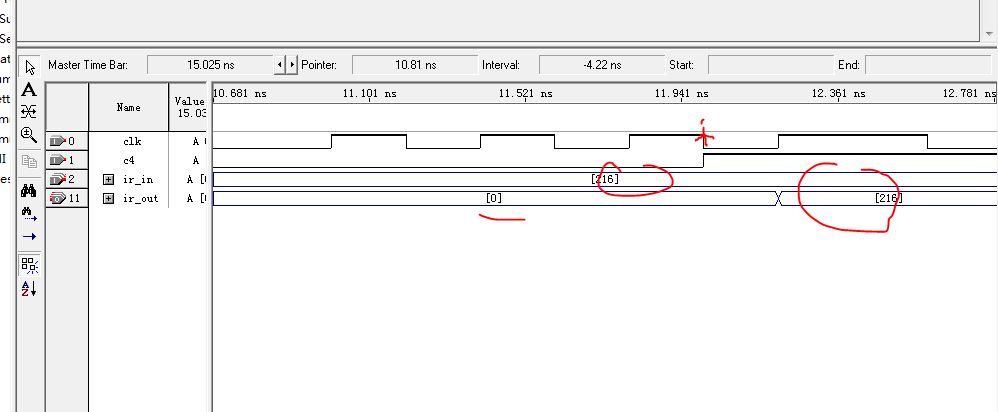
用来存放当前正在执行的指令。控制器根据其内容（操作码以及地址信息）产生所需要的各种微命令。为了提高计算机的处理速度，现代计算机一般将指令寄存器扩充为指令队列（指令栈）。形成了指令流水线处理方式.。指令流水线:一次预取多条指令，增加指令间执行的衔接速度和并行处理能力。该实体如上图所示，设计了三个输入，一个输出，有单独的时钟和使能端，这样能更好地控制。

其原理图如下：



Ir\_in是输入端，c4用来控制接不接受输入，

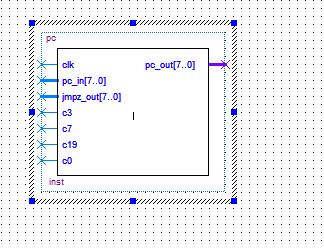
其仿真结果图如下：



2.2.3 程序计数器PC

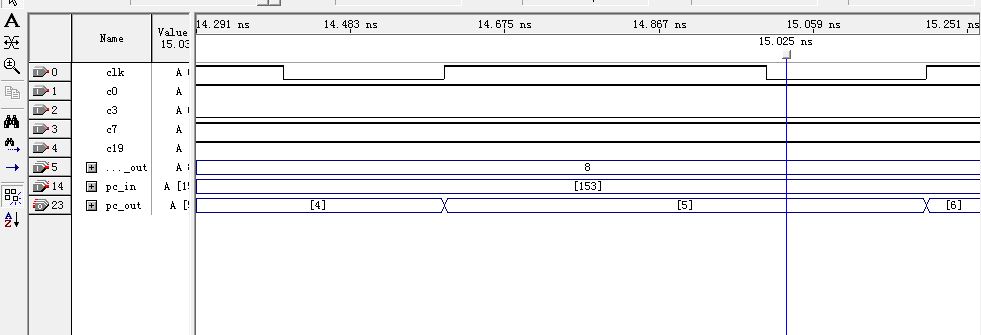
也叫指令计数器或指令指针。程序顺序执行时，每当从主存中取出一条指令后，PC内容就自动增量计数(pc=pc+1)，指向下一条指令。程序发生转移时，将转移地址送入PC，使其指向转移目的地的指令地址。该实体的输入不止一个，因为得考虑到转移指令，所以是要进行选择输入的。

其原理图如下：



由于pc输入的来源不同所以也需要使能端来控制，如图，c3,c7,c19,c0 都是用来控制pc来源的，分别控制pc的下一条指令地址是在内存还是跳转，还是寄存器里面。

其仿真结果图如下：



2.2.3程序状态字寄存器PSW

记录现行程序的运行状态和指示程序的工作方式。PSW包括特征状态位和方式控制位两部分。特征位：进位位C、溢出位V、零值位Z、符号位S、奇偶位P等。方式控制位：跟踪位T、中断允许位I、程序优先级字段、工作方式字段等。

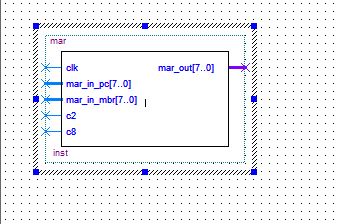
2.2.4显示模块

由于该CPU要显示运算结果和其他一些内容，内容较多。数码管使用简单，价格低廉，但一个数码管只能显示一个数字，要显示多位数据时要使用多个数码管，这就增加了硬件电路的复杂度和额外功耗。故不予考虑。液晶显示电路简单，使用方便，一个液晶显示器就可以同时满足要求，且更加直观明了。

2.2.5 地址寄存器MAR

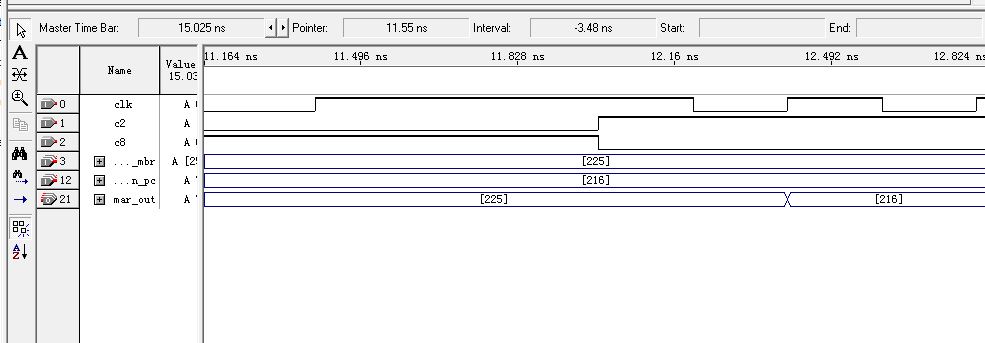
CPU访问主存时，暂时存放所访问单元地址。CPU取指令时，先将PC的内容送入MAR，再由MAR将指令的存放地址送往主存译码。CPU存取数据时，将数据的存放地址先送到MAR，再送往主存进行地址译码。该实体也是选择输入，因为其数据来源不止一处，要么是pc要么是mbr。

其原理图如下：



其数据来源有两处，所以得需要两个使能端类控制来源，如图，mar\_in\_pc表示数据来源地为pc，mar\_in\_mbr表示mbr，c2，c8分别为两个使能端。

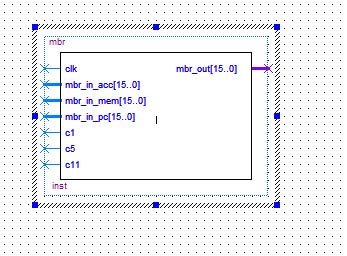
其仿真结果图如下：



2.2.6 数据缓冲寄存器MBR

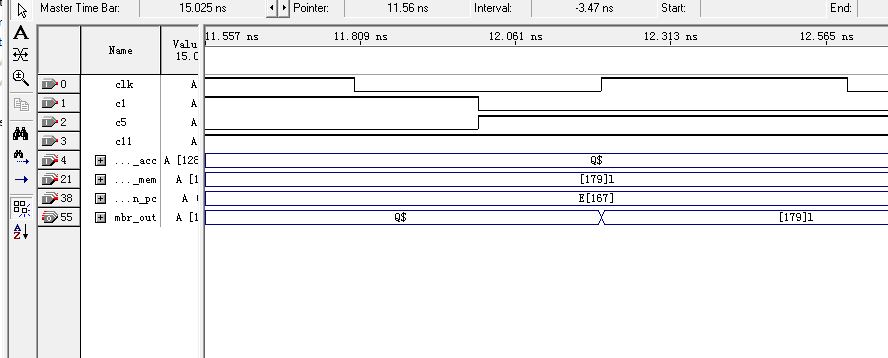
存放CPU与主存之间交换的数据。由CPU写入主存的数据先送入MBR，主存地址译码找到指定单元后，再将数据从MBR送入到该单元。从主存中读出的数据，先送入MBR，然后再送到指定的CPU内部某寄存器。该实体类似于MAR，只是它存放的是数据以及指令，而MAR存放的是地址。MAR和MBR是连接CPU与主存的桥梁，它们可使CPU与主存之间的数据通路变得比较单一。这两个寄存器不能编程访问，即对用户透明。

其原理图如下：



Mbr的来源有三处，不用说，所以也得需要三个使能端，如图所示即可。

其仿真结果图如下：



2.2.7控制器

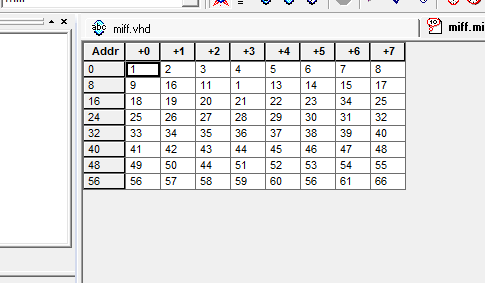
在计算机内部，程序的执行过程体现为信息的传送，即在计算机中形成控制流和数据流。这些信息的传送需要靠各种微命令来控制。CPU中设置了微命令产生部件，它根据指令提供的信息产生出各指令所需要的各种微命令序列。微命令产生部件可以是由一些组合逻辑电路组成，也可以由专门的存储逻辑组成。它们分别对应了组合逻辑控制方式和微程序控制方式。一条指令的读取与执行,需要分步:① 取指令② 取源操作数③ 取目的操作数④ 执行运算操作⑤ 存放结果每个阶段再分为若干步, 微命令也是分步产生。这个的话语下面的状态机对应了起来，因为一条指令的过程也就是这5步。

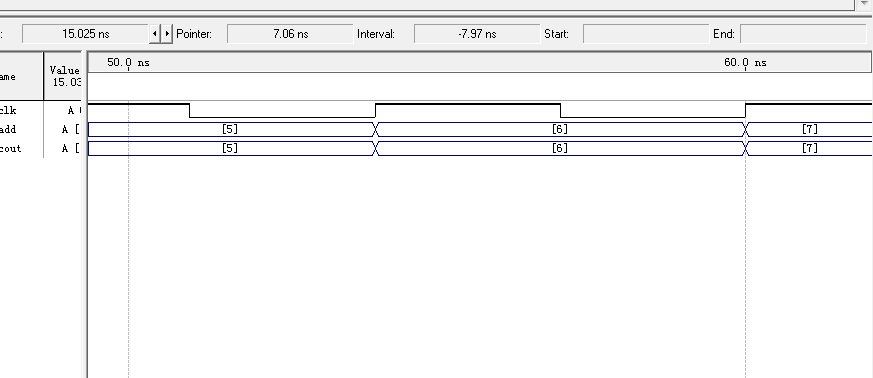
2.2.8时序系统

计算机能够有条不紊地自动执行程序，很重要的一个原因就是它有一个提供精确时间信号的时序系统。时序系统产生计算机所需要的各种时间信息，如周期、节拍、工作脉冲等。这些信号的产生依据是一个频率稳定的振荡器，以及计数分频器或倍频器。我们的FPGA芯片是48MHZ的，所以考虑到方便，所以分频为100k。

2.2.9内存ROM和RAM

用来存放指令和数据的。这里我们采用FPGA内部的宏模块LPM，宏模块是FPGA内部的独立的，里面的宏模块很多，RAM和ROM只是其中两种，不用的话就浪费了。当然我们也可以自己写一个实体存放数据，但是用宏模块的话方便一些而且速度更快些，因为用LPM的话，初始化数据我们可以用mif文件，而数据比较多的时候我们可以用其它高级语言生成mif。





2.2.10 输入模块

用键盘输入和数据和运算，当然一开始已经放有一个运算的程序在里面。

3、指令系统设计

也就是该CPU能做的运算。给每个指令相应的编码，到时候译码是高8位，为指令码，低8位为数据地址。

部分指令地址以及对应指令的操作

|  |  |  |
| --- | --- | --- |
| **指令** | **地址** | **描述** |
| STORE X | 10 | ACC🡪 [X] |
| LOAD X | 20 | [X]🡪ACC |
| ADD X | 30 | ACC+[X]🡪ACC |
| SUB X | 40 | ACC-[X]🡪ACC |
| JMPGEZ X | 50 | IF ACC>=0，THEN X🡪PC  ELSE PC+1🡪PC |
| Jmp x | 60 | Jump [x] |
| Halt | 70 | 0🡪ACC,0🡪PC |
| And | 80 | ACC AND BR 🡪ACC |
| Or | 90 | ACC OR BR🡪ACC |
| Not | A0 | NOT ACC🡪ACC |
| Sl | B0 | SHIFT ACC LEFT X BITS |
| Sr | C0 | SHIFT ACC RIGHT X BITS |
| Mpy | D0 | ACC \* BR🡪ACC |
| Div | E0 | ACC/BR🡪ACC |

这里为简单易懂起见，这里暂时说几条指令：

|  |  |  |  |
| --- | --- | --- | --- |
| 指令 | 运算吗 | 范例 | 描述 |
| LDA | 0000 | LDA 9H | 将9H的内容加载到累加器 |
| ADD | 0001 | ADD BH | 将BH内容和累加器相加，结果放到累加器 |
| SUB | 0010 | SUB CH | 将CH内容和累加器的相减，结果放到累加器 |
| OUT | 1110 | OUT | 将累加器的内容放到“输出寄存器”（给一个输出） |
| HLT | 1111 | HLT | Cpu停止运行 |

比如，

1，将下面程序翻译成机器码：

地址 指令

0H LDA 9H -----》0000 1001

1H ADD EH------》0001 1110

2H HLT----------》1111XXXX（但一般把X换成0，这个是因为EDA软件的问题）,

2，运算例子：10+20-15 （十六进制）

首先把数据10，20，15依次放入到9H，AH，BH内存中（这里是举例子，该程序中是放到LPM-ROM），然后首先将10加载到累加器，所以是LDA 9H，

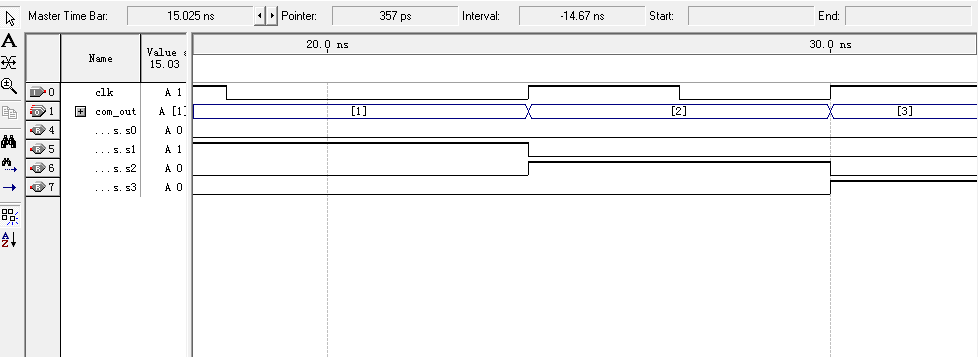
然后和20相加，所以是，ADD AH，结果放入了累加器，然后和15相减，所以是SUB BH，结果还是放入了累加器，最后输出结果，OUT ，CPU停止运行，HLT。

所以最后整个程序写在RAM中，就该是如下：

|  |  |  |
| --- | --- | --- |
| 地址 | 汇编语言 | 机器码 |
| 0H | LDA 9H | 0000 1001 |
| 1H | ADD AH | 0001 1010 |
| 2H | SUB BH | 0010 1011 |
| 3H | OUT | 1110 0000 |
| 4H | HLT | 1111 0000 |
| XXXX | XXXXXXX | XXXX XXXX |
| 8H | XXXX | XXXX XXXX |
| 9H | 10H | 0001 0000 |
| AH | 20H | 0010 0000 |
| BH | 15H | 0001 0101 |
| XXXX | XXXXX | XXXX XXXX |

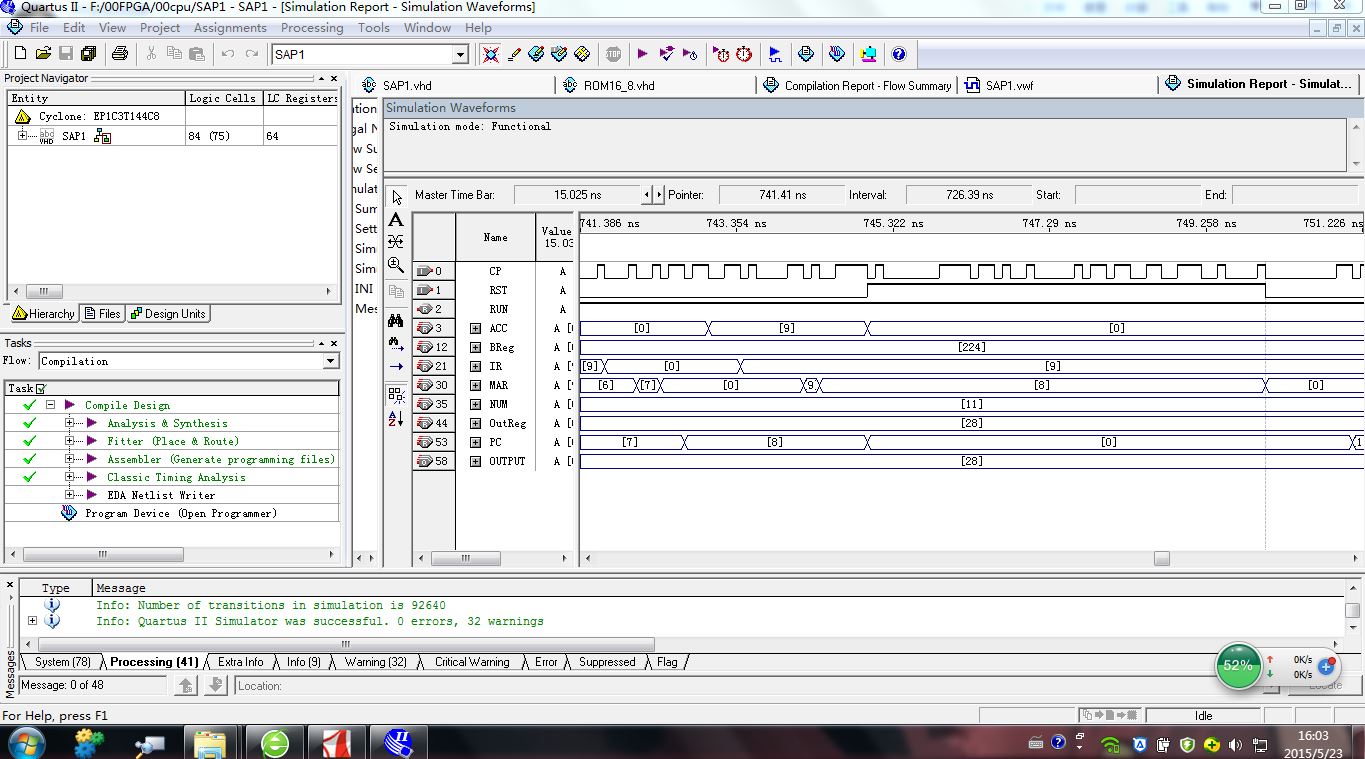
4,有限状态机设计

状态机就是在有效状态内，在时钟的驱动下，通过给定初始状态，能够自动完成状态间的循环和相应状态的输出的时序逻辑电路。因为CPU的处理过程很类似状态的转换，只是状态与状态之间的转换而已，而且牵涉到时序逻辑电路所以，为了消除和降低竞争和冒险，状态机是比较好的处理方法。在顶层实体里面写状态机进行各个状态的控制。



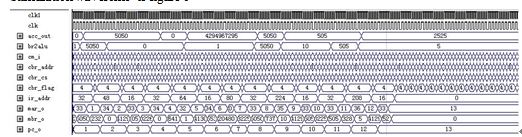
5，模拟仿真与验证

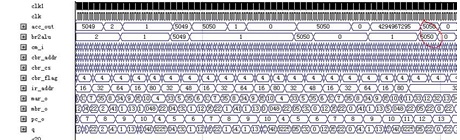
1. 计算--10+14+18-20=1C （16进制的）



可以看到上图结果是28，这是十进制的，换成十六进制也就是1C了，所以验证结果正确。

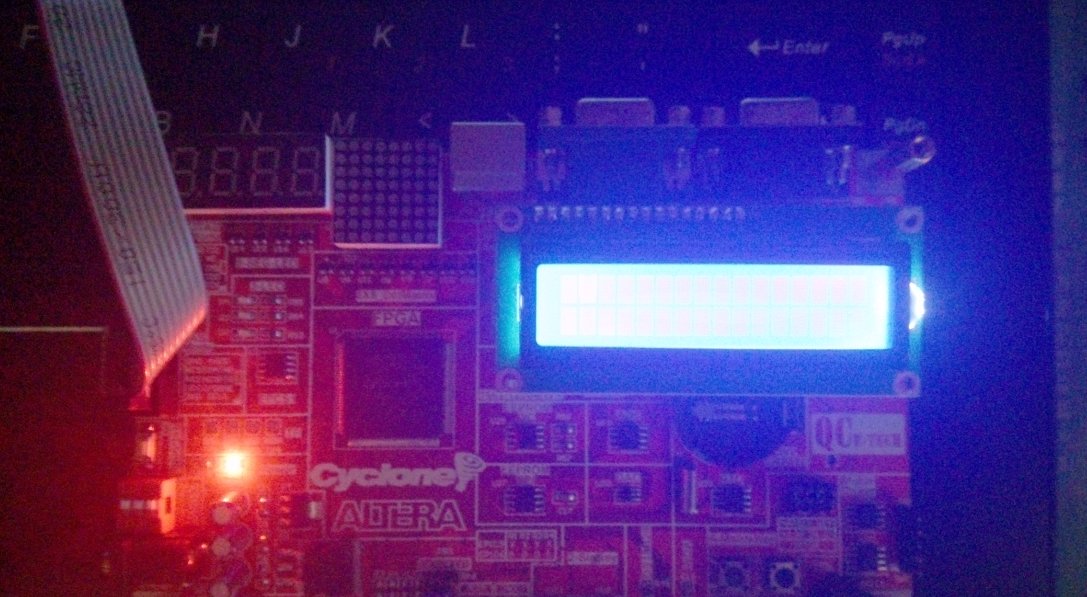
1. 计算1+2+3+……+100





经验证结果均正确。

1. 用LCD1602显示结果。



可能由于角度关系看不太清楚。

6，结论与展望

本系统以设计利用cyclone ep1c3t144c8n fpga为核心，用VHDL语言编写的一个简易的CPU，该CPU具备了CPU的基本功能，能进行基本的算数运算，和对外部设备的控制，该系统主要是用的有限状态机来解决了时序逻辑电路存在的毛刺现象，一开始的时候由于时钟的不稳定，所以结果不太准确，但后来稳定了就正确了。该系统的指令系统还不完善，对一些太过复杂的我们也进行了简化。但这并不影响其主要功能。

目前我们国家和其他国家的竞争是时刻存在的，谁掌握了核心技术，谁就能立不败之地，所以掌握了FPGA的设计对我们的军事和航天技术有很大的帮助，本文的CPU设计又恰恰是囊括FPGA知识点的最好的项目，当然如果能写得更好，研究的更加透测，那么我们就可以使用自己的CPU了，这样对我过改变CPU市场有很大的帮助，这样CPU也将具有更过阔的前景。

参考文献

1. FPGA开发
2. 邵贝贝.FPGA嵌入式应用的在线开发方法[M].北京:清华大学出版社,2005.40-42.
3. 卓晴,黄开胜,邵贝贝.[M].北京:北京航空航天大学出版社,2007.140-141.
4. 高吉祥.全国大学生电子设计大赛培训系列教程[M].电子工业出版社,2007.6.1.
5. 王志良.FPGA制作技术[M].机械工业出版社,2007.6.15.
6. 程鹏.FPGA原理[M].高等教育出版社. 2004.6.
7. 刘伟.传感器原理及其适用技术[J].电子工业出版社,2006.3.
8. 张植宝.电机原理与应用[M].化学工业出版社,2006.10.1.
9. 许大中.电机控制[M].浙江大学出版社,2002
10. 李华.FPGA实用接口技术[M].北京:北京航空航天大学出版社,1995.