JOURNAL OF ELECTRONIC TESTING: Theory and Applications,6,255-258 (1995)

Minimal Cost One-Dimensional Linear Hybrid Cellular Automata of Degree Through 500*

KEVIN CATTELL AND SHUJIAN ZHANG
Department of Computer Science, University of Victoria, Victoria, B.C, Canada V8W 3P6

Received July 5, 1994; Revised October 25, 1994

Editor: K.K. Saluja

Abstract. This letter is a supplement of the table of the minimal cost one-dimensional linear hybrid cellular automata with the maximum length cycle by Zhang, Miller, and Muzio *[IEE Electronics Letters,* 27(18):1625-1627, August 1991].

Keywords: linear feedback shift register, linear hybrid cellular automata, maximal length cycle, primitive polynomial

1 Introduction

Recently, one-dimensional linear hybrid cellular automata (LHCA) [1], [2] have been considered as alternatives to linear feedback shift registers (LFSRS) [3] for VLSI design and test, and in particular for Built-In Self-Test (BIST) [4]. When used as a test vector generator, an *n*-degree LFSR or LHCA with a primitive polynomial over GF(2) is desired since it yields a maximum length cycle. That is, the LFSR or LHCA in an autonomous mode of operation traverses all possible $(2^n - 1)$ non-zero states before returning to its initial non-zero state.

The LHCA considered are linear finite state machines (LFSMs), each composed of a one-dimensional array of cells. Cells are only able to communicate with their immediate neighbors. We examine only LHCA that consist of rule 90 and rule 150 cells, since it is shown in [2] that this is a necessary condition for the LHCA to have a maximum length cycle. The complete details of LHCA can be found in [1], [2].

In general, we should minimize the hardware cost of an LFSR or LHCA implementation. Fortunately, we can easily get an *n*-degree primitive LFSR with minimal cost for any practical use in BIST, since the minimal weight primitive polynomials of degree through 500 are contained in [3], [5]. The LFSRs are obtained via the one to one correspondence between n-degree LFSRs and polynomials of degree *n*. In this letter, we extend the table in [6], using an alternative method to produce the minimal-cost maximum-length LHCA. These LHCA, from degree 151 to 500, were not previously known.

2 Method and Result

The algorithm of determining whether a given *n*-degree LHCA has a maximum length cycle is as follows.
(a) Compute the characteristic polynomial of the LHCA using the recurrence relation in [2];
(b) Check if the characteristic polynomial is primitive; if so, the LHCA has the maximum length cycle.

For each degree, we first generate all of the LHCA with a single rule 150 cell. If this is not successful, we then generate all of the LHCA that have a pair of rule 150 cells. For each degree, the search is stopped at the first LHCA with the maximum length cycle. This search has never failed, meaning that for each degree up to 500, there is an LHCA with the maximal length cycle that has either one or two rule 150 cells.

* This work was supported in part by Research Grants and Postgraduate Scholarships from the Natural Sciences and Engineering Research Council of Canada and by an equipment loan from the Canadian Microelectronics Corporation.

The following table gives the maximum length LHCA with the minimal cost, one of each degree up to 500. For the completeness, we reproduce the table in [6] (the LHCA of degree 1 through 150). The entries in the table indicate which cells use rule 150. For example, the entry

(10) 2 7

represents a rule vector for an LHCA of degree 10

[0, 1, 0, 0, 0, 0, 1, 0, 0, 0],

where '1' denotes rule 150 and '0' denotes rule 90. That is, the LHCA has 10 cells[1], where cells 2 and 7 use rule 150 and the rest use rule 90.

Neither of the steps used in the algorithm to obtain the results is difficult. The recurrence in [2] allows the characteristic polynomial of an n-degree LHCA to be calculated with a linear (in n) number of polynomial operations. The polynomial, which has degree n, is then checked for irreducibility. This is done because

(a) if the polynomial is not irreducible (i.e., it is reducible), then it is not primitive, and

(b) irreducibility checking is significantly easier than primitivity checking.

The prime factorization (2^n-1) is required to determine whether a polynomial is primitive. Once these factors are known, the check for primitivity is straightforward, but may require a large number of polynomial operations. See [7] for a complete discussion of determining irreducibility and primitivity. A total of approximately three CPU days on a SPARC 10 cornputer is required to complete the table presented in this letter.

3 Conclusion

The proposed algorithm for determining whether a given n-degree LHCA has the maximum length cycle has been used to produce the minimal cost LHCA of degree up to 500 included in the letter. The contributed results will be useful for researchers working in the areas of VLSI design and test as well as other applications.

We have seen from the experimental results that for each degree n (<500), there exists an n-cell LHCA with the maximal length cycle that has at most two rule 150 cells. We conjecture that this is true in general.

Acknowledgments

The authors thank Dr. P.H. Bardell for providing prime factors of (2^n-1) which are used to determine whether a given polynomial of degree n is primitive, Dr. J.C. Muzio and Dr. D.M. Miller for proposing the work a few years ago, and the reviewers for their constructive comments.

References

1. K. Cattell and J.C. Muzio, "Synthesis of one-dimensional linear hybrid cellular automata," Submitted to *IEEE Transactions on Computer-Aided Design,* April 1994.
2. M. Serra, T. Slater, J.C. Muzio, and D.M. Miller, "The analysis of one-dimensional linear cellular automata and their aliasing properties," *IEEE Transactions on Computer-AidedDesign,* vol. 9, pp. 767-778, July 1990.
3. P.H. Bardell, W.H. McAnney, and J. Savir, *Built-In Test for VLSI: Pseudorandom Tecliniques,* John Wiley and Sons, 1987.
4. S. Zhang, R. Byme, J.C. Muzio, and D.M. Miller, "Why cellular automata are better than LFSRs as Built-In Self-Test generators for sequential-type faults," *Proceedings of IEEE International Symposium on Circuits and Systems,* pp. 69-72, May 1994.
5. P.H. Bardell, "Primitive polynomials of degree 301 through *500," Journal of Electronic Testing: Theory and Applications,* vol. 3, pp. 175-176, 1992.
6. S. Zhang, D.M. Miller, and J.C. Muzio, "Determination of minimal cost one-dimensional linear hybrid cellular automata," IEE *Electronics Letters,* vol. 27, no. 18, pp. 1625-1627, August 1991.
7. R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications,* Cambridge University Press, Cambridge, 1986.

Deg.	LHCA		Deg.	LHCA		Deg.	LHCA		Deg.	LHCA		Deg.	LHCA		Deg.	LHCA	
(1)		1	(51)		1	(101)	1	20	(151)		27	(201)	1	26	(251)		1
(2)		1	(52)	2	29	(102)		33	(152)		40	(202)		24	(252)	4	31
(3)		1	(53)		1	(103)		15	(153)		13	(203)		7	(253)		93
(4)	1	3	(54)		9	(104)	2	40	(154)		66	(204)	1	103	(254)		1
(5)		1	(55)		17	(105)		1	(155)		1	(205)		57	(255)		55
(6)		1	(56)	4	14	(106)		30	(156)	2	24	(206)	1	71	(256)	1	127
(7)		3	(57)		9	(107)		19	(157)		33	(207)	1	130	(257)		49
(8)	2	3	(58)		17	(108)	1	35	(158)		1	(208)		74	(258)	1	181
(9)		1	(59)	4	15	(109)	1	4	(159)		61	(209)		1	(259)		21
(10)	2	7	(60)	2	38	(110)		13	(160)	1	151	(210)		1	(260)	4	89
(11)		1	(61)	1	10	(111)		27	(161)	1	116	(211)		17	(261)		1
(12)	3	7	(62)		5	(112)	2	5	(162)		73	(212)	2	169	(262)	3	189
(13)		5	(63)		31	(113)		1	(163)	1	24	(213)	1	56	(263)		95
(14)		1	(64)	3	5	(114)		22	(164)		26	(214)		59	(264)	2	173
(15)		3	(65)		1	(115)		41	(165)	3	130	(215)		29	(265)		69
(16)	1	15	(66)	1	19	(116)		16	(166)		72	(216)	4	26	(266)		46
(17)		5	(67)		15	(117)		33	(167)	1	60	(217)		53	(267)		103
(18)	1	17	(68)		8	(118)		30	(168)	1	113	(218)		50	(268)	1	73
(19)		3	(69)		1	(119)		1	(169)		81	(219)		49	(269)	1	210
(20)	2	3	(70)	1	37	(120)	3	73	(170)		22	(220)		47	(270)		81
(21)	1	10	(71)		17	(121)		45	(171)		37	(221)		1	(271)		5
(22)		5	(72)	6	55	(122)		14	(172)		3	(222)	2	12	(272)	3	109
(23)		1	(73)		9	(123)		51	(173)		1	(223)		9	(273)		1
(24)	8	12	(74)		1	(124)		21	(174)		16	(224)		64	(274)	2	29
(25)		9	(75)		7	(125)		13	(175)	2	71	(225)		33	(275)		103
(26)		1	(76)	2	22	(126)		40	(176)	1	45	(226)	1	33	(276)		81
(27)	1	20	(77)	3	44	(127)		15	(177)	1	22	(227)		109	(277)	1	72
(28)		3	(78)	1	41	(128)	1	29	(178)	1	79	(228)		6	(278)		44
(29)		1	(79)		9	(129)		49	(179)		1	(229)		21	(279)		69
(30)		1	(80)	1	71	(130)	1	27	(180)		19	(230)		1	(280)	2	47
(31)		11	(81)		1	(131)		1	(181)		81	(231)		1	(281)		1
(32)	1	15	(82)	1	69	(132)		18	(182)		34	(232)		93	(282)	3	19
(33)		1	(83)		1	(133)	1	4	(183)		1	(233)		1	(283)		35
(34)	1	19	(84)		36	(134)		26	(184)	2	163	(234)	1	145	(284)		49
(35)		1	(85)	1	46	(135)		1	(185)		61	(235)		69	(285)	1	208
(36)		6	(86)		1	(136)	1	97	(186)	1	125	(236)		61	(286)	2	167
(37)		9	(87)		13	(137)	1	132	(187)		45	(237)		33	(287)		109
(38)		7	(88)		5	(138)		28	(188)	2	10	(238)		35	(288)	1	61
(39)		1	(89)		1	(139)		11	(189)		1	(239)		1	(289)		101
(40)		8	(90)		1	(140)		8	(190)	2	8	(240)	3	7	(290)		112
(41)		1	(91)		15	(141)		25	(191)		1	(241)	1	162	(291)		97
(42)		19	(92)	3	71	(142)		5	(192)	3	111	(242)		4	(292)		5
(43)		3	(93)		33	(143)		35	(193)		9	(243)		1	(293)		1
(44)	4	26	(94)		42	(144)		13	(194)		14	(244)	2	49	(294)	1	79
(45)		9	(95)		1	(145)	1	46	(195)		19	(245)		1	(295)		57
(46)	2	10	(96)		6	(146)		1	(196)		32	(246)		40	(296)		2
(47)		13	(97)	1	82	(147)	1	136	(197)		65	(247)	1	102	(297)	3	84
(48)		15	(98)		8	(148)		8	(198)	1	145	(248)	1	185	(298)		111
(49)	1	10	(99)		13	(149)	1	108	(199)		41	(249)		81	(299)		1
(50)		11	(100)	1	67	(150)	2	102	(200)		11	(250)	1	217	(300)	1	251

Deg.	**LHCA**		**Deg.**	**LHCA**		**Deg.**	**LHCA**		**Deg.**	**LHCA**		**Deg.**	**LHCA**		**Deg.**	**LHCA**	
(301)		81	(335)	1	12	(369)	1	74	(403)		179	(437)		5	(471)	2	185
(302)	2	105	(336)	4	35	(370)		75	(404)		68	(438)		187	(472)	2	212
(303)		1	(337)	1	100	(371)		1	(405)		117	(439)		171	(473)		1
(304)	4	130	(338)	1	41	(372)	1	257	(406)		116	(440)	4	135	(474)		61
(305)		73	(339)	2	9	(373)		5	(407)		95	(441)		1	(475)		45
(306)		1	(340)	2	37	(374)	2	313	(408)	2	239	(442)		138	(476)		25
(307)		123	(341)		145	(375)		1	(409)	4	97	(443)		1	(477)	1	266
(308)		14	(342)		69	(376)		62	(410)	3	181	(444)		184	(478)		86
(309)		1	(343)		99	(377)		149	(411)		1	(445)		197	(479)	1	140
(310)	1	43	(344)		13	(378)	2	12	(412)		185	(446)	1	443	(480)	8	239
(311)		7	(345)		49	(379)		33	(413)		1	(447)		223	(481)		225
(312)	1	295	(346)		32	(380)	4	102	(414)	4	137	(448)	1	153	(482)	1	285
(313)		57	(347)		13	(381)	1	44	(415)	1	276	(449)		209	(483)		1
(314)	3	41	(348)	1	25	(382)		138	(416)	1	219	(450)	2	137	(484)	3	237
(315)		123	(349)		117	(383)		77	(417)		129	(451)		11	(485)	1	272
(316)	1	25	(350)	2	63	(384)	1	215	(418)		36	(452)		29	(486)	1	181
(317)	2	127	(351)	1	64	(385)		161	(419)		1	(453)		1	(487)		159
(318)		16	(352)	1	145	(386)		1	(420)	1	323	(454)		18	(488)		188
(319)		21	(353)		97	(387)	1	176	(421)	1	108	(455)		31	(489)		109
(320)	3	79	(354)		24	(388)	1	339	(422)	1	21	(456)	2	209	(490)	2	287
(321)		97	(355)		69	(389)		89	(423)	2	221	(457)	1	220	(491)		1
(322)	2	74	(356)		40	(390)	2	56	(424)	2	80	(458)		91	(492)		15
(323)		1	(357)		73	(391)	1	252	(425)		37	(459)		19	(493)	1	184
(324)		19	(358)		9	(392)		173	(426)		1	(460)		32	(494)	2	3
(325)		33	(359)		1	(393)		33	(427)	2	91	(461)	1	90	(495)		1
(326)		1	(360)	1	97	(394)		59	(428)		184	(462)		138	(496)	3	69
(327)		25	(361)	1	282	(395)	1	86	(429)		1	(463)		33	(497)	1	200
(328)		80	(362)	2	154	(396)	1	113	(430)	3	93	(464)		82	(498)		63
(329)		1	(363)		51	(397)		113	(431)		1	(465)	1	202	(499)	1	174
(330)		1	(364)		27	(398)		1	(432)	1	49	(466)		90	(500)	2	78
(331)		21	(365)		85	(399)		61	(433)		45	(467)		95			
(332)	1	65	(366)	1	65	(400)		198	(434)		86	(468)	2	378			
(333)	1	52	(367)		93	(401)		185	(435)	2	281	(469)		113			
(334)		104	(368)		139	(402)		154	(436)		17	(470)		1			

[1] Here, we number the cells 1 to n from left to right. In fact, we can get another configuration by the reversal [I].