# 图像获取设计说明IG202003-T1

## 设计目标

！！！每个设计都需要一个明确且具体的目标。设计目标负责说明这个部分的内容。

### 1.1设计总目标

这个系统的总目标是完成一份1000字的英文文档的自动提取工作，需要提取的对象有数字、小写字母、大写字母，字符暂时不考虑。如果测试工作正常的话可以将其纳入测试对象，时间不足的话就暂时放弃。

### 1.2各部分目标

数字识别比较单一，只需要完成常见的一种数字即算通关。

字母识别的对象比较多，但是以待测试的文档作为标准进行测试。字符大小需要兼容预期。

符号识别暂时不做规划，设计顺利的话可以考虑。

## 设计原理

设计的原理还是比较简单的，使用FPGA完成字符定义的逻辑复刻，利用并行处理快速地得到字符的对应含义。设计的关键在于如何认识到字符和图片字符之间的对应关系以及构建对应关系。

## 设计思路

最简单的思路就是使用存储器，通过对不同字符的手动矫正完成对应关系的建立，进而使用对应关系完成需要的设计。比如用户输入一组标准字符和字符图像，由对应模块建立对应关系，然后启用对应关系实现字符图像向字符编码的快速转化。这种对应关系的建立是需要大量的用户干预的，且容易发生冲突，效率低下。最大的优点是结构简单，设计难度较低。实际上前面的设计中采用的方法就是这个。

基于前面的存储器记录对应关系的思路，可以考虑其他的思路。存储器通过记忆前面存在的状态构建对应关系，也可以预先确认存在的关系，通过区分加入到区分队列。但是两者的效率是不同的。单个关系的建立是比较简单的，一对多的关系建立是复杂的。复杂意味着更多的设计量。而且，没有办法保证压缩的数据冲突后的继续判断。即通过确认队列然后添加字符数据的方法是比较难实现的。

还有一种方法是图像比较。基于图像的特点进行特定结构的提取，通过比较判断对象中的数据所指代的字符。这种方法的难度在于如何设计大量的不同形状的提取模块。提取模块需要给出对应的匹配度，然后在最终的判断模块中确认。想想就难弄。但是这个却可以快速的延展。

还有一种思路是根据定义识别。这个识别需要关注字的要素，这个是比较难把握的。汉字一般可以采用这种思路，但是字符不行。

所以，字符还是需要采用更加简便的方法。字符的定义量有限，主要的问题在如何界定不同的字符体系的数据。根据当前的需求，能够完成一种就可以。存储格式的话采用分布式RAM存储即可。采用组合码的方式确认即可，主要需要处理的问题是如何管理关系，这里需要围绕对应关系构建一系列的内容。这里采用队列的思想构建设计。队列管理系统，也就是这个模块设计的最终目的。为了加以区分，将这种组合队列记为码表。

## 4.设计模块

#### 4.1新建码表

新建压缩码表主要是依靠图像压缩模块，获取到前导码后，通过调用组合编码功能，实现一个码表的建立。这种和压缩模块和组合模块有关的数据表就是后面设计所需要的码表。（理论上采用双压缩的模块不会出现一个编码对应多个字符的情况，但是对于某些区分度不足的字符，这个问题是可能存在的）。

#### 4.2使用码表

将图像获取的编码和码表对应，直接获取数据。这里对压缩方式和组合编码的方式都有一定的要求。考虑到FPGA和人交互速度的要求，可以直接采用页读完成数据比对。尤其可以采用分类比对的方法，这样可以快速地确认每一类中的比对结果以及标识出由于不同类别导致的差别。

#### 4.3读取码表

为了方便管理，必须可以读取目前设备内所存储的码表。这个直接使用一个读取模块就可以实现。

#### 4.4删除码表

这个是配合读取码表使用的，通过对码表的空白写入实现。由于00的编码对应的字符为空，所以删除码表不能简单的去除对应值。这里为了提高管理，可以采用主编码和次编码的方式，实现预期的设计。通过增减次编码实现编码数据的刷新。注意，如果可以的话考虑分布式的RAM，这个部分的数据量不会太多。存储格式的问题需要测试一下。

## 设计纲领

本次设计有一定的基础，需要的创新点也是有的。为了加快完成测试整体的设计，最好能够采用新的设计思路，确保设计可以顺利完成。设计时要求做好文档的记录、图片的绘制和工程的规范。整体设计采用git完成全流程的版本控制。用各种的手段保证进度。主要的是文档流、描述流和工程流三个部分的同步跟进。先测试这三个部分的能力，注意归类。在完成之后还有一个快捷部分的建立。