基于对称选择器的组合逻辑分析
Combinational Logic Analysis based on Symmetric Selector

王博文, Wang Bowen

September 2, 2021

摘 要

本文利用一种理论上计算可逆的数字逻辑最小完全集：对称选择器，通过快速而完备的方法，对外在形式各异的多比特输入单比特输出的组合数字逻辑电路进行了通用归约，并通过这种归约方式，将组合数字逻辑电路等价到一类正规形式，实现了完备的组合逻辑分析。该方法可作为布尔表达式可满足性判定的基础工具，应用于教学与工程应用之中。

In this paper, we propose a logic gate called ”Symmetric Selector” which satisfying reversible computation condition. Through a rapid and functionally complete method, we can reduce combinational logic circuits universally in normalized form, with the logic gate. Finally, we implement a complete combinational logic analysis, which can be applied in education and engineering.

Keywords: Logic Gate; Universal Logic Gates ; Sole Sufficient Operator ; Reversible Computation; Combinational Logic Circuits; SAT Solver

1 数字逻辑完全集：对称选择器

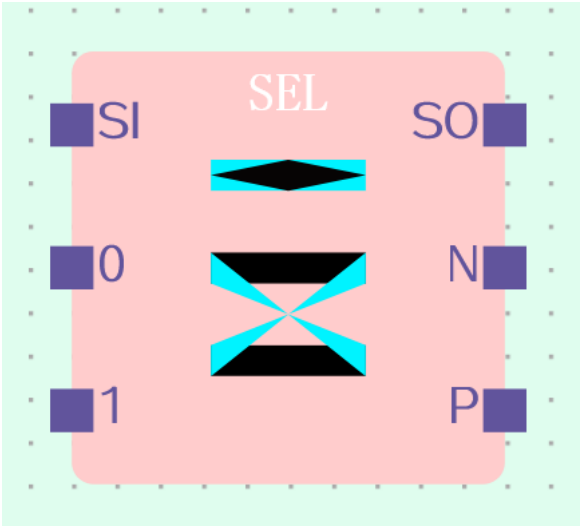


图 1: 对称选择器（其实是个升级版的 CrossBar-Switch [1]）

上图中，我们引入了一个可以用真值表表示的可逆数字逻辑基元——对称选择器：该基元有六个计算端口 SI、0、1、SO、N、P，每个端口均可输入或输出一个 {0, 1} 信号（即一个 bit），该基元的计算规则如下：

1. 端口 SI 的信号值传递到端口 SO
2. SI 的信号值为 0 时，端口 0 的信号值传递到端口 N，端口 1 的信号值传递到端口 P
3. SI 的信号值为 1 时，端口 0 的信号值传递到端口 P，端口 1 的信号值传递到端口 N
4. 以上信号传递均可反向
5. 传递延迟固定为 $\Delta\tau$

其运算真值表如下：

SI	“0”	“1”	SO	N	P
0	A	B	0	A	B
1	A	B	1	B	A

表 1: 对称选择器真值表

之后，用以下符号表达五个逻辑操作：

符号	逻辑操作名称
.	与（AND）
,	或（OR）
<	非（NOT）
>	推出（INFER）
=	等价/同或（EQUAL/NXOR）

表 2: 逻辑操作符号表

接下来，我们再引入逆波兰表达式，我们通过将这五个逻辑操作转化为“0”、“1”、“对称选择器”的输入输出组合，来证明这是一个数字逻辑最小完全集。同时，逆波兰表达式可以允许我们输入更复杂的符号组合，去表达一些更复杂的逻辑永真式。

1、与门逆波兰表达：a b .

对称选择器组合图像：

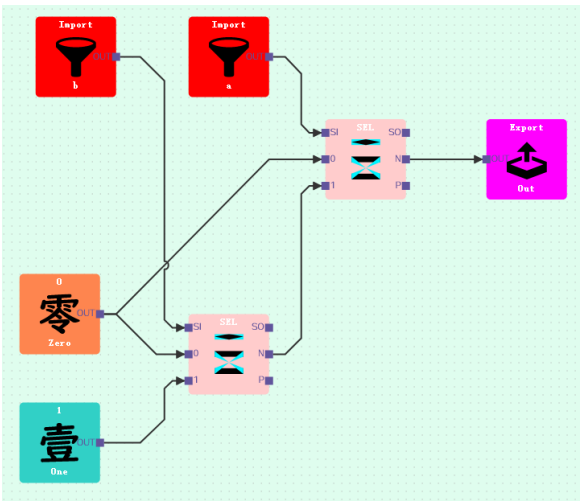


图 2: 由对称选择器组合成的与门

2、或门逆波兰表达：a b ,

对称选择器组合图像：

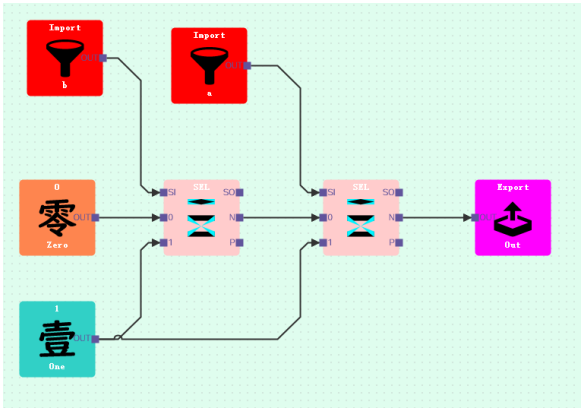


图 3: 由对称选择器组合成的或门

3、非门逆波兰表达：a <

对称选择器组合图像：

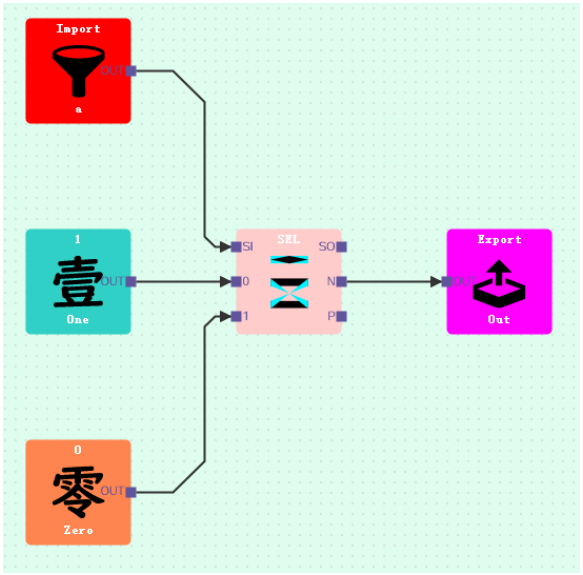


图 4: 由对称选择器组合成的非门

4、推出门逆波兰表达：a b >

对称选择器组合图像：

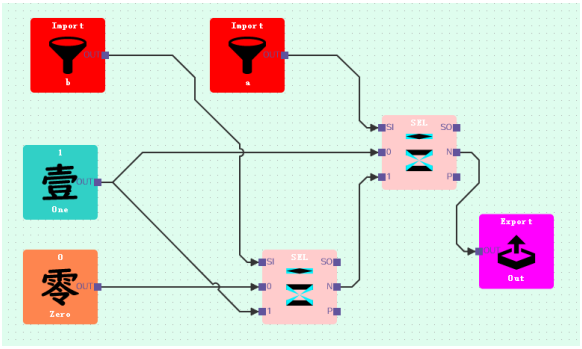


图 5: 由对称选择器组合成的推出门

5、等价门逆波兰表达：a b =
对称选择器组合图像：

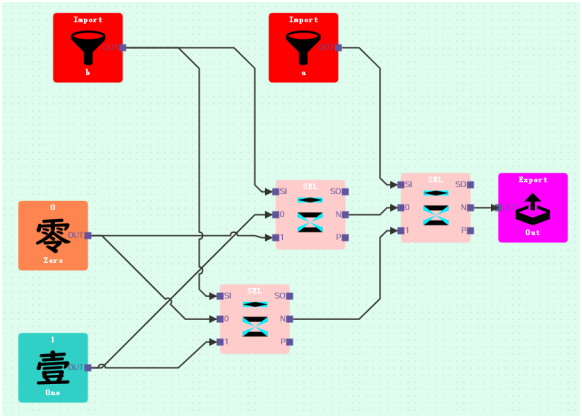


图 6: 由对称选择器组合成的等价门（同或门）

定义完以上逻辑门之后，我们就可以用逆波兰表达式表达一些更复杂的逻辑公式了，这里举几个例子：

1、等价律：

$$((a \to b).(b \to a)) = (a = b)$$

逆波兰表达：a b > b a > . a b = =

2、反证法：

$$(((a.\bar{b}) \to c).((a.\bar{b}) \to \bar{c})) = (a \to b)$$

逆波兰表达：a b < . c > a b < . c < > . a b > =

3、双前提：

$$(a \to (b \to c)) = ((a.b) \to c)$$

逆波兰表达：a b c > > a b . c > =

4、三段论：

$$((a \to b).(b \to c)) \to (a \to c)$$

逆波兰表达：a b > b c > . a c > >

5、等价传递：

$$((a = b).(b = c)) \to (a = c)$$

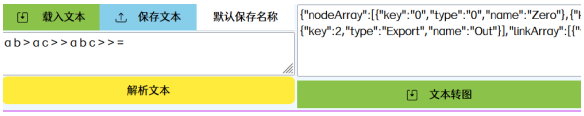
逆波兰表达：a b = b c = . a c = >

6、前提（上下文）转接：

$$((a \to b) \to (a \to c)) = (a \to (b \to c))$$

逆波兰表达：a b > a c > > a b c > > =

将这些逆波兰表达式输入到我们的归约系统中后，通过本文的规约方法，能统一得到如下的结果，由此可以验证各种谓词逻辑定律的正确性：



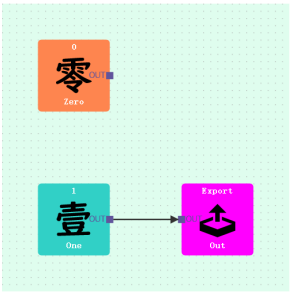


图 7: 对永真谓词命题的归约结果

实际上，本文所用的归约算法仅使用选择器即可实现，但这里使用**对称选择器**，是因为该计算结构是一个**可逆**计算单元。使用此种计算单元，可以在标准均一化的比特信号能量传输过程中，实现输入信号能量和输出信号能量的平衡。

通过合理使用对称选择器，可以一定程度上降低计算过程中的能量耗散（即兰道尔（landauer）原理 [2] ）。

2 算法实现原理：从未正规形式到正规形式的归约

这里使用逆波兰表达式 A B > A C > > A B C > > = 来举例说明该算法原理。

我们可以这个表达式分解成 A B > A C > > 和 A B C > > 这两个子表达式，再将这两个子表达式的输出结果输入到等价（=）如图6 这个计算式的 Import 端口。此时，我们得到了一个“未正规化”的计算模块，可以粗略地展现为下图（假设等价计算式内除了和最后的比特输出相接的对称选择器以外都已经变为了“正规化模块”）所示：

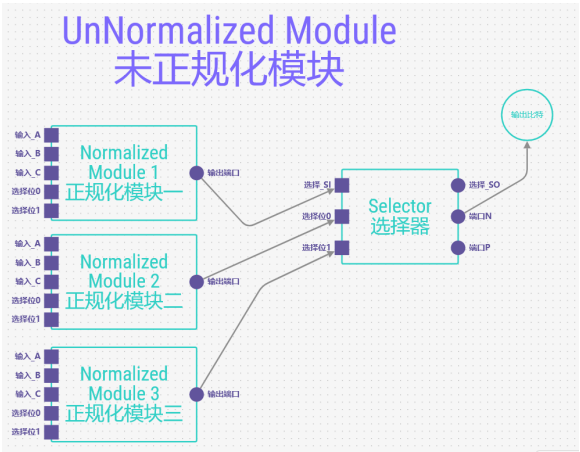


图 8: 未正规化模块，但这里假设选择器三个输入端均接上了“正规化”的计算模块。

这里的“正规化模块”，其实就是指一种模块，对于该模块内的所有对称选择器而言，它们的 SI 输入端口都是直接和 A、B、C 中的任一个输入信号相连。

然后，我们可以找到上图那个“未正规化模块”的等价形式，将它转换为“正规化模块”，如下图所示：

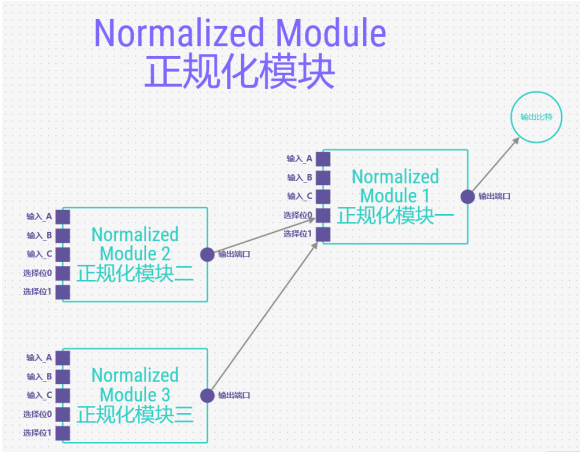


图 9: 从未正规化形式等效变换而来的正规化形式模块。

用该方法，我们可以通过递归的方式从最末端的对称选择器组合开始，逐步将“未正规化子模块”转为“正规化子模块”，最后完成整个计算模块的正规化。

在这个正规化模块里，其实存在很多无法到达输出端的，从“选择位 0”和“选择位 1”输入的信号通路。此时使用递归算法将这些不通的通路消去后，我们就能得到一个更清晰的正规化计算模块。

当这个正规化计算模块所有可能的选择输入，都将信号从“选择位 1”选入并输出到输出比特位置时，我们就得到了一个永真式的计算结构，并可将其简化为图7 的形式（同理，我们还能得到永假式的计算结构）。

3 对时序逻辑分析功能的展望

我们可以把数据记下（输入到输出）到回忆（输出到输入）的过程，用对称选择器表示成在数字逻辑电路中被广泛使用的“锁存”（Latch）[3]：

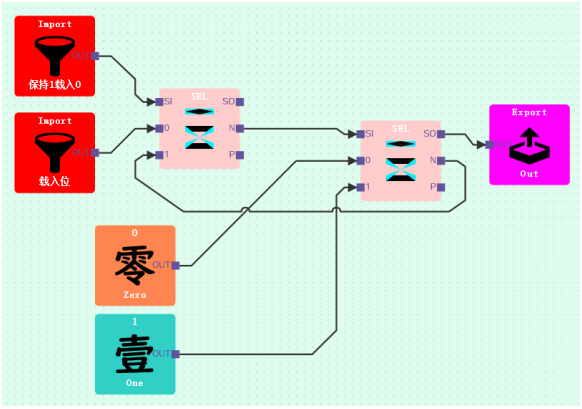


图 10: 用对称选择器制作的锁存结构

从图中，我们能看出，当“保持 1 载入 0”这个输入位的值为 1 时，这就允许了一个信号传输环路，并把信号“锁”在这个环路内。这个计算结构可以作为 D 触发器和 JK 触发器等存储与分频器件的基本工作单元。

我们实际上也能在这种计算结构中，制造振荡。

这里先假设信号通过一个对称选择器的时间均为 $\Delta\tau$，那么只需要在 $T = 2\Delta\tau$ 的时间内，将“保持 1 载入 0”这个输入位值保持为 0，再将“载入位”置 0 一段时间，然后将它置 1 一段时间，之后再将“保持 1 载入 0”这个输入位值修改为 1——这就产生了 01 振荡，实际上，我们能用这种方法产生周期为 $2\Delta\tau$ 的任意方波。

从这里可以看出，通过加入信号反馈，我们可以用有限个对称选择器组成的计算结构生成无限长的二进制序列。

可是实际上，对于有限个对称选择器组成的计算结构，通过这种时序逻辑的组合形式（信号反馈），我们只能得到有理数的二进制输出序列，也即，存在循环周期的序列。因为有限个对称选择器对应着有限个状态，而有限状态必然导致自发的状态转换在有限步后重复，也即存在循环周期。

但是，在数学构造的领域 [4] 中，我们可以用语言定义无理数，以及无理数（在可计算序列 [5] 的范围之内）的计算结构。

这种无理数的计算结构本质上是一种输入自然数（无理数二进制形式的数位位置）且输出一个比特（该数位位置对应的值）的算法，且作为该算法输入的自然数越大，为完成其计算所需要的存储单元数量就越多，且此数量不存在上限（存在的话，就简化为有限个对称选择器的情况了）。

这也意味着，我们可以用语言（有限编码）编写出这些对称选择器的组合形式，且通过引入自然数（有限编码）作为输入类型，这种组合形式默认能构建由无限多个对称选择器组合而出的计算结构。

为此，我们可以引入一种新的，基于符号定义的，与我们古老的自然文化语言保持一个相对独立性的语言，来构建并控制这种计算结构。

4 完备的结论

我们可以通过将不同的二元或多元逻辑操作符，转化为由对称选择器组成的等效正规化模块，再将这些逻辑操作符的组合转化为这些正规化模块的组合，从而得到一个新的更大的未正规化模块。

然后，再通过递归地使用本文所提出的归约方式，将未正规化的模块归约为正规化模块，从而得到一个组合逻辑表达式的完备分析结果。

该求解过程相比使用标准输入 CNF 形式（即 Conjunctive Normal Form，合取范式）的其他算法 [6] 更为简单且通用，能作为布尔表达式可满足性判定（SAT Solver）工具用于教学与工程领域。

同时，本文提出的数字逻辑最小完全集——对称选择器，也可以作为基元建造出一个完整的数字计算机系统 [7]。

参考文献

[1] FJ Scudder and JN Reynolds. Crossbar dial telephone switching system. The Bell System Technical Journal, 18(1):76–118, 1939.

[2] Rolf Landauer. Irreversibility and heat generation in the computing process. IBM journal of research and development, 5(3):183–191, 1961.

[3] Stephen Brown and Zvonko G. Vranesic. Fundamentals of digital logic with vhdl design. 2009.

[4] Nishanth Kumar. Construction of number systems. 2009.

[5] A M Turing. On computable numbers, with an application to the entscheidungsproblem. Proceedings of The London Mathematical Society, 41(1):230–265, 1937.

[6] 郭莹, 张长胜, 张斌, et al. 求解 sat 问题的算法的研究进展. 计算机科学, 43(3):8–17, 2016.

[7] Noam Nisan and Shimon Schocken. The elements of computing systems: building a modern computer from first principles. MIT press, 2021.