

Synchronization of a Line of Identical Processors at a Given Time^{*}

Salvatore La Torre and Margherita Napoli and Mimmo Parente

Dipartimento di Informatica ed Applicazioni Università degli Studi di Salerno 84081
Baronissi, Italy. e-mail: {sallat,mn,parente}@dia.unisa.it

Abstract. We are given a line of n identical processors (finite automata) that work synchronously. Each processor can transmit just one bit of information to the neighbour processors (if any) on the left and on the right. The computation starts at time 1 with the leftmost processor in a starting state and all other processors in a quiescent state. Given the time $f(n)$, the problem is to set (synchronize) all the processors in a particular state for the first time, at the very same instant $f(n)$.

This problem is also known as the *Firing Squad Synchronization Problem* and was introduced by Moore in 1964. Mazoyer has given a minimal time solution with the least number of different states (six) and very recently he has given a minimal time solution for the constrained problem in which adjacent processors can exchange only one bit.

In this paper we present solutions that synchronize the line at a given time expressed as a function of n . In particular we give solutions that synchronize at the times $n \log n$, $n\sqrt{n}$, n^2 and 2^n . Moreover we also show how to compose solutions in such a way to obtain synchronizing solutions for all times expressed by polynomials with nonnegative coefficients.

Clearly all such solutions work also in the general case when the 1-bit constraint is relaxed.

1 Introduction

We are given a line of n identical processors (finite automata) that work synchronously with discrete steps. Each processor can transmit just one bit of information to the neighbour processor (if any) on the left and to the neighbour processor (if any) on the right. The computation starts at time 1 with the leftmost processor in a given starting state and all other processors in a quiescent state. Given the time $f(n)$, the problem is to program the processors so that they all enter for the first time a particular state at the very same instant $f(n)$.

This problem is also known as the *Firing Squad Synchronization Problem* (shortly FSSP) as the processors can be seen like soldiers that have to fire simultaneously. The problem was introduced by Moore in 1964. However, in that version at each step each processor in the line can transmit its current state to the neighbour processors. Since then many solutions to the problem and to its

^{*} Work partially supported by progetto M.U.R.S.T. grant 60% "Modelli di Sistemi Concorrenti".

variations (see for example [3]) have been given. Minsky in [9] showed that a solution to the FSSP requires at least $2n - 1$ time units. Waksman in [10] gave the first minimal time solution and Mazoyer in [5] has given a minimal time solution with the least number of states: six.

Recently Mazoyer in [7] has given a minimal time solution for the problem where only one bit can be transmitted. Here we present solutions that synchronize the line of processors in a given time, not necessarily minimal, expressed as a function of n . Such problem was posed in [6], while facing the problem of the composition of different Cellular Automata. There the composition was reduced to *space-time constructibility* of Cellular Automata in the following sense: a pair of functions $(g(n), f(n))$ is space-time constructible if there exists a cellular automaton that synchronizes $g(n)$ cells at time $f(n)$, for all n .

In this paper we consider $g(n) = n$ and give algorithms for synchronizing in the times $f(n)$ of the following types: $n \log n$, $n\sqrt{n}$, n^2 and 2^n . Moreover we also show how to compose solutions in such a way to obtain synchronizing solutions for all linear times $an + b$, for any feasible a and b , and for all times expressed as polynomial with nonnegative coefficients. Clearly all these solutions are also solutions for the general case, where the 1-bit constraint is relaxed.

To achieve the above results we introduce the concept of signal that, informally speaking, is a particular set of cells that at a given time either receive the bit 1 from or send the bit 1 to the adjacent cells. Starting from basic signals we combine different signals to obtain others that allow to describe in a more natural way the synchronizing algorithms, (also [8] and [2] use signals, however there the settings are completely different from here).

The rest of the paper is organized as follows: in the next section, we give the definitions and introduce the concept of signal. In section 3 we give some basic results on how to compose signals and in section 4 we present the elementary signals that will allow us to easily describe the particular solutions presented in section 5 with time n^2 , 2^n , $n \log n$ and $n\sqrt{n}$. In section 6, we show how to obtain all linear time solutions and all solutions expressed by polynomials with nonnegative integer coefficients. Finally we give some conclusions in section 7. Due to lack of space some proofs (and some figures) are omitted, for a full version of the paper see [11].

2 Preliminaries

In this section we give the definitions and introduce the concept of *signal*, along with some examples. The line of processors is formally seen as a Cellular Automaton.

A one-bit cellular automaton (shortly 1-CA) is an array of n identical finite-state machines (*cells*) and is denoted by a tuple $(Q, \delta, \delta^L, \delta^R)$ where Q is a finite set of states, $\delta : \{0, 1\} \times Q \times \{0, 1\} \rightarrow \{0, 1\} \times Q \times \{0, 1\}$ is the transition function for cells from 2 to $n-1$, $\delta^L : Q \times \{0, 1\} \rightarrow Q \times \{0, 1\}$ and $\delta^R : \{0, 1\} \times Q \rightarrow \{0, 1\} \times Q$ are the transition functions for the first and last cell, respectively. In a 1-CA, the i -th cell is connected to the $(i-1)$ -th and $(i+1)$ -th cells, for all $i = 2, \dots, n-1$.

The first and the last cells have only one connection, respectively to the second and the $(n-1)$ -th cell. Informally each cell exchanges one bit of information with its adjacent cells and modifies its state depending on its current state and the bit received from the adjacent cells. In particular $\delta(a, p, b) = (c, q, d)$ means that a cell in the state p when receives the bit a from the left neighbour and the bit b from the right neighbour enters, at the next time step, the state q and sends the bit c and d to the left and right neighbours, respectively. In the following, Q, δ, δ^L and δ^R always refer to the set of states and to the transition functions of a given 1-CA. Moreover, if A_i is a 1-CA then Q_i is the set of states of A_i and δ_i, δ_i^L and δ_i^R are the transition functions.

We consider the time-unrolling of the 1-CA, that is we will speak of a space-time two dimensional array. A pair (k, t) of this array, with $1 \leq k \leq n$ and $t \geq 1$, is called a *site*, the state of the cell k at time t is denoted $state(k, t)$ and the bit sent to the neighbours are denoted $left(k, t)$ and $right(k, t)$. So, we have that $(left(k, t), state(k, t), right(k, t)) = \delta(right(k-1, t-1), state(k, t-1), left(k+1, t-1))$. A *configuration* of A is a mapping $C : \{1, 2, \dots, n\} \rightarrow \{0, 1\} \times Q \times \{0, 1\}$. A configuration at time t gives, for each cell k , the state entered and the two bit sent at this time. A starting configuration is a configuration at time 1. In the following we often write " (A, C) " to denote a 1-CA A starting on a configuration C . To avoid ambiguities, sometimes we use subscripts in *state*, *left* and *right* to refer to a given 1-CA and a given starting configuration. A site (k, t) is said *active* if at least one of the following conditions holds: $left(k+1, t-1) = 1$ or $right(k-1, t-1) = 1$ or $left(k, t) = 1$ or $right(k, t) = 1$ or $\delta(0, state(k, t), 0) \neq (0, state(k, t), 0)$. Let $Spectrum(A, C)$ denote the set of all the active sites of the 1-CA A with the starting configuration C . We let $Cell(A, C)$ denote the set of cells k such that the site (k, t) is active for some t .

Let A be a 1-CA and C be a configuration, (A, C) is *tailed* if there exists a subset of Q , called $tail(A, C)$, such that for all $k \in \{1, \dots, n\}$

$state(k, t) \in tail(A, C)$ if and only if $t = \max\{t' | (k, t') \in Spectrum(A, C)\}$.

A *simple signal* of (A, C) is a subset S of $Spectrum(A, C)$ such that if (i, t) and $(j, t+1)$ are in S then $j \in \{i-1, i, i+1\}$ and, if (A, C) is tailed, (k, t_k) belongs to S , where t_k is the maximum t such that (k, t) is active. The union of a finite number of simple signals of a given (tailed) (A, C) is called *signal* of (A, C) .

A graphical representation of a simple signal S of a (tailed) (A, C) is obtained by drawing a line between:

- (i) every pair of sites $(k, t) \in S$ and $(k, t+1) \in S$ and
- (ii) every pair of sites $(k, t) \in S$ and $(k+1, t+1) \in S$ (resp. $(k-1, t+1) \in S$) if $right(k, t) = 1$ (resp. $left(k, t) = 1$).

A graphical representation of a signal is obtained by the graphical representation of its simple signals.

The duration of a signal S is $t - t' + 1$ where $t = \max\{s | (i, s) \in S\}$ and $t' = \min\{s | (i, s) \in S\}$. In the following we sometimes will speak of a signal without specifying the 1-CA and the configuration.

The next examples introduce two signals: *COUNT* and *MAX* (see Fig. 1). The *COUNT* signal can be used to check whether the time elapsed between two

events (i.e. two signals crossing a given cell) is less than or equal to a given constant. The *MAX* signal is the “maximum rate” signal, which can be used to transmit the bit 1 from a cell to another as fast as possible.

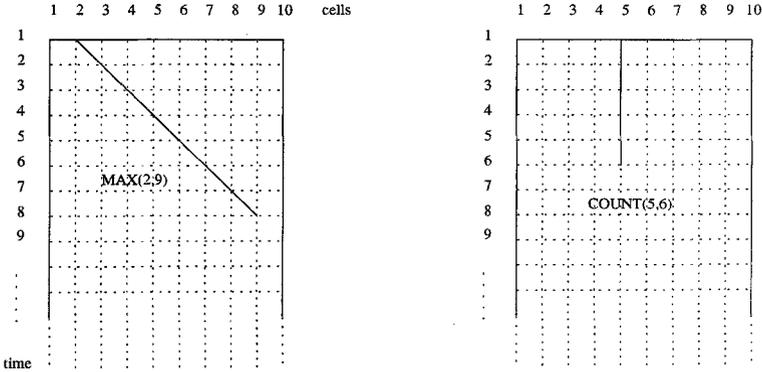


Fig. 1. Example of a *MAX* and a *COUNT* signal on a line of 10 cells.

Example 1. The set $COUNT(i, h)$ containing the sites (i, l) , $1 \leq l \leq h$, is a simple signal of duration k . The cell i can be seen as a counter from time 1 to k . This set is a signal of a tailed 1-CA whose starting configuration may be in such a way that the state of the cell i is different from the states of the other cells.

Example 2. Let $i \neq j$ and let $MAX(i, j)$ be the set containing the sites $(i + h, h + 1)$ if $i < j$, or the sites $(i - h, h + 1)$ otherwise, for $0 \leq h \leq |i - j| + 1$. This set is a simple signal, with duration $|i - j| + 1$, of a tailed 1-CA that starts from a configuration having the states of cells i and j different from all others.

A $t(n)$ -firing signal is a signal whose duration is $t(n)$ and contains the sites $(1, 1)$ and $(i, t(n))$ for $i = 1, \dots, n$.

Let us now introduce the notion of 1-bit solution to the FSSP.

Definition 1. Let $A = (Q, \delta, \delta^L, \delta^R)$ be a 1-CA such that Q contains three particular states, G, L and F , and L has the property that $\delta(0, L, 0) = (0, L, 0)$, $\delta^R(0, L) = (0, L)$ and $\delta^L(L, 0) = (L, 0)$. Let C be such that $C(1) = (0, G, 1)$ and $C(i) = (0, L, 0)$, for $i = 2, \dots, n$. The 1-CA A is a one-bit solution (or shortly solution) to the FSSP in time $t(n)$ if (A, C) is tailed, with $\text{tail}(A, C) = \{F\}$, and there exists a $t(n)$ -firing signal of (A, C) .

The starting configuration for a solution to the FSSP is called in what follows a *standard* configuration. Sometimes we will refer to the states G, L and F as the *General, Latent* and *Firing* states. Note that in the definition above the leftmost cell is in the General state: clearly a symmetric solution can be obtained by letting the cell n instead be in the General state.

Minsky in [9] has shown that $2n - 1$ is the minimal time for the solution to the FSSP. In [7] Mazoyer has presented a solution in time $2n - 1$.

3 Compositions of signals

In this section we show how to obtain a single signal as a composition of more signals under three different conditions.

From the first lemma it is possible to obtain a new signal from two signals with disjoint sets of active cells.

Lemma 2. *Given a 1-CA A , let S_1 and S_2 be two signals of A respectively on the configurations C_1 and C_2 . If there exists $i \leq j$, such that $\text{Cell}(A, C_1) \subseteq \{1, \dots, i\}$ and $\text{Cell}(A, C_2) \subseteq \{j, \dots, n\}$, then there exist a 1-CA A' and a configuration C' such that $S_1 \cup S_2$ is a signal of (A', C') . Moreover, if (A, C_1) and (A, C_2) are tailed then also (A', C') is tailed.*

Given two signals S_1 and S_2 , we denote with $S_1 \cup_s S_2$ the set

$$S_1 \cup \{(k, t + s - 1) | (k, t) \in S_2\}$$

and we say that it is the s -union of S_1 and S_2 .

A 1-CA A_2 on C_2 is said that *can follow* a tailed 1-CA A_1 on C_1 if the following conditions hold:

- i) $\{k | (k, 1) \in \text{Spectrum}(A_2, C_2)\} \subseteq \text{Cell}(A_1, C_1)$ and
- ii) if $\text{state}_{A_1, C_1}(k, t) = \text{state}_{A_1, C_1}(k', t') \in \text{tail}(A_1, C_1)$ then $C_2(k) = C_2(k')$.

Clearly when A_2 can follow A_1 , then the function $\text{foll}(q) = (a, p, b)$, where $q = \text{state}(k, t) \in \text{tail}(A_1, C_1)$ and $C_2(k) = (a, p, b)$, exists. Thus the set $\text{tail}(A_1, C_1)$ can be split into two disjoint sets: $\text{active}(A_1, C_1)$ and $\text{non_active}(A_1, C_1)$, such that $(k, 1) \in \text{Spectrum}(A_2, C_2)$ if and only if there exists a state $q \in \text{active}(A_1, C_1)$ such that $\text{foll}(q) = C_2(k)$.

The next lemma establishes when it is possible to design a 1-CA for the s -union of two signals.

Lemma 3. *Let A_2 be a 1-CA on C_2 that can follow a tailed 1-CA A_1 on C_1 . Let S_1 and S_2 be signals of (A_1, C_1) and (A_2, C_2) and $s = \min\{t | \text{state}_{A_1, C_1}(k, t) \in \text{active}(A_1, C_1)\}$. If $t' < t'' + s$ for all t' and t'' such that $(k, t') \in \text{Spectrum}(A_1, C_1)$ and $(k, t'') \in \text{Spectrum}(A_2, C_2)$ then for all integers $r > s$ there exists a 1-CA A and a configuration C , such that $S_1 \cup_r S_2$ is a signal of A on C . Moreover if (A_2, C_2) is tailed and $\text{Cell}(A_1, C_1) \subseteq \text{Cell}(A_2, C_2)$, then (A, C) is tailed too.*

Proof: First let us note that given the condition $t' < t'' + s$ and the definition of S , then all the cells k of (A_1, C_1) such that $(k, 1) \in \text{Spectrum}(A_2, C_2)$, enter a state belonging to $\text{tail}(A_1, C_1)$ at the same time s . Assume now, without loss of generality, that the sets Q_1 and Q_2 are disjoint. Define $C(k)$ equal to $C_1(k)$ if $k \in \text{Cell}(A_1, C_1)$ and equal to $C_2(k)$ otherwise.

Given an integer r , the behaviour of a 1-CA A for the signal $S_1 \cup_r S_2$ on C can be split in three phases: A initially behaves as A_1 , then at time s , if the site $(k, 1)$ of (A_2, C_2) is active, the site (k, s) enters $\text{state}_{A_1, C_1}(k, s) \in \text{tail}(A_1, C_1)$ and at time r the site (k, r) enters $\text{state}_{A_2, C_2}(k, 1)$ and from now on A behaves as A_2 . Note that on the cells k such that $\text{state}_{A_1, C_1}(k, s) \notin \text{active}(A_1, C_1)$ the 1-CA A immediately switches from A_1 to A_2 at the time t where $\text{state}_{A_1, C_1}(k, t) \in \text{non_active}(A_1, C_1)$. We omit the formal definition of A .

In Lemma 3 we have defined a 1-CA A that performs initially as A_1 and then as A_2 . Anyway, while switching from the first to the second 1-CA, some time $t > 0$ is elapsed: now we show that if two additional conditions hold then a similar construction is possible also for $t = 0$. The following Lemma gives the third approach to compose two signals.

Lemma 4. *Let A_2 be a 1-CA on C_2 that can follow a tailed 1-CA A_1 on C_1 . Let S_1 and S_2 be signals of (A_1, C_1) and (A_2, C_2) and $s = \min\{t \mid \text{state}_{A_1, C_1}(k, t) \in \text{active}(A_1, C_1)\}$. If $t' < t'' + s$ for all t' and t'' such that $(k, t') \in \text{Spectrum}(A_1, C_1)$ and $(k, t'') \in \text{Spectrum}(A_2, C_2)$ and for all sites (k, t) the following holds: $(k, s + t - 1) \in \text{Spectrum}(A_1, C_1)$ and $(k, t) \in \text{Spectrum}(A_2, C_2)$ implies $t = 1$, then there exists a 1-CA A and a configuration C such that $S_1 \cup_s S_2$ is a signal of A on C . Moreover, if (A_2, C_2) is tailed and $\text{Cell}(A_1, C_1) \subseteq \text{Cell}(A_2, C_2)$, then (A, C) is tailed too.*

Proof : The arguments are similar to those of Lemma 3. The behaviour of A can informally be described in two phases (with respect to the proof of Lemma 3, the second phase is skipped): A initially behaves as A_1 , then at time s , if the site $(k, 1)$ of (A_2, C_2) is active, the site (k, s) enters $\text{state}_{A_2, C_2}(k, 1)$ and from now on it behaves as A_2 . As in the preceding Lemma it may be the case that the sites (k', t') of A_1 and $(k', t' - s + 1)$ of A_2 and the sites (k'', t'') of A_1 and $(k'', t'' - s + 1)$ of A_2 , are all active. Here the additional constraint ensures that in such a case $t' = t'' = s$.

4 Some Elementary Signals

In this section we describe the elementary signals that will be used to obtain some solutions to the FSSP. In the following, we often refer to a (tailed) 1-CA without specifying the starting configuration and we call *tail* a state in $\text{tail}(A, C)$, for given A and C .

Given a signal S , to refer to the set of sites $\{(k, t + r - 1) \mid (k, t) \in S\}$ for some $r > 0$ we insert the parameter r in the descriptor of S , that is, for example, with $\text{COUNT}(i, h, r)$ we denote the set of the sites $\{(i, t + r - 1) \mid (i, t) \in \text{COUNT}(i, h)\}$. Obviously, $\text{COUNT}(i, h) = \text{COUNT}(i, h, 1)$.

Note that the 1-CA for the $\text{COUNT}(i, h)$ signal can be easily modified in order to halt the computation in the successive step if a bit 1 is received by the cell i at some time less than h . From now on, whenever we use a COUNT signal, the corresponding 1-CA has this additional feature.

The signal WAVE. Let $i \neq j$, the signal $\text{WAVE}(i, j)$ consists of the union of the sets of sites $\text{MAX}(i, j, 1)$ and $\text{MAX}(j, i, |i - j| + 1)$. Note that $\text{MAX}(i, j)$ and $\text{MAX}(j, i)$ are both signals of a tailed 1-CA starting from a configuration in which cells i and j can be distinguished from each other. Thus $\text{WAVE}(i, j) = \text{MAX}(i, j) \cup_r \text{MAX}(j, i)$, where $r = |i - j| + 1$, is a signal of a tailed 1-CA starting from the same configuration (Lemma 4) and its duration is $2|i - j| + 1$.

The signal MARK. Let $i \leq j - k$ and $k > 0$, the signal $\text{MARK}(i, j - k)$ is the union of the sets of sites $\text{MAX}(i, j, 1)$, $\text{MAX}(j, k, j - i + 1)$ and $\text{COUNT}(i +$

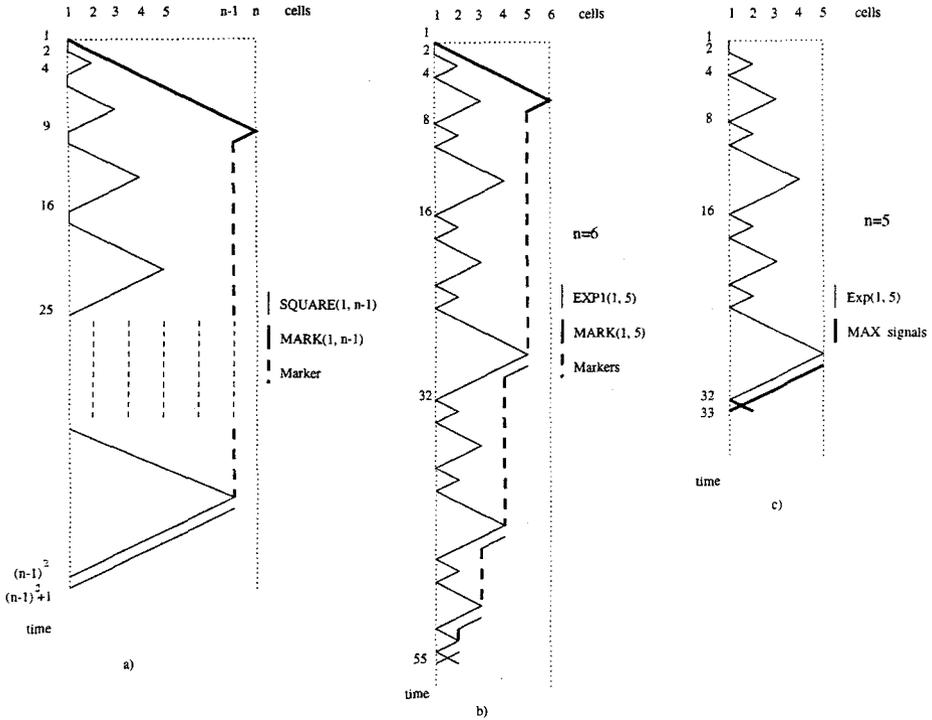


Fig. 2. The signals: a) $SQUARE(1, n-1)$ and the whole first phase of the solution to the FSSP in time n^2 , b) $EXP1(1, 5)$ with the signals of the whole first phase of the solution to the FSSP in time 2^n and c) $EXP2(1, 5)$.

$h, 2k + 1, h + 1)$ for $0 \leq h < j - i$. By Lemma 4 $MARK$ is a signal of a tailed 1-CA. Finally, the signal of $COUNT(j - k, 2k + 1, j - k - i + 1)$ ends on the site $(j - k, j - i - k)$ that is in $MAX(j, k, j - i + 1)$ and $COUNT(j - k, 2k + 1, j - k - i + 1)$ is the unique $COUNT$ signal ending on this MAX signal. Thus, the signal $MARK$ can be used to mark the cell $j - k$, for a fixed k . It is easy to verify that the duration of the signal $MARK$ is $|i - j| + k + 1$, see Fig. 2 (the signals $COUNT$ have been omitted).

The signal SQUARE. Let $i < j$ (resp. $i > j$), the signal $SQUARE(i, j)$ is the union of the sets of sites $COUNT(i, 2, 1)$, $WAVE(i, i + h, h^2 + 1)$ (resp. $WAVE(i, i - h, h^2 + 1)$) for all $1 \leq h \leq |i - j|$ and $MAX(j, i, (j - i)^2 + |j - i| + 2)$. $SQUARE(i, j)$ is a signal of a 1-CA A which can be described as follows (suppose for simplicity that $i < j$):

- first the cell i counts one time unit and sends a bit 1 to the right; then, if it receives a bit 1 from the right, it sends with a delay of one step, a bit 1 back to the right; finally, the cell i halts when it receives two consecutive bit 1;
- for $1 \leq h < |i - j|$, the cell $i + h$ sends a bit 1 to the left when it receives for

- the first time a bit 1 from the left; after, if the cell $i + h$ receives again a bit 1 from an adjacent cell, it sends a bit 1 to the other neighbour;
- the cell j sends two consecutive bit 1 to the left when it receives a bit 1 from the left.

The tailed 1-CA for $SQUARE(i, j)$ can be easily obtained from the previous one by the following observation: the cells from i to j can enter a tail state when they receive two consecutive bit 1. The duration of the $SQUARE$ signal is $(|i - j| + 1)^2 + 1$, (see Fig. 2).

The signals EXP1 and EXP2. For the signals $EXP1(i, j - 1)$ and $EXP2(i, j)$ we only consider the case $i < j$, the case $i > j$ is symmetric. First let us introduce the set of sites $Exp(i, j, t)$ that is recursively defined as:

$$Exp(i, j, t) = \begin{cases} WAVE(i, j, t) & \text{if } j = i + 1 \\ A \cup B \cup C - D & \text{otherwise.} \end{cases}$$

where the sets A, B, C and D are so defined: $A = Exp(i, j - 1, t)$, $B = Exp(i, j - 1, t + 2^{j-i} - 2)$, $C = WAVE(i, j, t + 2^{j-i+1} - 2(j-i) - 2)$ and $D = WAVE(i, j - 1, t + 2^{j-i+1} - 2(j-i) - 2)$.

Thus, $EXP1(i, j - 1)$ is the union of:

$$\begin{aligned} &COUNT(i, 2, 1), \\ &Exp(i, j, 2) - WAVE(i, j, 2^{j-i+1} - 2(j-i)), \\ &MAX(j-h, j-h-1, 3 + \sum_{l=1}^h (2^{j-i-l+1} - 2) - (j-i-h)), 1 \leq h \leq j-i-1, \\ &\text{and } MAX(1, 2, 2^{j-i+1} - 2(j-i)) \\ &\text{(see Fig. 2).} \end{aligned}$$

The signal $EXP2(i, j)$ is the union of:

$$COUNT(i, 2, 1), Exp(i, j, 2), MAX(j, i, 2^{j-i+1} - j + i) \text{ and } MAX(1, 2, 2^{j-i+1})$$

(see Fig. 2).

In the following description an *ending* cell is a cell which neither changes its state nor sends a bit 1 unless it receives a bit 1. At the beginning the only ending cell is the cell $j - 1$. The $EXP1(i, j - 1)$ is a signal of a 1-CA which can be described as follows:

- first the cell i counts one time unit and sends a bit 1 to the right; then, whenever the cell i receives the bit 1 from the right, immediately it replies sending back a bit 1; finally, if the cell i receives two consecutive bit 1 from the right then it changes into an ending cell;
- for $1 \leq h < (j - i - 1)$, if the cell $i + h$ receives a bit 1 from the left, then it alternates the following two behaviours:
 - it sends a bit 1 back to the left,
 - it sends a bit 1 to the right;
- for $1 \leq h < (j - i - 1)$, the cell $i + h$ sends a bit 1 to the left when it receives a bit 1 from the right and if, immediately after, the cell $i + h$ receives another bit 1 from its right neighbour, then it changes into an ending cell;
- an ending cell sends two consecutive bits 1 to the left when it receives a bit 1 for the first time.

Note that a tail state can be entered by an ending cell, so obtaining a tailed 1-CA.

The tailed 1-CA for $EXP2(i, j)$ is very similar to the previous one and we omit the description.

By induction on $j \geq (i + 1)$ one can show that the duration of the signal $Exp(i, j)$ is $2^{|i-j|+1} - 1$. From the definition of $EXP1(i, j-1)$ and the duration of $COUNT$, MAX and $WAVE$ signals, we have that the duration of $EXP1(i, j-1)$ is $1 + (2^{|i-j|+1} - 1) - (2|i-j| + 1) + 1 + 1 = 2^{|i-j|+1} - 2|i-j| + 1$. Analogously, the duration of $EXP2(i, j)$ is $2^{|i-j|+1} + 1$.

5 Some Particular Solutions

In this section we show the existence of solutions to the FSSP in time n^2 , 2^n , $n\lceil \log n \rceil$ and $n\lceil \sqrt{n} \rceil$. The first two solutions are obtained quite easily by using the elementary signals and the given results that allow us to combine signals. The other two solutions are more difficult.

Theorem 5. *There is a solution to the FSSP in time n^2 .*

Proof : The solution is divided into two phases: Initialization and Synchronization. The Initialization phase has duration $(n - 1)^2 + 1$ and consists of $MARK(1, n - 1) \cup_1 SQUARE(1, n - 1)$, see Fig. 2. By Lemma 4, this phase is a signal of a tailed 1-CA starting from a standard configuration.

The Synchronization phase consists of a minimal time solution to the FSSP. By Lemma 4, there is a 1-CA A such that the r -union of the two phases, for $r = (n - 1)^2 + 1$, is a n^2 - firing signal of A starting from a standard configuration. Thus A is a solution to the FSSP in time n^2 .

Theorem 6. *There is a solution to the FSSP in time 2^n .*

Proof : As in the proof above, the solution is divided into two phases: Initialization and Synchronization. The first phase consists of the signals $MARK(1, n - 1)$ and $EXP1(1, n - 1)$, see Fig. 2. By Lemma 4, Initialization is a signal of a tailed 1-CA starting from a standard configuration and has duration $2^n - 2n + 3$.

The Synchronization phase can be seen as a minimal time solution to the FSSP without the first configuration and thus its duration is $2n - 2$. It is easy to see that Synchronization is a signal that can follow the Initialization phase. By Lemma 3 there is a 1-CA A that initially performs the Initialization and then the Synchronization phases. Thus A is a solution to the FSSP in time 2^n .

Theorem 7. *There is a solution to the FSSP in time $n\lceil \log n \rceil$.*

Proof : The proof is quite involved, here it is only sketched. The solution has two different behaviours depending on whether $n \leq 4$ or $n > 4$. We can let the 1-CA behave in these two different manners as each cell i can determine at a time $t < 2n$ whether $n \leq 4$ and, in positive case, determine also the values

i, n and t . This way all the cells can move in the firing state exactly at time $2n$. Here for simplicity let us assume $n > 4$ and $\log(n - 1)$ not integer. The solution is then divided into three phases: the *Initialization*, the *Iterative* and the *Synchronization* phases. The Iterative phase is executed only if $n > 8$. Let us very briefly describe the whole solution. In the Initialization phase some cells are marked: the cells number $3, \lceil n/2 \rceil, \lfloor n/2 \rfloor + 1$ and $n - 2$. Moreover the test $n < 8$ is performed in such a way that at the end of the phase all the cells are aware of the result. If $n > 8$ this phase takes time $2n$, otherwise it takes $\max\{2n + 1, 2n - 2\lceil \log n \rceil + \lfloor n/2 \rfloor + 3\}$. In the Iterative phase the following two steps are iterated $(\log n - 2)$ times: during the i -th iteration, in the first step the test whether $i + 3 < \lceil \log n \rceil$ is performed, then in the second step the output of this test is spread to all the cells in the line. The time taken by this phase is $n(\log n - 2)$. The third and last phase is actually a minimal time solution on a line of $\lfloor n/2 \rfloor$ processors.

In each phase of the preceding solution the signal *EXP2* is intensively used. The solution in time $n\sqrt{n}$ can be obtained through exactly the same schema by merely substituting the *SQUARE* signals for the *EXP2* signals.

Theorem 8. *There is a solution to the FSSP in time $n\lceil\sqrt{n}\rceil$.*

6 Polynomial Time Solutions

In this section we show how to obtain solutions to the FSSP by composing other solutions. We also prove the existence of solutions in linear time and solutions whose time is expressed by polynomials with nonnegative integer coefficients.

In the following, if A_i is a solution to the FSSP, then G_i, L_i and F_i are the General, Latent and Firing states of A_i , respectively.

Lemma 9. *If A_i for $i = 1, 2$ are two solutions to the FSSP in time $t_i(n)$ and $d \geq 0$, then there is a solution to the FSSP in time $t_1(n) + t_2(n) + d$.*

Proof : Suppose that S_i is the $t_i(n)$ -firing signal of (A_i, C_0) , where C_0 is a standard configuration. From Lemma 3, if $r = t_1(n) + d + 1$, then there exists A such that $S_1 \cup_r S_2$ is a signal of (A, C_0) . Moreover, $S_1 \cup_r S_2$ is a $t(n)$ -firing signal with $t(n) = t_1(n) + t_2(n) + d$.

Lemma 10. *If A_i for $i = 1, 2$ are two solutions to the FSSP in time $t_i(n)$, then there is a solution to the FSSP in time $t_1(n)t_2(n)$.*

Proof : We define a solution A consisting of an Iterative phase with duration $t_1(n)$ which is executed $t_2(n)$ times. The set of states of A is $Q_1 \times Q_2 \times \{0, 1\}^2$, the General state is $(G_1, G_2, 0, 1)$, the Latent state is $(L_1, L_2, 0, 0)$ and the Firing state is $(F_1, F_2, 0, 0)$. In the Iterative phase, the 1-CA A modifies the first component of its state according to the transition functions of A_1 , until this component is F_1 . At the end of this phase A executes a transition step modifying the second component of the state according to the transition functions of

A_2 . The bit given as output by the transition functions of A_2 are saved in the third and fourth component. Moreover, in this same step, A replaces F_1 with either G_1 or L_1 (depending on whether the cell is the first in the line or not) in the first component. So the Iterative phase can start again, until the firing state is entered by all the cells. As consequence, the solution A_1 is iterated exactly $t_2(n)$ times.

Let A be a solution to the FSSP in time $t(n)$ and $X \subseteq \{1, \dots, n\}$, we say that A is X -detectable if for every $k \in X$ the set of states containing $state(k, t(n) - 1)$ is disjoint from the set of states containing $state(j, t(n) - 1)$, for $j \neq k$. Further, we say that A has the *parity property* if the following conditions hold:

- the set of states containing $state(1, t(n) - 1)$ in the case that n is even is disjoint from the set containing $state(1, t(n) - 1)$ in the case that n is odd;
- the set of states containing $state(n, t(n) - 1)$ in the case that n is even is disjoint from the set containing $state(n, t(n) - 1)$ in the case that n is odd.

Lemma 11. *Let $d \geq 0$ and $n \geq d$. Let A be a solution to the FSSP in time $t(n)$ with the parity property and X -detectable for a set $X = \{1, \dots, d\} \cup \{n - d + 1, \dots, n\} \cup \{n/2, n/2 + 1\}$, if n is even, and $X = \{1, \dots, d\} \cup \{n - d + 1, \dots, n\} \cup \{\lceil n/2 \rceil\}$, if n is odd. Then there exists a solution to the FSSP in time $t(n) + n - d$.*

Lemma 12. *Let A be a solution to the FSSP in time $t(n)$ with the parity property and X -detectable for a set $X = \{n/2, n/2 + 1\}$, if n is even and $X = \{\lceil n/2 \rceil\}$ if n is odd. Then there is a solution to the FSSP in time $nt(n)$.*

From [7] the following remark holds:

Remark. There exists a minimal time solution to the FSSP with the parity property and X -detectable for a set $X = \{1, 2, n - 1, n\} \cup \{n/2, n/2 + 1\}$, if n is even and $X = \{1, 2, n - 1, n\} \cup \{\lceil n/2 \rceil\}$ if n is odd.

Theorem 13. *Let a and b be two integer numbers. If $an + b \geq 2n - 1$ then there is a solution to the FSSP in time $an + b$.*

Proof : From the above Remark and the Lemma 11 (for $d = 2$) the existence of solutions to the FSSP in time $2n - 1 + k(n - 2)$ follows, for every $k \geq 0$. The condition $an + b \geq 2n - 1$ implies that $a \geq 2$ and $b \geq 3 - 2a$, so a one bit solution in time $an + b$ can be obtained by adding $b - 3 + 2a$ time units to a solution in time $2n - 1 + (a - 2)(n - 2)$.

Theorem 5 shows the existence of a solution to the FSSP in time n^2 which includes a minimal time solution, so, from [7], the following remark follows:

Remark. There exists a solution to the FSSP in time n^2 with the parity property and X -detectable for a set $X = \{n/2, n/2 + 1\}$, if n is even and $X = \{\lceil n/2 \rceil\}$ if n is odd.

Theorem 14. *Let $m \geq 2$ be an integer number and a_0, \dots, a_m natural numbers with $a_m \geq 1$. Then there is a 1-bit solution to the FSSP in time $a_m n^m + \dots + a_1 n^1 + a_0$.*

Proof : From the last Remark and the Lemma 12, a solution in time n^b can be obtained, for every $b \geq 2$ and then, from the Lemma 9 the theorem follows.

7 Conclusions

In this paper we have presented new techniques to synchronize a line of n identical processors at a given time expressed as a function of n . In particular we have given solutions in time n^2 , 2^n , $n\lceil\log n\rceil$, $n\lceil\sqrt{n}\rceil$, in all feasible linear times and in times expressed as polynomials with nonnegative coefficients. This problem was also discussed in [6] but here we have used a completely new approach. The novelty of our approach consists in the description of solutions by means of signals and signal compositions. As future research direction we think that many other solutions can be derived by introducing new signals of the type we have used here. In particular, solutions can be obtained in times expressed as polynomials with integer coefficients. Moreover, the technique used for the solutions in time $n\lceil\log n\rceil$ and $n\lceil\sqrt{n}\rceil$ can be extended to obtain solutions in time $nf^{-1}(n)$, if a solution in time $f(n)$ is known.

Acknowledgements

We thank Jacques Mazoyer for some useful comments on a very preliminary version of the paper.

References

1. R.Balzer, *An 8-states minimal time solution to the firing squad synchronization problem*. Information and Control 10 (1967), 22–42.
2. C.Choffrut and K.Culik II, *On Real Time Cellular Automata and Trellis Automata*. Acta Informatica, 21 (1984), 393–407.
3. K.Culik, *Variations of the firing squad problem and applications*. Information Processing Letters, 30 (1989), 153–157.
4. P.C.Fischer, *Generation of primes by a one-dimensional real-time iterative array*. Journal of the Association for computing Machinery 12 (1965), 388–394.
5. J.Mazoyer, *A six states minimal time solution to the firing squad synchronization problem*. Theoretical Computer Science 50 (1987), 183–238.
6. J.Mazoyer and N.Reimen, *A linear speed-up theorem for cellular automata*. Theoretical Computer Science 101 (1992), 59–98.
7. J.Mazoyer, *A Minimal Time Solution to the Firing Squad Synchronization Problem with only one bit of Information Exchanged*. To appear on Theoretical Computer Science.
8. J.Mazoyer and V.Terrier, *Signals in one dimensional cellular automata*. Research Report N.94–50, Ecole Normale Supérieure de Lyon, France, 1994.
9. F.Minsky, *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.
10. A.Waksman, *An optimum solution to the firing squad synchronization problem*. Information and Control 9 (1966), 66–78.
11. URL: www.unisa.it/papers/fssp.ps.gz