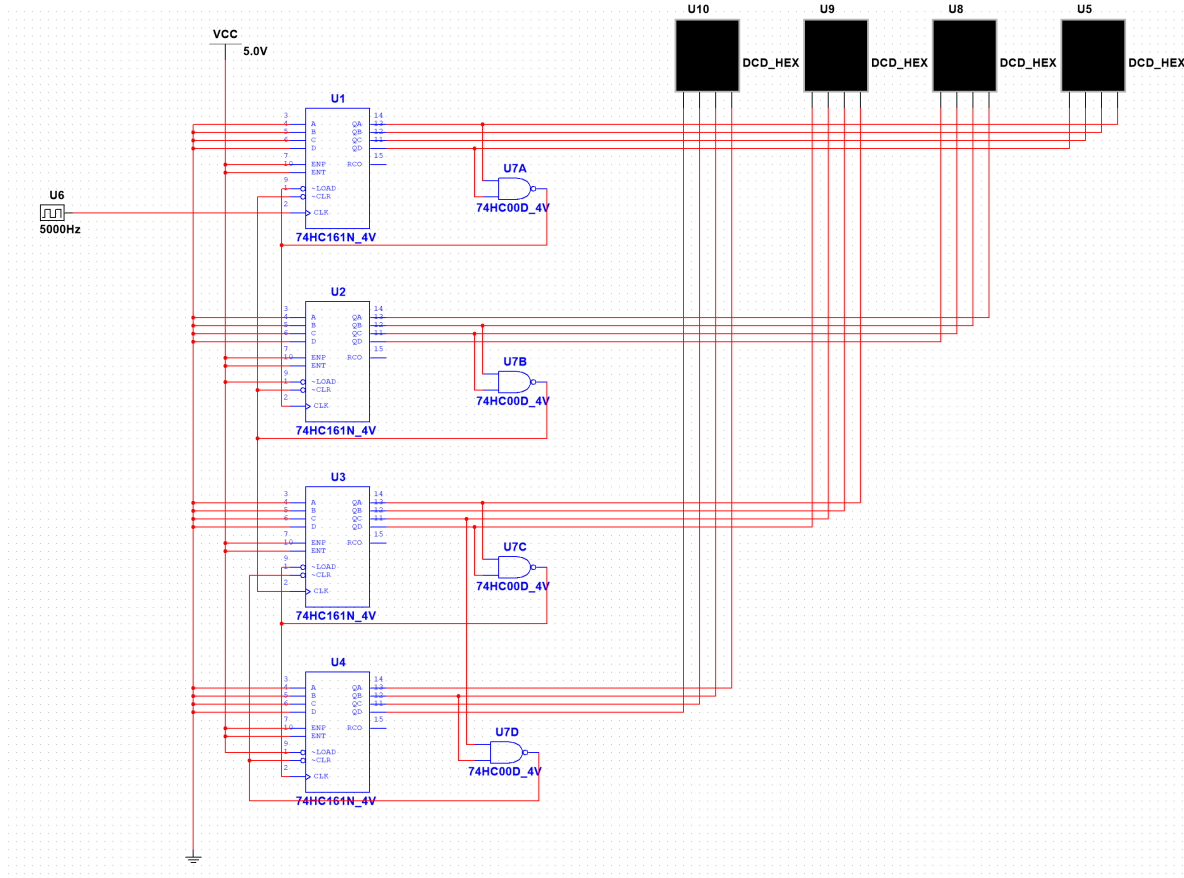


「简易硬件描述语言功能仿真工具」开题报告

组员：JS221103董懿璇 JS221102常睿心 JS221114田昊冬 JS221116倪哲振

1 契机

在本学期的「数字逻辑电路实验」课程中，我们需要自行设计出一些具有特定功能的组合逻辑芯片或时序逻辑芯片。在设计过程中，我们所使用的 Multisim 和 Quartus 等设计软件均采用图形化界面进行设计，需要手工连接大量引脚，工作繁琐不堪。同时，设计软件所自带的功能仿真工具过于臃肿，十分不便。



受到公开课 [From NAND To Tetris](#) 的启发，我们希望开发一款简易的 [硬件描述语言](#) 以及相应的功能仿真工具，用以简化数字逻辑电路设计过程，提升设计效率。

2 核心功能

硬件描述语言 CDL

用单个文件描述一个芯片的输入输出引脚以及内部逻辑，支持导入其他使用 CDL 语言描述的芯片。

示例：四输入与门 `AND4.cdl`

```
# 从 AND.cdl 中模块化导入 AND 芯片
import AND from "AND"
@Head
NAME AND4
# 实例化三个 AND 芯片
AND AND_1, AND_2, AND_3
```

```

# 声明输入输出引脚
INPUT a, b, c, d
OUTPUT out
@Body
# 指定 AND_1 芯片的输入输出引脚, a_and_b 是一个新的局部变量
AND_1(a = a, b = b) => (out = a_and_b)
AND_2(a = c, b = d) => (out = c_and_d)
AND_3(a = a_and_b, b = c_and_d) => (out = result)
@End
# 设置输出引脚
out = result

```

功能仿真工具

指定输入文件和 CDL 语言描述的芯片文件，计算并得到输出文件。

示例：四输入与门测试用例 `AN4.in`

```

# 测试用例
@Case
a = 0
b = 0
c = 1
d = 1

@Case
a = 1
b = 1
c = 1
d = 1

@Case
a = 01001
b = 00111
c = 01001
d = 10111

```

进行功能仿真后，得到的输出文件如下：

```

@Case
a  = 0
b  = 0
c  = 1
d  = 1
out = 0

@Case
a  = 1
b  = 1
c  = 1
d  = 1
out = 1

@Case

```

```
a = 01001
b = 00111
c = 01001
d = 10111
out = 00001
```

3 实现方法

导入分析

遍历芯片的导入层级结构，检查是否存在同名芯片或循环导入。

句法分析

分析所有文件中是否存在语法错误，例如使用未导入的芯片、使用未定义的引脚、未指定输入输出引脚等。

实例化 `symbol_definition` 类

对于每个芯片，实例化一个 `symbol_definition` 对象，记录该芯片的输入输出引脚、需要实例化的芯片等信息。

`symbol_definition` 对象需要提供 `create_instance()` 成员函数，用于返回一个实例化的 `symbol_instance` 对象。

`symbol_instance` 对象需要提供 `execute()` 成员函数，用于指定输入、返回输出。

为什么需要使用 `symbol_instance` 对象执行逻辑，而不直接使用 `symbol_definition` 执行逻辑？

这是为了应对时序逻辑芯片。时序逻辑芯片的输出除了取决于当前的输入以外，还取决于该芯片自身的状态。如果使用 `symbol_definition` 执行逻辑，将无法记录芯片自身状态。

实例化 `symbol_instance` 类

考虑 `create_instance()` 函数应该如何实现。`symbol_instance` 类在初始化时，应该按照 CDL 文件中的定义实例化若干芯片，并将它们设置为 `symbol_instance` 类的成员。

在调用 `execute()` 成员函数时，只需要按照逻辑体所指定的先后顺序，依次调用 `symbol_instance` 内部的“成员芯片”的 `execute()` 方法即可。