

論理回路シミュレータの使用法

1 はじめに

論理回路シミュレータであるが、バグをつぶしている間に、使用法が演習資料と若干異なってきた。そこで、主な変更点を以下で述べる。

2 NullNode について

回路の入力部に、必ず NullNode という Node を介すことにした。これは入力をそのまま出力するものである。simMain.cpp も NullNode を用いるように変更されている (図 1)。

NullNode をつける理由は、次章で見るように Edge を分岐させるのに Node が必要だからである。

```
Edge i1, i2, i3;  
Edge o1, o2, o3;  
Edge o4, o5, o6;
```

```
NullNode null1(o1, i1);  
NullNode null2(o2, i2);  
NullNode null3(o3, i3);
```

```
And and1( o4, o1, o2 );  
Or or1 ( o5, o3, o4 );  
Not not1( o6, o5 );
```

図 1: 回路の定義。

また、図 1 により定義される回路は図 2 のようになる。

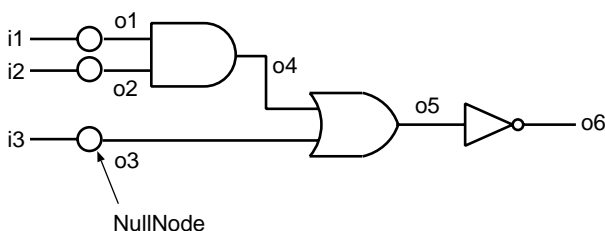


図 2: 定義された論理回路。

3 Edge の分岐に関して

論理回路を取り扱う際、図 3 (a) のように Edge を分岐させることがしばしばある。

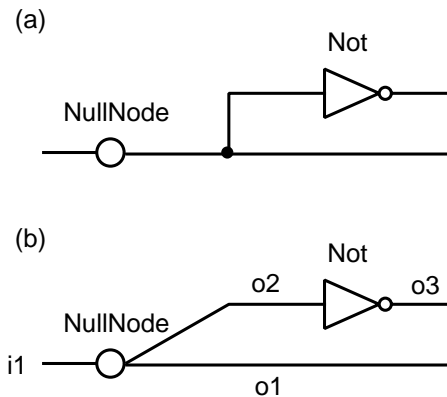


図 3: 分岐した Edge の取り扱い。

このような時は、図 3 (b) のように、前段の Node に複数の Edge をつなぐようにする。

方法は図 4 の通り。

```
NullNode null1(o1, i1);  
null1.outEdgeAdd(o2);  
Not not1(o3, o2);
```

図 4: Node に複数の Edge をつなぐ。

4 おわりに

以上、論理回路シミュレータの使用に関し、演習資料からの変更点を中心に解説した。以上を参考にすると、半加算器も実現できるはずである。

なお、配布したプログラムは必ずしも模範的な構造、記述になっていない。(バグなどの混乱からかわかと思いますが...)

このプログラムから、「一つのプログラムが、複数のクラスが組み合わされて実現され、動作していること」を実感するのが目的である。