GENERALIZED FIRING SQUAD SYNCHRONIZATION PROTOCOLS FOR ONE-DIMENSIONAL CELLULAR AUTOMATA — A SURVEY*

Hiroshi Umeo†, Naoki Kamikawa, Kouji Nishioka

University of Osaka Electro-Communication
Neyagawa-shi, Hatsu-cho, 18-8, 572-8530, Osaka, Japan

Shunsuke Akiguchi

Toyama National College of Maritime Technology
Imizu-shi, Ebie-Neriya, 1-2, 933-0293, Toyama, Japan

(Received February 18, 2010)

In the present article, we firstly examine the state transition rule sets for the generalized firing squad synchronization algorithms that give a finite-state protocol for synchronizing large-scale cellular automata. We focus on the fundamental generalized firing squad synchronization algorithms studied in recent fifty years, each operating in optimum- or non-optimum-steps on one-dimensional cellular arrays. A new eight-state algorithm is proposed. The eight-state optimum-step algorithm is the smallest one known at present in the class of generalized optimum-step firing squad synchronization protocols. A six-state non-optimum-step algorithm is also examined. We also construct a survey of the generalized synchronization algorithms and compare transition rule sets with respect to the number of internal states of each finite state automaton, the number of transition rules realizing the synchronization, and the number of state-changes on the array.

PACS numbers: 89.70.–a

1. Introduction

Cellular automata are considered to be a nice model of complex systems in which an infinite one-dimensional array of finite state machines (cells) updates itself in synchronous manner according to a uniform local rule. We

* Presented at the Summer Solstice 2009 International Conference on Discrete Models of Complex Systems, Gdańsk, Poland, June 22–24, 2009.

† Corresponding e-mail address: umeo@cyt.osakac.ac.jp

study a synchronization problem of giving a finite-state protocol for synchronizing a large scale of cellular automata. Synchronization of general network is a computing primitive of parallel and distributed computations. The synchronization in cellular automata has been known as *firing squad synchronization problem* (FSSP) since its development, in which it was originally proposed by J. Myhill in Moore [1964] to synchronize all parts of self-reproducing cellular automata. The problem has been studied extensively for more than 40 years.

In the present article, we firstly examine the state transition rule sets for the *generalized* firing squad synchronization algorithms that give a finite-state protocol for synchronizing large-scale one-dimensional cellular automata, where the term "*generalized*" means that an initial general can be located at any cell of the array. As a special case of the generalized firing squad synchronization algorithms, we have several FSSP algorithms that work only for one-dimensional arrays with a general at one end. They are Waksman's 16-state algorithm (Waksman [1966]), Balzer's 8-state algorithm (Balzer [1967]), Gerken's 7-state algorithm (Gerken [1987]), and Mazoyer's 6-state algorithm (Mazoyer [1987]). See Umeo, Hisaoka, and Sogabe [2005] for details. On the hand, many synchronization algorithms for two-dimensional arrays have been proposed, but here we do not treat them. See Umeo [2008].

In this paper, we focus on the fundamental generalized synchronization algorithms operating in optimum- and non-optimum-steps on one-dimensional cellular arrays. The generalized synchronization algorithms discussed herein are the first seventeen-state algorithm proposed by Moore and Langdon (Moore, Langdon [1968]), VMP ten-state algorithm (Varshavsky, Marakhovsky, Peschansky [1970]), Szwerinski's ten-state algorithm (Szwerinski [1982]), Settle and Simon's nine-state algorithm (Settle, Simon [2002]), UHMNM nine-state algorithm (Umeo *et al.* [2002]), and a new eight-state algorithm proposed in this paper. The eight-state optimum-step algorithm is the smallest one known at present in the class of generalized optimum-step firing squad synchronization protocols. A six-state non-optimum-step algorithm proposed by Umeo, Maeda, and Hongyo (Umeo, Maeda, Hongyo [2006]) is also examined. In addition, we construct a survey of the generalized synchronization algorithms and compare transition rule sets with respect to the number of internal states of each finite state automaton, the number of transition rules realizing the synchronization, and the number of state-changes on the array. Several new results and viewpoints are also given.

Specifically, we attempt to answer the following questions:

- First, what is the local transition rule set for those generalized FSSP algorithms?

- Are all previously presented transition rule sets correct?
- Do these rule sets contain redundant rules? If so, what is the exact rule set?
- How do the algorithms compare with each other?

In order to answer these questions, we implement all transition rule sets for the generalized synchronization algorithms mentioned above on a computer and check whether these rule sets yield successful firing configurations for any array of length n and any position k of the general from left end such that $2 \leq n \leq 250$ and $1 \leq k \leq n$.

2. Firing squad synchronization problem

2.1. Definition of the generalized firing squad synchronization problem

The firing squad synchronization problem (FSSP, for short) is formalized in terms of the model of cellular automata. Consider a one-dimensional array of finite state automata. All cells (except the end cells) are identical finite state automata. The array operates in lock-step mode such that the next state of each cell (except the end cells) is determined by both its own present state and the present states of its right and left neighbors. All cells (*soldiers*), except one *general* cell, are initially in the *quiescent* state at time $t = 0$ and have the property whereby the next state of a quiescent cell having quiescent neighbors is the quiescent state. At time $t = 0$ the *general* cell is in the *fire-when-ready* state, which is an initiation signal to the array. The *original* FSSP is stated as follows: Given an array of n identical cellular automata, including a *general* on the left end which is activated at time $t = 0$, we want to give the description (state set and next-state transition function) of the automata so that, *at some future time*, all of the cells will *simultaneously* and, *for the first time*, enter a special *firing* state. The initial general is on the left end of the array in the original FSSP.

The *generalized* FSSP (GFSSP, for short) studied in this paper is an extended version which allows the initial general to be located at any cell. Figure 1 shows a finite one-dimensional cellular array consisting of n cells, denoted by C_i, where $1 \leq i \leq n$. The set of states and the next-state transition function must be independent of n. Without loss of generality,

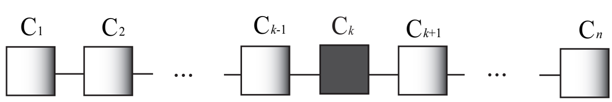

Fig. 1. One-dimensional cellular automaton with a general on C_k.

we assume $n \geq 2$. The tricky part of the problem is that the same kind of soldier having a fixed number of states must be synchronized, regardless of the position of the general and the length n of the array.

2.2. A brief history of the developments of optimum-time firing squad synchronization algorithms on one-dimensional arrays

The problem known as the FSSP was devised in 1957 by J. Myhill, and first appeared in print in a paper by E.F. Moore (Moore [1964]). This problem has been widely circulated, and has attracted much attention. The FSSP first arose in connection with the need to simultaneously turn on all parts of a self-reproducing machine. The problem was first solved by J. McCarthy and M. Minsky who presented a $3n$-step algorithm for n cells. In 1962, the first optimum-time, *i.e.* $(2n-2)$-step, synchronization algorithm was presented by Goto (Goto [1962]), with each cell having several thousands of states. Waksman (Waksman [1966]) presented a 16-state optimum-time synchronization algorithm. Afterward, Balzer (Balzer [1967]) and Gerken (Gerken [1987]) developed an eight-state algorithm and a seven-state synchronization algorithm, respectively, thus decreasing the number of states required for the synchronization. In 1987, Mazoyer (Mazoyer [1987]) developed a six-state synchronization algorithm which, at present, is the algorithm having the fewest states. Those algorithms mentioned above have been developed for one-dimensional arrays with the initial general at one end of the array. See Umeo, Hisaoka, Sogabe [2005] for the details of a survey on the original FSSP algorithms. On the other hand, the generalized FSSP has been also studied. Moore and Langdon (Moore and Langdon [1968]) first considered the problem and presented a 17-state optimum-time algorithm, *i.e.* $n-2+\max(k, n-k+1)$ steps for n cells with the general on the kth cell from left end of the array. Afterwards, Varshavsky, Marakhovsky, Peschansky [1970], Szwerinski [1982], Settle, Simon [2002], Umeo *et al.* [2002] presented ten-state and nine-state synchronization algorithms, respectively, thus decreasing the number of states required for the synchronization in GFFSP. A six-state non-optimum-step GFSSP algorithm was also proposed by Umeo, Maeda, Hongyo [2006].

2.3. Complexity measures and properties for synchronization algorithms

- **Time**

  Any solution to the original FSSP with the general at one end can be easily shown to require $(2n-2)$ steps for firing n cells, since signals on the array can propagate no faster than one cell per step, and the time from the general's instruction until the firing must be at least $2n-2$. See Balzer [1967], Mazoyer [1987] and Waksman [1966] for a proof. On the other hand, as for the GFSSP, it has been shown to be impossible to synchronize any array of length n in less than $n-2+\max(k, n-k+1)$ steps, where k is any integer such that $1 \leq k \leq n$ and the general is located on C_k. developed firstly a generalized optimum-time synchronization algorithm with 17 internal states that can synchronize any array of length n at exactly $n-2+\max(k, n-k+1)$ steps. See Moore, Langdon [1968] for details.

Theorem 1 (Moore, Langdon [1968]) (Lower Bounds) The minimum time in which the generalized firing squad synchronization could occur is $n-2+\max(k, n-k+1)$ steps, where the general is located on the kth cell from left end.

Theorem 2 (Moore, Langdon [1968]) There exists a 17-state cellular automaton that can synchronize any one-dimensional array of length n in optimum $n-2+\max(k, n-k+1)$ steps, where the general is located on the kth cell from left end.

- **Number of states**

The following three distinct states: the *quiescent* state, the *general* state, and the *firing* state, are required in order to define any cellular automaton that can solve the firing squad synchronization problem. Note that the boundary state for C_0 and C_{n+1} is not generally counted as an internal state. Balzer (Balzer [1967]) and Sanders (Sanders [1994]) showed that no four-state optimum-time solution exists. Umeo and Yanagihara (Umeo, Yanagihara [2009]), Yunès (Yunès [2008]), Umeo, Yunès and Kamikawa (Umeo, Yunès, Kamikawa [2008], Umeo, Kamikawa, Yunès [2009]) gave some 5- and 4-state *partial* solutions that can solve the synchronization problem for infinitely many sizes n, but not all, respectively. The solution is referred to as *partial* solution, which is compared with usual *full* solution that can solve the problem for all cells. Thus, the question that remains is: "What is the minimum number of states for an optimum-time full solution of the problem?" At present, that number is *five* or *six*.

Theorem 3 (Balzer [1967], Sanders [1994]) There is no four-state *full* solution that can synchronize n cells.

Berthiaume, Bittner, Perković, Settle and Simon (Berthiaume *et al.* [2004]) considered the state lower bound on ring-connected cellular automata. It is shown that there exists no three-state solution and no four-state symmetric solution for rings. The definition of the symmetry in transition rules can be found below.

Theorem 4 (Berthiaume *et al.* [2004]) There is no four-state symmetric optimum-time *full* solution for ring cellular automata.

Yunès [2008] and Umeo, Yunès, Kamikawa [2008], Umeo, Kamikawa, Yunès [2009] developed 4-state partial solutions based on Wolfram's rules 60 and 150. They can synchronize any array/ring of length $n = 2^k$ for any positive integer k. Details can be found in Yunès [2008], Umeo, Yunès, Kamikawa [2008] and Umeo, Kamikawa, Yunès [2009].

Theorem 5 (Yunès [2008], Umeo, Yunès, Kamikawa [2008], Umeo, Kamikawa, Yunès [2009]) There exist 4-state *partial* solutions to the firing squad synchronization problem for the rings.

- **Number of transition rules**

  Any k-state (excluding the boundary state) transition table for the synchronization has at most $(k-1)k^2$ entries in $(k-1)$ matrices of size $k \times k$. The number of transition rules reflects a complexity of synchronization algorithms.

- **Filled-in ratio**

  To measure the density of entries in the transition table, we introduce a measure *filled-in ratio* of the state transition table. The filled-in ratio of the state transition table $\mathcal{A}$ is defined as follows: $f_{\mathcal{A}} = e/e_{\text{total}}$, where e is the number of exact entries of the next state defined in the table $\mathcal{A}$ and e_{total} is the number of possible entries defined such that $e_{\text{total}} = (k-1)k^2$, where k is the number of internal states of the table $\mathcal{A}$.

- **Symmetry vs. asymmetry**

  Herman [1971, 1972] investigated the computational power of symmetrical cellular automata, motivated by a biological point of view. Szwerinski [1985] and Kobuchi [1987] considered a computational relation between symmetrical and asymmetrical CAs with von Neumann neighborhood. A transition table is said to be *symmetric* if and only if the transition table $\delta : \mathcal{Q}^3 \to \mathcal{Q}$ such that $\delta(x, y, z) = \delta(z, y, x)$ holds, for any state x, y, z in $\mathcal{Q}$. A symmetrical cellular automaton has a property that the next state of a cell depends on its present state and the states of its two neighbors, but it is same if the states of the left and right neighbors are interchanged. Thus, the symmetrical CA has no ability to distinguish between its left and right neighbors.

- **State-change complexity**

  Vollmar [1982] introduced a *state-change complexity* in order to measure the efficiency of cellular automata, motivated by energy consumption in certain physical memory systems. The state-change complexity is defined as the sum of *proper* state changes of the cellular space during the computations. Vollmar [1982] showed that $\Omega(n \log n)$ state-changes are required by the cellular space for the synchronization of n cells in $(2n-2)$ steps. Gerken [1987] presented an optimum-time $\Theta(n \log n)$ state-change synchronization algorithm.

  Theorem 6 (Vollmar [1982]) $\Omega(n \log n)$ state-change is necessary for synchronizing n cells.

  Theorem 7 (Gerken [1987]) $\Theta(n \log n)$ state-change is sufficient for synchronizing n cells in $2n-2$ steps.

2.4. Overview of the generalized firing squad synchronization algorithms

Figure 2 (left) is a space-time diagram for the original FSSP on which most of the optimum-time synchronization algorithms have been developed. The general at time $t = 0$ emits an infinite number of signals which propagate at $1/(2^{\ell+1}-1)$ speed, where ℓ is positive integer. These signals meet with a reflected signal at half point, quarter points, . . . , *etc.*, denoted by • in Fig. 2. It is noted that these cells indicated by • are synchronized. By increasing the number of "pre-synchronized" cells (not in firing state) exponentially, eventually all of the cells are synchronized at the last stage for the first time. A key idea behind the generalized FSSP algorithm proposed by Moore and Langdon [1968] is to reconstruct the original FSSP algorithm as if an initial general had been at the left or right end with being in the general state at time $t = -(k-1)$, where k is the number of cells between the general and the nearest end. Figure 2 (right) illustrates a space-time diagram for the generalized FSSP. The initial general emits a left- and right-going signal with 1/1 speed and keeps its position by marking a special symbol. The propagated signals generate a new general at each end. On reaching the end, they generate the necessary signals assuming that that end is the far end. The special marking symbol tells the first 1/1 signal generated by the left and right end generals that that side was the just nearest end. At that point the slope 1/1 signal is generated and it changes the slope of all the preceding signals to the next higher one, that is, $1/(2^{\ell}-1)$ becomes $1/(2^{\ell+1}-1)$. Note that the original optimum-time solution is working below the dotted line in Fig. 2 (right). All of the GFSSP algorithms presented in this paper are based on the space-time diagram shown in Fig. 2. Therefore, the optimum-time complexity for GFSSP is $\min(k-1, n-k)$ steps smaller than the original FSSP with a general at one end. Thus, the time complexity is $2n-2-\min(k-1, n-k) = n-1+\max(k-1, n-k) = n-2+\max(k, n-k+1)$.

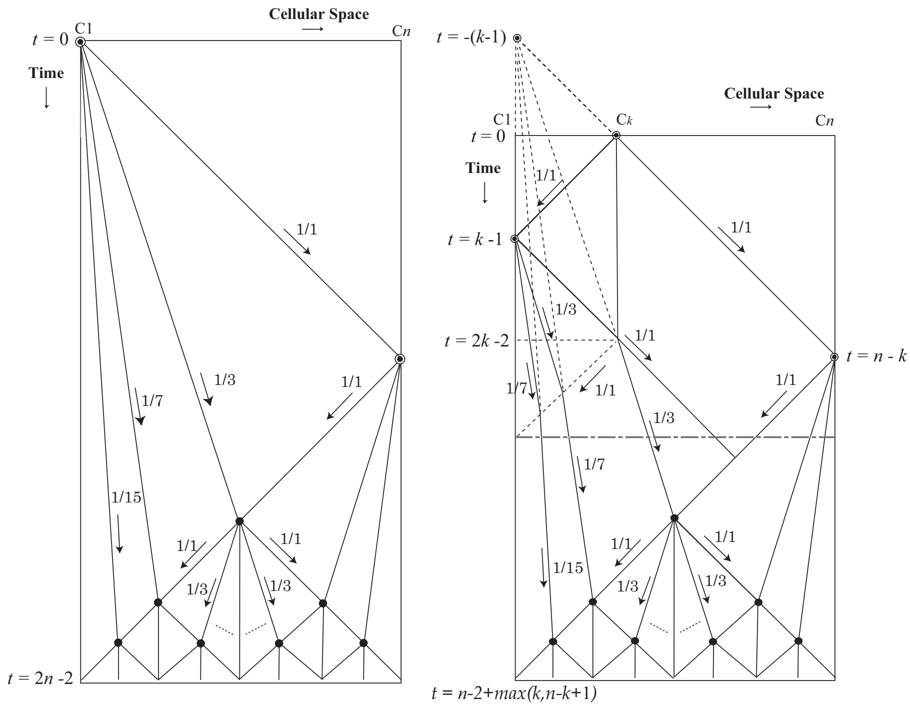

Fig. 2. Space-time diagram for the original (left) and the generalized (right) firing squad synchronization algorithms on one-dimensional array of length n.

3. State transition tables for generalized FSSP

In this section, we examine the state transition rule sets for the generalized firing squad synchronization protocols developed so far. Each state on the first row (column) indicates a state of right (left) neighbor cell, respectively. Each entry of the sub-tables shows a state at the next step. The state "*" that appears in the state transition table is a border state for the left and right end cells. It is noted that according to the conventions in FSSP, the border state "*" is not counted in the number of states. We have tested the validity of those tables for any array of length n and any position k of the general from the left end such that $2 \leq n \leq 250$ and $1 \leq k \leq n$. It reveals that all of the rule tables tested in this paper include no redundant rules.

3.1. Moore and Langdon's optimum-time 17-state table

Moore and Langdon [1968] constructed firstly a 17-state transition table that can synchronize any array in optimum-step. The transition table presented in (Moore, Langdon [1968]) has a compressed description that can be extended in a usual 4-tuple `W X Y` $\rightarrow$ `Z` which represents a state transition rule that an automaton in currently in state `X`, with its left neighbor in state `W` and the right neighbor in state `Y` will enter state `Z` at the next step. To

Q	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	Q	G	L	L	Q	Q	Q	Q	R	Q	G	Q	Z	Q	Q	Z	Q
K	I	K			X	I	I	I		I			I			I	
P	S																K
L	Q	Y				Q							Z			Z	
S	S																K
D	Q	G			Q						G						
A	Q	G					Q		R		G	Q		Q	Q		
B	Q	G						Q	R	Q	G				Q		
R	Q	G						Q			G						
H	H						H	H									
G	Q						Q										
I	I					I	I	I		I			I			I	
Y	Q						Q										
X	W	G			W						G						
W	W	G			W						G						
Z	Q						Q	Q									
*	Q		K	K													

K	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	K	K								K							K
K	K	T					K		K			K					T
P																	
L																	K
S																	
D																	K
A		K					K			K							K
B																	
R	K						K			K							
H		K							K								K
G		K										K					K
I																	
Y																	K
X																	
W																	
Z																	
*	K	T			K	K	K		K			K		K			

S	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	Q																
K	Q																
P																	
L																	
S																	
D	Q																
A																	
B																	
R																	
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*																	

D	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	D										Y						
K		K			X												
P																	
L		Y			D												
S																	
D																	
A																	
B																	
R																	
H																	
G																	
I	X										K						
Y																	
X																	
W																	
Z																	
*																	

R	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	Q	G						Q			G	Q		Q	Q		
K		K						A				A		A			
P																	
L																	
S																	
D																	
A	B										K				B		
B	A										K				A		
R																	
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*																	

H	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	Q						B	A									
K	I	K															
P																	
L																	
S																	
D																	
A																	
B	Q	A															
R																	
H																	
G	Q	A															
I	I						K	K									
Y	Q	A															
X																	
W																	
Z	Q						B	A									
*																	

Fig. 3. Transition table for Moore and Langdon's 17-state protocol (to be continued).

implement their transition table on a computer for its checking, we extend the compressed table into a set of 4-tuples. However, they yield unsuccessful synchronizations for relatively large number of arrays with several positions of the initial general.

P	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	D																K
K																	
P																	
L																	
S																	
D																	
A																	
B																	
R																	
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*	K																

Y	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	A	A								B							
K																	
P																	
L																	
S																	
D																	
A																	
B																	
R																	
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*																	

A	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	A	A							R		K				A		
K	A	K						A	R		K				A		
P																	
L																	
S																	
D																	
A																	
B		A															
R																	
H	H	H															
G																	
I	K	K															
Y																	
X																	
W																	
Z	A	A															
*																	

X	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	A																
K	A																
P																	
L																	
S																	
D																	
A																	
B																	
R	B																
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*																	

G	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	H	H								Q							
K																	
P																	
L																	
S																	
D	H	H								Q							
A	K	K								Q							
B	H	H								K							
R	H	H								Q							
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*																	

W	Right State																
Left State	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	R																
K																	
P																	
L																	
S																	
D																	
A	R																
B	R																
R	Q																
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*																	

Fig. 4. Transition table for Moore and Langdon's 17-state protocol (to be continued).

L (Left State \ Right State)	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	Q	Q				Q											
K																	
P																	
L																	
S																	
D																	
A																	
B																	
R																	
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*																	

B (Left State \ Right State)	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	B						B		Q	B	K				B		
K																	
P																	
L																	
S																	
D																	
A	B								Q		K				B		
B																	
R	B										K						
H	Q						Q										
G																	
I	K						K			K							
Y																	
X																	
W																	
Z	B						B										
*																	

I (Left State \ Right State)	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	R					R	K	R		R							
K	R					R	K	R		R							
P																	
L																	
S																	
D																	
A																	
B																	
R	Q					Q	Q	K		Q							
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*																	

Z (Left State \ Right State)	Q	K	P	L	S	D	A	B	R	H	G	I	Y	X	W	Z	*
Q	H						H	H		Q							
K																	
P																	
L																	
S																	
D																	
A																	
B																	
R																	
H																	
G																	
I																	
Y																	
X																	
W																	
Z																	
*																	

Fig. 5. Transition table for Moore and Langdon's 17-state protocol (continued).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	Q	Q	Q	Q	Q	Q	Q	Q	P	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
1	Q	Q	Q	Q	Q	Q	Q	L	D	S	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
2	Q	Q	Q	Q	Q	Q	L	Q	D	Q	S	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
3	Q	Q	Q	Q	Q	L	Q	Q	D	Q	Q	S	Q	Q	Q	Q	Q	Q	Q	Q	Q
4	Q	Q	Q	Q	L	Q	Q	Q	D	Q	Q	Q	S	Q	Q	Q	Q	Q	Q	Q	Q
5	Q	Q	Q	L	Q	Q	Q	Q	D	Q	Q	Q	Q	S	Q	Q	Q	Q	Q	Q	Q
6	Q	Q	L	Q	Q	Q	Q	Q	D	Q	Q	Q	Q	Q	S	Q	Q	Q	Q	Q	Q
7	Q	L	Q	Q	Q	Q	Q	Q	D	Q	Q	Q	Q	Q	Q	S	Q	Q	Q	Q	Q
8	K	Q	Q	Q	Q	Q	Q	Q	D	Q	Q	Q	Q	Q	Q	Q	S	Q	Q	Q	Q
9	K	I	Q	Q	Q	Q	Q	Q	D	Q	Q	Q	Q	Q	Q	Q	Q	S	Q	Q	Q
10	K	R	I	Q	Q	Q	Q	Q	D	Q	Q	Q	Q	Q	Q	Q	Q	Q	S	Q	Q
11	K	A	Q	I	Q	Q	Q	Q	D	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	S	Q
12	K	A	Q	R	I	Q	Q	Q	D	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	K
13	K	A	R	Q	Q	I	Q	Q	D	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	G	K
14	K	R	B	Q	Q	R	I	Q	D	Q	Q	Q	Q	Q	Q	Q	Q	Q	G	H	K
15	K	A	B	Q	R	Q	Q	I	D	Q	Q	Q	Q	Q	Q	Q	Q	G	Q	A	K
16	K	A	B	R	Q	Q	Q	R	X	Q	Q	Q	Q	Q	Q	Q	G	H	Q	A	K
17	K	A	Q	A	Q	Q	R	Q	B	W	Q	Q	Q	Q	Q	G	Q	Q	H	A	K
18	K	A	Q	A	Q	R	Q	Q	B	R	W	Q	Q	Q	G	H	Q	Q	B	H	K
19	K	A	Q	A	R	Q	Q	Q	Q	A	Q	W	Q	G	Q	Q	H	Q	B	A	K
20	K	A	Q	R	B	Q	Q	Q	Q	A	Q	R	G	H	Q	Q	Q	H	B	A	K
21	K	A	R	Q	B	Q	Q	Q	Q	A	R	G	Q	Q	H	Q	Q	A	Q	A	K
22	K	R	B	Q	B	Q	Q	Q	Q	R	K	H	Q	Q	Q	H	Q	A	Q	A	K
23	K	A	B	Q	B	Q	Q	Q	R	G	K	I	H	Q	Q	Q	H	A	Q	A	K
24	K	A	B	Q	B	Q	Q	R	G	H	K	R	I	H	Q	Q	B	H	Q	A	K
25	K	A	B	Q	B	Q	R	G	Q	A	K	A	Q	I	H	Q	B	Q	H	A	K
26	K	A	B	Q	B	R	G	H	Q	A	K	A	Q	R	I	H	B	Q	B	H	K
27	K	A	B	Q	Q	K	Q	Q	H	A	K	A	R	Q	Q	K	Q	Q	B	A	K
28	K	A	B	Q	G	K	I	Q	B	H	K	R	B	Q	G	K	I	Q	B	A	K
29	K	A	B	G	H	K	R	I	B	A	K	A	B	G	H	K	R	I	B	A	K
30	K	A	K	K	A	K	A	K	K	A	K	A	K	K	A	K	A	K	K	A	K
31	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K
32	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	P	Q	Q	Q	Q
1	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	L	D	S	Q	Q	Q
2	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	L	Q	D	Q	S	Q	Q
3	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	L	Q	Q	D	Q	Q	S	Q
4	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	L	Q	Q	Q	D	Q	Q	Q	K
5	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	L	Q	Q	Q	Q	D	Q	Q	G	K
6	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	L	Q	Q	Q	Q	Q	D	Q	G	H	K
7	Q	Q	Q	Q	Q	Q	Q	Q	Q	L	Q	Q	Q	Q	Q	Q	D	G	Q	A	K
8	Q	Q	Q	Q	Q	Q	Q	Q	L	Q	Q	Q	Q	Q	Q	Q	Y	H	Q	A	K
9	Q	Q	Q	Q	Q	Q	Q	L	Q	Q	Q	Q	Q	Q	Q	Z	B	Q	H	A	K
10	Q	Q	Q	Q	Q	Q	L	Q	Q	Q	Q	Q	Q	Q	Z	H	B	Q	B	H	K
11	Q	Q	Q	Q	Q	L	Q	Q	Q	Q	Q	Q	Q	Z	Q	A	Q	Q	B	A	K
12	Q	Q	Q	Q	L	Q	Q	Q	Q	Q	Q	Q	Z	H	Q	A	Q	Q	B	A	K
13	Q	Q	Q	L	Q	Q	Q	Q	Q	Q	Q	Z	Q	Q	H	A	Q	Q	B	A	K
14	Q	Q	L	Q	Q	Q	Q	Q	Q	Q	Z	H	Q	Q	B	H	Q	Q	B	A	K
15	Q	L	Q	Q	Q	Q	Q	Q	Q	Z	Q	Q	H	Q	B	Q	H	Q	B	A	K
16	K	Q	Q	Q	Q	Q	Q	Q	Z	H	Q	Q	Q	H	B	Q	Q	H	B	A	K
17	K	I	Q	Q	Q	Q	Q	Z	Q	Q	H	Q	Q	A	Q	Q	Q	A	Q	A	K
18	K	R	I	Q	Q	Q	Z	H	Q	Q	Q	H	Q	A	Q	Q	Q	A	Q	A	K
19	K	A	Q	I	Q	Z	Q	Q	H	Q	Q	Q	H	A	Q	Q	Q	A	Q	A	K
20	K	A	Q	R	I	H	Q	Q	Q	H	Q	Q	B	H	Q	Q	Q	A	Q	A	K
21	K	A	R	Q	Q	I	H	Q	Q	Q	H	Q	B	Q	H	Q	Q	A	Q	A	K
22	K	R	B	Q	Q	R	I	H	Q	Q	Q	H	B	Q	Q	H	Q	A	Q	A	K
23	K	A	B	Q	R	Q	Q	I	H	Q	Q	A	Q	Q	Q	Q	H	A	Q	A	K
24	K	A	B	R	Q	Q	Q	R	I	H	Q	A	Q	Q	Q	Q	B	H	Q	A	K
25	K	A	Q	A	Q	Q	R	Q	Q	I	H	A	Q	Q	Q	Q	B	Q	H	A	K
26	K	A	Q	A	Q	R	Q	Q	Q	R	K	H	Q	Q	Q	Q	B	Q	B	H	K
27	K	A	Q	A	R	Q	Q	Q	R	G	K	I	H	Q	Q	Q	B	Q	B	A	K
28	K	A	Q	R	B	Q	Q	R	G	H	K	R	I	H	Q	Q	B	Q	B	A	K
29	K	A	R	Q	B	Q	R	G	Q	A	K	A	Q	I	H	Q	B	Q	B	A	K
30	K	R	B	Q	B	R	G	H	Q	A	K	A	Q	R	I	H	B	Q	B	A	K
31	K	A	B	Q	Q	K	Q	Q	H	A	K	A	R	Q	Q	K	Q	Q	B	A	K
32	K	A	B	Q	G	K	I	Q	B	H	K	R	B	Q	G	K	I	Q	B	A	K
33	K	A	B	G	H	K	R	I	B	A	K	A	B	G	H	K	R	I	B	A	K
34	K	A	K	K	A	K	A	K	K	A	K	A	K	K	A	K	A	K	K	A	K
35	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K
36	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

Fig. 6. Snapshots for synchronization operations for the Moore and Langdon's algorithm on 21 cells.

The set of 17 internal states for the table of Moore, Langdon [1968] is $\mathcal{Q}$={Q, P, T, K, L, S, D, A, B, R, H, G, I, X, Y, W, Z}, where the state Q is the *quiescent* state, P is the initial *general* states, and T is the *firing* state, respectively. The other 14 states {K, L, S, D, A, B, R, H, G, I, X, Y, W, Z} are used as auxiliary states for the synchronization. The state transition table is shown in Fig. 3, 4, and 5 as 16 sub-tables for each state. The table itself, consisting of 252 4-tuple rules, is constructed newly in this paper. The readers find that the table is very sparse in a sense that each table has many empty entries. The filled-in ratio of the algorithm is $f_{\text{Moore,Langdon[1968]}} = 252/16 \times 17 \times 17 = 5.4$ (%). Figure 6 shows some snapshots for synchronization operations for 21 cells with the general on C_9 and C_{17}, respectively.

A	Right State									
Left State	A	B	C<	C>	D<	D>	E1	E2	R	*
A	A	B	C<	D<	D<	A	A	A	C<	A
B	B									R
C<	D>					A	D>	D>		
C>	C>		R			C>	C>	C>		
D<	A		C<	A	D<	A	A		C<	
D>	D>				A	D>	D>	D>		
E1	A		C<	D<	D<	A	A		C<	
E2	A		C<	D<	D<			A	C<	
R	C>					C>	C>	C>		
*	A	R								

B	Right State									
Left State	A	B	C<	C>	D<	D>	E1	E2	R	*
A	B	D>	C<			D>			C<	R
B	D<	A							C<	
C<										
C>	C>									
D<	D<									
D>										
E1										
E2										
R	C>	C>							R	
*	R									

C<	Right State									
Left State	A	B	C<	C>	D<	D>	E1	E2	R	*
A					D>	D<			D>	
B	A							A	E2	
C<										
C>	C>				D>	D<	C>	C>	D>	
D<	A				D>	D<	A	A	D>	
D>	A				E2	E1			E2	
E1					D>	R			D>	
E2					R	D<			R	
R	C>							C>	R	
*										

E1	Right State									
Left State	A	B	C<	C>	D<	D>	E1	E2	R	*
A	E1		R		A	E1		E1		
B										
C<						E1				
C>	R					R		R		
D<	E1		R	E1				E1		
D>	A							A		
E1										
E2	E1		R		A	E1				
R										
*										

E2	Right State									
Left State	A	B	C<	C>	D<	D>	E1	E2	R	*
A	E2		R	E2	D<	E2			E2	
B										
C<	E2								E2	
C>	R								R	
D<	E2								E2	
D>	D>								D>	
E1									E2	
E2										
R	E2		R	E2	D<	E2	E2		R	
*										

R	Right State									
Left State	A	B	C<	C>	D<	D>	E1	E2	R	*
A					R	R			R	
B										R
C<				R					R	R
C>										R
D<	R					R		R		R
D>	R				R				R	R
E1										
E2						R		R	R	R
R	R			R	R			R	F	F
*		R	R	R	R	R		R	F	

D<	Right State									
Left State	A	B	C<	C>	D<	D>	E1	E2	R	*
A	A	A	C<	A			A		C<	
B										
C<	A							A		
C>	D>	D>	C<		C>				C<	
D<	D>	D>	C<		C>				C<	
D>	D<	D<	C<				D>	D>	C<	
E1	E2		R		E2					
E2	E1		R		E1					
R	C>			E2			E2	C>	R	
*										

C>	Right State									
Left State	A	B	C<	C>	D<	D>	E1	E2	R	*
A		A	C<		A	A			C<	
B										
C<										
C>										
D<	D>		D>		E1	D>	R	D>		
D>	D<		D<		E2	D<	D<	R		
E1			C<			A				
E2		A	C<			A			C<	
R	D<	E2	D<		E2	D<	D<	R	R	
*										

D>	Right State									
Left State	A	B	C<	C>	D<	D>	E1	E2	R	*
A	A		D<	A	D>	D<	E2	E1	C<	
B	A		D<		D>	D<				
C<	A								E2	
C>	C>		C>		C>	C>	R	R		
D<										
D>			C<			C<	E2	E1		
E1	A				D<				E2	
E2				A	D<				C<	
R	C>		C>		C>	C>			R	
*										

Fig. 7. Newly constructed transition table for VMP ten-state protocol.

3.2. Optimum-time ten-state VMP table

Varshavsky, Marakhovsky and Peschansky [1970] constructed a ten-state synchronization algorithm. They gave the transition table as a transition function with many wild cards. In order to define the accurate transition table we expanded the function given in Varshavsky, Marakhovsky, Peschansky [1970]. We found that the expanded table yields numerous unsuccessful synchronizations as was indicated in the Varshavsky, Marakhovsky, Peschansky [1970] that they did not give any complete table. The set of 10 internal states for Varshavsky, Marakhovsky, Peschansky [1970] is $\mathcal{Q}$={`A`, `B`, `F`, `C>`, `C<`, `E1`, `E2`, `R`, `D>`, `D<`}, where the state `A` is the *quiescent* state, `B` is the initial *general* states, and `F` is the *firing* state, respectively. Here we present our new construction of the table consisting of 273 rules as 9 sub-tables for each state in Fig. 7. The table itself is constructed newly in this paper. The filled-in ratio of the table is $f_{\mathrm{Varshavsky,Marakhovsky,Peschansky[1970]}} = 273/9 \times 10 \times 10 = 30.3$ (%). Figure 8 shows some snapshots for synchronization operations on 21 cells with the general on C_9 and C_{17}, respectively.

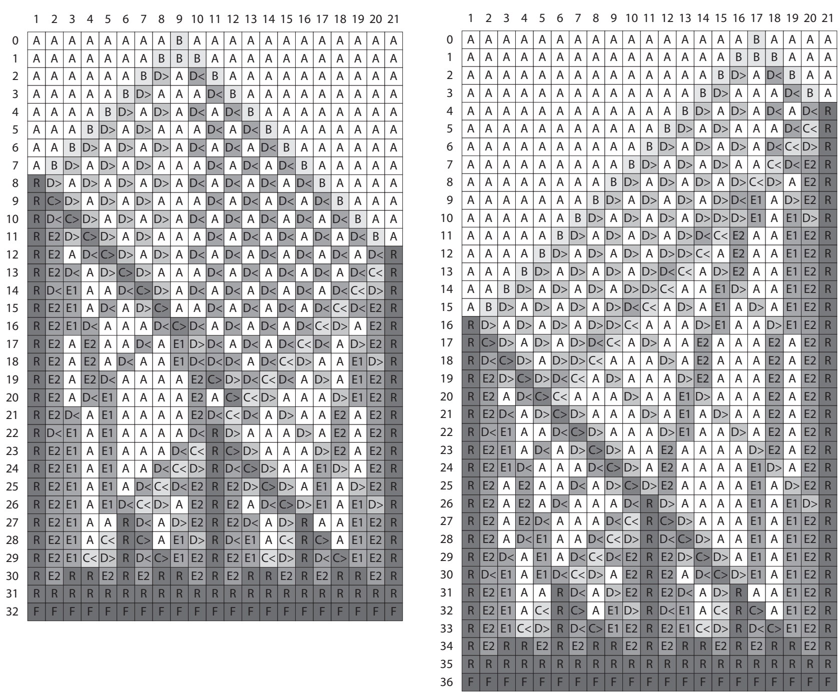

Fig. 8. Snapshots of the VMP 10-state algorithm on 21 cells.

3.3. *Szwerinski's optimum-time ten-state table*

Szwerinski [1982] constructed a ten-state, 345-rule synchronization algorithm. The set of 10 internal states for Szwerinski [1982] is $\mathcal{Q}$={`z`, `m`, `f`, `a`, `p`, `d`, `r`, `b`, `q`, `g`}, where the state `z` is the *quiescent* state, `m` is the initial *general* state, and `f` is the *firing* state, respectively. In our computer examination, no errors were found in the table, however, 21 rules were found to be redundant. Figure 9 gives a list of the transition rules for Szwerinski's table. The 21 redundant rules are marked by shaded small squares in the table. The filled-in ratio is $f_{\text{Szwerinski[1982]}} = 324/9 \times 10 \times 10 = 36\,(\%)$. Figure 10 shows some snapshots for synchronization operations on 21 cells with the general on C_9 and C_{17}, respectively.

z	Right State									
Left State	z	a	b	d	r	p	q	m	g	*
z	z	a	z		r	z	z	a	a	z
a	a		a	a	p	r	a		r	g
b	z	a	z	z	r	z	z		a	
d		a	z		r	z	z		b	
r	r	p	r	r			r			
p	z	r	z	z			z			
q	z	a	z	z	r	z			a	
m	a									g
g	a	r	a	b			a		g	g
*	z	g						g	g	

a	Right State									
Left State	z	a	b	d	r	p	q	m	g	*
z	z	z	g	r	p	r	r		r	
a	z	g	g				z		g	g
b	g	g			p	r			r	
d	r									
r	p		p		p	r	p	q	r	
p	r		r		r		r			
q	r	z			p	r			r	
m					q					
g	r	g	r		r		r			
*		g								

b	Right State									
Left State	z	a	b	d	r	p	q	m	g	*
z	b	g	b	d	p	b	b		b	
a	g		g			g	g		g	
b	b	g			p				b	
d	d								b	
r	p		p			q			b	
p	b	g			q				b	
q	b	g							b	
m										
g	b	g	b	b	b	b	b		g	
*										

d	Right State									
Left State	z	a	b	d	r	p	q	m	g	*
z			p		d	d	d	g		
a						g	g			
b	p		p	p	d	q	p	g		
d			p			q				
r	d		d		d	d	d			
p	d	g	q	q	d			g		
q	d	g	p		d			g		
m	g		g			g	g			
g										
*										

r	Right State									
Left State	z	a	b	d	r	p	q	m	g	*
z		a			q	q	q		b	
a	a	q	g	m		q	a		b	
b		g					b			
d		m				q	b			
r	q				z	q			a	
p	q	q		q	q		q	q	a	
q	q	a	b	b		q	q		b	
m						q			b	
g	b	b			a	a	b	b	g	
*										

p	Right State									
Left State	z	a	b	d	r	p	q	m	g	*
z			p	p	r		r		a	
a			p				p			
b	p	p	p	p	r		r	p	a	
d	p		p							
r	r		r				r			
p										
q	r	p	r		r		r	p	a	
m			p				p			
g	a		a				a			
*										

g	Right State									
Left State	z	a	b	d	r	p	q	m	g	*
z	g				g				g	g
a		g							g	g
b			g		g				g	g
d										
r	g		g		g	a	g		g	g
p					a	a	a		a	a
q					g	a	a		g	
m										
g	g	g	g		g	a	g		f	f
*	g	g	g		g	a			f	

m	Right State									
Left State	z	a	b	d	r	p	q	m	g	*
z	r									g
a		g					b			
b										
d					g	g				
r				g						
p				g						
q		b					q			
m										
g										
*	g									

q	Right State									
Left State	z	a	b	d	r	p	q	m	g	*
z	q	d			z			z		
a	d	g			a	d				
b			p		z	q	p		a	
d					z	q				
r	z	a	z	z	m	z	z		a	
p		d	q	q	z	q	z		a	
q			p		z	z			a	
m	z									
g			a		a	a	a			
*										

Fig. 9. Transition table for the Szwerinski's ten-state protocol. Shaded entries show the redundant rules included in the original paper (Szwerinski [1982]).

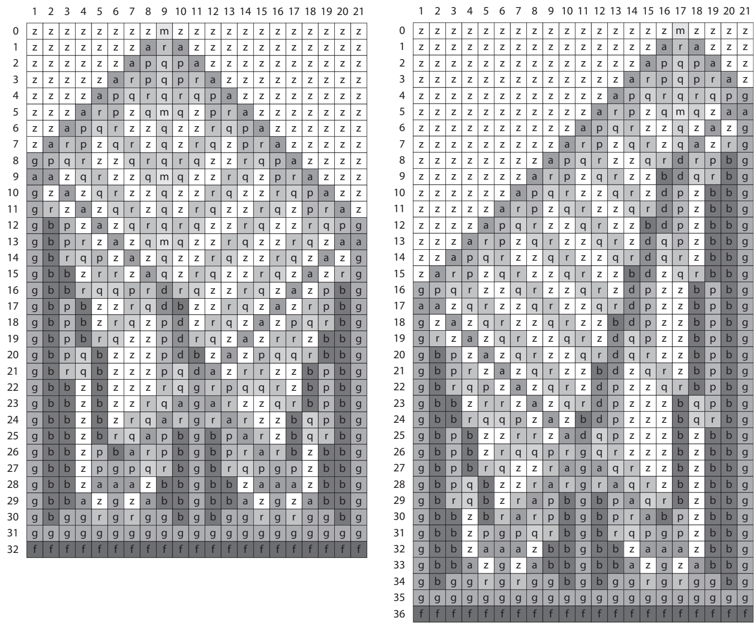

Fig. 10. Snapshots for the Szwerinski's synchronization operation on 21 cells.

3.4. Settle and Simon's optimum-time nine-state table

Settle and Simon [2002] constructed a nine-state, 326-rule synchronization algorithm. The set of nine internal states for Settle, Simon [2002] is $\mathcal{Q}$={Z, D, F, Q, A, B, R, P, G}, where state Z is the *quiescent* state, D is the initial *general* state, and F is the *firing* state, respectively. In our computer examination, no errors were found, however, 16 rules were found to be redundant. Figure 11 gives a list of the transition rules for the Settle and Simon's table. The 16 redundant rules are marked by shaded squares in the table. The filled-in ratio is $f_{\mathrm{Settle,Simon[2002]}} = 310/8\times9\times9$ $(326/8\times9\times9) =$ 47.8 (50.3) (%), respectively. Figure 12 shows some snapshots for synchronization operations on 21 cells with the general on C_9 and C_{17}, respectively.

3.5. Optimum-time nine-state UHMNM table

Umeo *et al.* [2002] constructed a nine-state, 203-rule synchronization algorithm. The set of nine internal states is $\mathcal{Q}$={Q, G, F, [,], A, H, R, W}, where state Q is the *quiescent* state, G is the initial *general* state, and F is the *firing* state, respectively. Figure 13 gives a list of the transition rules for UHMNM table consisting of 203 transition rules. The filled-in ratio is

Q (Left State \ Right State)	Q	Z	A	B	D	R	P	G	*
Q				P		Z	Z	A	
Z		Q	D		Z	Z			
A		D	G			A	D		
B	P			P		Z	Q	A	
D		Z				Z	Q		
R	Z	Z	A	Z	Z	D	Z	A	
P	Z		D	Q	Q	Z	Q	A	
G	A			A		A	A		
*									

Z (Left State \ Right State)	Q	Z	A	B	D	R	P	G	*
Q		Z	A	Z	Z	R	Z	A	
Z	Z	Z	A	Z	A	R	Z	A	Z
A	A	A		A		P	R	R	G
B	Z	Z	A	Z	Z	R	Z	A	
D	Z	A		Z		R		B	G
R	R	R	P	R	R				
P	Z	Z	R	Z					
G	A	A	R	A	B			G	G
*		Z	G		G			G	

A (Left State \ Right State)	Q	Z	A	B	D	R	P	G	*
Q		R	Z			P	R	R	
Z	R	Z	Z	G	R	P	R	R	
A	Z	Z	G	G				G	G
B		G	G			P	R	R	
D		R				Q			
R	P	P		P	Q	P	R	R	
P	R	R		R		R			
G	R	R	G	R		R			
*			G						

B (Left State \ Right State)	Q	Z	A	B	D	R	P	G	*
Q		B	G					B	
Z	B	B	G	B	D	P	B		
A	G	G		G			G	G	
B		B	G			P		B	
D		D						B	
R		P		P			Q	B	
P		B	G			Q		B	
G	B		G	B	B	B	B	G	
*									

D (Left State \ Right State)	Q	Z	A	B	D	R	P	G	*
Q	Q	D	B	P	G	D			
Z	D	R		P	G	D	D		G
A	B		G			G	G		
B	P	P		P	G	P	R		
D	G	G		G		G	G		
R	D	D	G	P	G	D	D		
P		D	G	R	G	D			
G									
*		G							

R (Left State \ Right State)	Q	Z	A	B	D	R	P	G	*
Q	Q	Q	A	B	B				
Z	Q		A			Q	Q	B	
A	A	A	Q	G	D		Q	B	
B	B		G			B			
D	B		D			Z	R	B	
R		Q		B	Z	Z	Q	A	
P		Q	Q		R	Q		A	
G		B	B		B	A	A	G	
*									

P (Left State \ Right State)	Q	Z	A	B	D	R	P	G	*
Q		R	P	R	P	R		A	
Z	R			P	P	R		A	
A	P			P					
B	R	P	P	P	P	R		A	
D	P	P		P					
R	R	R		R					
P									
G	A	A		A					
*									

G (Left State \ Right State)	Q	Z	A	B	D	R	P	G	*
Q	A					G		G	
Z		G				G		G	G
A			G					G	G
B				G		G		G	G
D									
R	G	G		G		G	A	G	G
P						A	A	A	A
G	G	G	G	G		G	A	F	F
*		G	G	G		G	A	F	

Fig. 11. Transition table for the Settle and Simon's nine-state protocol. Shaded entries show the redundant rules included in the original paper (Settle, Simon [2002]).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	Z	Z	Z	Z	Z	Z	Z	Z	D	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
1	Z	Z	Z	Z	Z	Z	Z	A	R	A	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
2	Z	Z	Z	Z	Z	Z	A	P	Q	P	A	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
3	Z	Z	Z	Z	Z	A	R	P	Q	P	R	A	Z	Z	Z	Z	Z	Z	Z	Z	Z
4	Z	Z	Z	Z	A	P	Q	R	Q	R	Q	P	A	Z	Z	Z	Z	Z	Z	Z	Z
5	Z	Z	Z	A	R	P	Z	Q	D	Q	Z	P	R	A	Z	Z	Z	Z	Z	Z	Z
6	Z	Z	A	P	Q	R	Z	Z	Q	Z	Z	R	Q	P	A	Z	Z	Z	Z	Z	Z
7	Z	A	R	P	Z	Q	R	Z	Q	Z	R	Q	Z	P	R	A	Z	Z	Z	Z	Z
8	G	P	Q	R	Z	Z	Q	R	Q	R	Q	Z	Z	R	Q	P	A	Z	Z	Z	Z
9	A	A	Z	Q	R	Z	Z	Q	D	Q	Z	Z	R	Q	Z	P	R	A	Z	Z	Z
10	G	Z	A	Z	Q	R	Z	Z	Q	Z	Z	R	Q	Z	Z	R	Q	P	A	Z	Z
11	G	R	Z	A	Z	Q	R	Z	Q	Z	R	Q	Z	Z	R	Q	Z	P	R	A	Z
12	G	B	P	Z	A	Z	Q	R	Q	R	Q	Z	Z	R	Q	Z	Z	R	Q	P	G
13	G	B	P	R	Z	A	Z	Q	D	Q	Z	Z	R	Q	Z	Z	R	Q	Z	A	A
14	G	B	R	Q	P	Z	A	Z	Q	Z	Z	R	Q	Z	Z	R	Q	Z	A	Z	G
15	G	B	B	Z	R	R	Z	A	Q	Z	R	Q	Z	Z	R	Q	Z	A	Z	R	G
16	G	B	B	R	Q	Q	P	R	D	R	Q	Z	Z	R	Q	Z	A	Z	P	B	G
17	G	B	P	B	Z	Z	R	R	D	B	Z	Z	R	Q	Z	A	Z	R	P	B	G
18	G	B	P	B	Z	R	Q	Z	P	D	Z	R	Q	Z	A	Z	P	Q	R	B	G
19	G	B	P	B	R	Q	Z	Z	P	D	R	Q	Z	A	Z	R	R	Z	B	B	G
20	G	B	P	Q	B	Z	Z	Z	P	D	B	Z	A	Z	P	Q	Q	R	B	B	G
21	G	B	R	Q	B	Z	Z	Z	P	R	D	A	Z	R	R	Z	Z	B	P	B	G
22	G	B	B	Z	B	Z	Z	Z	R	R	G	R	P	Q	Q	R	Z	B	P	B	G
23	G	B	B	Z	B	Z	Z	R	Q	A	G	A	R	Z	Z	Q	R	B	P	B	G
24	G	B	B	Z	B	Z	R	Q	A	R	G	R	A	R	Z	Z	B	Q	P	B	G
25	G	B	B	Z	B	R	Q	A	P	B	G	B	P	A	R	Z	B	Q	R	B	G
26	G	B	B	Z	P	B	A	R	P	B	G	B	P	R	A	R	B	Z	B	B	G
27	G	B	B	Z	P	G	P	Q	R	B	G	B	R	Q	P	G	P	Z	B	B	G
28	G	B	B	Z	A	A	A	Z	B	B	G	B	B	Z	A	A	A	Z	B	B	G
29	G	B	B	A	Z	G	Z	A	B	B	G	B	B	A	Z	G	Z	A	B	B	G
30	G	B	G	G	R	G	R	G	G	B	G	B	G	G	R	G	R	G	G	B	G
31	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
32	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	D	Z	Z	Z	Z
1	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	A	R	A	Z	Z	Z
2	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	A	P	Q	P	A	Z	Z
3	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	A	R	P	Q	P	R	A	Z
4	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	A	P	Q	R	Q	R	Q	P	G
5	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	A	R	P	Z	Q	D	Q	Z	A	A
6	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	A	P	Q	R	Z	Z	Q	Z	A	Z	G
7	Z	Z	Z	Z	Z	Z	Z	Z	Z	A	R	P	Z	Q	R	Z	Q	A	Z	R	G
8	Z	Z	Z	Z	Z	Z	Z	Z	A	P	Q	R	Z	Z	Q	R	D	R	P	B	G
9	Z	Z	Z	Z	Z	Z	Z	A	R	P	Z	Q	R	Z	Z	B	D	R	R	B	G
10	Z	Z	Z	Z	Z	Z	A	P	Q	R	Z	Z	Q	R	Z	D	P	Z	B	B	G
11	Z	Z	Z	Z	Z	A	R	P	Z	Q	R	Z	Z	Q	R	D	P	Z	B	B	G
12	Z	Z	Z	Z	A	P	Q	R	Z	Z	Q	R	Z	Z	B	D	P	Z	B	B	G
13	Z	Z	Z	A	R	P	Z	Q	R	Z	Z	Q	R	Z	D	R	P	Z	B	B	G
14	Z	Z	A	P	Q	R	Z	Z	Q	R	Z	Z	Q	R	D	R	R	Z	B	B	G
15	Z	A	R	P	Z	Q	R	Z	Z	Q	R	Z	Z	B	D	Z	Q	R	B	B	G
16	G	P	Q	R	Z	Z	Q	R	Z	Z	Q	R	Z	D	P	Z	Z	B	P	B	G
17	A	A	Z	Q	R	Z	Z	Q	R	Z	Z	Q	R	D	P	Z	Z	B	P	B	G
18	G	Z	A	Z	Q	R	Z	Z	Q	R	Z	Z	B	D	P	Z	Z	B	P	B	G
19	G	R	Z	A	Z	Q	R	Z	Z	Q	R	Z	D	R	P	Z	Z	B	P	B	G
20	G	B	P	Z	A	Z	Q	R	Z	Z	Q	R	D	R	R	Z	Z	B	P	B	G
21	G	B	P	R	Z	A	Z	Q	R	Z	Z	B	D	Z	Q	R	Z	B	P	B	G
22	G	B	R	Q	P	Z	A	Z	Q	R	Z	D	P	Z	Z	Q	R	B	P	B	G
23	G	B	B	Z	R	R	Z	A	Z	Q	R	D	P	Z	Z	Z	B	Q	P	B	G
24	G	B	B	R	Q	Q	P	Z	A	Z	B	D	P	Z	Z	Z	B	Q	R	B	G
25	G	B	P	B	Z	Z	R	R	Z	A	D	R	P	Z	Z	Z	B	Z	B	B	G
26	G	B	P	B	Z	R	Q	Q	P	R	G	R	R	Z	Z	Z	B	Z	B	B	G
27	G	B	P	B	R	Q	Z	Z	R	A	G	A	Q	R	Z	Z	B	Z	B	B	G
28	G	B	P	Q	B	Z	Z	R	A	R	G	R	A	Q	R	Z	B	Z	B	B	G
29	G	B	R	Q	B	Z	R	A	P	B	G	B	P	A	Q	R	B	Z	B	B	G
30	G	B	B	Z	B	R	A	R	P	B	G	B	P	R	A	B	P	Z	B	B	G
31	G	B	B	Z	P	G	P	Q	R	B	G	B	R	Q	P	G	P	Z	B	B	G
32	G	B	B	Z	A	A	A	Z	B	B	G	B	B	Z	A	A	A	Z	B	B	G
33	G	B	B	A	Z	G	Z	A	B	B	G	B	B	A	Z	G	Z	A	B	B	G
34	G	B	G	G	R	G	R	G	G	B	G	B	G	G	R	G	R	G	G	B	G
35	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
36	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

Fig. 12. Snapshots for Settle and Simon's synchronization operations on 21 cells with the general on C_9 and C_{17}, respectively.

$f_{\text{UHMNM [2002]}} = 203/8 \times 9 \times 9 = 31.3$ (%). Figure 14 shows some snapshots for synchronization operations on 21 cells with the general on C_9 and C_{17}, respectively.

G	G	Q	R	]	[	H	A	W	*
G	Q	H						[	
Q	A	G			A	H	A	[	W
R									
]		H							
[									
H		H				H			
A		A					A		
W	]	]						W	
*		W							

Q	G	Q	R	]	[	H	A	W	*
G	Q	G			[				W
Q	G	Q							Q
R							A		
]	]				W		]		
[							A		
H			H	H	[	H		[	
A						Q	A		
W							]		
*	W	Q							

R	G	Q	R	]	[	H	A	W	*
G									
Q					W	R			
R					R	]	R		
]		W	R	H	W	R	W		
[					A		A		
H			R	H	W	R			
A		R	[		R		R		
W									
*									

A	G	Q	R	]	[	H	A	W	*
G	G	G	A		R	A	H		
Q			[		R		Q		
R		A	A				A		
]	]	]	]		]	]	]		
[					A		A		
H			A		A	Q	A		
A		A	A		A		A		
W	]	]	]		]	]	]		
*									

]	G	Q	R	]	[	H	A	W	*
G									
Q		H	W		H	R	H		
R					W	]			
]		H	W		H	R	H		
[									
H		Q	Q		W	]	Q		
A									
W	]	]	]	]	W	]	]	W	
*									

[	G	Q	R	]	[	H	A	W	*
G								[	
Q		A			A		Q	[	
R		W			W		Q	[	
]		A	W		A		W	W	
[								[	
H		A			A		Q	[	
A		R	[		R		[	[	
W								W	
*									

H	G	Q	R	]	[	H	A	W	*
G	G				[			[	
Q	G		H		[	H		[	
R	H	]	H		[	H	H	[	
]	R	R		H	[	H	H	[	
[									
H	A	Q	H	H	[	H	H	[	
A	H				[		Q	[	
W									
*									

W	G	Q	R	]	[	H	A	W	*
G									W
Q		W					W		
R									
]					W			W	W
[				W				W	W
H		W					W	W	W
A									
W				W	W		W	F	F
*	W			W	W		W	F	

Fig. 13. Transition table for the UHMNM nine-state protocol.

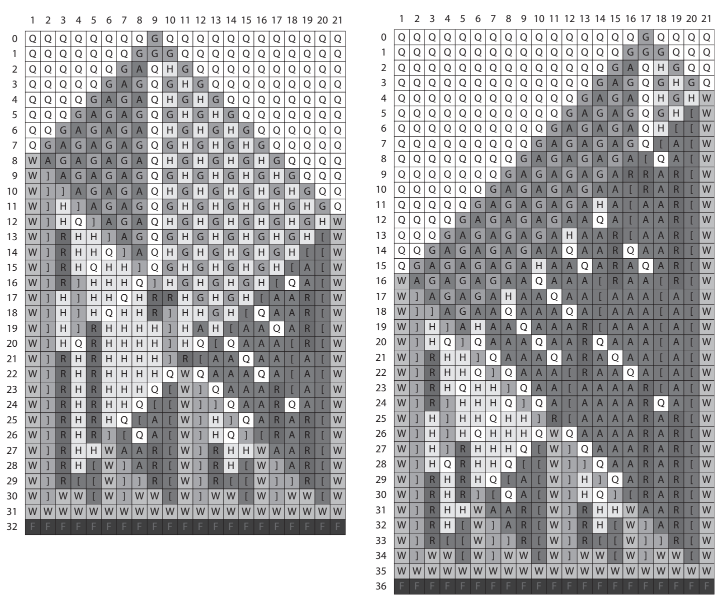

Fig. 14. Snapshots for UHMNM nine-state synchronization operations on 21 cells with the general on C_9 and C_{17}, respectively.

3.6. Optimum-time eight-state table

Finally, we propose a new eight-state, 222-rule synchronization algorithm. The set of eight internal states is $\mathcal{Q}$={`Q`, `G`, `F`, `[`, `]`, `A`, `B`, `C`}, where state `Q` is the *quiescent* state, `G` is the initial *general* state, and `F` is the *firing* state, respectively. Figure 15 gives a list of the 222 transition rules. The

G	Right State							
Left State	G	Q	[	]	A	B	C	*
G								
Q		Q	C		G		[	C
[			B			B		
]		C	C	A	G	C	[	
A			C	A	G		Q	
B		G	G			G		
C		]	]			Q		F
*		C					F	

Q	Right State							
Left State	G	Q	[	]	A	B	C	*
G	C	]		]		B		G
Q	[	Q	[		A	B	A	Q
[	[			Q		B	C	
]		]	C		]	]	]	C
A	A	A	[	A	A		[	
B		B	[		Q	B		
C		B	]	C		]		
*	G	Q	C					

[	Right State							
Left State	G	Q	[	]	A	B	C	*
G		C	Q	]		Q	[	
Q		B	[	B		Q	]	
[		B	B	[		Q	[	]
]	C	B	G	G		C	C	
A		B		B		Q	]	
B	[		[			[	[	
C		]	]	]		Q	C	
*		C		C				

]	Right State							
Left State	G	Q	[	]	A	B	C	*
G			C		]			
Q	C	A	A	A		A	[	C
[	[	A	G	]		A	[	C
]	Q	]	G	A	]		[	
A	Q	Q	C	Q	]	Q	Q	
B								
C	]	[	C	]	]	[	C	
*				[				

A	Right State							
Left State	G	Q	[	]	A	B	C	*
G	A	]	[		A		[	
Q	A	Q	[	A	A		[	
[								
]	A	G	[	A	A		[	
A	A	Q	[	A	A		[	
B								
C								
*								

B	Right State							
Left State	G	Q	[	]	A	B	C	*
G	B	B	B			B		
Q	[	Q	G			Q		
[		B	B			B		
]	]	]	]			]		
A								
B	B	B	B			B		
C	]	]	]			]		
*								

C	Right State							
Left State	G	Q	[	]	A	B	C	*
G						C		F
Q		C	C	C		C		[
[		C	C	C		C	C	C
]		C	C	C		C	C	C
A	C	C	C	C		C	C	C
B								
C			C	C		C	F	F
*	F	]	C	C		C	F	

Fig. 15. Transition table for the newly proposed eight-state protocol.

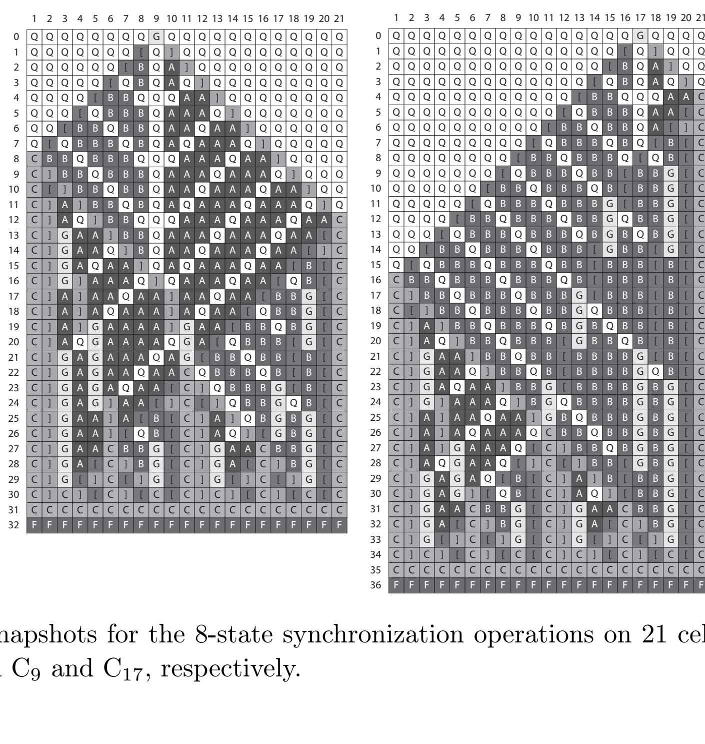

Fig. 16. Snapshots for the 8-state synchronization operations on 21 cells with the general on C_9 and C_{17}, respectively.

filled-in ratio is $f_{\text{this paper}} = 222/7 \times 8 \times 8 = 45.1$ (%). Figure 16 shows some snapshots for synchronization operations on 21 cells with the general on C_9 and C_{17}, respectively.

3.7. Non-optimum-time six-state UMH table

Umeo, Maeda, and Hongyo [2006] proposed a non-optimum-time six-state synchronization algorithm. See the details in (Umeo, Maeda, Hongyo [2006]). The set of six internal states is $\mathcal{Q}$={`Q`, `M`, `F`, `P`, `R`, `Z`}, where the state `Q` is the *quiescent* state, `M` is the initial *general* state, and `F` is the *firing* state, respectively. Figure 17 gives a list of the transition rules for the six-state non-optimum-time algorithm consisting of 115 transition rules. The filled-in ratio is $f_{\text{Umeo,Maeda,Hongyo[2006]}} = 115/5 \times 6 \times 6 = 63.9$ (%). Figure 18 shows some snapshots for synchronization operations on 21 cells with a general on C_9 and C_{17}, respectively.

Q	Right State					
Left State	Q	P	R	M	Z	*
Q	Q	P	Q	M	Q	Q
P	P	P				P
R	Q		Q			Q
M	M					M
Z	Q				Q	Q
*	Q	P	Q	M	Q	

Z	Right State					
Left State	Q	P	R	M	Z	*
Q	P	Q	Q	Q	P	
P	Q	Z			Z	Q
R	Q		Q		Q	Q
M	Q			Q	Q	Q
Z	P	Z	Q	Q	F	F
*		Q	Q	Q	F	

P	Right State					
Left State	Q	P	R	M	Z	*
Q	Z	Z	R		R	
P	Z		Z		Z	
R	R	Z	Z			Z
M				P	R	
Z	R	Z		R	Z	Z
*			Z		Z	

R	Right State					
Left State	Q	P	R	M	Z	*
Q			R	Z	P	
P			M	M	R	
R	R	M		M		
M	Z	M	M	M	R	Z
Z	P	R		R		
*				Z		

M	Right State					
Left State	Q	P	R	M	Z	*
Q	P	M	R	M		R
P	M			M	Z	Z
R	R		R	R	Z	Z
M	M	M	R	M	Z	Z
Z		Z	Z	Z		
*	R	Z	Z	Z		

Fig. 17. Transition table for the six-state protocol (Umeo, Maeda, Hongyo [2006]).

3.8. A comparison of quantitative aspects of generalized synchronization algorithms

Here we present Table I which is based on a quantitative comparison of optimum-time and non-optimum-time generalized synchronization algorithms and their transition tables discussed above with respect to the number of internal states of each finite state automaton, the number of transition rules realizing the synchronization, and the number of state-changes on the array.

The readers can see the symmetry/asymmetry property in the table by checking each state transition table. Those tables proposed by Szwerinski [1982], Settle, Simon [2002] and Umeo, Maeda, Hongyo [2006] are symmetric, other ones are asymmetric. As for the state change complexity all GFSSP algorithms discussed above are based on several original FSSP algorithms with $O(n^2)$ state-change complexities. For example, Moore, Langdon [1968] is based on Waksman's 16-state algorithm with $O(n^2)$ state-change complexity. For details on the state-change complexity in the original FSSP algorithms. See Umeo, Hisaoka, Sogabe [2005]. We have examined the average state-change complexity per cell for each GFSSP table for any array

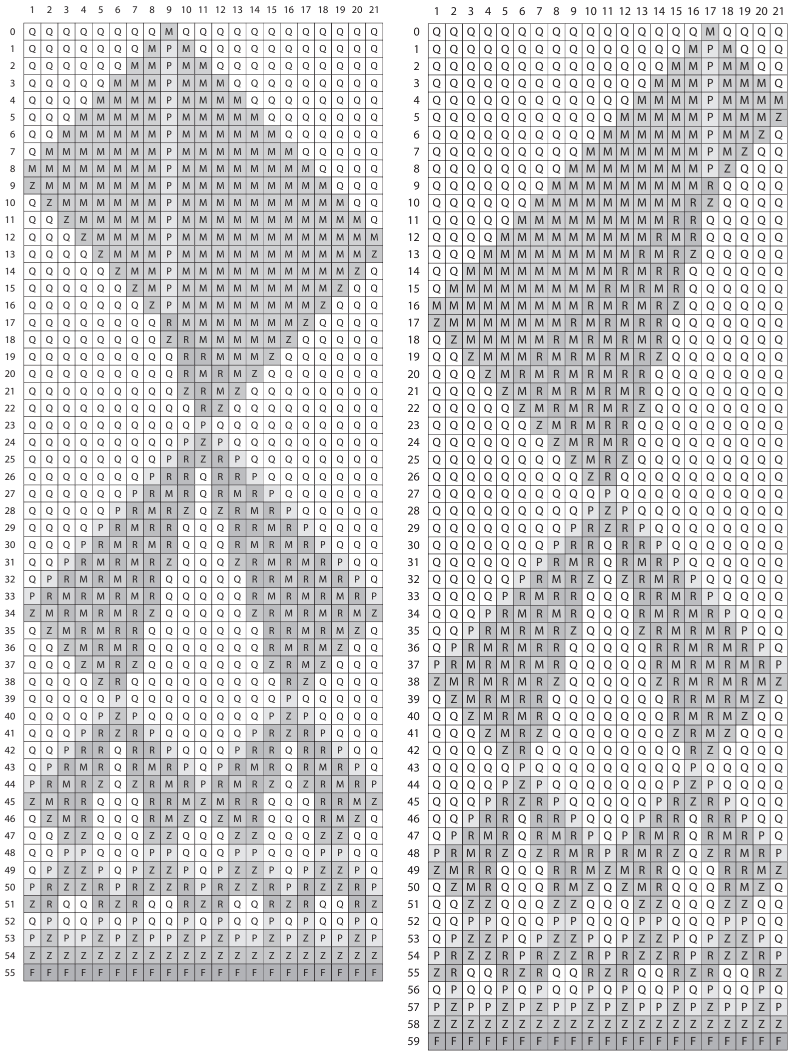

Fig. 18. Snapshots for 6-state non-optimum-time synchronization operations on 21 cells with a general on C_9 and C_{17}, respectively.

size n and any position k of the initial general such that $2 \le n \le 1000$ and $1 \le k \le n$. In Fig. 19 we give a comparison of the average state-changes per cell in those algorithms discussed above. For reference and comparison, we also show an average state-change of Gerken's 156-state FSSP algorithm, working only on the arrays with a general at one end, with $O(\log n)$ state-change complexity per cell.

TABLE I

Quantitative comparison of transition rule sets for optimum- and non-optimum-time generalized firing squad synchronization algorithms. The symbol "(–)" in the fourth column means that the number of transition rules is not given in the original construction.

Algorithm	Time complexity	No of states	No of newly constructed/ revised (original) transition rules	Symmetry/ asymmetry	Filled-in ratio (%)	State-change complexity in cellular space
Moore Langdon [1968]	optimum	17	252 (—)	asymmetric	5.4	$O(n^2)$
VMP [1969]	optimum	10	273 (—)	asymmetric	30.3	$O(n^2)$
Szwerinski [1982]	optimum	10	324 (345)	symmetric	36	$O(n^2)$
Settle Simon [2002]	optimum	9	310 (326)	symmetric	47.8	$O(n^2)$
UHMNM [2002]	optimum	9	203	asymmetric	31.3	$O(n^2)$
Umeo, Maeda, Hongyo [2006]	non-optimum	6	115	symmetric	63.9	$O(n^2)$
this paper	optimum	8	222	asymmetric	45.1	$O(n^2)$

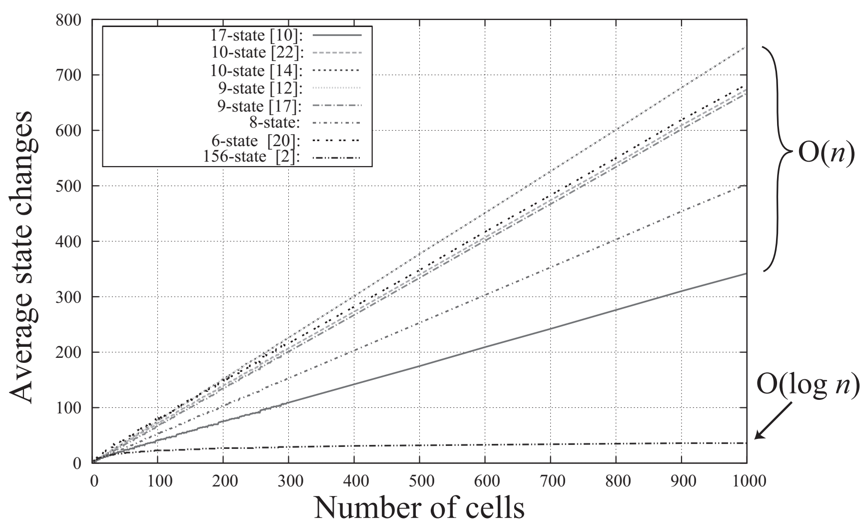

Fig. 19. Comparison of state-changes in those generalized firing squad synchronization protocols. An $O(\log n)$ state-change complexity of Gerken's 156-state FSSP algorithm, which is not a GFSSP, is also shown, for comparison.

4. Summary and future directions

A rich variety of GFSSP algorithms and protocols have been proposed and developed, however almost no work has been done on the implementations and computer examinations of those algorithms. In the present article, we have examined via computer the state transition rule sets for which generalized firing squad synchronization algorithms have been designed over the past forty years. The smallest transition rule sets for the well-known firing squad synchronization algorithms are useful and important for researchers who might have interests in those transition rule sets that realize the classical optimum-time and non-optimum-time generalized firing algorithms quoted frequently in the literatures. We have presented a survey and a comparison of the quantitative aspects of the generalized synchronization algorithms developed thus far for one-dimensional cellular arrays. Several new results and new viewpoints have been shown. Yunès [2008] studied a very interesting extension of the original FSSP with one-end-general that every state of the protocol can be an initial general with adding some supplemental rules. The following question remains to be checked: Can we extend the GFSSP tables discussed in this paper so that each state in the table could be an initial general state, excluding the quiescent and the firing state, by adding some new rule set?

The authors would like to express their thanks to reviewers for useful comments. A part of this work has been supported by the Kayamori Foundation of Informational Science Advancement.

REFERENCES

Balzer, R. 1967, *Information and Control* **10**, 22 .

Gerken, H.D. 1987, Diplomarbeit.

Goto, E. 1962, *Dittoed Course Notes for Applied Mathematics* 298, Harvard University, pp. 52–59, with an illustration in color.

Herman, G.T. 1971, *International Journal of Systems Sciences* **2**, 271.

Herman, G.T. 1972 *International Journal of Systems Sciences* **3**, 149.

Kobuchi, Y. 1987, *Information Processing Letters* **25**, 413.

Mazoyer, J. 1987, *Theor. Comput. Sci.* **50**, 183.

Minsky, M. 1967, *Computation: Finite and Infinite Machines*, Prentice Hall, pp. 28–29.

Moore, E.F. 1964, *Sequential Machines, Selected Papers* (E.F. Moore, ed.), Addison-Wesley, Reading MA 1964, pp. 213–214.

Moore, F.R., Langdon, G.G. 1968, *Information and Control*, **12**, 212.

Sanders, P. 1994, *Proc. of the VI International Workshop on Parallel Processing by Cellular Automata and Arrays*, eds. C. Jesshope, V. Jossifov, W. Wilhelmi, Akademie-Verlag, pp. 99–108.

Settle, A., Simon, J. 2002, *Theor. Comput. Sci.* **276**, 83.

Szwerinski, H. 1985, *Information and Control* **67**, 163.

Szwerinski, H. 1982, *Theor. Comput. Sci.* **19**, 305.

Umeo, H. 2008, *Journal of Cellular Automata* **4**, 1.

Umeo, H. 2009, *Encyclopedia of Complexity and System Science*, Vol. 4 Ed. R.A. Meyers, Springer, 2009, pp. 3537–3574.

Umeo, H., Hisaoka, M., Michisaka, K., Nishioka, K., Maeda, M. 2002, Proc. of the 3rd International Conference on Unconventional Models of Computation: UMC 2002, LNCS 2509, pp. 276–286.

Umeo, H., Hisaoka, M., Sogabe, T. 2005, *Intern. J. of Unconventional Computing* **1**, 403.

Umeo, H., Kamikawa, N., Yunès, J.B. 2009, *Parallel Processing Letters* **19**, 299.

Umeo, H., Maeda, M., Hongyo, K. 2006, Proc. of 7th International Conference on Cellular Automata for Research and Industry, ACRI2006, LNCS 4173, pp. 157–168.

Umeo, H., Yanagihara, T. 2009, *Fundamenta Informaticane* **91**, 161.

Umeo, H., Yunès, J.B., Kamikawa, N. 2008, Proc. of 8th International Conference on Cellular Automata for Research and Industry: ACRI 2008, LNCS 5191, (2008) pp. 108–113.

Varshavsky, V.I., Marakhovsky, V.B., Peschansky, V.A. 1970, *Mathematical Systems Theory* **4**, 212.

Vollmar, R. 1981, *Computing, Suppl.* **3**, 181.

Vollmar, R. 1982, *Inter. J. Theor. Phys.* **21**, 1007.

Waksman, A. 1966, *Information and Control* **9**, 66 (1966).

Yunès, J.B. 2008 *Information Processing Letters* **107**, 71; *J. of Cellular Automata* **4**, 147.