1. 模块结构列表
2. class Module的结构

{

list<Block \*> blocks; //存放包含的block

list<InputNet \*> inputs; //输入端口

list<OutputNet \*> outputs; //输出端口

list<OutputNet \*> internalNets; //内部网络

}

1. struct Block的结构

{

vector<class Net \*> inputs; //输入网络口

vector<class OutputNet \*> outputs; //输出网络口

Librarycell \*cell;

int moduleNumber;

}

1. struct InputNet : struct Net的结构

{

double requiredMinLength;

double allowedMaxLength;

map<OutputNet \*,double> controllableOutputs; //通表，防止闭环

list<Terminal> sinks;

}

1. struct OutputNet : struct Net的结构

{

Terminal source;

double maxLength;

list<Terminal> sinks;

}

1. Terminal的结构

typedef pair<Block \*, unsigned int> Terminal

1. Module生成的两种模式
2. Module(Librarycell \*cell) //block生成Module,针对LibrarycellNode类型结点
3. latch类型

InputNet

{

double requiredMinLength = 0

double allowedMaxLength = 40

map<OutputNet \*,double> controllableOutputs; //通表，防止闭环

list<Terminal> sinks;

}

OutputNet

{

Terminal source;

double maxLength = 0

list<Terminal> sinks;

}

1. gate类型

InputNet

{

double requiredMinLength = 0

double allowedMaxLength = 39

map<OutputNet \*,double> controllableOutputs; //通表，防止闭环

{controllableOutputs[ouputnet] = 1}

list<Terminal> sinks;

}

OutputNet

{

Terminal source;

double maxLength = 1

list<Terminal> sinks;

}

1. Module(Module \*modA, Module \*modB, ModuleType \*modType) //两个Module聚合,针对CompoundNode类型结点