可重构非线性布尔函数利用率模型研究与硬件设计

戴紫彬[①] 王周闯[①] 李 伟*[②] 李嘉敏[①] 南龙梅[②]
[①](解放军信息工程大学 郑州 450001)
[②](复旦大学专用集成电路与系统国家重点实验室 上海 200433)

摘 要：为解决序列密码中非线性布尔函数(Non-Linear Boolean Function, NLBF)硬件资源利用率低的问题，该文对以查找表(Look-Up Table, LUT)为基本构件的利用率模型进行研究，并结合适配算法的前期处理结果确定影响硬件利用率的 3 个基本参数(LUT 大小、单元规模和输入端口数目)；在此基础上，以变量频次为约束实现 NLBF 的映射，完成非线性运算单元的设计，单元支持多路并行处理。在 SMIC 180 nm 下进行逻辑综合，并行度为 32 时，工作频率达到 241 MHz，吞吐率为 7.71 Gb/s；对不同 NLBF 进行利用率评估，利用率均达到 91.14%以上，并且随着并行度增加，利用率不断增大。

关键词：序列密码；可重构计算；非线性布尔函数；查找表
中图分类号： TN918; TP311 **文献标识码**：A **文章编号**：1009-5896(2017)05-1226-07
DOI: 10.11999/JEIT160733

Hardware Implementation and Utilization Model Research for Reconfigurable Non-linear Boolean Function

DAI Zibin[①] WANG Zhouchuang[①] LI Wei[②] LI Jiamin[①] Nan Longmei[②]
[①](*The PLA Information Engineering University*, *Zhengzhou* 450001, *China*)
[②](*State Key Laboratory of ASIC and System*, *Fudan University*, *Shanghai* 200433, *China*)

Abstract: In order to solve the problem that the Non-Linear Boolean Function (NLBF) unit in sequence cryptogram possesses poor hardware resource utilization, the utilization model of basic component composed by Look-Up Table (LUT) is studied and three essential parameters (LUT size, cluster scale and the number of input ports) which impact hardware utilization are decided combined with the early processing results of adaption algorithm. On the basis, the mapping of NLBF limited to variable frequency is realized and the design of nonlinear computing unit is implemented, which can support multi-way parallel processing. The circuit is developed and synthesized in SMIC 180 nm. Its working frequency realizes 241 MHz and it achieves the maximum throughput of 7.71 Gb/s in parallelism of 32. The results after evaluating the utilization of various NLBFs show that all utilization can reach over 91.14% and the utilization increases continually as the parallelism increases.

Key words: Sequence cryptogram; Reconfigurable computing; Non-Linear Boolean Function (NLBF); Look-Up Table (LUT)

1 引言

由非线性布尔函数(Non-Linear Boolean Function, NLBF)作为基本构件的序列密码广泛应用于线上通信[1]、多媒体通信[2]和军事通信[3]等领域。基于 LUT 的硬件设计方式因实现原理简单、对函数形式无要求，成为 NLBF 设计的主流方法。

文献[4]提出一种基于 LUT 的 NLBF 处理单元 ALM(Adaptive Logic Module)，利用共享变量及查找表的拆分，可实现 1 个 6 变量和两个 5 变量布尔表达式，但未对 NLBF 变量特征进行分析，导致共享变量设置不够合理，硬件性能不能充分发挥；文献[5]利用 Shannon 分解定理，依据分解结果设计不同硬件结构，异构化设计可能导致利用率不高，尤其当 NLBF 表达式单一时，资源利用率会更低；文献[6]通过对 NLBF 可重构并行化技术的研究，将 NLBF 分成线性部分和非线性部分，本质仍是异构化设计且只对线性部分进行优化，仍存在利用率不高的问题。可重构计算不仅包含体系结构、硬件电路，还包含相应的软件(系统、应用软件)，而以上研究均未考虑软件的适配能力，以最大限度地提升资源利用率。

收稿日期：2016-07-08；改回日期：2016-12-12；网络出版：2017-02-09
*通信作者：李伟 liwei12@fudan.edu.cn
基金项目：国家自然科学基金(61404175)
Foundation Item: The National Natural Science Foundation of China (61404175)

将 NLBF 的变量特征与硬件性能相结合也许是提升单元利用率的一种有效方式。为获得较高的资源利用率，首先研究可编程现场门阵列(Programmed Field Gate Array, FPGA)中 查找表(Look-Up Table, LUT)的性能模型 —— 完全利用率(complete logic utilization)模型，将面向通用计算领域的 LUT 性能规律应用于专用处理领域；同时根据 NLBF 自身的特点，采用全局定向搜索适配算法对不同的 NLBF 进行适配处理，依据适配结果确定影响运算单元性能的基本参数，最终完成面向序列密码 NLBF 运算单元的设计——可重构序列密码逻辑单元 (Reconfigurable Logic Unit for Sequence Cryptographic, RSCLU)。

2　LUT 利用率影响因子

与非门、选择器、LUT 等作为可重构硬件设计的常用部件[7]，通过不同的组合形式可完成不同的功能。基于 LUT 的运算单元只需改变配置信息即可完成函数功能的可重构，因此，研究与分析 LUT 的性能影响因子对提升单元利用率将具有很大的作用。

2.1 LUT 利用率模型

目前，学术界有许多以 LUT 作为基本单元对其性能进行的研究。文献[8]研究了分簇式逻辑块利用率与 LUT 大小、共享输入间的关系，对于包含 N 个 LUT 的逻辑块，当输入端口数 I 与规模 N 的关系满足式(1)时，逻辑块将获得完全利用率(complete logic utilization)，同时对不同输入 LUT 进行性能对比，4 输入 LUT 组成的逻辑块具有完全利用率。

$$I = 2N + 2 \tag{1}$$

文献[9]通过对 2~7 输入 LUT、规模 N 从 1~10 的逻辑块进行测试，研究了 LUT 大小、逻辑块规模与端口数目对速度、面积、功耗的影响，对 28 个基准电路进行测试分析，并对所有可能的 LUT 大小和规模进行评估，得出达到完全利用率的等式关系：

$$I = \frac{K}{2} \times (N + 1) \tag{2}$$

式(1)及式(2)中 K 为 LUT 输入数目，K 输入 LUT 表示为 LUT-K；N 为运算簇包含的 LUT 数目，即逻辑块规模，I 为逻辑块输入端口数目。同时通过性能测试：在 $N \geqslant 4$ 时，小 LUT(LUT-2 及 LUT-3)具有较好的利用率，但速度却较大 LUT(4 输入以上)要差，并且 4~6 输入 LUT 在规模为 4~10 时具有较好的面积与延时。而在近期的研究中[10−16]，完全利用率模型也得到了证实。

2.2 基本参数确定

定义 1　f_0 和 f_1 为变量个数均不超过 K 的布尔表达式，由于 K 输入 LUT 能实现任意 K 变量布尔函数，若 f_0 的变量均包含在 f_1 中，则 f_0 和 f_1 可用同一个 LUT 实现，称 f_0 可以被 f_1 “吸收”，反之亦然；若 f_0 和 f_1 含有不同变量且总变量个数不超过 K，则 f_0 和 f_1 仍可用同一个 LUT 实现，称 f_0 和 f_1 可“合并”。

文献[17]提出一种适配 NLBF 的全局定向搜索算法，采用全搜索方式对各布尔表达式进行“吸收”处理，虽然映射于基本型 LUT 上，而非专用的 NLBF 处理单元，但适配结果仍值得借鉴。对不同密码算法中的 NLBF 进行“吸收”处理，得到各类型布尔表达式的比率，表 1 中第 2 列为 NLBF 各次项在“吸收”前的比率，“吸收”后各次项比率如第 4 列所示。选择出现频率高的 LUT 类型将有利于 NLBF 的实现，2 次项及 4 次项所占比率最高为 43.49%和 32.98%。但 LUT-2 作为基本单元存在两个问题：一是实现更高次的布尔表达式时级联复杂，增加了实现的困难度；二是 LUT-2~3 比 LUT-4~6 的性能要差[7, 8]。因此采用 LUT-4 作为基本运算单元，即将 K 值定为 4。

表1 “吸收”前后各次项比率(%)

类型	吸收前比率	类型	吸收后比率
1 次项	19.12	1 次项	17.44
2 次项	35.66	2 次项	43.49
3 次项	18.22	3 次项	0
4 次项	22.61	4 次项	32.98
5 次项	4.83	5 次项	3.62
6 次及以上	1.26	6 次及以上	0.77

RSCLU 的设计是以提升资源利用率为目标，高次项(4 次及以上)可通过 LUT-4 直接或级联实现，利用率会较高，但低次项需一定的处理才能达到较高利用率。为此，适配时将低次项“合并”成 4 变量形式，以确定 RSCLU 端口数目。在进行“合并”时遵循以下原则：

(1)为合理分配输入端口并降低端口数量，含有共享变量的布尔表达式优先“合并”。

(2)“合并”要满足多数 NLBF 的处理，不能针对特定的密码算法或某个固定的 NLBF。

依据适配原则，对密码算法中的 NLBF 进行“合并”，得出各 NLBF 低次项包含的端口数目，如表 2。

可以看出，端口数目通常在 11 以下，由于采用 LUT-4 作为基本运算单元，根据式(2)可知，端口数目 $I=(K/2)\times(N+1)=2(N+1)$必是偶数，因此将端口数目定为 10，对于大于 10 个变量的低次项，如

表2 低次项端口数目分布

算法	端口数目	算法	端口数目
Ach-A0	8	Ach-A10	4
Ach-A1	11	Ach-A11	10
Ach-A2	10	Ach-A12	10
Ach-A3	11	Decim	–
Ach-A4	8	Dicing	–
Ach-A5	9	LILI-128	13
Ach-A6	11	Grain-80	3
Ach-A7	9	Grain-128	19
Ach-A8	10	Toy-hs1	128
Ach-A9	7	Trivium	15

Grain-128 共有 19 个数据端口，可采用两个 RSCLU 实现；LILI-128 变量个数虽然较少只有 13 个，但布尔表达式却十分复杂，适配过程中发现，在 10 输入 RSCLU 上的适配也能达到很高的资源利用率；Toy-hs1 变量个数较多且无共享变量，适配时需要 13 个 10 输入 RSCLU，且利用率也较高；Trivium 算法共有 15 个变量，但分属 3 个不同的 5 变量 NLBF，采用 LUT-4 级联实现，也会具有较高的利用率。

将 I=10, K=4 代入式(2)，可得到 RSCLU 的规模 N=4。即 RSCLU 由 4 个 LUT-4 构成，外部含有 10 个输入端口。由此确定了专用序列密码运算单元 RSCLU 的 3 个基本参数，但 RSCLU 内多个 LUT 如何级联及级联后完成的功能需要进一步确定。

3 非线性运算单元结构

3.1 共享变量映射

在进行 RSCLU 变量映射之前，要确定 RSCLU 具备的功能。从表 1 各次项比率分布上可以知道，6 次及 6 次以上的布尔表达式，所占比率极少，依据 Shannon 分解定理：任何一个 n 变量的布尔函数均可采用两个(n–1)次项布尔函数来构造完成，如式(3)所示。因此，RSCLU 在实现 NLBF 时，不再直接实现 6 变量布尔表达式，而通过两个 5 变量布尔表达式异或完成，式中 $\bar{x}_1$ 表示取反操作，$\oplus$ 为异或操作。

$$f(x_1,x_2,\cdots,x_n)=x_1f(1,x_2,\cdots,x_n)\oplus\bar{x}_1f(0,x_2,\cdots,x_n) \tag{3}$$

最终确定 RSCLU 具备的功能包括：(1)每个 LUT 单独实现一个 4 变量布尔函数，即 RSCLU 实现 4 个 4 变量布尔函数；(2)两个 LUT 可通过输入控制选择，完成一个 5 变量布尔函数，即 RSCLU 可实现两个 5 变量布尔函数；(3)RSCLU 同时具备实现 6 输入布尔函数的能力。

在基本电路结构确定之前，需对 NLBF 的 5 变量及 4 变量表达式进行变量映射。RSCLU 可完成两个 5 变量或 4 个 4 变量布尔函数，分别对 NLBF 中任 4 个 5 变量及任 4 个 4 变量布尔表达式进行适配。对于 5 变量布尔函数，主要考察与项的共享变量情况，由于 5 变量布尔函数所占比率较少为 3.62%，事实上，NLBF 中的 5 变量布尔函数通常不超过 4 个，可采用直接遍历的方式。对 NLBF 中任两个 5 变量布尔函数进行搜索，可得到共享变量的分布，如表 3 所示，共享变量个数通常为 3 或 4 个。

表3 5变量布尔函数适配后特征分布

算法	f个数	个数	算法	f个数	个数
Ach-A0	2	4	Ach-A10	0	–
Ach-A1	3	3/4	Ach-A11	2	4
Ach-A2	4	4	Ach-A12	0	–
Ach-A3	0	–	Decim	0	–
Ach-A4	0	–	Dicing	4	3/4
Ach-A5	0	–	LILI-128	0	–
Ach-A6	5	3/4	Grain-80	2	0
Ach-A7	2	4	Grain-128	0	–
Ach-A8	2	4	Toy-hs1	0	–
Ach-A9	0	–	Trivium	0	–

而 NLBF 中 4 变量布尔函数的个数较多，若对任意的 4 个 4 变量布尔函数进行逐一对比，将造成适配效率大大降低，而同一个 NLBF 的布尔表达式遵循相同或相似的设计思想，布尔表达式间的共享变量也将具有相同或相似的分布。同时，由于各变量在布尔表达式中出现的频次不同，因此，利用变量频次的差异来简化映射过程。以 Ach-128 算法的 A3 为例说明适配流程。A3 的 4 变量布尔函数共有 12 个，图 1 左侧所示，右侧为适配过程：

(1)频次搜索：遍历所有 4 变量布尔函数得出变量频次，如黑色序号 1 所示，x_1，x_2，x_5，x_{15}，x_7，x_6，x_9，x_{14}，x_{16} 频次分别为 2, 8, 8, 6, 8, 3, 6, 3, 3。由于频次大的变量成为共享变量的可能性更大，为此选中频次为 8 的 x_2，x_5 作为共享变量(不选取 x_7 也作为共享变量的原因在于很可能搜索不到 4 个 4 变量布尔函数)；

(2)函数遍历：以 x_2，x_5 作为共享变量遍历 4 变量布尔函数，得到含有该共享变量的 4 个 4 变量表达式，如图 1 中(1)所示，将这 4 个 4 变量布尔表达式从表达式队列中删除，并更新变量频次，如黑色序号 2 所示。按照同样方式得到其余共享变量及对应 4 变量布尔函数。

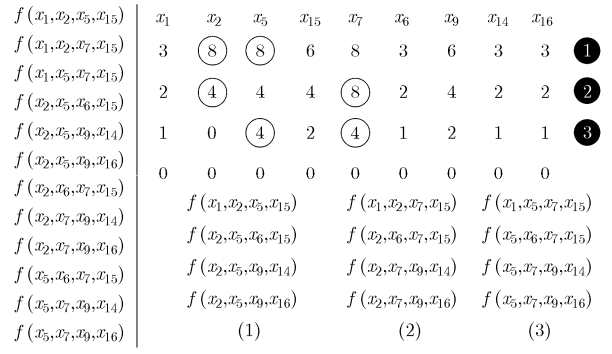


图 1　4 变量布尔函数适配流程示意

依据上述适配流程，Ach-128 算法 A3 中的 4 变量布尔函数可分为 3 组，每组均包含 4 个 4 变量布尔函数，图 1 中(1)、(2)和(3)所示，共享变量分别为 x_2, x_5，x_2, x_7 和 x_5, x_7。对其他 NLBF 进行适配，发现 4 变量布尔函数的共享变量特征为：(1)4 个布尔函数间通常至少有一个共享变量；(2)若将 4 个布尔函数分成 2 组，则每两个 4 变量布尔函数至少有 2 个共享变量。

3.2 运算单元结构

基于 LUT 的利用率模型，并通过 2.2 节对 NLBF 前期的处理结果，确定了影响运算单元的基本参数，结合 3.1 节不同类型布尔表达式间共享变量的研究，也明确了各 LUT 间的连接关系，可完成 RSCLU 基本结构的设计，如图 2 所示(图中选择器的控制信号未画出，圆圈表示取反)。

RSCLU 由 4 个相互级联的 LUT 构成，通过选择器的控制和 LUT 的配置信息可完成不同的函数功能。

(1)可以同时完成 4 个 4 变量布尔函数，表达式分别表示为 $f_0 = f(x_0, x_1, x_2, x_3)$，$f_1 = f(x_0, x_1, x_2, x_9)$，$f_1 = f(x_0, x_1, x_3, x_4)$ 或 $f_1 = f(x_0, x_1, x_4, x_9)$，$f_2 = f(x_0, x_5, x_6, x_7)$，$f_3 = f(x_0, x_5, x_6, x_9)$，$f_3 = f(x_0, x_5, x_7, x_8)$ 或 $f_3 = f(x_0, x_5, x_8, x_9)$；

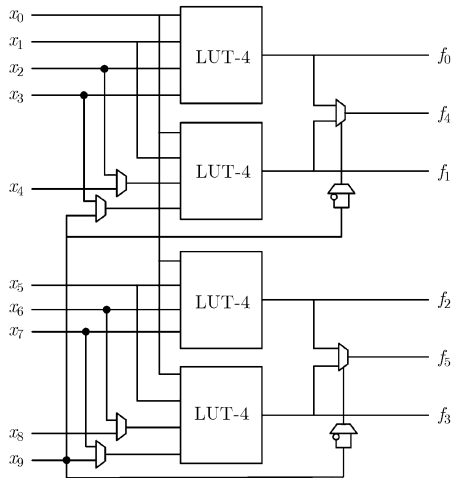


图 2 基于 LUT 的 RSCLU 结构

(2)可以同时完成两个 5 变量布尔函数，表达式可表示为 f_4=$f(x_0, x_1, x_2, x_3, x_9)$ 或 f_4=$f(x_0, x_1, x_2, x_3, \overline{x}_9)$，$f_5 = f(x_0, x_5, x_6, x_7, x_9)$ 或 $f_5 = f(x_0, x_5, x_6, x_7, \overline{x}_9)$；

(3)可以完成一个 6 变量布尔函数，依据式(3)采用两个 5 变量布尔函数在可编程异或输出时实现，表达式可表示为 $f = x_9 f(1, x_0, x_1, x_2, x_3, x_4) \oplus \overline{x}_9 f(0, x_0, x_1, x_2, x_3, x_4)$，此时 $x_5 = x_1$，$x_6 = x_2$，$x_7 = x_3$ 且 $x_8 = x_4$。

4 性能评估与分析

4.1 硬件性能评估

采用 Verilog 硬件描述语言进行 RTL 级实现，并在 SMIC 180 nm 工艺下，进行逻辑综合，各项性能指标如表 4 所示。在并行度为 32 时，延时为 4.14 ns，面积为 1097573.49 μm^2，等效两输入与非门约为 10.29 万，此时的时钟频率为 241 MHz。

表4　NLBF整体结构性能

并行度	面积 (μm^2)	等效门数 (万门)	延时 (ns)	频率 (MHz)
8	258536.76	2.58	3.21	311
16	515928.53	5.15	3.87	258
32	1097573.49	10.29	4.14	241

将系统性能与以往设计进行对比，如表 5 所示。文献[18]提出的可重构结构采用流水线设计，可在一定程度上提升工作频率，并行度为 1 时，最大时钟频率可达 360 MHz，但资源消耗过大，在并行度为 8 时等效门数约为 262.3 万门，若进行 32 路并行设计，则高达 1049 万门，最大吞吐率为 2.88 Gb/s。文献[19]和文献[6]依据可重构并行化设计原理，实现不同并行度的 NLBF 结构设计，系统性能均明显优于文献[18]，并行度为 32 时，等效门数分别为 41.51 万和 12.84 万，但其频率较文献[18]却有很大的下降，分别为 142 MHz 和 172 MHz，吞吐率最高为 4.54 Gb/s 和 5.50 Gb/s。本文提出的 NLBF 可重构运算结构立足于序列密码特点和 NLBF 的特征，在对比文献中所需电路资源最小，工作频率要略大于 RPNF 和 TNF 而小于 RNF，并行度 32 时，吞吐率可达 7.71 Gb/s。

4.2 资源利用率评估

由于 RSCLU 采用 4 个 LUT 级联的方式完成相应功能，在评估资源利用率时，依据文献[20]提出的 LUT 利用率等价转换公式来评估。

表5 性能对比结果

文献	发表时间	工艺	并行度(级数)	面积(mm^2)	等效门数(万门)	延时(ns)	时钟频率(MHz)	吞吐率(Gb/s)
RNF[16]	2003	180	8	26.23	262.30	2.73	360	2.88
			16	52.46	524.60	–		–
			32	104.92	1049.20	–		–
RPNF[18]	2009	180	8	1.04	10.49	6.06	142	1.14
			16	1.92	19.23	6.54		2.27
			32	4.14	41.51	7.03		4.54
TNF[2]	2013	180	8	0.34	3.48	5.30	172	1.37
			16	0.66	6.56	5.46		2.25
			32	1.28	12.84	5.79		5.50
本文	2016	180	8	0.26	2.58	3.21	241	1.93
			16	0.52	5.15	3.87		3.86
			32	1.03	10.29	4.14		7.71

$$\text{LUT利用率}=\frac{\text{LUT使用个数}}{\text{RSCLU使用个数}\times 4}\times 100\% \quad (4)$$

由于每个 RSCLU 中包含 4 个 LUT-4，故在计算利用率时将 RSCLU 的个数乘以 4。依据适配流程，对 NLBF 在 RSCLU 上进行映射，得出各 NLBF 在并行度为 1 时的资源消耗及利用率情况，如图 3 所示(图中只给出了部分典型 NLBF 的性能评估数据)。其中横坐标为密码算法或 NLBF 的名称，柱状图表示消耗的 RSCLU 的个数，曲线中黑色方框表示利用率变化(图 4 同)。

图 3 中不同 NLBF 消耗资源略有不同，其中 Toy-hs1 算法消耗的资源最多，为 13 个 RSCLU，其余算法或 NLBF 均不超过 5 个，且以 4 个居多。从利用率上可以看出，除 Trivium 算法资源利用率为 75.0%外，其余均在 80%以上，部分 NLBF 的利用率在理论上可达到 100%(利用率达到 100%表示 RSCLU 中的所有 LUT-4 均被使用)。

密码算法通常具有一定的并行度，对不同 NLBF 进行并行度分析，Ach-A0～A12 并行度至少为 8, Grain-80 并行度为 16, Grain-128 并行度为 32, Trivium 并行度最大为 64，由于本设计可支持并行度为 8, 16 和 32 的 NLBF 运算，即对不同并行度的

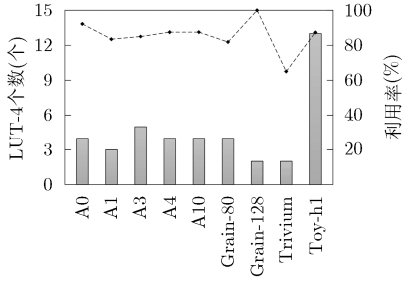


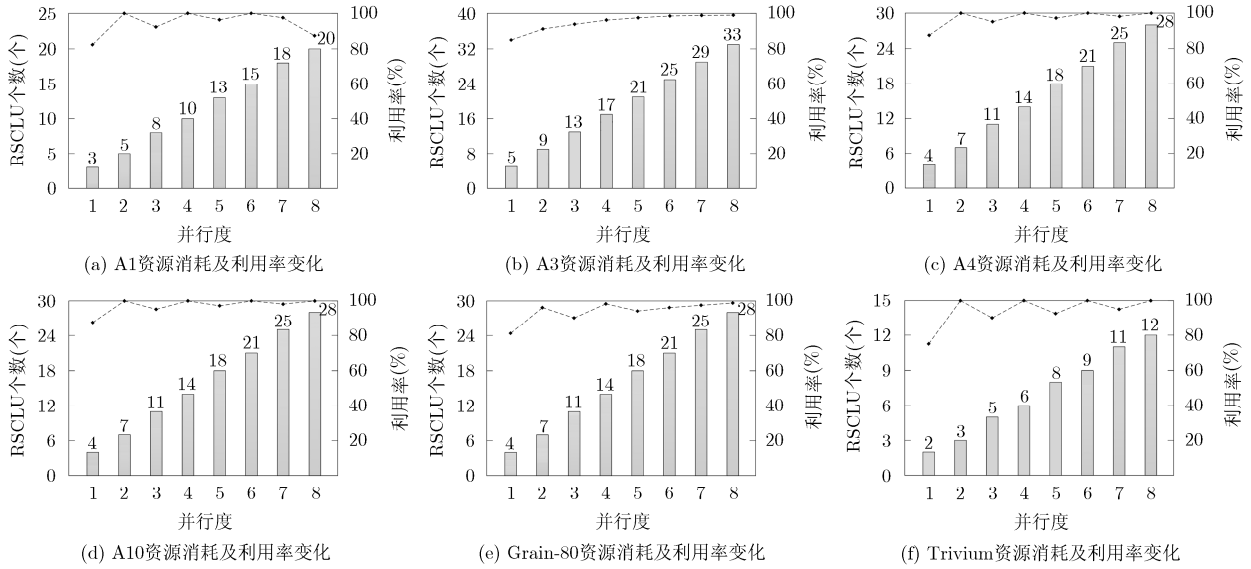
图3 不同NLBF资源及利用率

NLBF 具有一定的适应性，因此，可以对不同并行下的资源利用率进行评估。对资源利用率低于 91.14%[8]的 NLBF 在不同并行度下进行性能评估，各 NLBF 资源消耗和利用率变化情况如图 4 所示。

由图 4 可以看出，并行度大于 1 时，A1, A3, A4, 和 A10 的利用率均达到 91.14%以上，Grain-80 和 Trivium 在并行度大于 3 时利用率也超过 91.14%。并且，有一个现象值得关注：并行度为偶数时，许多 NLBF 的利用率达到 100%；并行度为奇数时，利用率呈逐渐增长的趋势。这是由于并行度为 1 时，依据适配结果，NLBF 需求的 RSCLU 中有两个 LUT-4 未使用，并行度为偶数时，恰好可完全被 NLBF 使用，利用率达到 100%；并行度为奇数时，未使用的 LUT-4 可以被部分使用，随着并行度增大，消耗的 RSCLU 资源必然越多，未使用的 LUT-4 个数将减少，由式(4)知道，资源利用率会呈增大趋势，同时 LUT-4 不能被完全使用，所以利用率无法达到 100%。

5 结束语

基于 LUT 利用率模型和搜索适配算法，本文对 NLBF 进行特征分析，确定了影响利用率的基本参数，完成面向序列密码的 NLBF 单元设计，可支持不同并行度，具有一定扩展性；将利用率模型与 NLBF 软件适配相结合的设计方式，对其他领域 NLBF 的设计具有一定的推广意义，如面向多媒体处理的 NLBF 运算可采用类似方式完成。同时，在研究过程中也发现一些问题：适配算法在适配时采取分函数类型的方式进行变量搜索，这给软件实现及 NLBF 适配增加了难度，需要找到一种不区分函数类型的高效适配算法。



(a) A1资源消耗及利用率变化　(b) A3资源消耗及利用率变化　(c) A4资源消耗及利用率变化

(d) A10资源消耗及利用率变化　(e) Grain-80资源消耗及利用率变化　(f) Trivium资源消耗及利用率变化

图 4 不同并行度下资源消耗及利用率变化

参 考 文 献

[1] ZENG G, DONG X, and BORNEMANN J. Reconfigurable feedback shift register based stream cipher for wireless sensor networks[J]. *IEEE Wireless Communications Letters*, 2013, 2(5): 559–562. doi: 10.1109/wcl.2013.13.130292.

[2] 禹思敏, 吕金虎, 李澄清. 混沌密码及其在保密通信中应用的进展[J]. 电子与信息学报, 2016, 38(3): 735–752. doi: 10.11999/JEIT151356.

YU Simin, LU Jinhu, and LI Chengqing. Some progresses of chaotic cipher and its application in multimedia secure communications[J]. *Journal of Electranics & Information Technology*, 2016, 38(3): 735–752. doi: 10.11999/JEIT151356.

[3] 丁群, 彭喜元, 杨自恒. 基于神经网络算法的组合序列密码芯片[J]. 电子学报, 2006, 34(3): 409–412. doi: 10.3321/j.issn:0372-2112.2006.03.006.

DING Qun, PENG Xiyuan, and YANG Ziheng. The cipher chip of combining stream based on the neural network algorithm[J]. *Acta Electronica Sinica,* 2006, 34(3): 409–412. doi: 10.3321/j.issn:0372-2112.2006.03.006.

[4] MIKE H and JAY S. Improving FPGA performance and area using an adaptive logic module[J]. *Leuven Belgium, Spring Berlin Heidelberg*, 2004, 32(03): 135–144. doi: 10.1007/978-3-540-30117-2_16.

[5] ANDERSON J H and QIANG W. Area-efficient FPGA logic elements: Architecture and synthesis[C]. 16th Asia and South Pacific Design Automation Conference, Yokohama, 2011: 369–375. doi: 10.1109/aspdac.2011.5722215.

[6] 陈韬, 杨萱, 戴紫彬, 等. 面向序列密码的非线性反馈移位寄存器可重构并行化设计[J]. 上海交通大学学报, 2013, 47(1): 28–32.

CHEN Tao, YANG Xuan, DAI Zibin, *et al*. Design of a reconfigurable parallel nonlinear feedback shift register structure targeted at stream cipher[J]. *Journal of Shanghai Jiao Tong University*, 2013, 47(1): 28–32.

[7] SATWANT S, JONATHAN R, PAUL C, *et al*. The effect of logic block architecture on FPGA performance[J]. *IEEE Journal of Solid-State Circuits*, 1992, 27(3): 281–287. doi: 10.1109/4.121549.

[8] ELIAS A and JONATHAN R. The effect of LUT and cluster size on deep-submicron FPGA performance and density[J]. *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*, 2004, 12(3): 288–298. doi: 10.1109/tvlsi.2004.824300.

[9] KUON L and JONATHAN R. Measuring the gap between FPGAs and ASICs[C]. Acm/sigda International Symposium on Field Programmable Gate Arrays, Monterey, 2015: 21–30. doi: 10.1145/1116201.1117205.

[10] MANA P, TALATI N, RISWADKAR A, *et al*. Stateful-NOR based reconfigurable architecture for logic implementation[J]. *Microelectronics Journal*, 2015, 46(6): 551–562. doi: 10.1016/j.mejo.2015.03.021.

[11] TANG X and WANG L. The effect of LUT size on nanometer FPGA architecture[C]. IEEE 11th International on Solid-State and Integrated Circuit Technology, Xi'an, 2012: 1–4. doi: 10.1109/icsict.2012.6467767.

[12] DICKIN D and SHANNON L. Exploring FPGA technology mapping for fracturable LUT minimization[C]. International Conference on Field-Programmable Technology, New Delhi, 2011: 1–8. doi: 10.1109/fpt.2011.6132691.

[13] FAROOQ U and ASLAM M. Design and implementation of basic building blocks of FPGA using memristor-transistor hybrid approach[C]. 2015 Fifth International Conference on Innovative Computing Technology, Vigo, 2015: 142–147. doi:

10.1109/intch.2015.7173484.

[14] ASLAM M H, FAROOQ U, AWAIS M, *et al*. Exploring the effect of LUT size on the area and power consumption of a novel memristor-transistor hybrid FPGA architecture[J]. *Arabian Journal for Science & Engineering*, 2016, 41(8): 3035–3049. doi: 10.1007/s13369-016-2068-8.

[15] TANG Xifan and DE-MICHELI G. Pattern-based FPGA logic block and clustering algorithm[P]. US, 20160063168. 2016. doi: 10.1109/fpl.2014.6927429.

[16] SAXENA S and TIWARI A. A comparative study of leakage reduction techniques used in, FGPA for optimized area and power consumption[J]. *International Journal of Engineering Research & Applications*, 2014, 4(2): 89–94.

[17] 王周闯, 戴紫彬, 李伟. 高效适配 NLBF 序列密码的全局定向搜索算法[J]. 计算机应用, 2016, 36(9): 65–69. doi: 10.11772/j.issn 1001-9081.2016.09.

WANG Zhouchuang, DAI Zibin, and LI Wei. Global directional search algorithm adapting NLBF sequence cryptogram efficiently[J]. *Journal of Computer Applications*, 2016, 36(9): 65–69. doi: 10.11772/j.issn 1001-9081.2016.09.

[18] 秦晓懿, 王瀚晟, 曾烈光. 线性和非线性寄存器系统的并行化技术[J]. 电子学报, 2003, 31(3): 406–410. doi: 10.3321/j.issn: 0372-2112.2003.03.023.

QIN Xiaoyi, WANG Hansheng, and ZENG Lieguang. Paralleling techniques for linear and nonlinear register systems[J]. *Acta Electronica Sinica*, 2003, 31(3): 406–410. doi: 10.3321/j.issn:0372-2112.2003.03.023.

[19] 李伟. 面向序列密码的反馈移位寄存器可重构并行化设计技术研究[D]. [硕士论文], 解放军信息工程大学, 2009.

LI Wei. Research on technology of reconfigurable parallel feedback shift register targeted at stream cipher[D]. [Master dissertation], PLA Information Engineering University, 2009.

[20] REBEIRO C and MUKHOPADHYAY D. High speed compact elliptic curve cryptoprocessor for FPGA platforms [C]. International Conference Progress in Cryptology-Indocrypt 2008, Kharagpur, 2008: 376–388. doi: 10.1007/978-3-540-89754-5_29.

戴紫彬： 男，1966 年生，教授，博士生导师，研究方向为专用集成电路设计、芯片安全防护、信息安全芯片技术研究等.

王周闯： 男，1992 年生，硕士生，研究方向为信息安全、专用集成电路设计、安全芯片设计等.

李 伟： 男，1983 年生，副教授，研究方向为体系结构、安全芯片设计、集成电路技术研究等.