

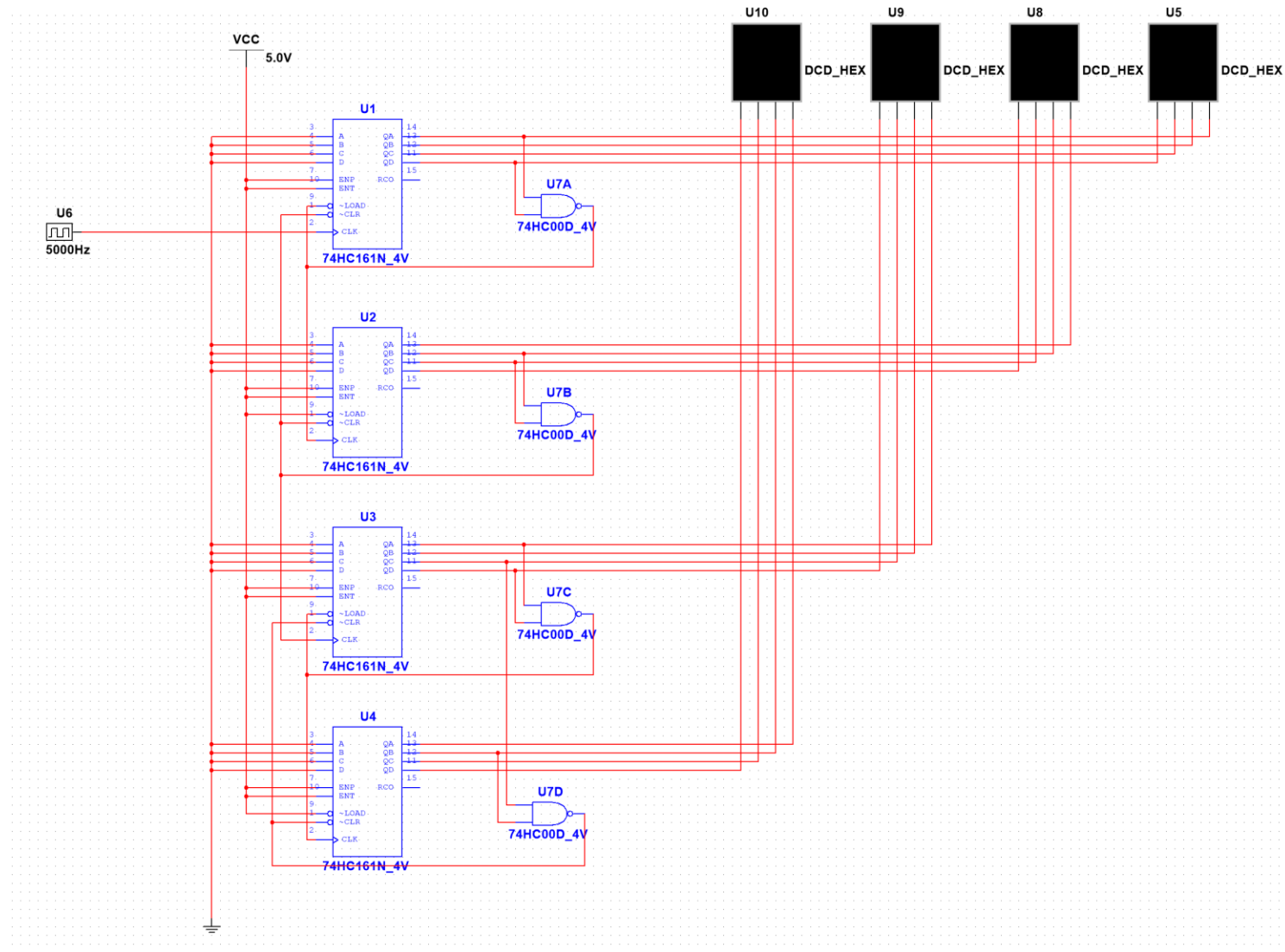
Contras HDL Simulator

硬件描述语言功能仿真工具

程序设计基础及语言 II 课程设计

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

1 契机



2 核心功能 – 硬件描述语言 CDL

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 AND1, AND2, AND3
```

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

```
AND1(a = a, b = b) => (out = ab)
```

```
AND2(a = c, b = d) => (out = cd)
```

```
AND3(a = ab, b = cd) => (out = result)
```

```
@End
```

设置输出引脚

```
out = result
```

2 核心功能 – 功能仿真工具

AND4.in

@Case

a = 0

b = 0

c = 1

d = 1

out = ?

@End

@Case

a = 01001

b = 00111

c = 01001

d = 10111

out = ?

@End

2 核心功能 – 功能仿真工具

AND4.out

@Case

a[0] = 0

b[0] = 0

c[0] = 1

d[0] = 1

out[0] = 0

@Case

a[0] = 01001

b[0] = 00111

c[0] = 01001

d[0] = 10111

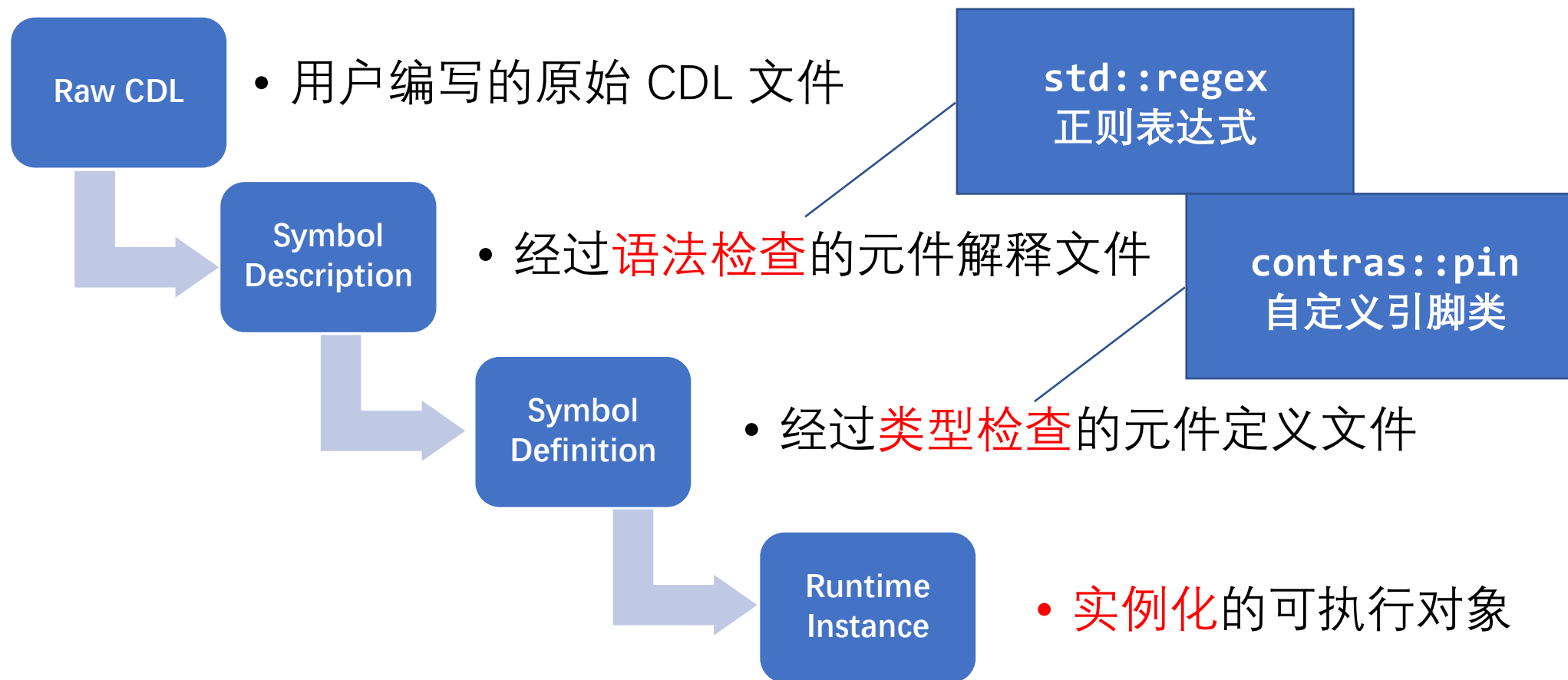
out[0] = 00001

3 设计思路 – 系统设计

技术难点：

1. CDL 文件语义分析
2. 将单文件的元件 CDL 文件转换为实例化的 C++ 对象

3 设计思路 – 系统设计



3 设计思路 – Symbol Definition

CDL 导入结构：有向无环图 (DAG)

拓扑排序

3 设计思路 – Runtime Instance

- 使用 `std::map` 构建元件名与 `Symbol Definition` 的映射

```
static std::shared_ptr<runtime_instance>
create_runtime_instance(
    const std::shared_ptr<symbol_definition> &def,
    const std::map<std::string,
std::shared_ptr<symbol_definition>> &def_map);
```

3 设计思路 – Utils

- 自定义日志输出工具
- 自定义异常处理

```
__CONTRAS_LOG(debug, "Put your debug messages here");  
__CONTRAS_THROW(contras::exception_type::invalid_argument,  
"Throw an exception");  
  
int main(int argc, char *argv[]) {  
#ifdef CONTRAS_USE_LOGGER  
    contras::use_logger();  
#endif  
...  
}
```

4 演示

- 使用 CMake 构建

```
mkdir build
cd build
cmake .. -G Ninja -DCONTRAS_USE_LOGGER=true
cmake --build .
cd ../bin
```

```
mkdir build
cd build
cmake .. -G Ninja -DCONTRAS_USE_LOGGER=false
cmake --build .
cd ../bin
```

4 演示

```
cd bin  
./contras And4.cdl test/And4.in test/And4.out  
./contras Or4.cdl test/Or4.in test/Or4.out
```

5 改进空间

1. 语义分析
2. 时序逻辑电路不穩定状态的仿真

Thank you!

- [GitHub Repo](#)
- [Coursera: Nand2Tetris](#)
- [From Nand To Tetris](#)