

Article

A Distributed Supervisor Architecture for a General Wafer Production System

Fotis N. Koumboulis ¹, Dimitrios G. Fragkoulis ^{2,*} and Panteleimon Georgakopoulos ²¹ Department of Digital Industry Technologies, School of Science, National and Kapodistrian University of Athens, Euripus Campus, 34400 Euboea, Greece² Core Department, National and Kapodistrian University of Athens, Euripus Campus, 34400 Euboea, Greece

* Correspondence: dfragkoulis@uoa.gr

Abstract: The current trend in the wafer production industry is to expand the production chain with more production stations, more buffers, and robots. The goal of the present paper is to develop a distributed control architecture to face this challenge by controlling wafer industrial units in a general production chain, with a parametric number of production stations, one robot per two stations where each robot serves its two adjacent production stations, and one additional robot serving a parametric number of stations. The control architecture is analyzed for individual control units, one per robot, monitoring appropriate event signals from the control units of the adjacent robots. Each control unit is further analyzed to individual supervisors. In the present paper, a modular parametric discrete event model with respect to the number of production stations, the number of buffers, and the number of robotic manipulators is developed. A set of specifications for the total system is proposed in the form of rules. The specifications are translated and decomposed to a set of local regular languages for each robotic manipulator. The distributed supervisory control architecture is developed based on the local regular languages, where a set of local supervisors are designed for each robotic manipulator. The desired performance of the total manufacturing system, the realizability, and the nonblocking property of the proposed architecture is guaranteed. Finally, implementation issues are tackled, and the complexity of the distributed architecture is determined in a parametric formula. Overall, the contribution of the present paper is the development of a parametric model of the wafer manufacturing systems and the development of a parametric distributed supervisory control architecture. The present results provide a ready-to-hand solution for the continuously expanding wafer production industry.

Keywords: discrete event systems; distributed supervisory control; semiconductor industry

Citation: Koumboulis, F.N.; Fragkoulis, D.G.; Georgakopoulos, P. A Distributed Supervisor Architecture for a General Wafer Production System. *Sensors* **2023**, *23*, 4545. <https://doi.org/10.3390/s23094545>

Academic Editor: Carlo Alberto Avizzano

Received: 5 March 2023

Revised: 27 April 2023

Accepted: 5 May 2023

Published: 7 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

One of the main goals of Industry 4.0 is to improve existing systems and processes via the automatization of manufacturing processes and the interoperability of corresponding applications using cyber-physical systems and the Internet of Things (IoT). The adaptation of these tools in many systems' processes, including installed robotic manipulators, and the direction for more flexible and easily expandable manufacturing systems increase the need for smarter and more efficient control systems. One of the manufacturing sectors into which these tools are introduced at many different stages of the manufacturing process is the semiconductor manufacturing industry [1].

In semiconductor industry, wafer processing units have an important role. The wafer processing unit (see [2–9]) is a manufacturing unit consisting of a set of integrated production stations (process chambers), a set of one-slot buffers, one loading dock (input) of the raw products, one loading dock (output) of the manufactured products, and a set of robotic manipulators that transfer the products between the stations, the buffers, and the input and output docks. For every production station, there is one robotic manipulator dedicated to

serve the production station. The production station and the robotic manipulator constitute an individual manufacturing unit, called a cluster tool. The first studies of wafer manufacturing units (see [2–7]) have been limited to single cluster tool systems up to four cluster tools systems. The systems that have more than one cluster tool are called multicluster tool systems. The current trend in the wafer industry is to increase the number of cluster tools participating in the process (see [8,9]), or/and to increase the number of production stations in each cluster tool (see [10,11]).

For the case of multicluster tool systems, with a parametric number of production stations in each cluster, the number of studies towards synchronization of the cluster tool and scheduling analysis of the manufacturing unit is increasing (indicatively, see [10–14]). The goal of these studies is to increase the production speed and optimize the job scheduling of the manufacturing unit. According to [10], multicluster tool systems can be viewed either from the point of view of the robots or from the point of view of the wafer product flow. In the first case, the production times are considered to be neglectable. Therefore, the study is focused on the supervision of the pick-and-place tasks of the robotic manipulators. In the second case, the processing time is considered to be non-neglectable, while the pick-and-place times are considered to be neglectable. Therefore, the study focuses on the flow of wafers and the processing timing of the wafers in the production sections. In the present paper, the study focuses on the supervision of the robotic manipulators, and the production time is considered to be zero.

The architecture of a cluster tool in the semiconductor industry has been presented in [2], where linear wafer processing has been introduced. In [2], the system's analysis focuses on the time needed for the manufacturing of the wafers to be completed. In [3,4], a wafer processing unit, consisting of nine production stations, three buffers, and four robotic manipulators, is studied. In [3], the wafer processing units are modelled using finite deterministic automata. The automaton of the stations and the buffers has a deadlock. In [3], a set of local supervisors is proposed to avoid the deadlock and coordinate the transfer of the wafers from one production unit to another. In [4], the models developed in [3] have been used and a set of supervisors is proposed to avoid deadlock and coordinate the overall manufacturing unit. In [5,6], a wafer manufacturing unit with two production stations, one buffer, and two robotic manipulators is studied. Furthermore, a set of local supervisors is proposed for the coordination of the unit. In [7], an extended version of the wafer manufacturing unit, consisting of eleven production stations, four buffers, and five robotic manipulators, is studied. In [7], the models of the stations and the buffers are not used and the supervisor design is based on the models of the robotic manipulators. The models of the manipulators can be viewed as controlled versions of the models in [3,4]. In [7], about 70 local supervisors are derived to prevent overflow of the stations and buffers as well as unnecessary use of manipulators. In [8,9], a wafer production line, with a parametric number of robots, is studied. Each robot serves two production stations, except for the last robot, which serves three production stations. The design specifications in [8,9] guarantee nonblocking through a distributed supervisory control scheme including a set of four two-state supervisors for each robotic manipulator and a global supervisor counting the wafers. In [8,9], the design of supervisors is based on abstractions of the automata models of the process in [3]. Both works focus on the time needed for the completion of manufacturing.

In the present paper, a modular discrete event model of a wafer manufacturing unit, being parametric with respect to the number of production stations, the number of buffers, and the number of robotic manipulators, is developed using discrete event systems (DES) and especially the Ramadge–Wonham (RW) framework (see [15,16]). The configuration of the present wafer manufacturing unit consists of a chain of a parametric number of robots, where each robot serves two production stations, and two buffers. The last manipulator serves another parametric number of production stations. The model covers the models in [8,9] as special cases. This general model is the first contribution of the paper. The desired specification is expressed in the form of five rules. The specifications are translated and

decomposed to appropriate sets of local regular languages, one per robotic manipulator. A set of parametric supervisors in the form of finite deterministic automata and in the RW framework is developed. Furthermore, a distributed supervisory control architecture is developed. For distributed supervisory control and its properties, see [15]. The architecture consists of appropriate control units, one per robotic manipulator, monitoring appropriate event signals of the adjacent control units. The development of the present distributed supervisory control, with parametric supervisors, is the second contribution of the paper. The desired performance of the controlled total manufacturing system, the realizability of the control structure (for the notion of realizability see [17]), and the nonblocking property (see [15]) of the proposed architecture are guaranteed. Finally, implementation issues are tackled, and the complexity of the distributed architecture is determined in a parametric formula. The final contribution of the paper is the development of a clear and sufficient method to face possible extensions of the number of production stations and robotic manipulators in the wafer industry. The motivations of the present research were (a) the introduction of a generic modelling for wafer manufacturing in the semiconductor industry, and (b) the cover of the current industrial trends by the extension of the number of the production stations and the number of the serving robotic manipulators. An additional significant motivation was to provide a generic distributed supervisory control architecture, with respect to the system's devices, that will be able to adopt a system's specifications to any possible extensions of the system.

This study is structured as follows: In Section 2, the DES models of the components of the parametric wafer manufacturing system are presented. In Section 3, the desired behavior of the manufacturing process is expressed in the form of a set of desired regular languages. In Section 4, the distributed supervisory control architecture is analyzed. In Section 5, the supervisor automata realizing the desired regular languages are presented. In Section 6, the required system properties of the controlled automaton of the wafer manufacturing system are investigated. In Section 7, simulation results of the controlled automaton regarding a practical sequence of operating commands are presented. In Section 8, the total complexity of the proposed distributed supervisory scheme is calculated. Finally, in Section 9, implementation aspects of the proposed scheme are presented, and the implementation of the proposed supervisors in the ladder diagram (LD) framework are developed.

2. Modelling of the Manufacturing Process

2.1. Parametric Notation and Configuration of the Process

As already mentioned, the wafer manufacturing system consists of production stations, buffers, two loading docks (one for the raw product and one for the manufactured products), and a set of robotic manipulators dedicated to the product transfer between the stations and the two docks. Here, the process is considered in parametric form. The number of robotic manipulators is equal to $n + 1$, where $n \in \mathbb{Z}_+$ and $\mathbb{Z}_+$ comprise the set of positive integers. Each robotic manipulator is indexed by the integer i , where

$$i \in \{1, \dots, n + 1\} \quad (1a)$$

The i -th robotic manipulator is denoted by R_i . It is important to mention that if $i \in \{1, \dots, n\}$, then every R_i serves two production stations, configured as a couple in the production line, while if $i = n + 1$, then R_i serves a parametric number of $m \in \mathbb{Z}_+$ production stations. The total number of production stations is equal to $2n + m$. Furthermore, each production station is indexed by the integers i and j , where

$$j \in \mathbb{J}(i); \mathbb{J}(i) = \begin{cases} \{1, 2\}, & \text{if } i \in \{1, \dots, n\} \\ \{1, 2, \dots, m\}, & \text{if } i = n + 1 \end{cases} \quad (1b)$$

According to (1b), the set $\mathbb{J}(i)$, namely the range of j , depends upon i . The index i is the index of the robot manipulator serving the (i, j) station. The index j is the index of the

production station served by the i -th robot. So, each production station is denoted by $C_{i,j}$. The number of buffers is equal to n . Each buffer is indexed by the integer ν , where

$$\nu \in \{1, \dots, n\} \quad (1c)$$

The symbol B_ν represents the ν -th buffer. The first loading dock is the input dock of the process and is represented by L_{in} . The second loading dock is the output dock of the process and is represented by L_{out} .

To illustrate the above integer definitions, an example is presented. Let $n = 3$ and $m = 3$. Therefore it holds that $\mathbb{J}(1) = \{1, 2\}$, $\mathbb{J}(2) = \{1, 2\}$, $\mathbb{J}(3) = \{1, 2\}$, and $\mathbb{J}(4) = \{1, 2, 3\}$. Thus, there are four robotic manipulators, denoted by R_1, R_2, R_3 , and R_4 , and three buffers, denoted by B_1, B_2 , and B_3 . The number of production stations is equal to nine, and they are denoted by $C_{1,1}, C_{1,2}, C_{2,1}, C_{2,2}, C_{3,1}, C_{3,2}, C_{4,1}, C_{4,2}$, and $C_{4,3}$.

In Figure 1, a simplified drawing of the configuration of the components of the wafer production process is presented. The configuration of the array of the production stations as well as the array of the robotic manipulators and the array of the buffers allows appropriate transfer of products. In particular, this process enables the following:

- (i) R_1 to transfer products from L_{in} to $C_{1,1}$, from $C_{1,1}$ and $C_{1,2}$ to B_1 , and from $C_{1,2}$ to L_{out} ,
- (ii) R_i , where $i \in \{2, \dots, n\}$, to transfer products from $C_{i,1}$ and $C_{i,2}$ to B_i and B_{i-1} , and vice versa, i.e., from B_i and B_{i-1} to $C_{i,1}$ and $C_{i,2}$,
- (iii) R_{n+1} to transfer products from $C_{n+1,1}, \dots, C_{n+1,m}$ to B_n and vice versa.

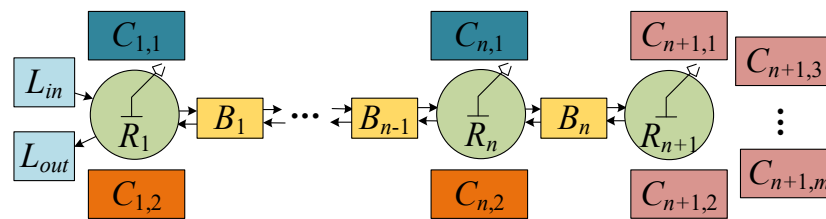


Figure 1. Drawing of the configuration of the wafer manufacturing system.

In the following subsections, the physical entities R_i , $C_{i,j}$, B_ν , L_{in} , and L_{out} , composing the production unit, will be modelled by mathematical entities in DES forms. In particular, the DES model of R_i is denoted by ${}^{ij}\mathbf{G}_R$, the DES model of $C_{i,j}$ is denoted by ${}^{ij}\mathbf{G}$, the DES model of B_ν is denoted by ${}^{ij}\mathbf{G}$, the DES model of L_{in} is denoted by $\mathbf{G}_L$, and the DES model of L_{out} is denoted by $\mathbf{G}_O$.

2.2. The Models of the Production Stations

The model of the production station $C_{i,j}$ is described by the 6-tuple

$${}^{ij}\mathbf{G} = ({}^{ij}\mathbf{Q}, {}^{ij}\mathbf{E}, {}^{ij}f, {}^{ij}\mathbf{H}, {}^{ij}x_0, {}^{ij}\mathbf{Q}_m) \quad (2)$$

For the description of finite deterministic automata in the form of a 6-tuple, see [17–21]. The set of the states of the automaton is

$${}^{ij}\mathbf{Q} = \{{}^{ij}q_1, {}^{ij}q_2, {}^{ij}q_3, {}^{ij}q_4\} \quad (3)$$

The state ${}^{ij}q_1$ describes the case where there is no product at $C_{i,j}$ and the station is in idle mode (standby). The state ${}^{ij}q_2$ describes the case where there is a product at the station, dropped by R_i . The state ${}^{ij}q_3$ describes the case where the processing of the product at the station has been completed and the processed product has not yet been picked up by R_i . The state ${}^{ij}q_4$ describes two malfunction cases. The first is the case where R_i tried to pick a product when there was no processed product at the station. The second is the case where R_i dropped a product to the station when there was a product at the station. Clearly, ${}^{ij}q_4$

describes a functional faulty situation. The initial state is $^{ij}x_0 = ^{ij}q_1$. The set of the marked states of the automaton is $^{ij}Q_m = \{^{ij}q_1, ^{ij}q_2, ^{ij}q_3\}$.

The alphabet of the automaton is

$$^{ij}\mathbb{E} = \{^{ij}e_p, ^{ij}e_D, ^{ij}e_u\} \quad (4)$$

The event $^{ij}e_p$ is the command to R_i to pick a product from the station. The event $^{ij}e_D$ is the command to R_i to drop a product at the station. The event $^{ij}e_u$ takes place when the processing of the product has been completed. The set of the controllable events of the automaton is $^{ij}\mathbb{E}_c = \{^{ij}e_D, ^{ij}e_p\}$. Hence, the set of the uncontrollable events is $^{ij}\mathbb{E}_{uc} = \{^{ij}e_u\}$.

The values of the transition function of the automaton are:

$$\begin{aligned} ^{ij}f(^{ij}q_1, ^{ij}e_D) &= ^{ij}q_2, \quad ^{ij}f(^{ij}q_2, ^{ij}e_u) = ^{ij}q_3, \quad ^{ij}f(^{ij}q_3, ^{ij}e_p) = ^{ij}q_1, \\ ^{ij}f(^{ij}q_1, ^{ij}e_p) &= ^{ij}q_4, \quad ^{ij}f(^{ij}q_2, ^{ij}e_D) = ^{ij}q_4, \quad ^{ij}f(^{ij}q_2, ^{ij}e_p) = ^{ij}q_4, \\ ^{ij}f(^{ij}q_3, ^{ij}e_D) &= ^{ij}q_4. \end{aligned}$$

The active event sets of the automaton are:

$$^{ij}\mathbb{H}(^{ij}q_1) = \{^{ij}e_D, ^{ij}e_p\}, \quad ^{ij}\mathbb{H}(^{ij}q_2) = \{^{ij}e_D, ^{ij}e_p, ^{ij}e_u\}, \quad ^{ij}\mathbb{H}(^{ij}q_3) = \{^{ij}e_D, ^{ij}e_p\}, \quad ^{ij}\mathbb{H}(^{ij}q_4) = \emptyset$$

The closed behavior of the automaton is

$$\mathbb{L}(^{ij}\mathbf{G}) = \overline{(^{ij}e_D ^{ij}e_u ^{ij}e_p)^* (^{ij}e_p + ^{ij}e_D (^{ij}e_D + ^{ij}e_p + ^{ij}e_u ^{ij}e_D))} \quad (5)$$

and the marked behavior of the automaton is

$$\mathbb{L}_m(^{ij}\mathbf{G}) = \overline{(^{ij}e_D ^{ij}e_u ^{ij}e_p)^*} \quad (6)$$

where the symbol “*” denotes the Kleene star of a language. Regarding the Kleene star, see [7,16]. Regarding the definition and properties $\mathbb{L}(\cdot)$ and $\mathbb{L}_m(\cdot)$, namely the closed and the marked behavior of the argument automaton, see [7]. The automaton $^{ij}\mathbf{G}$ in (2) is a blocking automaton, as $\mathbb{L}(^{ij}\mathbf{G}) \neq \overline{\mathbb{L}_m(^{ij}\mathbf{G})}$.

In Figure 2, the state diagram of $^{ij}\mathbf{G}$ is presented. The state diagram of $^{ij}\mathbf{G}$ is the same as the state diagram of the model of the production station proposed in [4].

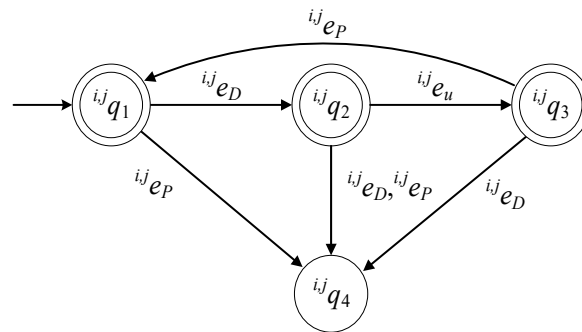


Figure 2. State diagram of the model of C_{ij} .

2.3. The Models of the Buffers

The model of B_v is described by the 6-tuple

$$^v\mathbf{G} = (^vQ, ^v\mathbb{E}, ^vf, ^v\mathbb{H}, ^vx_0, ^vQ_m) \quad (7)$$

The set of the states of the automaton is

$$^vQ = \{^vq_1, ^vq_2, ^vq_3, ^vq_4\} \quad (8)$$

The state ${}^vq_1$ describes the case where the buffer is empty. The state ${}^vq_2$ describes the case where there is a product at the buffer, dropped by R_v . The state ${}^vq_3$ describes the case where there is a product at the buffer, dropped by R_{v+1} . The state ${}^vq_4$ describes two faulty cases. The first is the case where R_v (or R_{v+1}) tried to pick a product when the buffer was empty. The second is the case where R_v (or R_{v+1}) dropped a product when the buffer was full. The initial state of the automaton is ${}^vx_0 = {}^vq_1$. The set of the marked states of the automaton is ${}^vQ_m = \{{}^vq_1, {}^vq_2, {}^vq_3\}$.

The alphabet of the automaton is

$${}^v\mathbb{E} = \{{}^{v,v}e_{BP}, {}^{v,v}e_{BD}, {}^{v+1,v}e_{BP}, {}^{v+1,v}e_{BD}\} \quad (9)$$

The event ${}^{v,v}e_{BP}$ is the command to R_v to pick a product from B_v . The event ${}^{v,v}e_{BD}$ is the command to R_v to drop a product to B_v . The event ${}^{v+1,v}e_{BP}$ is the command to R_{v+1} to pick a product from B_v . The event ${}^{v+1,v}e_{BD}$ is the command to R_{v+1} to drop a product to B_v . The set of the controllable events of the automaton is ${}^v\mathbb{E}_c = {}^v\mathbb{E}$. Hence, the set of the uncontrollable events is ${}^v\mathbb{E}_{uc} = \emptyset$.

The values of the transition functions of the automaton are:

$$\begin{aligned} {}^vf({}^vq_1, {}^{v,v}e_{BD}) &= {}^vq_2, {}^vf({}^vq_1, {}^{v+1,v}e_{BD}) = {}^vq_3, {}^vf({}^vq_2, {}^{v+1,v}e_{BP}) = {}^vq_1, {}^vf({}^vq_3, {}^{v,v}e_{BP}) = {}^vq_1, \\ {}^vf({}^vq_1, {}^{v+1,v}e_{BP}) &= {}^vq_4, {}^vf({}^vq_1, {}^{v,v}e_{BP}) = {}^vq_4, {}^vf({}^vq_2, {}^{v,v}e_{BP}) = {}^vq_4, {}^vf({}^vq_2, {}^{v,v}e_{BD}) = {}^vq_4, {}^vf({}^vq_2, {}^{v+1,v}e_{BD}) = {}^vq_4, \\ {}^vf({}^vq_3, {}^{v,v}e_{BD}) &= {}^vq_4, {}^vf({}^vq_3, {}^{v+1,v}e_{BP}) = {}^vq_4, {}^vf({}^vq_3, {}^{v+1,v}e_{BD}) = {}^vq_4 \end{aligned}$$

The active event sets of the automaton are

$${}^v\mathbb{H}({}^vq_1) = {}^v\mathbb{H}({}^vq_2) = {}^v\mathbb{H}({}^vq_3) = {}^v\mathbb{E}, {}^v\mathbb{H}({}^vq_4) = \emptyset$$

The closed behavior of the automaton is

$$\mathbb{L}({}^v\mathbf{G}) = \overline{({}^{v,v}e_{BD}{}^{v+1,v}e_{BP} + {}^{v+1,v}e_{BD}{}^{v,v}e_{BP})^*({}^{v,v}e_{BP} + {}^{v+1,v}e_{BP} + {}^{v,v}e_{BD}({}^{v,v}e_{BD} + {}^{v,v}e_{BP} + {}^{v+1,v}e_{BD}) + {}^{v+1,v}e_{BD}({}^{v,v}e_{BD} + {}^{v+1,v}e_{BD} + {}^{v+1,v}e_{BP}))} \quad (10)$$

and the marked behavior of the automaton is

$$\mathbb{L}_m({}^v\mathbf{G}) = \overline{({}^{v,v}e_{BD}{}^{v+1,v}e_{BP} + {}^{v+1,v}e_{BD}{}^{v,v}e_{BP})^*} \quad (11)$$

The automaton ${}^v\mathbf{G}$ in (7) is a blocking automaton, as $\mathbb{L}({}^v\mathbf{G}) \neq \mathbb{L}_m({}^v\mathbf{G})$.

In Figure 3, the state diagram of ${}^v\mathbf{G}$ is presented. The state diagram of ${}^v\mathbf{G}$ is the same as the model of the buffer proposed in [4].

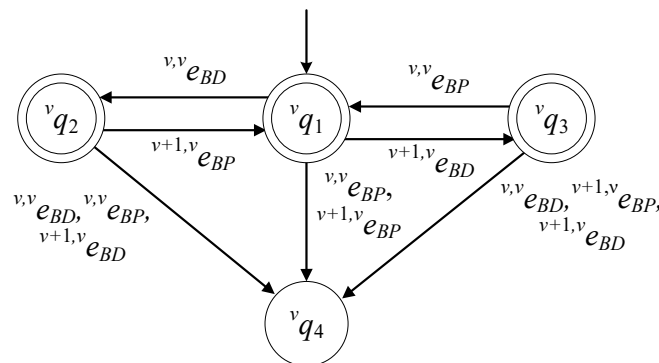


Figure 3. State diagram of the model of B_v .

2.4. The Models of the Loading Docks

The model of L_{in} (input loading dock) is described by the 6-tuple

$$\mathbf{G}_I = (\mathbb{Q}_I, \mathbb{E}_I, f_I, \mathbb{H}_I, x_{I,0}, \mathbb{Q}_{I,m}) \quad (12)$$

The set of the states of the automaton is $\mathbb{Q}_I = \{q_{I,1}\}$. The state $q_{I,1}$ is the only state of the automaton. The initial state of the automaton is $x_{I,0} = q_{I,1}$. The set of the marked states is $\mathbb{Q}_{I,m} = \{q_{I,1}\}$.

The alphabet of the automaton is $\mathbb{E}_I = \{e_I\}$. The event e_I is the command to R_1 to pick a product from L_{in} . The set of the controllable events is $\mathbb{E}_{I,c} = \{e_I\}$. Hence, the set of the uncontrollable events is $\mathbb{E}_{I,uc} = \emptyset$. The transition function is $f_I(q_{I,1}, e_I) = q_{I,1}$. The active event set is $\mathbb{H}_I(q_{I,1}) = \{e_I\}$.

The closed behavior of the automaton is equal to the marked behavior of the automaton, i.e., $\mathbb{L}(\mathbf{G}_I) = \mathbb{L}_m(\mathbf{G}_I) = e_I^*$. Thus, $\mathbf{G}_I$ in (12) is a nonblocking automaton. In Figure 4, the state diagram of $\mathbf{G}_I$ is presented.

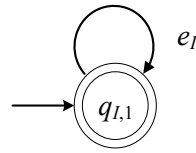


Figure 4. State diagram of the model of L_{in} .

The model of L_{out} (output loading dock) is described by the 6-tuple

$$\mathbf{G}_O = (\mathbb{Q}_O, \mathbb{E}_O, f_O, \mathbb{H}_O, x_{O,0}, \mathbb{Q}_{O,m}) \quad (13)$$

The set of the states of the automaton is $\mathbb{Q}_O = \{q_{O,1}\}$. The state $q_{O,1}$ is the only state of the automaton. The initial state of the automaton is $x_{O,0} = q_{O,1}$. The set of the marked states of the automaton is $\mathbb{Q}_{O,m} = \{q_{O,1}\}$.

The alphabet of the automaton is $\mathbb{E}_O = \{e_O\}$. The event e_O is the command to R_1 to drop a product to L_{out} . The set of the controllable events is $\mathbb{E}_{O,c} = \{e_O\}$. Hence, the set of the uncontrollable events is $\mathbb{E}_{O,uc} = \emptyset$. The transition function of the automaton is $f_O(q_{O,1}, e_O) = q_{O,1}$. The active events set is $\mathbb{H}_O(q_{O,1}) = \{e_O\}$.

The closed behavior of the automaton is equal to the marked behavior of the automaton, i.e., $\mathbb{L}(\mathbf{G}_O) = \mathbb{L}_m(\mathbf{G}_O) = e_O^*$. Thus, $\mathbf{G}_O$ in (13) is a nonblocking automaton. In Figure 5, the state diagram of $\mathbf{G}_O$ is presented.

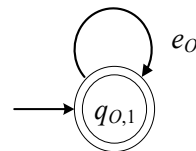


Figure 5. State diagram of the model of L_{out} .

The state diagrams of each of the two loading docks, in Figures 4 and 5, are the same with the respective models in [4].

2.5. The Models of the Robotic Manipulators

The model of R_1 is described by the 6-tuple

$${}^1\mathbf{G}_R = ({}^1\mathbb{Q}_R, {}^1\mathbb{E}_R, {}^1f_R, {}^1\mathbb{H}_R, {}^1x_{R,0}, {}^1\mathbb{Q}_{R,m}) \quad (14)$$

The set of the states of the automaton is ${}^1\mathbb{Q}_R = \{{}^1q_{R,1}, {}^1q_{R,2}\}$. The state ${}^1q_{R,1}$ describes the case where the gripper of R_1 is empty. The state ${}^1q_{R,2}$ describes the case where a product is gripped by R_1 . The initial state of the automaton is ${}^1x_{R,0} = {}^1q_{R,1}$. The set of the marked states is ${}^1\mathbb{Q}_{R,m} = \{{}^1q_{R,1}\}$.

The alphabet of the automaton is

$${}^1\mathbb{E}_R = \{e_I, e_O, {}^{1,1}e_P, {}^{1,1}e_D, {}^{1,2}e_P, {}^{1,2}e_D, {}^{1,1}e_{BP}, {}^{1,1}e_{BD}\} \quad (15)$$

The events of the above alphabet have been defined in the previous subsections. It is important to mention that all events of ${}^1\mathbb{E}_R$ in (15) are controllable.

The values of the transition function of the automaton are

$${}^1f_R({}^1q_{R,1}, e) = {}^1q_{R,2}, \forall e \in \{e_I, {}^{1,1}e_P, {}^{1,2}e_P, {}^{1,1}e_{BP}\}, {}^1f_R({}^1q_{R,2}, e) = {}^1q_{R,1}, \forall e \in \{e_O, {}^{1,1}e_D, {}^{1,2}e_D, {}^{1,1}e_{BD}\}$$

The active event sets of the automaton are

$${}^1\mathbb{H}_R({}^1q_{R,1}) = \{e_I, {}^{1,1}e_P, {}^{1,2}e_P, {}^{1,1}e_{BP}\}, {}^1\mathbb{H}_R({}^1q_{R,2}) = \{e_O, {}^{1,1}e_D, {}^{1,2}e_D, {}^{1,1}e_{BD}\}$$

The closed behavior of the automaton is

$$\mathbb{L}({}^1\mathbf{G}_R) = \overline{((e_I + {}^{1,1}e_P + {}^{1,2}e_P + {}^{1,1}e_{BP})(e_O + {}^{1,1}e_D + {}^{1,2}e_D + {}^{1,1}e_{BD}))^*} \quad (16)$$

and the marked behavior of the automaton is

$$\mathbb{L}_m({}^1\mathbf{G}_R) = \left((e_I + {}^{1,1}e_P + {}^{1,2}e_P + {}^{1,1}e_{BP})(e_O + {}^{1,1}e_D + {}^{1,2}e_D + {}^{1,1}e_{BD}) \right)^* \quad (17)$$

Thus, ${}^1\mathbf{G}_R$ in (14) is a nonblocking automaton.

In Figure 6, the state diagram of ${}^1\mathbf{G}_R$ is presented.

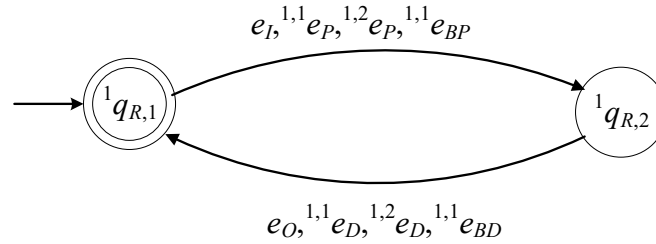


Figure 6. State diagram of the model of R_1 .

The model of R_i where $i = \{2, \dots, n\}$ is described by the 6-tuple

$${}^i\mathbf{G}_R = ({}^i\mathbb{Q}_R, {}^i\mathbb{E}_R, {}^if_R, {}^i\mathbb{H}_R, {}^ix_{R,0}, {}^i\mathbb{Q}_{R,m}) \quad (18)$$

The set of the states of the automaton is

$${}^i\mathbb{Q}_R = \{{}^iq_{R,1}, {}^iq_{R,2}\} \quad (19)$$

The state ${}^iq_{R,1}$ describes the case where the gripper of R_i is empty. The state ${}^iq_{R,2}$ describes the case where a product is gripped by R_i . The initial state of the automaton is ${}^ix_{R,0} = {}^iq_{R,1}$. The set of the marked states of the automaton is ${}^i\mathbb{Q}_{R,m} = \{{}^iq_{R,1}\}$.

The alphabet of the automaton is

$${}^i\mathbb{E}_R = \{{}^{i,1}e_P, {}^{i,1}e_D, {}^{i,2}e_P, {}^{i,2}e_D, {}^{i,i-1}e_{BP}, {}^{i,i-1}e_{BD}, {}^{i,i}e_{BP}, {}^{i,i}e_{BD}\} \quad (20)$$

The events of the above alphabet have been defined in the previous subsections. It is important to mention that all events of ${}^i\mathbb{E}_R$ in (20) are controllable.

The values of the transition function of the automaton are

$${}^if_R({}^iq_{R,1}, e) = {}^iq_{R,2}, \forall e \in \{{}^{i,1}e_P, {}^{i,2}e_P, {}^{i,i-1}e_{BP}, {}^{i,i}e_{BP}\}, {}^if_R({}^iq_{R,2}, e) = {}^iq_{R,1}, \forall e \in \{{}^{i,1}e_D, {}^{i,2}e_D, {}^{i,i-1}e_{BD}, {}^{i,i}e_{BD}\}$$

The active event sets of the automaton are

$${}^i\mathbb{H}_R({}^iq_{R,1}) = \{{}^{i,1}e_P, {}^{i,2}e_P, {}^{i,i-1}e_{BP}, {}^{i,i}e_{BP}\}, {}^i\mathbb{H}_R({}^iq_{R,2}) = \{{}^{i,1}e_D, {}^{i,2}e_D, {}^{i,i-1}e_{BD}, {}^{i,i}e_{BD}\}$$

The closed behavior of the automaton is

$$\mathbb{L}(^i\mathbf{G}_R) = \overline{((^i1e_P + ^i2e_P + ^{i,i-1}e_{BP} + ^{i,i}e_{BP})(^i1e_D + ^i2e_D + ^{i,i-1}e_{BD} + ^{i,i}e_{BD}))^*} \quad (21)$$

and the marked behavior of the automaton is

$$\mathbb{L}_m(^i\mathbf{G}_R) = \left((^i1e_P + ^i2e_P + ^{i,i-1}e_{BP} + ^{i,i}e_{BP})(^i1e_D + ^i2e_D + ^{i,i-1}e_{BD} + ^{i,i}e_{BD}) \right)^* \quad (22)$$

Hence, $^i\mathbf{G}_R$ in (18) is a nonblocking automaton.

In Figure 7, the state diagram of $^i\mathbf{G}_R$ is presented.

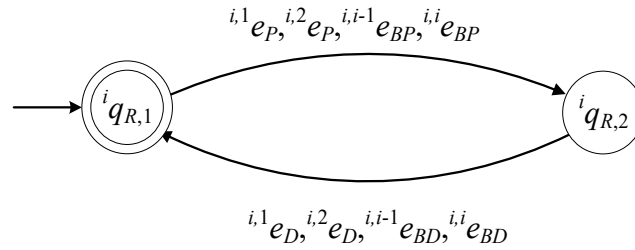


Figure 7. State diagram of the model of R_i .

The model of R_{n+1} is described by the 6-tuple

$$^{n+1}\mathbf{G}_R = (^{n+1}\mathbb{Q}_R, ^{n+1}\mathbb{E}_R, ^{n+1}f_R, ^{n+1}\mathbb{H}_R, ^{n+1}x_{R,0}, ^{n+1}\mathbb{Q}_{R,m}) \quad (23)$$

The set of the states is $^{n+1}\mathbb{Q}_R = \{^{n+1}q_{R,1}, ^{n+1}q_{R,2}\}$. The state $^{n+1}q_{R,1}$ describes the case where the gripper of R_{n+1} is empty. The state $^{n+1}q_{R,2}$ describes the case where a product is gripped by R_{n+1} . The initial state of the automaton is $^{n+1}x_{R,0} = ^{n+1}q_{R,1}$. The set of the marked states of the automaton is $^{n+1}\mathbb{Q}_{R,m} = \{^{n+1}q_{R,1}\}$.

The alphabet of the automaton is

$$^{n+1}\mathbb{E}_R = \bigcup_{j=1}^m \left(\{^{n+1,j}e_P, ^{n+1,j}e_D\} \right) \cup \{^{n+1,n}e_{BP}, ^{n+1,n}e_{BD}\} \quad (24)$$

The events of the above alphabet have been defined in the previous subsections. It is important to mention that all events of $^{n+1}\mathbb{E}_R$ in (24) are controllable.

The values of the transition function of the automaton are

$$^{n+1}f_R(^{n+1}q_{R,1}, e) = ^{n+1}q_{R,2}, \forall e \in \bigcup_{j=1}^m \{^{n+1,j}e_P\} \cup \{^{n+1,n}e_{BP}\}, \quad ^{n+1}f_R(^{n+1}q_{R,2}, e) = ^{n+1}q_{R,1}, \forall e \in \bigcup_{j=1}^m \{^{n+1,j}e_D\} \cup \{^{n+1,n}e_{BD}\}$$

The active event sets of the automaton are

$$^{n+1}\mathbb{H}_R(^{n+1}q_{R,1}) = \bigcup_{j=1}^m \{^{n+1,j}e_P\} \cup \{^{n+1,n}e_{BP}\}, \quad ^{n+1}\mathbb{H}_R(^{n+1}q_{R,2}) = \bigcup_{j=1}^m \{^{n+1,j}e_D\} \cup \{^{n+1,n}e_{BD}\}$$

The closed behavior of the automaton is

$$\mathbb{L}(^{n+1}\mathbf{G}_R) = \overline{\left(\left(\bigoplus_{j=1}^m ^{n+1,j}e_P + ^{n+1,n}e_{BP} \right) \left(\bigoplus_{j=1}^m ^{n+1,j}e_D + ^{n+1,n}e_{BD} \right) \right)^*} \quad (25)$$

and the marked behavior of the automaton is

$$\mathbb{L}_m(^{n+1}\mathbf{G}_R) = \left(\left(\bigoplus_{j=1}^m ^{n+1,j}e_P + ^{n+1,n}e_{BP} \right) \left(\bigoplus_{j=1}^m ^{n+1,j}e_D + ^{n+1,n}e_{BD} \right) \right)^* \quad (26)$$

Hence, $^{n+1}\mathbf{G}_R$ in (23) is a nonblocking automaton.

In Figure 8, the state diagram of $^{n+1}\mathbf{G}_R$ is presented.

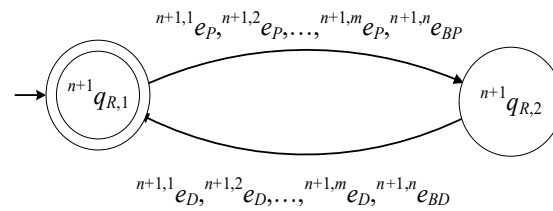


Figure 8. State diagram of the model of R_{n+1} .

For the case where $n = 3$ and $m = 3$, the models of the robotic manipulators, presented in this subsection, are reduced to those presented by respective state diagrams in [4].

3. Desired Behavior

3.1. The Rules of the Desired Behavior

The desired behavior of the wafer manufacturing system can be described in the form of rules as follows:

- Only if a station (or a buffer) is empty will the respective serving robotic manipulator drop a product to the station (or the buffer).
- Only if a station has a manufactured product (or a buffer has a product) will the respective serving robotic manipulator pick a product from the station (or the buffer).
- The desired production sequence, namely the sequence of the placements of a wafer, is presented in the transportation diagram in Figure 9.
- Once a wafer is picked up by R_i , where $i \in \{1, \dots, n\}$, if $C_{i+1,1}$ is full, then the transportation of the wafer to B_i may cause blocking. Hence, a reasonable requirement to avoid blocking is to guarantee an empty slot in $C_{i+1,1}$ before R_i initiates the wafer transportation to B_i .
- Once a wafer is picked up by R_i , where $i \in \{2, \dots, n+1\}$, if $C_{i-1,2}$ is full, then the transportation of the wafer to B_{i-1} may cause blocking. Hence, a reasonable requirement to avoid blocking is to guarantee an empty slot in $C_{i-1,2}$ before R_i initiates the wafer transportation to B_{i-1} .

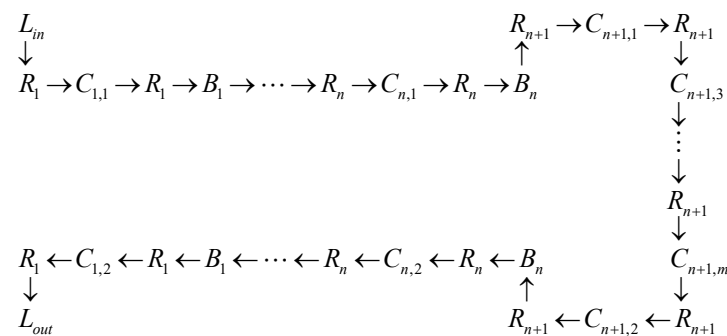


Figure 9. Desired production sequence.

The first rule is a modification of the respective rule in [7]. It is noted that the uncontrollable events, signaling completion of the production process, have not been considered in the model of [7]. The second rule is more restrictive than the respective rule in [7], in the sense that it does not allow the respective robotic manipulator to pick a product if the destination buffer is full. According to the respective rule in [7], the robotic manipulator is not allowed to drop a product to the buffer only if the buffer is full. The third rule is introduced here to guarantee the serial structure of the wafer manufacturing system. This rule is in accordance with the production sequence presented in [7] for the case where $n = 4$ and $m = 3$. The fourth rule has been introduced first here. The aim of the fourth rule is to avoid blocking when transporting products from the machines $C_{i,1}$ to the respective buffer. The fifth rule was first introduced in [7].

3.2. The Desired Behavior in the Form of Regular Languages

For $C_{i,j}$, where $i \in \{1, \dots, n+1\}$ and $j \in \mathbb{J}(i)$, which is given in (1), the first and second rule can be expressed in the form of the following regular languages:

$${}^{1,1}\mathbb{K} = \overline{((e_I + {}^{1,1}e_u)^{*} {}^{1,1}e_D {}^{1,1}e_u {}^{1,1}e_u^* {}^{1,1}e_P)^*} \quad (27a)$$

$${}^{1,2}\mathbb{K} = \overline{(({}^{1,1}e_{BP} + {}^{1,2}e_u)^{*} {}^{1,2}e_D {}^{1,2}e_u {}^{1,2}e_u^* {}^{1,2}e_P)^*} \quad (27b)$$

$${}^{i,1}\mathbb{K} = \overline{((i^{i-1}e_{BP} + i^{i,1}e_u)^{*} i^{i,1}e_D i^{i,1}e_u i^{i,1}e_u^* i^{i,1}e_P)^*}; i \in \{2, \dots, n\} \quad (27c)$$

$${}^{i,2}\mathbb{K} = \overline{((i^{i,2}e_{BP} + i^{i,2}e_u)^{*} i^{i,2}e_D i^{i,2}e_u i^{i,2}e_u^* i^{i,2}e_P)^*}; i \in \{2, \dots, n\} \quad (27d)$$

$${}^{n+1,1}\mathbb{K} = \overline{((n+1,n)e_{BP} + n+1,1e_u)^{*} n+1,1e_D n+1,1e_u n+1,1e_u^* n+1,1e_P)^*} \quad (27e)$$

$${}^{n+1,2}\mathbb{K} = \overline{((n+1,1e_P + n+1,3e_u)^{*} n+1,3e_D n+1,3e_u n+1,3e_u^* n+1,3e_P)^*} \quad (27f)$$

$${}^{n+1,\lambda}\mathbb{K} = \overline{((n+1,\lambda e_P + n+1,\lambda+1e_u)^{*} n+1,\lambda+1e_D n+1,\lambda+1e_u n+1,\lambda+1e_u^* n+1,\lambda+1e_P)^*}; \lambda \in \{3, \dots, m-1\} \quad (27g)$$

$${}^{n+1,m}\mathbb{K} = \overline{((n+1,m e_P + n+1,2e_u)^{*} n+1,2e_D n+1,2e_u n+1,2e_u^* n+1,2e_P)^*} \quad (27h)$$

It is observed that ${}^{i,j}\mathbb{K} \subseteq \mathbb{L}_m({}^{i,j}\mathbf{G})$, where $i \in \{1, \dots, n+1\}$ and $j \in \mathbb{J}(i)$.

For B_i , where $i \in \{1, \dots, n\}$, the combination of the first and second rule is expressed in the form of two specifications. The first specification is as follows: once R_i drops a product to B_i , then the product can be picked only by R_{i+1} . The second specification is as follows: once R_{i+1} drops a product to B_i , then the product can be picked only by R_i . For the two specifications, the corresponding regular languages are:

$${}^i\mathbb{K} = \overline{{}^i c}; i \in \{1, \dots, n\}, {}^i\mathbb{K}_B = \overline{{}^i c_B}; i \in \{2, \dots, n+1\} \quad (28)$$

where the regular expressions ${}^i c$ and ${}^i c_B$ are

$$\begin{aligned} {}^i c &= \left((i^{i+1}e_{BP} + i^{i+1}e_{BD})^* (i^{i,1}e_P (i^{i+1}e_{BP} + i^{i+1}e_{BD} + i^{i+1,2}e_P)^* (i^{i,2}e_{BD}) (i^{i+1}e_{BD} + i^{i+1,2}e_P)^* (i^{i+1}e_{BP}) + \right. \\ &\quad \left. + i^{i+1,2}e_P (i^{i+1}e_{BP} + i^{i+1,2}e_P)^* (i^{i+1}e_{BD}) (i^{i+1}e_{BP} + i^{i+1}e_{BD} + i^{i+1,2}e_P)^* (i^{i,2}e_{BP})) \right), \\ {}^i c_B &= \left((i^{i-1,i-1}e_{BP} + i^{i-1,i-1}e_{BD})^* (i^{i,2}e_P (i^{i-1,1}e_P + i^{i-1,i-1}e_{BP} + i^{i-1,i-1}e_{BD})^* (i^{i,i-1}e_{BD}) (i^{i-1,1}e_P + i^{i-1,i-1}e_{BD})^* (i^{i-1,i-1}e_{BP}) + \right. \\ &\quad \left. + i^{i-1,1}e_P (i^{i-1,1}e_P + i^{i-1,i-1}e_{BP})^* (i^{i-1,i-1}e_{BD}) (i^{i-1,1}e_P + i^{i-1,i-1}e_{BP} + i^{i-1,i-1}e_{BD})^* (i^{i,i-1}e_{BP})) \right)^*. \end{aligned}$$

Regarding the third rule, namely the coordination of the production sequence through the control of the robotic manipulators, the desired behavior of each robotic manipulator is expressed by the following regular languages:

$${}^1\mathbb{K}_R = \left(e_I {}^{1,1}e_D + {}^{1,1}e_P {}^{1,1}e_{BD} + {}^{1,1}e_{BP} {}^{1,2}e_D + {}^{1,2}e_P e_O \right)^* \quad (29a)$$

$${}^i\mathbb{K}_R = \left(i^{i-1}e_{BP} i^{i,1}e_D + i^{i,1}e_P i^{i,2}e_{BD} + i^{i,2}e_{BP} i^{i,2}e_D + i^{i,2}e_P i^{i-1}e_{BD} \right)^*; i \in \{2, \dots, n\}, \quad (29b)$$

$${}^{n+1}\mathbb{K}_R = \left(n+1,n e_{BP} n+1,1e_D + n+1,1e_P n+1,3e_D + \left(\sum_{\lambda=3}^{m-1} (n+1,\lambda e_P n+1,\lambda+1e_D) \right) + n+1,m e_P n+1,2e_D + n+1,2e_P n+1,n e_{BD} \right)^*. \quad (29c)$$

In the formulation of the third rule as the above set of regular languages, appropriate characteristics of the fourth and fifth rules have been used. It is observed that ${}^i\mathbb{K}_R \subseteq \mathbb{L}_m({}^i\mathbf{G}_R)$ and the alphabet of ${}^i\mathbb{K}_R$ is the alphabet ${}^i\mathbb{E}_R$, where $i \in \{1, \dots, n+1\}$.

Proposition 1. *The regular language ${}^i\mathbb{K}_R$, where $i \in \{1, \dots, n+1\}$, is $\mathbb{L}_m({}^i\mathbf{G}_R)$ -closed.*

Proof of Proposition 1. According to [15,16], the language ${}^i\mathbb{K}_R$ is $\mathbb{L}_m({}^i\mathbf{G}_R)$ -closed if ${}^i\mathbb{K}_R = \overline{{}^i\mathbb{K}_R} \cap \mathbb{L}_m({}^i\mathbf{G}_R)$. Clearly, every word of $\overline{{}^i\mathbb{K}_R}$, not belonging to ${}^i\mathbb{K}_R$, ends with a pick event. Furthermore, in every word of $\mathbb{L}_m({}^i\mathbf{G}_R)$, after a pick event, a drop event follows. This drop event activates the transition of ${}^i\mathbf{G}_R$ to its initial state, being the only marked state of ${}^i\mathbf{G}_R$. Thus, $\overline{{}^i\mathbb{K}_R} \cap \mathbb{L}_m({}^i\mathbf{G}_R) \subseteq {}^i\mathbb{K}_R$. Since, ${}^i\mathbb{K}_R \subseteq \overline{{}^i\mathbb{K}_R}$ and ${}^i\mathbb{K}_R \subseteq \mathbb{L}_m({}^i\mathbf{G}_R)$, the proof has been completed. $\square$

The fourth rule is expressed by the following regular language:

$${}^{i,1}\mathbb{K}_D = \overline{({}^{i+1,1}e_p^* {}^{i,1}e_p {}^{i+1,1}e_p)^*}; i \in \{1, \dots, n\} \quad (30)$$

The fifth rule is expressed by the following regular language:

$${}^{i,2}\mathbb{K}_D = \overline{({}^{i-1,2}e_p^* {}^{i,2}e_p {}^{i-1,2}e_p)^*}; i \in \{2, \dots, n+1\} \quad (31)$$

Overall and mainly due to the second rule, the presented design specifications are less permissive than the design specifications proposed in [7] for the wafer manufacturing system with $n = 4$ and $m = 3$. It is noted that in such production lines, blocking can take place in cases where there are more products to be transported to the next stations and buffers than their number. In [7], appropriate coordination supervisors are proposed to avoid blocking. Each coordinator is essentially a product counter controlling only one robotic manipulator. In the present work, blocking is avoided using an alternative approach where, through the second rule, the robotic manipulator is permitted to receive products from a station only if the destination buffer and the respective production station following the buffer are empty.

4. Distributed Supervisory Control Architecture

The control of the wafer manufacturing process will be implemented by local control units, e.g., programmable logic controllers (PLCs), remote terminal units (RTUs), and microcontrollers. Each local control unit is denoted as U_i , where $i \in \{1, \dots, n+1\}$. Each U_i is dedicated to controlling the allowability of the influence of a particular set of controllable events. This event set is disjoint to the respective set of any other local control unit. This way, inspired by the respective decentralized architecture proposed in [7] for $n = 4$ and $m = 3$, a distributed control configuration is proposed. Details regarding distributed supervisory control can be found in [15]. In Section 5, the design specifications, defined by the regular languages presented in Section 3.2, will be realized through appropriate supervisor automata implemented to the local control units U_i , where $i \in \{1, \dots, n+1\}$.

Particularly, the supervisor automata implemented in U_i , where $i \in \{1, \dots, n\}$, are the supervisors ${}^{i,1}\mathbf{S}$, ${}^{i,2}\mathbf{S}$, ${}^i\mathbf{S}$, ${}^i\mathbf{S}_B$, ${}^i\mathbf{S}_R$, ${}^{i,1}\mathbf{S}_D$, and ${}^{i,2}\mathbf{S}_D$. The supervisors realize the languages ${}^{i,1}\mathbb{K}$, ${}^{i,2}\mathbb{K}$, ${}^i\mathbb{K}$, ${}^i\mathbb{K}_B$, ${}^i\mathbb{K}_R$, ${}^{i,1}\mathbb{K}_D$, and ${}^{i,2}\mathbb{K}_D$, respectively. Note that the languages ${}^1\mathbb{K}_B$ and ${}^{1,2}\mathbb{K}_D$ do not correspond to any rule of desired behavior and are introduced exclusively for uniformity reasons to be ${}^1\mathbb{K}_B = {}^1\mathbb{E}_R^*$ and ${}^{1,2}\mathbb{K}_D = {}^1\mathbb{E}_R^*$.

The input signals of U_i , where $i \in \{1, \dots, n\}$, are:

- (i) The events of R_i ;
- (ii) the single uncontrollable event of $C_{i,j}$, namely the event ${}^{i,j}e_u$, where $j \in \{1, 2\}$;
- (iii) The events ${}^{i+1,i}e_{BP}$, ${}^{i+1,i}e_{BD}$, ${}^{i+1,1}e_p$, and ${}^{i+1,2}e_p$, of R_{i+1} ;
- (iv) the events ${}^{i-1,i-1}e_{BP}$, ${}^{i-1,i-1}e_{BD}$, ${}^{i-1,1}e_p$, and ${}^{i-1,2}e_p$ of R_{i-1} , where $i \in \{2, \dots, n\}$.

Particularly, the supervisor automata implemented in U_{n+1} are the supervisors ${}^{n+1,1}\mathbf{S}$, $\dots$, ${}^{n+1,m}\mathbf{S}$, ${}^{n+1}\mathbf{S}$, ${}^{n+1}\mathbf{S}_B$, ${}^{n+1}\mathbf{S}_R$, ${}^{n+1,1}\mathbf{S}_D$, and ${}^{n+1,2}\mathbf{S}_D$. The supervisors realize the languages ${}^{n+1,1}\mathbb{K}$, $\dots$, ${}^{n+1,m}\mathbb{K}$, ${}^{n+1}\mathbb{K}$, ${}^{n+1}\mathbb{K}_B$, ${}^{n+1}\mathbb{K}_R$, ${}^{n+1,1}\mathbb{K}_D$, and ${}^{n+1,2}\mathbb{K}_D$, respectively. Note that the languages ${}^{n+1}\mathbb{K}$ and ${}^{n+1,1}\mathbb{K}_D$ do not correspond to any rule of desired behavior and are introduced exclusively for uniformity reasons to be ${}^{n+1}\mathbb{K} = {}^{n+1}\mathbb{E}_R^*$ and ${}^{n+1,1}\mathbb{K}_D = {}^{n+1}\mathbb{E}_R^*$.

The input signals of U_{n+1} are:

- (i) The events of R_{n+1} ;
- (ii) The uncontrollable events of $C_{n+1,j}$, namely the events ${}^{n+1,j}e_u$, where $j \in \{1, \dots, m\}$;
- (iii) The events ${}^{n,n}e_{BP}$, ${}^{n,n}e_{BD}$, ${}^{n,1}e_P$, and ${}^{n,2}e_P$ of R_n .

Clearly, the events of the categories (iii) and (iv) of U_i , where $i \in \{1, \dots, n\}$, and the events of the category (iii) of U_{n+1} cannot be restricted by the control unit and serve only as monitoring events. The events ${}^{i,j}e_u$, where $i \in \{1, \dots, n+1\}$ and $j \in \mathbb{J}(i)$, being uncontrollable events of the production stations, cannot be restricted by the control unit U_i . Furthermore, the events ${}^{i+1,i}e_{BP}$, ${}^{i+1,i}e_{BD}$, ${}^{i+1,1}e_P$, ${}^{i+1,2}e_P$, and the events ${}^{i-1,i-1}e_{BP}$, ${}^{i-1,i-1}e_{BD}$, ${}^{i-1,1}e_P$, and ${}^{i-1,2}e_P$, where $i \in \{2, \dots, n+1\}$, are used by U_i exclusively for monitoring purposes and are not allowed to be restricted by the control unit U_i . Based on this interconnection, the alphabet of U_i will be augmented as compared to the alphabet of the automaton ${}^i\mathbf{G}_R$. The augmented alphabet is the following:

$${}^i\widetilde{\mathbb{E}}_R = {}^i\mathbb{E}_R \cup \{{}^{i+1,i}e_{BP}, {}^{i+1,i}e_{BD}, {}^{i+1,1}e_P, {}^{i+1,2}e_P, {}^{i-1,i-1}e_{BP}, {}^{i-1,i-1}e_{BD}, {}^{i-1,1}e_P, {}^{i-1,2}e_P, {}^{i,1}e_u, {}^{i,2}e_u\} \quad (32)$$

where $i \in \{1, \dots, n\}$, and

$${}^{n+1}\widetilde{\mathbb{E}}_R = {}^{n+1}\mathbb{E}_R \cup \{{}^{n,n}e_{BP}, {}^{n,n}e_{BD}, {}^{n,1}e_P, {}^{n,2}e_P\} \bigcup_{j=1}^m \{{}^{n+1,j}e_u\} \quad (33)$$

It is important to mention that the set of the events that can be restricted by U_i (controllable event set), where $i \in \{1, \dots, n+1\}$, is ${}^i\widetilde{\mathbb{E}}_{R,u} = {}^i\mathbb{E}_R$. Regarding these event sets, it holds that

$${}^1\widetilde{\mathbb{E}}_{R,u} \cap {}^k\widetilde{\mathbb{E}}_{R,u} = \emptyset, \forall k \in \{3, \dots, n+1\} \quad (34a)$$

$${}^i\widetilde{\mathbb{E}}_{R,u} \cap {}^k\widetilde{\mathbb{E}}_{R,u} = \emptyset, \forall i, k \in \{2, \dots, n\} : (k > i+1) \wedge (k < i-1) \quad (34b)$$

$${}^{n+1}\widetilde{\mathbb{E}}_{R,u} \cap {}^k\widetilde{\mathbb{E}}_{R,u} = \emptyset, \forall k \in \{1, \dots, n-1\} \quad (34c)$$

Also, regarding the adjacent control units, it holds that

$${}^i\widetilde{\mathbb{E}}_R \cap {}^{i+1}\widetilde{\mathbb{E}}_R = \{{}^{i+1,i}e_{BP}, {}^{i+1,i}e_{BD}, {}^{i+1,1}e_P, {}^{i+1,2}e_P\}; i \in \{1, \dots, n\} \quad (35a)$$

$${}^i\widetilde{\mathbb{E}}_R \cap {}^{i-1}\widetilde{\mathbb{E}}_R = \{{}^{i-1,i-1}e_{BP}, {}^{i-1,i-1}e_{BD}, {}^{i-1,1}e_P, {}^{i-1,2}e_P\}; i \in \{2, \dots, n+1\} \quad (35b)$$

Before presenting the design requirements of the present supervisory control scheme, the alphabets of the desired languages will be presented. The alphabets of ${}^{1,1}\mathbb{K}$ and ${}^{1,2}\mathbb{K}$ are ${}^{1,1}\widetilde{\mathbb{E}} = {}^{1,1}\mathbb{E} \cup \{e_I\}$ and ${}^{1,2}\widetilde{\mathbb{E}} = {}^{1,2}\mathbb{E} \cup \{e_O\}$, respectively. The alphabets of ${}^{i,1}\mathbb{K}$ and ${}^{i,2}\mathbb{K}$ are ${}^{i,1}\widetilde{\mathbb{E}} = {}^{i,1}\mathbb{E} \cup \{{}^{i,i-1}e_{BP}\}$ and ${}^{i,2}\widetilde{\mathbb{E}} = {}^{i,2}\mathbb{E} \cup \{{}^{i,i}e_{BP}\}$, where $i \in \{2, \dots, n\}$, respectively. The alphabet of ${}^{n+1,1}\mathbb{K}$ is ${}^{n+1,1}\widetilde{\mathbb{E}} = {}^{n+1,1}\mathbb{E} \cup \{{}^{n+1,n}e_{BP}\}$, the alphabet of ${}^{n+1,2}\mathbb{K}$ is ${}^{n+1,2}\widetilde{\mathbb{E}} = {}^{n+1,2}\mathbb{E} \cup \{{}^{n+1,1}e_P\}$, the alphabet of ${}^{n+1,\lambda}\mathbb{K}$ is ${}^{n+1,\lambda}\widetilde{\mathbb{E}} = {}^{n+1,\lambda+1}\mathbb{E} \cup \{{}^{n+1,\lambda}e_P\}$, where $\lambda \in \{3, \dots, m-1\}$, and the alphabet of ${}^{n+1,m}\mathbb{K}$ is ${}^{n+1,m}\widetilde{\mathbb{E}} = {}^{n+1,2}\mathbb{E} \cup \{{}^{n+1,m}e_P\}$. The alphabet of ${}^i\mathbb{K}$ is ${}^i\widetilde{\mathbb{E}} = \{{}^{i,i}e_{BP}, {}^{i,i}e_{BD}, {}^{i,1}e_P, {}^{i+1,2}e_P, {}^{i+1,i}e_{BP}, {}^{i+1,i}e_{BD}\}$, where

$i \in \{1, \dots, n\}$. The alphabet of ${}^{n+1}\mathbb{K}$ is ${}^{n+1}\tilde{\mathbb{E}} = {}^{n+1}\mathbb{E}_R$. The alphabet of ${}^i\mathbb{K}_B$ is ${}^i\tilde{\mathbb{E}}_B = \{{}^{i,i-1}e_{BP}, {}^{i,i-1}e_{BP}, {}^{i,2}e_P, {}^{i-1,1}e_P, {}^{i-1,i-1}e_{BP}, {}^{i-1,i-1}e_{BD}\}$, where $i \in \{2, \dots, n+1\}$. The alphabet of ${}^1\mathbb{K}_B$ is ${}^1\tilde{\mathbb{E}}_B = {}^1\mathbb{E}_R$. The alphabet of ${}^i\mathbb{K}_R$ is ${}^i\mathbb{E}_R$, where $i \in \{1, \dots, n+1\}$. The alphabet of ${}^{i,1}\mathbb{K}_D$ is ${}^{i,1}\tilde{\mathbb{E}}_D = \{{}^{i,1}e_P, {}^{i+1,1}e_P\}$, where $i \in \{1, \dots, n\}$. The alphabet of ${}^{n+1,1}\mathbb{K}_D$ is ${}^{n+1,1}\tilde{\mathbb{E}}_D = {}^{n+1}\mathbb{E}_R$. Finally, the alphabet of ${}^{i,2}\mathbb{K}_D$ is ${}^{i,2}\tilde{\mathbb{E}}_D = \{{}^{i-1,2}e_P, {}^{i,2}e_P\}$, where $i \in \{2, \dots, n+1\}$. The alphabet of ${}^{1,2}\mathbb{K}_D$ is ${}^{1,2}\tilde{\mathbb{E}}_D = {}^1\mathbb{E}_R$.

The design requirements of the supervisors of U_i , where $i \in \{1, \dots, n\}$, are expressed in terms of a respective controlled automaton, denoted by ${}^i\mathbf{G}_c$, being of the following structure:

$$\begin{aligned} {}^i\mathbf{G}_c &= \mathbf{G}_I || \mathbf{G}_O || {}^i\mathbf{G}_R || {}^{i,1}\mathbf{G} || {}^{i,2}\mathbf{G} || {}^i\mathbf{G} || {}^{i-1}\mathbf{G} || {}^i\mathbf{S}_R || {}^{i,1}\mathbf{S} || {}^{i,2}\mathbf{S} || {}^i\mathbf{S} || {}^i\mathbf{S}_B || {}^{i,1}\mathbf{S}_D || {}^{i,2}\mathbf{S}_D = \\ &= {}^i\mathbf{G}_R || {}^{i,1}\mathbf{G} || {}^{i,2}\mathbf{G} || {}^i\mathbf{G} || {}^{i-1}\mathbf{G} || {}^i\mathbf{S}_R || {}^{i,1}\mathbf{S} || {}^{i,2}\mathbf{S} || {}^i\mathbf{S} || {}^i\mathbf{S}_B || {}^{i,1}\mathbf{S}_D || {}^{i,2}\mathbf{S}_D \end{aligned} \quad (36)$$

where $||$ denotes the synchronous product [15] (or the parallel connection [16]) of two automata, and where ${}^0\mathbf{G} = (\{q_1\}, \emptyset, \emptyset, \emptyset, q_1, \{q_1\})$. Clearly, for every automaton $\mathbf{G}$ it holds that ${}^0\mathbf{G} || \mathbf{G} = \mathbf{G}$. It is important to mention that the final equality in (36) results from the structure of the automata of the loading docks, having only one state and being marked, and a self-transition activated by all events of the automaton's alphabet. The closed and the marked behavior of the synchronous product in (36), where $i \in \{1, \dots, n\}$, are in the following form:

$$\begin{aligned} \mathbb{L}({}^i\mathbf{G}_c) &= {}^iP_R^{-1} (\mathbb{L}({}^i\mathbf{G}_R) \cap \mathbb{L}({}^i\mathbf{S}_R)) \cap {}^iP^{-1} (\mathbb{L}({}^i\mathbf{G})) \cap {}^iP_B^{-1} (\mathbb{L}({}^{i-1}\mathbf{G})) \cap {}^i\tilde{P}^{-1} (\mathbb{L}({}^i\mathbf{S})) \cap {}^i\tilde{P}_B^{-1} (\mathbb{L}({}^i\mathbf{S}_B)) \cap \\ &\quad \bigcap_{j=1}^2 \left({}^{i,j}P^{-1} (\mathbb{L}({}^{i,j}\mathbf{G})) \cap {}^{i,j}\tilde{P}^{-1} (\mathbb{L}({}^{i,j}\mathbf{S})) \cap {}^{i,j}P_D^{-1} (\mathbb{L}({}^{i,j}\mathbf{S}_D)) \right), \end{aligned} \quad (37a)$$

$$\begin{aligned} \mathbb{L}_m({}^i\mathbf{G}_c) &= {}^iP_R^{-1} (\mathbb{L}_m({}^i\mathbf{G}_R) \cap \mathbb{L}_m({}^i\mathbf{S}_R)) \cap {}^iP^{-1} (\mathbb{L}_m({}^i\mathbf{G})) \cap {}^iP_B^{-1} (\mathbb{L}_m({}^{i-1}\mathbf{G})) \cap {}^i\tilde{P}^{-1} (\mathbb{L}_m({}^i\mathbf{S})) \cap {}^i\tilde{P}_B^{-1} (\mathbb{L}_m({}^i\mathbf{S}_B)) \cap \\ &\quad \bigcap_{j=1}^2 \left({}^{i,j}P^{-1} (\mathbb{L}_m({}^{i,j}\mathbf{G})) \cap {}^{i,j}\tilde{P}^{-1} (\mathbb{L}_m({}^{i,j}\mathbf{S})) \cap {}^{i,j}P_D^{-1} (\mathbb{L}_m({}^{i,j}\mathbf{S}_D)) \right), \end{aligned} \quad (37b)$$

where ${}^iP_R$ is the projection of ${}^i\tilde{\mathbb{E}}_R^*$ to ${}^i\mathbb{E}_R^*$; ${}^{i,j}P$ and ${}^{i,j}\tilde{P}$ are the projections of ${}^i\tilde{\mathbb{E}}_R^*$ to ${}^{i,j}\mathbb{E}^*$ and ${}^{i,j}\tilde{\mathbb{E}}^*$, respectively; ${}^iP$ and ${}^i\tilde{P}$ are the projections of ${}^i\tilde{\mathbb{E}}_R^*$ to ${}^i\mathbb{E}^*$ and ${}^i\tilde{\mathbb{E}}^*$, respectively; ${}^iP_B$ and ${}^i\tilde{P}_B$ are the projections of ${}^i\tilde{\mathbb{E}}_R^*$ to ${}^{i-1}\mathbb{E}^*$ and ${}^{i-1}\tilde{\mathbb{E}}^*$, respectively; ${}^{i,1}P_D$ is the projection of ${}^i\tilde{\mathbb{E}}_R^*$ to ${}^{i,1}\tilde{\mathbb{E}}_D^*$; and ${}^{i,2}P_D$ is the projection of ${}^i\tilde{\mathbb{E}}_R^*$ to ${}^{i,2}\tilde{\mathbb{E}}_D^*$.

For $i = n+1$, the controlled automaton is required to be of the following structure:

$${}^{n+1}\mathbf{G}_c = {}^{n+1}\mathbf{G}_R || \left(\bigcap_{j=1}^m ({}^{n+1,j}\mathbf{G} || {}^{n+1,j}\mathbf{S}) \right) || {}^n\mathbf{G} || {}^{n+1}\mathbf{S}_R || {}^{n+1}\mathbf{S}_B || {}^{n+1,2}\mathbf{S}_D \quad (38)$$

Clearly, the closed and the marked behavior of the synchronous product in (38) are in the following form:

$$\begin{aligned} \mathbb{L}({}^{n+1}\mathbf{G}_c) &= {}^{n+1}P_R^{-1} (\mathbb{L}({}^{n+1}\mathbf{G}_R) \cap \mathbb{L}({}^{n+1}\mathbf{S}_R)) \cap {}^{n+1}P_B^{-1} (\mathbb{L}({}^n\mathbf{G})) \cap {}^{n+1}\tilde{P}_B^{-1} (\mathbb{L}({}^{n+1}\mathbf{S}_B)) \cap \\ &\quad \bigcap_{j=1}^m \left({}^{n+1,j}P^{-1} (\mathbb{L}({}^{n+1,j}\mathbf{G})) \cap {}^{n+1,j}\tilde{P}^{-1} (\mathbb{L}({}^{n+1,j}\mathbf{S})) \right) \cap {}^{n+1,2}P_D^{-1} (\mathbb{L}({}^{n+1,2}\mathbf{S}_D)), \end{aligned} \quad (39a)$$

$$\begin{aligned} \mathbb{L}_m({}^{n+1}\mathbf{G}_c) &= {}^{n+1}P_R^{-1} (\mathbb{L}_m({}^{n+1}\mathbf{G}_R) \cap \mathbb{L}_m({}^{n+1}\mathbf{S}_R)) \cap {}^{n+1}P_B^{-1} (\mathbb{L}_m({}^n\mathbf{G})) \cap {}^{n+1}\tilde{P}_B^{-1} (\mathbb{L}_m({}^{n+1}\mathbf{S}_B)) \cap \\ &\quad \bigcap_{j=1}^m \left({}^{n+1,j}P^{-1} (\mathbb{L}_m({}^{n+1,j}\mathbf{G})) \cap {}^{n+1,j}\tilde{P}^{-1} (\mathbb{L}_m({}^{n+1,j}\mathbf{S})) \right) \cap {}^{n+1,2}P_D^{-1} (\mathbb{L}_m({}^{n+1,2}\mathbf{S}_D)), \end{aligned} \quad (39b)$$

where ${}^{n+1}P_R$ is the projection of ${}^{n+1}\tilde{\mathbb{E}}_R^*$ to ${}^{n+1}\mathbb{E}_R^*$; ${}^{n+1,j}P$ and ${}^{n+1,j}\tilde{P}$ are the projections of ${}^{n+1}\tilde{\mathbb{E}}_R^*$ to ${}^{n+1,j}\mathbb{E}^*$ and ${}^{n+1,j}\tilde{\mathbb{E}}^*$ ($\forall j \in \mathbb{J}(i)$), respectively; ${}^{n+1}P_B$ and ${}^{n+1}\tilde{P}_B$ are the projections of ${}^{n+1}\tilde{\mathbb{E}}_R^*$ to ${}^{n+1}\mathbb{E}^*$ and ${}^{n+1}\tilde{\mathbb{E}}^*$, respectively; and ${}^{n+1,2}P_D$ is the projection of ${}^{n+1}\tilde{\mathbb{E}}_R^*$ to ${}^{n+1,2}\tilde{\mathbb{E}}_D^*$.

In accordance with [7,16], the present supervisors are designed in a way that all supervisors' states are marked. Thus, the closed and the marked behaviors of the supervisors for $i \in \{1, \dots, n+1\}$ and $j \in \mathbb{J}(i)$ are required to be

$$\mathbb{L}(^{i,j}\mathbf{S}) = \mathbb{L}_m(^{i,j}\mathbf{S}) = \overline{^{i,j}\mathbb{K}} = ^{i,j}\mathbb{K} \quad (40a)$$

$$\mathbb{L}(^i\mathbf{S}) = \mathbb{L}_m(^i\mathbf{S}) = \overline{^i\mathbb{K}} = ^i\mathbb{K} \quad (40b)$$

$$\mathbb{L}(^i\mathbf{S}_B) = \mathbb{L}_m(^i\mathbf{S}_B) = \overline{^i\mathbb{K}_B} = ^i\mathbb{K}_B \quad (40c)$$

$$\mathbb{L}(^{i,1}\mathbf{S}_D) = \mathbb{L}_m(^{i,1}\mathbf{S}_D) = \overline{^{i,1}\mathbb{K}_D} = ^{i,1}\mathbb{K}_D \quad (40d)$$

$$\mathbb{L}(^{i,2}\mathbf{S}_D) = \mathbb{L}_m(^{i,2}\mathbf{S}_D) = \overline{^{i,2}\mathbb{K}_D} = ^{i,2}\mathbb{K}_D \quad (40e)$$

$$\mathbb{L}(^i\mathbf{S}_R) = \mathbb{L}_m(^i\mathbf{S}_R) = \overline{^i\mathbb{K}_R} \quad (40f)$$

According to Equation (40a–f), the closed and marked behaviors of the respective supervisors are designed to be equal to the prefix closure of the desired languages. Furthermore, regarding Equation (40a–e), it is observed that since the languages $^{i,j}\mathbb{K}$, $^i\mathbb{K}$, $^i\mathbb{K}_B$, $^{i,1}\mathbb{K}_D$, and $^{i,2}\mathbb{K}_D$ are prefix-closed, then the equality between the closed and marked behaviors of the respective supervisors is derived. Here, it is important to recall that, according to Proposition 1, the language $^i\mathbb{K}_R$ is $\mathbb{L}_m(^i\mathbf{G}_R)$ -closed. For the definition and properties of prefixed-closed languages, see [15,16].

In addition to the above design requirements, the marked behavior of the controlled automaton $^i\mathbf{G}_c$, for $i \in \{1, \dots, n\}$, is required to be

$$\mathbb{L}_m(^i\mathbf{G}_c) = \bigcap_{j=1}^2 \left(^{i,j}P^{-1}(^{i,j}\mathbb{K}) \cap ^{i,j}P_D^{-1}(^{i,j}\mathbb{K}_D) \right) \cap ^iP_R^{-1}(^i\mathbb{K}_R) \cap ^iP^{-1}(^i\mathbb{K}) \cap ^iP_B^{-1}(^i\mathbb{K}_B) \quad (41)$$

For $i = n+1$, the marked behavior of the controlled automaton $^{n+1}\mathbf{G}_c$ is required to be

$$\mathbb{L}_m(^{n+1}\mathbf{G}_c) = \bigcap_{j=1}^m \left(^{n+1,j}P^{-1}(^{n+1,j}\mathbb{K}) \right) \cap ^{n+1}P_R^{-1}(^{n+1}\mathbb{K}_R) \cap ^{n+1}P_B^{-1}(^{n+1}\mathbb{K}_B) \cap ^{n+1,2}P_D^{-1}(^{n+1,2}\mathbb{K}_D) \quad (42)$$

In addition, for all $i \in \{1, \dots, n+1\}$, the marked behaviors of the controlled automata are required to satisfy the nonblocking requirement, i.e.,

$$\mathbb{L}(^i\mathbf{G}_c) = \overline{\mathbb{L}_m(^i\mathbf{G}_c)} \quad (43)$$

Finally, it is required for the supervisors embedded in U_i , where $i \in \{1, \dots, n+1\}$ and $j \in \mathbb{J}(i)$, to be physical realizable (PR) (see [17]) with respect to the automata $^i\mathbf{G}_R$, $^{i,j}\mathbf{G}$, $^i\mathbf{G}$, and $^{i-1}\mathbf{G}$ through the synchronous products in (36) and (38).

The satisfaction of the design requirements (36), (38) and (40)–(43) will be accomplished through the distributed control architecture proposed here, providing separate development and implementation of the control algorithm of each control unit U_i . Furthermore, the control algorithm of each U_i is analyzed for the conjunction of several supervisors providing modularity at the local control layer. The distributed-control architecture, designed for the present wafer system and being in accordance with the definition of distributed architectures presented in [7], is analyzed for a set of local controllers, i.e., one controller per robotic manipulator and the respective served production stations. Each local controller is fed by the events of the respective controlling subsystem and a set of distinct events coming from the adjacent subsystems through a “transfer line” (shared resources). It is mentioned that in [3,4], a distributed control architecture is proposed for the case where

$n = 3$ and $m = 3$. To the best of our knowledge the presented control architecture is the first distributed control architecture for the general parametric model of wafer manufacturing systems.

In Figure 10, a simplified drawing of the configuration of the distributed control architecture is presented. In each control unit, the respective supervisors to be implemented and the respective participating alphabets are depicted. Clearly, each of the supervisors $^{n+1}\mathbf{S}$, $^1\mathbf{S}_B$, $^{n+1,1}\mathbf{S}_D$, and $^{1,2}\mathbf{S}_D$, introduced for uniformity purposes, can be realized by an automaton with one state being marked and self-transitions for all the events of the corresponding alphabets. Nevertheless, the implementation of these supervisors is not necessary as they do not impose any additional desired performance. However, in the analysis in Section 6, they will be used for uniformity reasons.

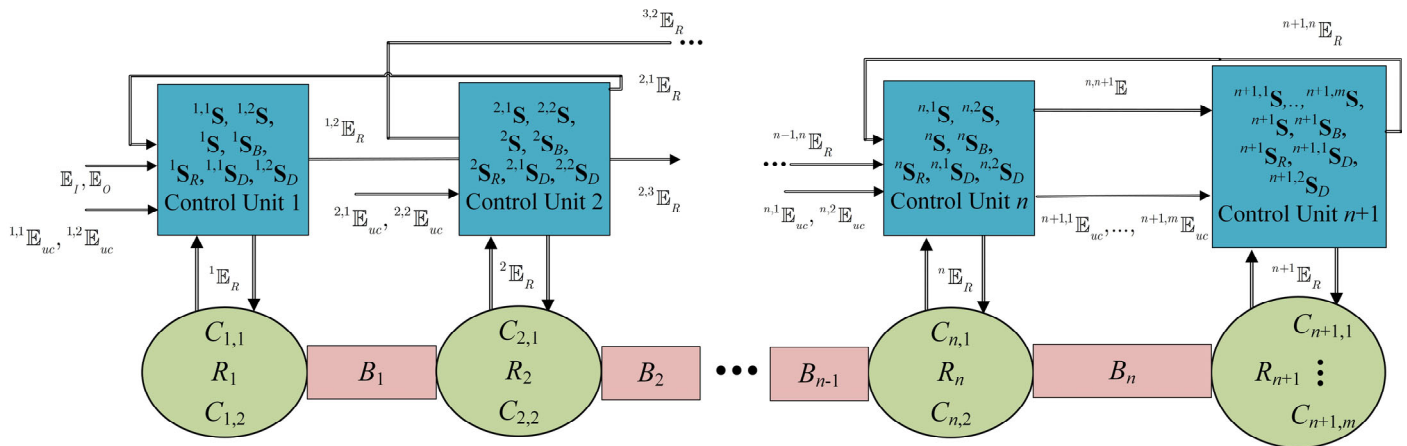


Figure 10. Distributed supervisory control architecture.

As already mentioned, in the special case where $n = 4$ and $m = 3$, the resulting model of the system is similar to the one in [7]. In Section 3, differences and similarities of the proposed desired behavior, in comparison to the one in [7], have been presented. The main difference is imposed by the combination of the second, fourth, and fifth specifications, regarding blocking avoidance. In the present paper, a more restrictive rule is proposed. Particularly, if a buffer or a manufacturing station (in any flow direction) is full, then the respective robotic manipulator is not allowed to pick a product from the respective station.

The present method can be effective in cases of station faults, i.e., if a fault takes place in a station, then the robotic manipulator is not allowed to pick or place products in this station. This way, the respective buffers are kept available to receive products from the reverse product flow.

The realization of the supervisors, depicted in Figure 10 and satisfying the design requirements (40a–f), will be determined in Section 5. In Section 6, the design requirements (40)–(43) and the physical realizability of the distributed supervisor design scheme will be proven to be satisfied.

5. Supervisor Realization

5.1. Supervisors of the Production Stations

The supervisors realizing the languages $^{i,1}\mathbb{K}$, where $i \in \{1, \dots, n\}$, are in a 6-tuple form (see [17–21]), as follows:

$$^{i,1}\mathbf{S} = (^{i,1}\mathbb{Q}_S, ^{i,1}\widetilde{\mathbb{E}}, ^{i,1}f_S, ^{i,1}\mathbb{H}_S, ^{i,1}x_{S,0}, ^{i,1}\mathbb{Q}_{S,m}) \quad (44)$$

The set of the states of the supervisor automaton is ${}^{i,1}\mathbb{Q}_S = \{{}^{i,1}q_{S,1}, {}^{i,1}q_{S,2}, {}^{i,1}q_{S,3}\}$. The initial state is ${}^{i,1}x_{S,0} = {}^{i,1}q_{S,1}$. The set of the marked states is ${}^{i,1}\mathbb{Q}_{S,m} = {}^{i,1}\mathbb{Q}_S$. The values of the transition functions of the supervisor automaton are

$$\begin{aligned} {}^{i,1}f_S({}^{i,1}q_{S,1}, {}^{i,i-1}e_{BP}) &= {}^{i,1}q_{S,1}, {}^{i,1}f_S({}^{i,1}q_{S,1}, {}^{i,1}e_u) = {}^{i,1}q_{S,1}, {}^{i,1}f_S({}^{i,1}q_{S,1}, {}^{i,1}e_D) = {}^{i,1}q_{S,2}, \\ {}^{i,1}f_S({}^{i,1}q_{S,2}, {}^{i,1}e_u) &= {}^{i,1}q_{S,3}, {}^{i,1}f_S({}^{i,1}q_{S,3}, {}^{i,1}e_P) = {}^{i,1}q_{S,1} \end{aligned}$$

The active events sets are

$$\begin{aligned} {}^{i,1}\mathbb{H}_S({}^{i,1}q_{S,1}) &= \{{}^{i,i-1}e_P, {}^{i,1}e_D, {}^{i,1}e_u\}, {}^{i,1}\mathbb{H}_S({}^{i,1}q_{S,2}) = \{{}^{i,1}e_u\}, \\ {}^{i,1}\mathbb{H}_S({}^{i,1}q_{S,3}) &= \{{}^{i,1}e_P, {}^{i,1}e_u\} \end{aligned}$$

The closed behavior and the marked behavior of the automaton satisfy the requirements (40a). In Figure 11, the state diagram of the supervisor is presented.

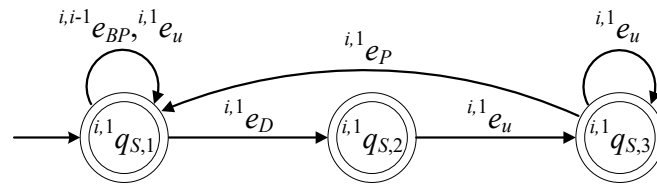


Figure 11. State diagram of supervisor ${}^{i,1}\mathbf{S}$.

The supervisor ${}^{1,1}\mathbf{S}$ realizing the language ${}^{1,1}\mathbb{K}$, the supervisor ${}^{i,2}\mathbf{S}$ realizing the language ${}^{i,2}\mathbb{K}$, where $i \in \{1, \dots, n\}$, and the supervisor ${}^{n+1,j}\mathbf{S}$ realizing ${}^{n+1,j}\mathbb{K}$, where $j \in \{1, \dots, m\}$, have the same state diagram configuration as that in Figure 11.

The supervisor automaton realizing ${}^{i,1}\mathbb{K}_D$, where $i \in \{1, \dots, n\}$, is the 6-tuple

$${}^{i,1}\mathbf{S}_D = ({}^{i,1}\mathbb{Q}_{D,S}, {}^{i,1}\mathbb{E}_D, {}^{i,1}f_{D,S}, {}^{i,1}\mathbb{H}_{D,S}, {}^{i,1}x_{D,S,0}, {}^{i,1}\mathbb{Q}_{D,S,m}) \quad (45)$$

The set of the states of the supervisor automaton is ${}^{i,1}\mathbb{Q}_{D,S} = \{{}^{i,1}q_{D,S,1}, {}^{i,1}q_{D,S,2}\}$. The initial state is ${}^{i,1}x_{D,S,0} = {}^{i,1}q_{D,S,1}$. The set of the marked states is ${}^{i,1}\mathbb{Q}_{D,S,m} = {}^{i,1}\mathbb{Q}_{D,S}$. The values of the transition function of the supervisor automaton are

$${}^{i,1}f_{D,S}({}^{i,1}q_{D,S,1}, {}^{i+1,1}e_P) = {}^{i,1}q_{D,S,1}, {}^{i,1}f_{D,S}({}^{i,1}q_{D,S,1}, {}^{i,1}e_P) = {}^{i,1}q_{D,S,2}, {}^{i,1}f_{D,S}({}^{i,1}q_{D,S,2}, {}^{i+1,1}e_P) = {}^{i,1}q_{D,S,1}$$

The active events sets are

$${}^{i,1}\mathbb{H}_{D,S}({}^{i,1}q_{D,S,1}) = \{{}^{i,1}e_P, {}^{i+1,1}e_P\}, {}^{i,1}\mathbb{H}_{D,S}({}^{i,1}q_{D,S,2}) = \{{}^{i+1,1}e_P\}$$

The closed behavior and the marked behavior of the automaton satisfy the requirements (40d). In Figure 12, the state diagram of the supervisor is presented.

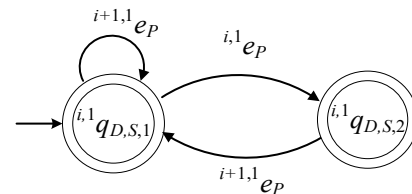


Figure 12. State diagram of supervisor ${}^{i,1}\mathbf{S}_D$.

The supervisor ${}^{i,2}\mathbf{S}_D$ of the language ${}^{i,2}\mathbb{K}_D$, where $i \in \{2, \dots, n+1\}$, has the same state diagram configuration as that in Figure 12. It is mentioned that a realization of the supervisors ${}^{n+1,1}\mathbf{S}_D$ and ${}^{1,2}\mathbf{S}_D$ has already been proposed in Section 4.

5.2. Supervisors of the Buffers

The supervisors for the buffers, namely the supervisors realizing the language ${}^i\mathbb{K}$, where $i \in \{1, \dots, n\}$, are in the form

$${}^i\mathbf{S} = ({}^i\mathbb{Q}_S, {}^i\widetilde{\mathbb{E}}, {}^if_S, {}^i\mathbb{H}_S, {}^ix_{S,0}, {}^i\mathbb{Q}_{S,m}) \quad (46)$$

The set of the states is ${}^i\mathbb{Q}_S = \{{}^iq_{S,1}, {}^iq_{S,2}, {}^iq_{S,3}, {}^iq_{S,4}, {}^iq_{S,5}\}$. The initial state is ${}^ix_{S,0} = {}^iq_{S,1}$. The set of the marked states is ${}^i\mathbb{Q}_{S,m} = {}^i\mathbb{Q}_S$. The values of the transition functions are

$$\begin{aligned} {}^if_S({}^iq_{S,1}, {}^{i+1,i}e_{BP}) &= {}^iq_{S,1}, {}^if_S({}^iq_{S,1}, {}^{i+1,i}e_{BD}) = {}^iq_{S,1}, {}^if_S({}^iq_{S,1}, {}^{i+1,i}e_P) = {}^iq_{S,2}, \\ {}^if_S({}^iq_{S,1}, {}^{i+1,2}e_P) &= {}^iq_{S,4}, {}^if_S({}^iq_{S,2}, {}^{i+1,i}e_{BP}) = {}^iq_{S,2}, {}^if_S({}^iq_{S,2}, {}^{i+1,i}e_{BD}) = {}^iq_{S,2}, \\ {}^if_S({}^iq_{S,2}, {}^{i+1,2}e_P) &= {}^iq_{S,2}, {}^if_S({}^iq_{S,2}, {}^{i,i}e_{BD}) = {}^iq_{S,3}, {}^if_S({}^iq_{S,3}, {}^{i+1,i}e_{BP}) = {}^iq_{S,3}, \\ {}^if_S({}^iq_{S,3}, {}^{i+1,2}e_P) &= {}^iq_{S,3}, {}^if_S({}^iq_{S,3}, {}^{i+1,i}e_{BD}) = {}^iq_{S,1}, {}^if_S({}^iq_{S,4}, {}^{i+1,i}e_{BP}) = {}^iq_{S,4}, \\ {}^if_S({}^iq_{S,4}, {}^{i+1,2}e_P) &= {}^iq_{S,4}, {}^if_S({}^iq_{S,4}, {}^{i+1,i}e_{BD}) = {}^iq_{S,5}, {}^if_S({}^iq_{S,5}, {}^{i+1,i}e_{BP}) = {}^iq_{S,5}, \\ {}^if_S({}^iq_{S,5}, {}^{i+1,i}e_{BD}) &= {}^iq_{S,5}, {}^if_S({}^iq_{S,5}, {}^{i+1,2}e_P) = {}^iq_{S,5}, {}^if_S({}^iq_{S,5}, {}^{i,i}e_{BP}) = {}^iq_{S,1} \end{aligned}$$

The active events sets are

$$\begin{aligned} {}^i\mathbb{H}_S({}^iq_{S,1}) &= \{{}^{i,1}e_P, {}^{i+1,2}e_P, {}^{i+1,i}e_{BP}, {}^{i+1,i}e_{BD}\}, {}^i\mathbb{H}_S({}^iq_{S,2}) = \{{}^{i,i}e_{BD}, {}^{i+1,2}e_P, {}^{i+1,i}e_{BP}, {}^{i+1,i}e_{BD}\} \\ {}^i\mathbb{H}_S({}^iq_{S,3}) &= \{{}^{i+1,2}e_P, {}^{i+1,i}e_{BP}, {}^{i+1,i}e_{BD}\}, {}^i\mathbb{H}_S({}^iq_{S,4}) = \{{}^{i+1,2}e_P, {}^{i+1,i}e_{BP}, {}^{i+1,i}e_{BD}\}, \\ {}^i\mathbb{H}_S({}^iq_{S,5}) &= \{{}^{i,i}e_{BP}, {}^{i+1,2}e_P, {}^{i+1,i}e_{BP}, {}^{i+1,i}e_{BD}\} \end{aligned}$$

The closed behavior and the marked behavior of the automaton satisfy the requirements (40b). In Figure 13, the state diagram of the supervisor is presented.

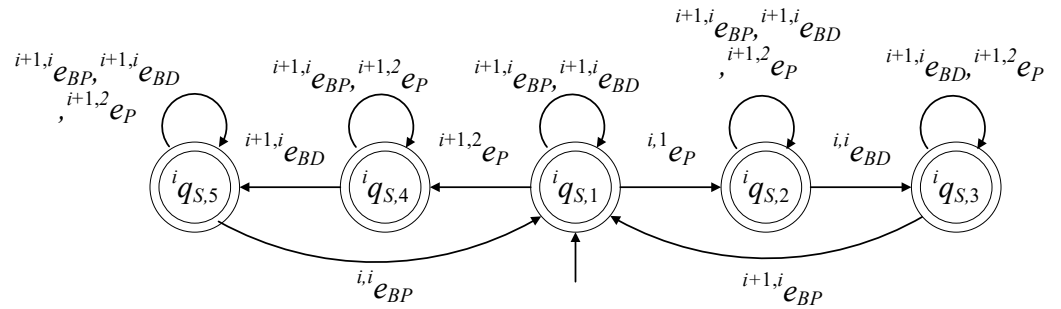


Figure 13. State diagram of supervisor ${}^i\mathbf{S}$.

The state diagram of the supervisor ${}^i\mathbf{S}_B$, where $i \in \{2, \dots, n+1\}$, has the same configuration as the state diagram of ${}^i\mathbf{S}$, where $i \in \{1, \dots, n\}$. The closed behavior and the marked behavior of the automaton satisfy the requirements (6c). In Figure 14, the state diagram of the supervisor ${}^i\mathbf{S}_B$, where $i \in \{2, \dots, n+1\}$, is presented.

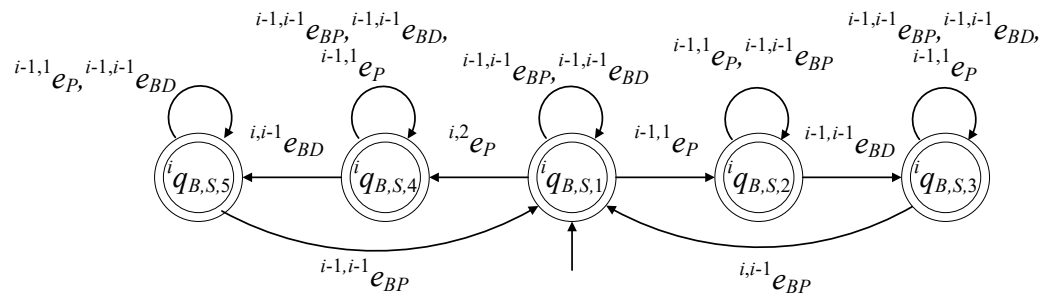


Figure 14. State diagram of supervisor ${}^i\mathbf{S}_B$.

It is mentioned that the supervisors $^{n+1}\mathbf{S}$ and $^1\mathbf{S}_B$, mentioned in Section 4 to be supervisors realizing the languages $^{n+1}\mathbb{K}$ and $^1\mathbb{K}_B$, respectively, are single-state automata, with self-transitions activated by all events of the alphabet $^{n+1}\mathbb{E}_R$ and the alphabet $^1\mathbb{E}_R$, respectively. It is noted that the single states of the supervisor automata are marked.

5.3. Supervisors of the Robotic Manipulators

The supervisor of the robotic manipulator R_i , namely the supervisor realizing the language $^i\mathbb{K}_R$, where $i \in \{2, \dots, n\}$, is in the form

$$^i\mathbf{S}_R = (^i\mathbb{Q}_{R,S}, ^i\mathbb{E}_R, ^if_{R,S}, ^i\mathbb{H}_{R,S}, ^ix_{R,S,0}, ^i\mathbb{Q}_{R,S,m}) \quad (47)$$

The set of the states is $^i\mathbb{Q}_{R,S} = \{^iq_{R,S,1}, ^iq_{R,S,2}, ^iq_{R,S,3}, ^iq_{R,S,4}, ^iq_{R,S,5}\}$. The initial state is $^ix_{R,S,0} = ^iq_{R,S,1}$. The set of the marked states is $^i\mathbb{Q}_{R,S,m} = ^i\mathbb{Q}_{R,S}$. The values of the transition function of the supervisor automaton are

$$\begin{aligned} ^if_{R,S}(^iq_{R,S,1}, ^{i,i-1}e_{BP}) &= ^iq_{R,S,2}, \quad ^if_{R,S}(^iq_{R,S,1}, ^{i,1}e_P) = ^iq_{R,S,3} \\ ^if_{R,S}(^iq_{R,S,1}, ^{i,i}e_{BP}) &= ^iq_{R,S,4}, \quad ^if_{R,S}(^iq_{R,S,1}, ^{i,2}e_P) = ^iq_{R,S,5}, \quad ^if_{R,S}(^iq_{R,S,2}, ^{i,1}e_D) = ^iq_{R,S,1}, \\ ^if_{R,S}(^iq_{R,S,3}, ^{i,i}e_{BD}) &= ^iq_{R,S,1}, \quad ^if_{R,S}(^iq_{R,S,4}, ^{i,2}e_D) = ^iq_{R,S,1}, \\ ^if_{R,S}(^iq_{R,S,5}, ^{i,i-1}e_{BD}) &= ^iq_{R,S,1} \end{aligned}$$

The active events sets are

$$\begin{aligned} ^i\mathbb{H}_{R,S}(^iq_{R,S,1}) &= \{^{i,i-1}e_{BP}, ^{i,1}e_P, ^{i,i}e_{BP}, ^{i,2}e_P\}, \quad ^i\mathbb{H}_{R,S}(^iq_{R,S,2}) = \{^{i,1}e_D\}, \\ ^i\mathbb{H}_{R,S}(^iq_{R,S,3}) &= \{^{i,i}e_{BD}\}, \quad ^i\mathbb{H}_{R,S}(^iq_{R,S,4}) = \{^{i,2}e_D\}, \quad ^i\mathbb{H}_{R,S}(^iq_{R,S,5}) = \{^{i,i-1}e_{BD}\} \end{aligned}$$

The closed behavior and the marked behavior of the automaton satisfy the requirements (40f). In Figure 15, the state diagram of the supervisor is presented.

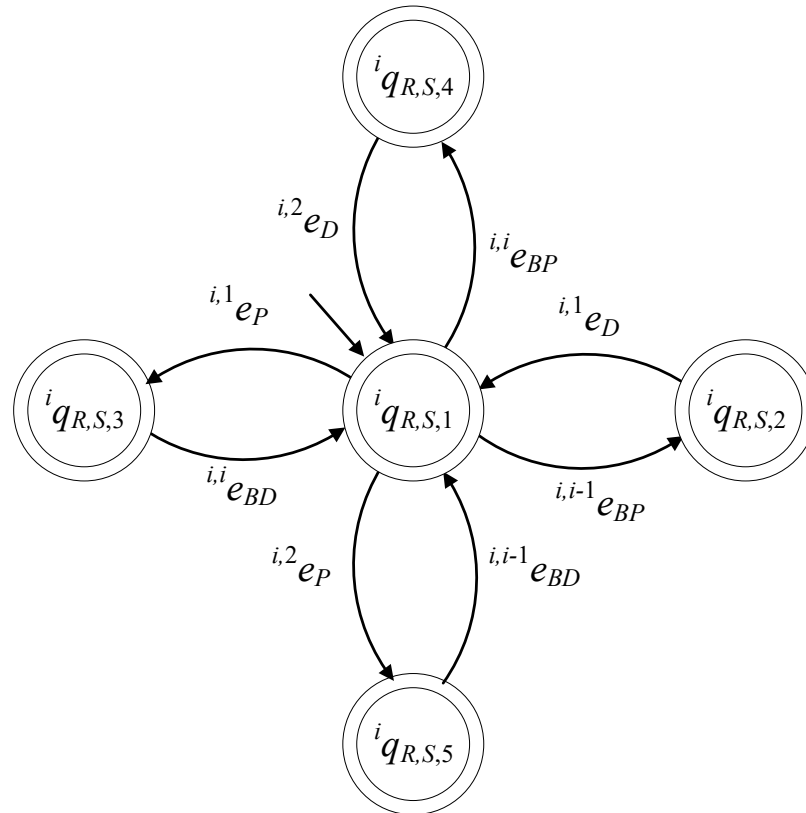


Figure 15. State diagram of robotic manipulator $^i\mathbf{S}_R$.

The supervisor ${}^1S_R$ of the language ${}^1K_R$ has the same state diagram configuration as that in Figure 15. In Figure 16, the state diagram of the supervisor ${}^1S_R$ is presented.

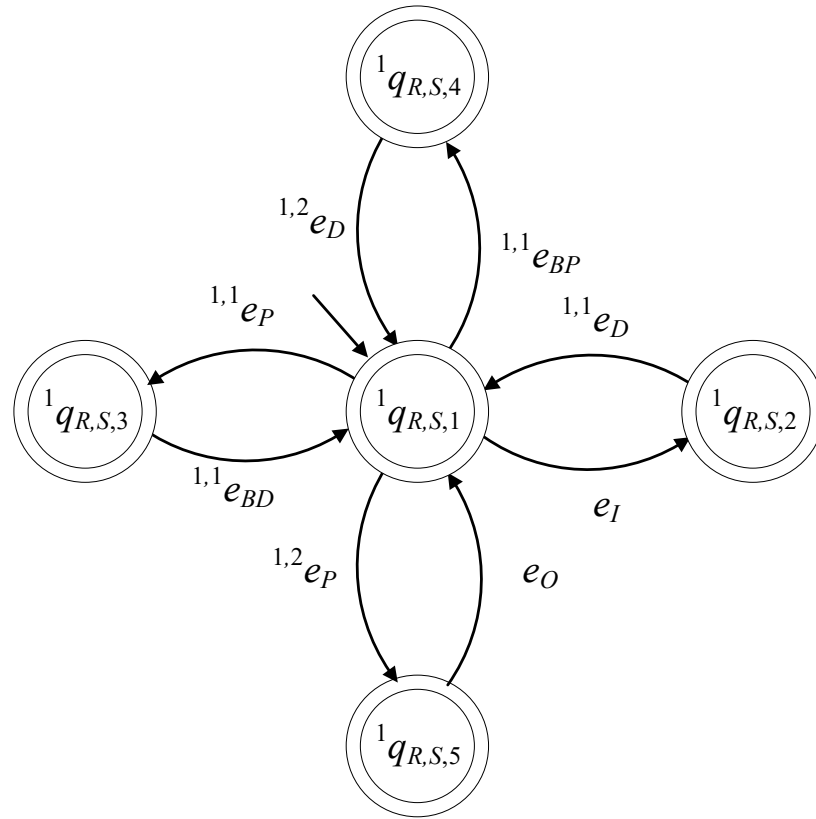


Figure 16. State diagram of robotic manipulator ${}^1S_R$.

The supervisor of the robotic manipulator R_{n+1} , namely the supervisor realizing the language ${}^{n+1}K_{R,D}$, is in the form

$${}^{n+1}S_R = ({}^{n+1}Q_{R,S}, {}^{n+1}E_R, {}^{n+1}f_{R,S}, {}^{n+1}H_{R,S}, {}^{n+1}x_{R,S,0}, {}^{n+1}Q_{R,S,m}) \quad (48)$$

The set of the states is ${}^{n+1}Q_{R,S} = \{{}^{n+1}q_{R,S,1}, {}^{n+1}q_{R,S,2}, \dots, {}^{n+1}q_{R,S,m+2}\}$. The initial state is ${}^{n+1}x_{R,S,0} = {}^{n+1}q_{R,S,1}$. The set of the marked states is ${}^{n+1}Q_{R,S,m} = {}^{n+1}Q_{R,S}$. The values of the transition functions of the supervisor automaton are

$$\begin{aligned} {}^{n+1}f_{R,S}({}^{n+1}q_{R,S,1}, {}^{n+1,n}e_{BP}) &= {}^{n+1}q_{R,S,2}, \quad {}^{n+1}f_{R,S}({}^{n+1}q_{R,S,1}, {}^{n+1,1}e_P) = {}^{n+1}q_{R,S,3}, \\ {}^{n+1}f_{R,S}({}^{n+1}q_{R,S,1}, {}^{n+1,j}e_P) &= {}^{n+1}q_{R,S,j+1}, \quad \forall j \in \{3, 4, \dots, m\} \\ {}^{n+1}f_{R,S}({}^{n+1}q_{R,S,1}, {}^{n+1,2}e_P) &= {}^{n+1}q_{R,S,m+2}, \quad {}^{n+1}f_{R,S}({}^{n+1}q_{R,S,2}, {}^{n+1,1}e_D) = {}^{n+1}q_{R,S,1}, \\ {}^{n+1}f_{R,S}({}^{n+1}q_{R,S,j}, {}^{n+1,j}e_D) &= {}^{n+1}q_{R,S,1}, \quad \forall j \in \{3, 4, \dots, m\}, \quad {}^{n+1}f_{R,S}({}^{n+1}q_{R,S,m+1}, {}^{n+1,2}e_D) = {}^{n+1}q_{R,S,1}, \\ {}^{n+1}f_{R,S}({}^{n+1}q_{R,S,m+2}, {}^{n+1,n}e_{BD}) &= {}^{n+1}q_{R,S,1} \end{aligned}$$

The active events sets are

$$\begin{aligned} {}^{n+1}H_{R,S}({}^{n+1}q_{R,S,1}) &= \{{}^{n+1,n}e_{BP}, {}^{n+1,1}e_P, \dots, {}^{n+1,m}e_P\} \\ {}^{n+1}H_{R,S}({}^{n+1}q_{R,S,2}) &= \{{}^{n+1,1}e_D\} \quad {}^{n+1}H_{R,S}({}^{n+1}q_{R,S,j}) = \{{}^{n+1,j}e_D\}, \quad \forall j \in \{3, 4, \dots, m\}, \\ {}^{n+1}H_{R,S}({}^{n+1}q_{R,S,m+1}) &= \{{}^{n+1,2}e_D\} \quad {}^{n+1}H_{R,S}({}^{n+1}q_{R,S,m+2}) = \{{}^{n+1,n}e_{BD}\} \end{aligned}$$

The closed behavior and the marked behavior of the automaton satisfy the requirements (40f). In Figure 17, the state diagram of the supervisor is presented.

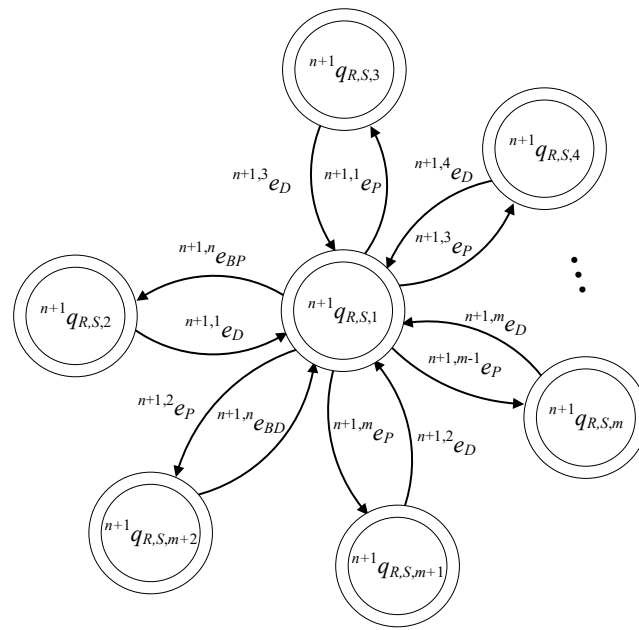


Figure 17. State diagram of robotic manipulator $^{n+1}\mathbf{S}_R$.

6. Establishment of the Satisfactory Performance of the Controlled Automaton

In the present section, two lemmas, five theorems and two remarks regarding the implementability of the control architecture proposed in Section 4 will be established using the supervisor realizations derived in Section 5. In particular, the controllability of the desired languages will be proven through the proof of the physical realizability (see [17]) of the derived supervisors. Furthermore, the nonblocking property of the resulting controlled automaton will be proven. Finally, using these theorems and remarks, the satisfactory performance of the controlled automaton will be guaranteed. The satisfactory transition sequence of the controlled automaton will be determined in terms of the closed and the marked behavior of the controlled automaton. According to the distributed control architecture, proposed in Section 4, the supervisors embedded in U_i , where $i \in \{1, \dots, n+1\}$, are designed in order to satisfy the physical realizability of the supervisors with respect to the wafer manufacturing process, the nonblocking property of the resulting controlled automaton, and the satisfactory closed and marked behaviors of the controlled automaton.

Regarding the physical realizability, the investigation is limited to the physical realizability of the supervisors of U_i with respect to the robotic manipulator R_i , the production station $C_{i,j}$, where $j \in \mathbb{J}(i)$, and the buffer B_i , where $i \in \{1, \dots, n\}$, through the synchronous products in (36) and (38). The physical realizability of the supervisors of U_1 with respect to the loading docks is not investigated as all events in their alphabets are controllable.

Regarding nonblocking, the investigation will start with the nonblocking property of the automata in the synchronous products (36) and (38). Then the nonblocking property of the overall system will be investigated. Regarding the loading docks, it is observed that they cannot cause blocking to the controlled automaton, as they have only one state, being marked, and a single event self-transition.

As already mentioned, in the present section, two lemmas and five theorems will be presented. In the two lemmas, the exclusion of non-marked states in appropriate synchronous products, due to non-accessibility, will be proven. Regarding non-accessible states, see [16]. In the first four theorems of the present section, the issues of physical realizability and nonblocking will be investigated with respect to the influence of the supervisors of U_i to the devices controlled by the respective control unit. Finally, in the fifth theorem, the nonblocking property of the overall controlled wafer manufacturing system is investigated through the synchronous product of the participating controlled automata.

In the first theorem, the physical realizability of the supervisors, embedded to U_i , where $i = \{2, \dots, n\}$, with respect to the automata of the production stations, the robot, and the buffer affected by the events of the control unit U_i , through the synchronous product (36) will be proven. All automata affected by the control unit U_i are composed as the synchronous product ${}^iG_R || {}^iG || {}^{i-1}G || {}^{i,1}G || {}^{i,2}G$. In the second theorem, the physical realizability of the supervisors, embedded to U_{n+1} , with respect to the automaton ${}^{n+1}G_R || {}^nG || \prod_{j=1}^m {}^{n+1,j}G$, through the synchronous product (38) will be proven. Furthermore, the first of the two remarks of the section will cover the physical realizability of the supervisors, embedded to U_1 .

Theorem 1. *The synchronous product in (36), where $i = \{2, \dots, n\}$, is physical realizable (PR), with respect to ${}^iG_R || {}^iG || {}^{i-1}G || {}^{i,1}G || {}^{i,2}G$.*

Proof of Theorem 1. Regarding the supervisors embedded in the control unit U_i , the several events of the automaton ${}^iG_R || {}^iG || {}^{i-1}G || {}^{i,1}G || {}^{i,2}G$ are not allowed to be restricted by the supervisors. These events, being uncontrollable by U_i , are the uncontrollable events of the two production stations, i.e., the events ${}^{i,1}e_u$ and ${}^{i,2}e_u$, and the monitoring events of the adjacent control units, i.e., the events ${}^{i+1,i}e_{BP}$, ${}^{i+1,2}e_P$, ${}^{i+1,i}e_{BD}$, ${}^{i-1,1}e_P$, ${}^{i-1,i-1}e_{BD}$, ${}^{i-1,i-1}e_{BP}$, ${}^{i+1,1}e_P$, and ${}^{i-1,2}e_P$. Here, it will be investigated if each supervisor of U_i is able to restrict these 10 events. To this end, first it is observed that all events in the alphabet of ${}^iS_R$ are controllable. Thus, ${}^iS_R$ does not restrict any of the 10 events. Second, it is observed that the uncontrollable events in the alphabet of ${}^iS$, being the events ${}^{i+1,i}e_{BP}$, ${}^{i+1,2}e_P$, and ${}^{i+1,i}e_{BD}$, satisfy the following properties:

$${}^{i+1,i}e_{BP} \in \cap_{\lambda=1}^5 \left({}^{i\mathbb{H}_S}(q_{S,\lambda}) \right), {}^{i+1,2}e_P \in \cap_{\lambda=1}^5 \left({}^{i\mathbb{H}_S}(q_{S,\lambda}) \right), {}^{i+1,i}e_{BD} \in \cap_{\lambda=1}^3 \left({}^{i\mathbb{H}_S}(q_{S,\lambda}) \right)$$

The above relations reveal that the three uncontrollable events of ${}^iS$ belong to the active even sets of all states of ${}^iS$. Thus, ${}^iS$ does not restrict any of the 10 events. Third, it is observed that the uncontrollable events in the alphabet of ${}^iS_B$, being the events ${}^{i-1,1}e_P$, ${}^{i-1,i-1}e_{BD}$, and ${}^{i-1,i-1}e_{BP}$, satisfy the following properties

$${}^{i-1,1}e_P \in \cap_{\lambda=1}^5 \left({}^{i\mathbb{H}_{B,S}}(q_{B,S,\lambda}) \right), {}^{i-1,i-1}e_{BD} \in \cap_{\lambda=1}^5 \left({}^{i\mathbb{H}_{B,S}}(q_{B,S,\lambda}) \right), {}^{i-1,i-1}e_{BP} \in \cap_{\lambda=1}^5 \left({}^{i\mathbb{H}_{B,S}}(q_{B,S,\lambda}) \right)$$

The above relations reveal that the three uncontrollable events of ${}^iS_B$ belong to the active even sets of all states of ${}^iS_B$. Thus, ${}^iS_B$ does not restrict any of the 10 events.

Fourth, it is observed that there is only one uncontrollable event in the alphabet of ${}^{i,1}S_D$, being the event ${}^{i+1,1}e_P$, where ${}^{i+1,1}e_P \in \cap_{\lambda=1}^2 \left({}^{i,1\mathbb{H}_{D,S}}(q_{D,S,\lambda}) \right)$. The above relation reveals that the uncontrollable event of ${}^{i,1}S_D$ belongs to the active even sets of all states of ${}^{i,1}S_D$. Thus, ${}^{i,1}S_D$ does not restrict any of the 10 events. Fifth, it is observed that there is only one uncontrollable event in the alphabet of ${}^{i,2}S_D$, being the event ${}^{i-1,2}e_P$ where ${}^{i-1,2}e_P \in \cap_{\lambda=1}^2 \left({}^{i,2\mathbb{H}_{D,S}}(q_{D,S,\lambda}) \right)$. The above relation reveals that the uncontrollable event of ${}^{i,2}S_D$ belongs to the active even sets of all states of ${}^{i,2}S_D$. Thus, ${}^{i,2}S_D$ does not restrict any of the 10 events. Sixth, it is observed that there is only one uncontrollable event in the alphabet of ${}^{i,j}S$, where $j \in \mathbb{J}(i)$, being the event ${}^{i,j}e_u$ where ${}^{i,j}e_u \in \cap_{\lambda=1}^3 \left({}^{i,j\mathbb{H}_S}(q_{S,\lambda}) \right)$. The above relation reveals that the uncontrollable event of ${}^{i,j}S$ belongs to the active even sets of all states of ${}^{i,j}S$. Thus, ${}^{i,j}S$ does not restrict any of the 10 events. Hence, according to Corollary 1 in [17], all supervisors are PR with respect to ${}^iG_R || {}^iG || {}^{i-1}G || {}^{i,1}G || {}^{i,2}G$, through (36). $\square$

Remark 1. *The proof of the physical realizability for U_1 is similar to the one of U_i , for $i \in \{2, \dots, n\}$, if the events of the loading docks are considered, i.e., if the events e_I and e_O are used instead of the events ${}^{i,i-1}e_{BP}$ and ${}^{i,i-1}e_{BD}$ in the proof of Theorem 1.*

Theorem 2. *The synchronous product in (38) is physically realizable (PR) with respect to ${}^{n+1}\mathbf{G}_R || {}^n\mathbf{G} \parallel \prod_{j=1}^m {}^{n+1,j}\mathbf{G}$.*

Proof of Theorem 2. The uncontrollable events of the automaton ${}^{n+1}\mathbf{G}_R || {}^n\mathbf{G} \parallel \prod_{j=1}^m {}^{n+1,j}\mathbf{G}$ are the m uncontrollable events of the last m production stations, i.e., the events ${}^{i,j}e_u$ where $j \in \{1, \dots, m\}$ and the monitoring events of the adjacent control units considered as uncontrollable events. The monitoring events are the events ${}^{n,1}e_P$, ${}^{n,2}e_P$, ${}^{n,n}e_{BD}$, and ${}^{n,n}e_{BP}$. Here, it will be investigated if each supervisor of U_{n+1} is able to restrict these $m + 4$ events. To this end, first, it is observed that all event of ${}^{n+1}\mathbf{S}_R$ are controllable. Thus, ${}^{n+1}\mathbf{S}_R$ does not restrict any of the $m + 4$ events. Second, it is observed that the uncontrollable events in the alphabet of ${}^{n+1}\mathbf{S}_B$, being the events ${}^{n,1}e_P$, ${}^{n,n}e_{BD}$ and ${}^{n,n}e_{BP}$, satisfy the following properties:

$${}^{n,1}e_P \in \cap_{\lambda=1}^5 \left({}^{n+1}\mathbb{H}_{B,S}(i q_{B,S,\lambda}) \right), \quad {}^{n,n}e_{BD} \in \cap_{\lambda=1}^5 \left({}^{n+1}\mathbb{H}_{B,S}(i q_{B,S,\lambda}) \right), \quad {}^{n,n}e_{BP} \in \cap_{\lambda=1}^5 \left({}^{n+1}\mathbb{H}_{B,S}(i q_{B,S,\lambda}) \right)$$

The above relations reveal that the three uncontrollable events of ${}^{n+1}\mathbf{S}_B$ belong to the active even sets of all states of ${}^{n+1}\mathbf{S}_B$. Thus, ${}^{n+1}\mathbf{S}_B$ does not restrict any of the $m + 4$ events. Third, it is observed that there is only one uncontrollable event in the alphabet of ${}^{n+1,2}\mathbf{S}_D$, being the event ${}^{n,2}e_P$ where ${}^{n,2}e_P \in \cap_{\lambda=1}^2 \left({}^{n+1,2}\mathbb{H}_{D,S}({}^{n+1,2}q_{D,S,\lambda}) \right)$. The above relation reveals that the uncontrollable event of ${}^{n+1,2}\mathbf{S}_D$ belongs to the active even sets of all states of ${}^{n+1,2}\mathbf{S}_D$. Thus, ${}^{n+1,2}\mathbf{S}_D$ does not restrict any of the $m + 4$ events. Fourth, it is observed that there is only one uncontrollable event in the alphabet of ${}^{n+1,j}\mathbf{S}$, being the event ${}^{n+1,j}e_u$ where ${}^{n+1,j}e_u \in \cap_{\lambda=1}^3 \left({}^{n+1,j}\mathbb{H}_S({}^{n+1,j}q_{S,\lambda}) \right)$. The above relation reveals that the uncontrollable event of ${}^{n+1,j}\mathbf{S}$ belongs to the active even sets of all states of ${}^{n+1,j}\mathbf{S}$. Thus, ${}^{n+1,j}\mathbf{S}$ does not restrict any of the $m + 4$ events. Hence, according to Corollary 1 in [17], all supervisors in (38) are PR with respect to ${}^{n+1}\mathbf{G}_R || {}^n\mathbf{G} \parallel \prod_{j=1}^m {}^{n+1,j}\mathbf{G}$, through (38).

Since the uncontrollable alphabets of all supervisors are disjoint sets, the conjunction of all supervisors is PR with respect to ${}^{n+1}\mathbf{G}_R || {}^n\mathbf{G} \parallel \prod_{j=1}^m {}^{n+1,j}\mathbf{G}$, through (38). $\square$

In the following two lemmas, a subautomaton of the synchronous product of the automaton of the production station and one of the proposed supervisors, as well as a subautomaton of the synchronous product of the automaton of the buffer and two of the proposed supervisors, are presented. All states of these two subautomata will be proven to be marked. Regarding subautomata, see [15,16].

Lemma 1. *All states of the automaton ${}^{i,j}\mathbf{G}_c = {}^{i,j}\mathbf{G} || {}^{i,j}\mathbf{S}$, for $i \in \{1, \dots, n+1\}$ and $j \in \mathbb{J}(i)$, are marked states.*

Proof of Lemma 1. For the automaton ${}^{i,j}\mathbf{G}_c$ it holds that

$$\begin{aligned} {}^{i,j}f_c(({}^{i,j}q_{S,1}, {}^{i,j}q_1), e) &= ({}^{i,j}q_{S,1}, {}^{i,j}q_1); e \in {}^{i,j}\widetilde{\mathbb{E}} - {}^{i,j}\mathbb{E}, \quad {}^{i,j}f_c(({}^{i,j}q_{S,1}, {}^{i,j}q_1), {}^{i,j}e_D) = ({}^{i,j}q_{S,2}, {}^{i,j}q_2), \\ {}^{i,j}f_c(({}^{i,j}q_{S,2}, {}^{i,j}q_2), {}^{i,j}e_u) &= ({}^{i,j}q_{S,3}, {}^{i,j}q_3), \quad {}^{i,j}f_c(({}^{i,j}q_{S,3}, {}^{i,j}q_3), {}^{i,j}e_P) = ({}^{i,j}q_{S,1}, {}^{i,j}q_1) \end{aligned}$$

where ${}^{i,j}f_c(\cdot, \cdot)$ is the transition function and ${}^{i,j}\mathbb{H}_c(\cdot)$ is the set of the active events of the automaton ${}^{i,j}\mathbf{G}_c$. From the above it is observed that any pair of states, where the first element is ${}^{i,j}q_4$ and the second is a state of ${}^{i,j}\mathbf{S}$, is not a state of ${}^{i,j}\mathbf{G}_c$, in the sense that this pair is non-accessible. The rest states of ${}^{i,j}\mathbf{G}$, i.e., the states ${}^{i,j}q_1$, ${}^{i,j}q_2$, and ${}^{i,j}q_3$, are all marked. Furthermore, the states of ${}^{i,j}\mathbf{S}$ are all marked. Thus, the proof has been completed. $\square$

Corollary 1. *The automaton ${}^i\mathbf{G}_c$ is a nonblocking automaton.*

Before presenting the following lemma, it is important to mention that the alphabets of the buffers contain events of the adjacent control units, controlling the buffer. Thus, the controlled automaton of the buffer is a result of the joint action of the adjacent control units.

Lemma 2. *All accessible states of the automaton ${}^i\mathbf{G}$, under the influence of the supervisors ${}^i\mathbf{S}$ and ${}^{i+1}\mathbf{S}_B$, for $i \in \{1, \dots, n\}$, are marked states.*

Proof of Lemma 2. First, the properties of the automaton ${}^i\mathbf{G}||{}^i\mathbf{S}$ will be examined. Regarding ${}^i\mathbf{G}$, it is observed that the arrival to the non-marked state ${}^iq_4$, having no transitions, can be accomplished only by transitions from ${}^iq_1$, ${}^iq_2$, and ${}^iq_3$. Regarding ${}^iq_1$, the respective transitions in ${}^i\mathbf{G}||{}^i\mathbf{S}$ are

$${}^if_{G||S}(({}^iq_1, {}^iq_{S,1}), {}^{i+1}e_{BP}) = ({}^iq_4, {}^iq_{S,1}), {}^if_{G||S}(({}^iq_1, {}^iq_{S,2}), {}^{i+1}e_{BP}) = ({}^iq_4, {}^iq_{S,2}), {}^if_{G||S}(({}^iq_1, {}^iq_{S,4}), {}^{i+1}e_{BP}) = ({}^iq_4, {}^iq_{S,4})$$

where ${}^if_{G||S}(\cdot, \cdot)$ is the transition function of the automaton ${}^i\mathbf{G}||{}^i\mathbf{S}$. Regarding ${}^iq_2$, the respective transition in ${}^i\mathbf{G}||{}^i\mathbf{S}$ is ${}^if_{G||S}(({}^iq_2, {}^iq_{S,3}), {}^{i+1}e_{BD}) = ({}^iq_4, {}^iq_{S,3})$. Regarding ${}^iq_3$, the respective transitions in ${}^i\mathbf{G}||{}^i\mathbf{S}$ are ${}^if_{G||S}(({}^iq_3, {}^iq_{S,5}), {}^{i+1}e_{BD}) = ({}^iq_4, {}^iq_{S,5})$ and ${}^if_{G||S}(({}^iq_3, {}^iq_{S,5}), {}^{i+1}e_{BP}) = ({}^iq_4, {}^iq_{S,5})$. At first, it is observed that ${}^i\mathbf{G}||{}^i\mathbf{S}$ is a blocking automaton, where all events provoking a transition to the non-marked state ${}^iq_4$ are events of U_{i+1} . Next, the properties of ${}^i\mathbf{G}||{}^{i+1}\mathbf{S}_B$ will be examined. It is observed that the transitions to the non-marked state ${}^iq_4$, from the state ${}^iq_1$, are

$${}^if_{G||B}(({}^iq_1, {}^iq_{B,S,1}), {}^ie_{BP}) = ({}^iq_4, {}^iq_{B,S,1}), {}^if_{G||B}(({}^iq_1, {}^iq_{B,S,2}), {}^ie_{BP}) = ({}^iq_4, {}^iq_{B,S,2}), \\ {}^if_{G||B}(({}^iq_1, {}^iq_{B,S,4}), {}^ie_{BP}) = ({}^iq_4, {}^iq_{B,S,4})$$

where ${}^if_{G||B}(\cdot, \cdot)$ is the transition function of the automaton ${}^i\mathbf{G}||{}^{i+1}\mathbf{S}_B$. The respective transitions from ${}^iq_2$ are

$${}^if_{G||B}(({}^iq_2, {}^iq_{B,S,5}), {}^ie_{BP}) = ({}^iq_4, {}^iq_{B,S,5}), {}^if_{G||B}(({}^iq_2, {}^iq_{B,S,5}), {}^ie_{BD}) = ({}^iq_4, {}^iq_{B,S,5})$$

The transition from ${}^iq_3$ is ${}^if_{G||B}(({}^iq_3, {}^iq_{B,S,3}), {}^ie_{BP}) = ({}^iq_4, {}^iq_{B,S,3})$. Thus, the automaton ${}^i\mathbf{G}||{}^{i+1}\mathbf{S}_B$ is a blocking automaton, where all events provoking a transition to the non-marked state ${}^iq_4$ are events of U_i . From the above it is concluded that the joint action of the control units U_i and U_{i+1} to ${}^i\mathbf{G}$ restricts all transitions to ${}^iq_4$. Since, ${}^iq_4$ is the only non-marked state of ${}^i\mathbf{G}$, it is concluded that ${}^i\mathbf{G}$, under the influence of the supervisors ${}^i\mathbf{S}$ and ${}^{i+1}\mathbf{S}_B$, namely the resulting controlled automaton, has all its accessible states marked. Clearly, the controlled automaton is a nonblocking automaton. $\square$

From Lemma 2 it is concluded that the non-marked state of the automaton of the buffer B_i is non-accessible, under the influence of the control actions imposed by U_i and U_{i+1} . Define the automaton ${}^i\tilde{\mathbf{G}}$, being a subautomaton of ${}^i\mathbf{G}$ and derived by removing the single non-marked state ${}^iq_4$ and the corresponding transitions of ${}^i\mathbf{G}$. Clearly, the set of the states of ${}^i\tilde{\mathbf{G}}$ is the set ${}^i\mathbb{Q}_m$. Using the present definition, the following synchronous products, being subautomata of the controlled automata in (36) and (38), are derived

$${}^i\tilde{\mathbf{G}}_c = {}^i\mathbf{G}_R||{}^{i,1}\mathbf{G}_c||{}^{i,2}\mathbf{G}_c||{}^i\tilde{\mathbf{G}}||{}^{i-1}\tilde{\mathbf{G}}||{}^i\mathbf{S}_R||{}^i\mathbf{S}||{}^i\mathbf{S}_B||{}^{i,1}\mathbf{S}_D||{}^{i,2}\mathbf{S}_D, i \in \{1, \dots, n\} \quad (49a)$$

$${}^{n+1}\tilde{\mathbf{G}}_c = {}^{n+1}\mathbf{G}_R||\left(\bigparallel_{j=1}^m ({}^{n+1,j}\mathbf{G}_c)\right)||{}^n\tilde{\mathbf{G}}||{}^{n+1}\mathbf{S}_R||{}^{n+1}\mathbf{S}_B||{}^{n+1,2}\mathbf{S}_D \quad (49b)$$

In the following two theorems, the property of nonblocking of ${}^i\tilde{G}_c$, where $i = \{2, \dots, n\}$, and the nonblocking property of ${}^{n+1}\tilde{G}_c$ will be proven. Following [15] (Section 4.9, p. 190), the proof will be accomplished by investigating if the controlled automaton always has an active transition from any non-marked state to a marked state. Finally, a remark, being the second remark of the present section, will cover the nonblocking property of ${}^1\tilde{G}_c$.

Theorem 3. *The synchronous product in (49a), namely the controlled automaton ${}^i\tilde{G}_c$, where $i \in \{2, \dots, n\}$, is a nonblocking automaton.*

Proof of Theorem 3. For the proof of Theorem 3, see Appendix A. $\square$

Remark 2. *The proof of the nonblocking property of ${}^1\tilde{G}_c$ follows the one of Theorem 3, if the events of the loading docks are considered, i.e., if the events e_I and e_O are used instead of the events ${}^{i,i-1}e_{BP}$ and ${}^{i,i-1}e_{BD}$ in the proof of Theorem 3.*

Theorem 4. *The synchronous product in (49b), namely the controlled automaton ${}^{n+1}\tilde{G}_c$, is a nonblocking automaton.*

Proof of Theorem 4. For the proof of Theorem 4, see Appendix B. $\square$

In the following theorem, the nonblocking property of the overall manufacturing system will be proven.

Theorem 5. *The automaton ${}^1G_c || {}^2G_c || \dots || {}^{n+1}G_c$ is a nonblocking automaton.*

Proof of Theorem 5. Consider the automata ${}^iG_c$ and ${}^{i+1}G_c$, where $i \in \{1, \dots, n\}$. Using Lemma 1, the synchronous product ${}^iG_c || {}^{i+1}G_c$ of the two automata is expressed as follows:

$$\begin{aligned} {}^iG_c || {}^{i+1}G_c = & {}^iG_R || \left(\begin{array}{c} |\mathbb{J}(i)| \\ || \\ i,j \\ j=1 \end{array} {}^iG_c \right) || {}^iG || {}^{i-1}G || {}^iS_R || {}^iS || {}^iS_B || {}^{i,1}S_D || {}^{i,2}S_D || {}^{i+1}G_R || \\ & \left(\begin{array}{c} |\mathbb{J}(i+1)| \\ || \\ i+1,j \\ j=1 \end{array} {}^{i+1}G_c \right) || {}^{i+1}G || {}^iG || {}^{i+1}S_R || {}^{i+1}S || {}^{i+1}S_B || {}^{i+1,1}S_D || {}^{i+1,2}S_D. \end{aligned} \quad (50)$$

where $|\cdot|$ denotes the cardinality of the argument set. Using the supervisors ${}^{i+1}S_B$ and ${}^iS$ twice in the above formula, it is observed that the closed and the marked language of ${}^iG_c || {}^{i+1}G_c$ are equal to the closed and the marked language of the automaton:

$$\begin{aligned} {}^iG_R || \left(\begin{array}{c} |\mathbb{J}(i)| \\ || \\ i,j \\ j=1 \end{array} {}^iG_c \right) || {}^iG || {}^{i-1}G || {}^iS_R || {}^iS || {}^{i+1}S_B || {}^iS_B || {}^{i,1}S_D || {}^{i,2}S_D || {}^{i+1}G_R || \\ \left(\begin{array}{c} |\mathbb{J}(i+1)| \\ || \\ i+1,j \\ j=1 \end{array} {}^{i+1}G_c \right) || {}^{i+1}G || {}^iG || {}^{i+1}S_R || {}^{i+1}S || {}^{i+1}S_B || {}^iS || {}^{i+1,1}S_D || {}^{i+1,2}S_D. \end{aligned} \quad (51)$$

Using the remarks before relation (49a,b), it is observed that the closed and the marked behavior of the above automaton are equal to the closed and the marked behavior of the following automaton:

$$\begin{aligned} {}^iG_R || \left(\begin{array}{c} |\mathbb{J}(i)| \\ || \\ i,j \\ j=1 \end{array} {}^iG_c \right) || {}^i\tilde{G} || {}^{i-1}G || {}^iS_R || {}^iS || {}^{i+1}S_B || {}^iS_B || {}^{i,1}S_D || {}^{i,2}S_D || {}^{i+1}G_R || \\ \left(\begin{array}{c} |\mathbb{J}(i+1)| \\ || \\ i+1,j \\ j=1 \end{array} {}^{i+1}G_c \right) || {}^{i+1}G || {}^i\tilde{G} || {}^{i+1}S_R || {}^{i+1}S || {}^{i+1}S_B || {}^iS || {}^{i+1,1}S_D || {}^{i+1,2}S_D. \end{aligned} \quad (52)$$

Using (49a,b), it is observed that the closed and the marked behavior of ${}^1\mathbf{G}_c || {}^2\mathbf{G}_c || \dots || {}^{n+1}\mathbf{G}_c$ are equal to the closed and the marked behavior of the subautomaton ${}^1\tilde{\mathbf{G}}_c || {}^2\tilde{\mathbf{G}}_c || \dots || {}^{n+1}\tilde{\mathbf{G}}_c$. In Theorems 3 and 4, the nonblocking property of ${}^i\tilde{\mathbf{G}}_c$, where $i \in \{1, \dots, n+1\}$, has been proved. Furthermore, in Theorems 1 and 2 it has been shown that all the events from the adjacent control units are considered as uncontrollable events, and they are not restricted by the respective supervisors. Thus, the supervisors of the control unit U_i do not restrict the events of ${}^i\tilde{\mathbf{G}}_c$, being common with ${}^{i-1}\tilde{\mathbf{G}}_c$ or ${}^{i+1}\tilde{\mathbf{G}}_c$. Furthermore, ${}^i\tilde{\mathbf{G}}_c$ does not have any common event with the rest automata. Thus, the nonblocking property of ${}^i\tilde{\mathbf{G}}_c$ is not affected by the rest-controlled automata in the synchronous product ${}^1\tilde{\mathbf{G}}_c || {}^2\tilde{\mathbf{G}}_c || \dots || {}^{n+1}\tilde{\mathbf{G}}_c$. Hence, ${}^1\tilde{\mathbf{G}}_c || {}^2\tilde{\mathbf{G}}_c || \dots || {}^{n+1}\tilde{\mathbf{G}}_c$ is a nonblocking automaton, and consequently ${}^1\mathbf{G}_c || {}^2\mathbf{G}_c || \dots || {}^{n+1}\mathbf{G}_c$ is also a nonblocking automaton. $\square$

In Theorems 1 and 2 and in Remark 1, the physical realizability of the supervisors implemented in U_i with respect to the respective manufacturing devices related to U_i , namely the devices B_i , R_i and $C_{i,j}$ where $j \in \mathbb{J}(i)$, through the respective synchronous products, has been proved. The events of the adjacent control units have been considered as uncontrollable events and they cannot be restricted by U_i . The latter implies that the events of U_{i+1} used in U_i have not been restricted and vice versa. Thus, the PR property of the synchronous product ${}^1\mathbf{G}_c || {}^2\mathbf{G}_c || \dots || {}^{n+1}\mathbf{G}_c$ is derived.

From relations (40) and (49), as well as the five theorems and the remarks, presented above, it is concluded that the system performance of the controlled automaton ${}^i\mathbf{G}_c$ is satisfactory for every $i \in \{1, \dots, n+1\}$. Recall that the languages ${}^{i,j}\mathbb{K}$ are prefix-closed languages and ${}^{i,j}\tilde{P}^{-1}({}^{i,j}\mathbb{K}) \subseteq {}^{i,j}P^{-1}(\mathbb{L}({}^{i,j}\mathbf{G}))$, $\forall i \in \{1, \dots, n+1\}$, and $\forall j \in \mathbb{J}(i)$, as well as that the languages ${}^i\mathbb{K}_R$ are $\mathbb{L}_m({}^i\mathbf{G}_R)$ -closed (see Proposition 1) and ${}^i\mathbb{K}_R \subseteq \mathbb{L}_m({}^i\mathbf{G}_R)$, for every $i \in \{1, \dots, n+1\}$. The satisfactory performance of the closed and the marked behaviors of the controlled automata, i.e., the satisfactory performance of the languages $\mathbb{L}({}^i\mathbf{G}_c)$ and $\mathbb{L}_m({}^i\mathbf{G}_c)$, for every $i \in \{1, \dots, n+1\}$, is validated by the following relations:

$$\begin{aligned} \mathbb{L}({}^i\mathbf{G}_c) &= \mathbb{L}({}^i\tilde{\mathbf{G}}_c) = {}^iP_R^{-1}(\mathbb{L}({}^i\mathbf{G}_R) \cap {}^iP_R^{-1}(\overline{{}^i\mathbb{K}_R})) \cap {}^iP^{-1}(\mathbb{L}({}^i\tilde{\mathbf{G}})) \cap {}^iP_B^{-1}(\mathbb{L}({}^{i-1}\tilde{\mathbf{G}})) \cap \\ &\quad \bigcap_{j=1}^{|\mathbb{J}(i)|} ({}^{i,j}P^{-1}(\mathbb{L}({}^{i,j}\mathbf{G})) \cap {}^{i,j}\tilde{P}^{-1}(\overline{{}^{i,j}\mathbb{K}})) \cap {}^i\tilde{P}^{-1}(\overline{{}^i\mathbb{K}}) \cap {}^i\tilde{P}_B^{-1}(\overline{{}^i\mathbb{K}_B}) \cap \bigcap_{j=1}^2 ({}^{i,j}P_D^{-1}(\overline{{}^{i,j}\mathbb{K}_D})) = \\ &= {}^iP_R^{-1}(\overline{{}^i\mathbb{K}_R}) \cap {}^i\tilde{P}^{-1}(\overline{{}^i\mathbb{K}}) \cap {}^i\tilde{P}_B^{-1}(\overline{{}^i\mathbb{K}_B}) \cap \bigcap_{j=1}^2 ({}^{i,j}P_D^{-1}(\overline{{}^{i,j}\mathbb{K}_D})) \cap \bigcap_{j=1}^{|\mathbb{J}(i)|} {}^{i,j}\tilde{P}^{-1}(\overline{{}^{i,j}\mathbb{K}}), \end{aligned} \quad (53a)$$

$$\begin{aligned} \mathbb{L}_m({}^i\mathbf{G}_c) &= \mathbb{L}_m({}^i\tilde{\mathbf{G}}_c) = {}^iP_R^{-1}(\mathbb{L}_m({}^i\mathbf{G}_R) \cap {}^iP_R^{-1}(\overline{{}^i\mathbb{K}_R})) \cap {}^iP^{-1}(\mathbb{L}_m({}^i\tilde{\mathbf{G}})) \cap {}^iP_B^{-1}(\mathbb{L}_m({}^{i-1}\tilde{\mathbf{G}})) \cap \\ &\quad \bigcap_{j=1}^{|\mathbb{J}(i)|} ({}^{i,j}\tilde{P}^{-1}(\overline{{}^{i,j}\mathbb{K}}) \cap {}^{i,j}P^{-1}(\mathbb{L}_m({}^{i,j}\mathbf{G}))) \cap {}^i\tilde{P}^{-1}(\overline{{}^i\mathbb{K}}) \cap {}^i\tilde{P}_B^{-1}(\overline{{}^i\mathbb{K}_B}) \cap \bigcap_{j=1}^2 ({}^{i,j}P_D^{-1}(\overline{{}^{i,j}\mathbb{K}_D})) = \\ &= {}^iP_R^{-1}(\overline{{}^i\mathbb{K}_R}) \cap {}^i\tilde{P}^{-1}(\overline{{}^i\mathbb{K}}) \cap {}^i\tilde{P}_B^{-1}(\overline{{}^i\mathbb{K}_B}) \cap \bigcap_{j=1}^2 ({}^{i,j}P_D^{-1}(\overline{{}^{i,j}\mathbb{K}_D})) \cap \left(\bigcap_{j=1}^{|\mathbb{J}(i)|} {}^{i,j}\tilde{P}^{-1}(\overline{{}^{i,j}\mathbb{K}}) \right), \end{aligned} \quad (53b)$$

where for the derivation of the above relations, the relations ${}^i\tilde{P}^{-1}({}^i\mathbb{K}) \subseteq {}^iP^{-1}(\mathbb{L}_m({}^i\tilde{\mathbf{G}}))$ and ${}^i\tilde{P}_B^{-1}({}^i\mathbb{K}_B) \subseteq {}^iP_B^{-1}(\mathbb{L}_m({}^{i-1}\tilde{\mathbf{G}}))$ have been used.

7. Simulation Results

To illustrate the performance of the closed and marked behaviors of the controlled automata of the proposed control scheme, simulation results for the controlled automata of the overall wafer manufacturing system will be presented. Consider the case of the wafer manufacturing process where $n = 4$ and $m = 3$. In addition, consider the case where the following concatenation of events took place:

$$e_I^{1,1}e_D^{1,1}e_P^{1,1}e_u^{1,1}e_P^{1,1}e_D^{1,1}e_{BD}e_I^{2,1}e_{BP}^{1,1}e_D^{2,1}e_D^{1,1}e_u^{1,1}e_P^{2,1}e_u^{2,1}e_P^{1,1}e_P^{2,1}e_{BD}^{1,1}e_{BD}$$

The above event concatenation corresponds to a sequence of operation commands appropriately enriched with sensor signals. To interpret the above word, appropriate prefixes of the word will be analyzed. First, consider the prefix $e_I^{1,1}e_D^{1,1}e_P$. This prefix is analyzed as follows: R_1 is commanded to pick a product from the loading dock; R_1 is commanded to drop the product to $C_{1,1}$; then, R_1 is commanded to pick the product from $C_{1,1}$. The latter command would lead $C_{1,1}$ to malfunction, but the supervisor ^1S has deactivated this command. Thus, the supervisor allows the station to manufacture the dropped product and the sensor signal, indicating that $C_{1,1}$ finished the manufacturing of the product, to take place. Thus, the prefix $e_I^{1,1}e_D^{1,1}e_P^{1,1}e_u$ has been interpreted. Continuing, the prefix $e_I^{1,1}e_D^{1,1}e_P^{1,1}e_u^{1,1}e_P^{1,1}e_D^{1,1}e_{BD}$ is analyzed as follows: After the sensor signal, indicating that $C_{1,1}$ finished the manufacturing of the product, R_1 is commanded to pick the product from $C_{1,1}$. After, R_1 is commanded to drop the product again to $C_{1,1}$. However, the latter command is disabled by the supervisors ^1S and ^1S_R . So, the next command, namely the command to drop the product to B_1 , is executed. Continuing with the prefix $e_I^{1,1}e_D^{1,1}e_P^{1,1}e_u^{1,1}e_P^{1,1}e_D^{1,1}e_{BD}e_I^{2,1}e_{BP}^{1,1}e_D^{2,1}e_D$, the remaining commands of the prefix are analyzed as follows: R_1 is commanded to pick another product from the loading dock; R_2 is commanded to pick a product from B_2 ; R_1 is commanded to drop a product to $C_{1,1}$; R_2 is commanded to drop a product to $C_{2,1}$. Since $C_{1,1}$ has already finished the manufacturing of the product, R_1 is commanded to pick the product from $C_{1,1}$, but the supervisor ^1S_D has deactivated the command ^1e_P till $C_{2,1}$ is empty. Thus, the prefix $e_I^{1,1}e_D^{1,1}e_P^{1,1}e_u^{1,1}e_P^{1,1}e_D^{1,1}e_{BD}e_I^{2,1}e_{BP}^{1,1}e_D^{2,1}e_D^{1,1}e_u$ has been interpreted. After $C_{2,1}$ finished the manufacturing of the product, the remaining commands of the whole word are interpreted as follows: R_2 is commanded to pick a product from $C_{2,1}$, R_1 is commanded to pick a product from $C_{1,1}$, R_2 is commanded to drop a product to B_2 , and R_1 is commanded to drop a product to B_1 .

In Table 1, the influence of the previous events concatenation to the automata of the overall controlled wafer manufacturing system are presented. In the first column of Table 1, the events that take place are presented. In the next six columns of Table 1, the current states of the automata of the robotic systems R_1 and R_2 , the production stations $C_{1,1}$ and $C_{2,1}$, and the buffers B_1 and B_2 are presented.

Table 1. Simulation of the wafer manufacturing process for a sequence of events.

Event	Robot R_1	Robot R_2	Station $C_{1,1}$	Station $C_{2,1}$	Buffer B_1	Buffer B_2
-	$^1q_{R,1}$	$^2q_{R,1}$	$^{1,1}q_1$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$
e_I	$^1q_{R,2}$	$^2q_{R,1}$	$^{1,1}q_1$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$
$^{1,1}e_D$	$^1q_{R,1}$	$^2q_{R,1}$	$^{1,1}q_2$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$
$^{1,1}e_P$	$^1q_{R,1}$	$^2q_{R,1}$	$^{1,1}q_2$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$
$^{1,1}e_u$	$^1q_{R,1}$	$^2q_{R,1}$	$^{1,1}q_3$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$
$^{1,1}e_P$	$^1q_{R,2}$	$^2q_{R,1}$	$^{1,1}q_1$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$
$^{1,1}e_D$	$^1q_{R,2}$	$^2q_{R,1}$	$^{1,1}q_1$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$
$^{1,1}e_{BD}$	$^1q_{R,1}$	$^2q_{R,1}$	$^{1,1}q_1$	$^{2,1}q_1$	$^1q_2$	$^{2,1}q_1$
e_I	$^1q_{R,2}$	$^2q_{R,1}$	$^{1,1}q_1$	$^{2,1}q_1$	$^1q_2$	$^{2,1}q_1$
$^{2,1}e_{BP}$	$^1q_{R,2}$	$^2q_{R,2}$	$^{1,1}q_1$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$
$^{1,1}e_D$	$^1q_{R,1}$	$^2q_{R,2}$	$^{1,1}q_2$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$
$^{2,1}e_D$	$^1q_{R,1}$	$^2q_{R,1}$	$^{1,1}q_2$	$^{2,1}q_2$	$^1q_1$	$^{2,1}q_1$
$^{1,1}e_u$	$^1q_{R,1}$	$^2q_{R,1}$	$^{1,1}q_1$	$^{2,1}q_2$	$^1q_1$	$^{2,1}q_1$
$^{1,1}e_P$	$^1q_{R,1}$	$^2q_{R,1}$	$^{1,1}q_3$	$^{2,1}q_2$	$^1q_1$	$^{2,1}q_1$
$^{2,1}e_u$	$^1q_{R,1}$	$^2q_{R,1}$	$^{1,1}q_3$	$^{2,1}q_3$	$^1q_1$	$^{2,1}q_1$
$^{2,1}e_P$	$^1q_{R,1}$	$^2q_{R,2}$	$^{1,1}q_3$	$^{2,1}q_1$	$^1q_1$	$^{2,1}q_1$

Table 1. Cont.

Event	Robot R_1	Robot R_2	Station $C_{1,1}$	Station $C_{2,1}$	Buffer B_1	Buffer B_2
${}^{1,1}e_P$	${}^1q_{R,2}$	${}^2q_{R,2}$	${}^{1,1}q_3$	${}^{2,1}q_1$	${}^1q_1$	${}^{2,1}q_1$
${}^{2,1}e_{BD}$	${}^1q_{R,2}$	${}^2q_{R,1}$	${}^{1,1}q_3$	${}^{2,1}q_1$	${}^1q_1$	${}^{2,1}q_2$
${}^{1,1}e_{BD}$	${}^1q_{R,1}$	${}^2q_{R,1}$	${}^{1,1}q_3$	${}^{2,1}q_1$	${}^1q_2$	${}^{2,1}q_2$

The colored rows in Table 1 indicate that an event that was restricted by the supervisors took place. According to the third row, the event ${}^{1,1}e_P$ has been restricted to take place by the supervisor ${}^{1,1}S$. According to the sixth row, the event ${}^{1,1}e_D$ has been restricted to take place by the supervisors ${}^1S$ and ${}^1S_R$. Finally, according to the 13th row, the event ${}^{1,1}e_P$ has been restricted to take place by the supervisor ${}^{1,1}S_D$. The event ${}^{1,1}e_P$ will not provoke a transition in the system until the event ${}^{2,1}e_P$ takes place that empties the production machine $C_{2,1}$. As already mentioned in Section 3, the supervisor ${}^{1,1}S_D$ is not included in the supervisory control scheme proposed in [7]. However, the decentralized architecture proposed in [7] guarantees that if there are free slots (buffers and/or production machines), after $C_{1,1}$ and B_1 , then the product will eventually move on; a possible malfunction in the production machine $C_{2,1}$ may cause product blocking in the buffer B_1 .

8. Complexity of the Control Architecture

The complexity of an automaton is described by the triad including the number of states, the number of events, and the number of transitions of the automaton; indicatively, see [21–23]. The complexity triads of ${}^{i,1}S$ and ${}^{i,2}S$, where $i \in \{1, \dots, n\}$, are equal to (3, 4, 6). The complexity triad of ${}^iS$, where $i \in \{1, \dots, n\}$, is equal to (5, 6, 18). The complexity triad of ${}^iS_B$, where $i \in \{2, \dots, n+1\}$, is equal to (5, 6, 18). The complexity triad of ${}^{i,1}S_D$, where $i \in \{1, \dots, n\}$, is equal to (2, 2, 3). The complexity triad of ${}^{i,2}S_D$, where $i \in \{2, \dots, n+1\}$, is (2, 2, 3). The complexity triad of ${}^iS_R$, where $i \in \{1, \dots, n\}$, is (5, 8, 8). The complexity triad of the supervisor ${}^{n+1}S_R$ is $(m+2, 2m+2, 2m+2)$. Regarding the supervisors ${}^{n+1,1}S_D$, ${}^{1,2}S_D$, ${}^{n+1}S$, and ${}^1S_B$, it is mentioned that is not necessary to be realized. Hence, the total complexity triad of all supervisors in U_1 is computed to be (18, 24, 39). The total complexity triad of all supervisors in U_i , where $i \in \{2, \dots, n\}$, is computed to be (25, 32, 50). Similarly, the total complexity triad of all supervisors in U_{n+1} is computed to be $(4m+9, 6m+10, 8m+23)$. Hence, the total complexity of the proposed distributed supervisor design scheme is computed to be $(25n+4m+2, 32n+6m+2, 50n+8m+12)$. It is mentioned that the total complexity of the proposed supervisory scheme is linear with respect to the number of robots and the number of stations. It is also mentioned that the complexity of the present wafer manufacturing process is equal to $(10n+4m+4, 15n+5m+4, 27n+9m+4)$. In comparison to the complexity of the wafer manufacturing process, it is observed that the complexity of the supervisor scheme proposed here is of analogous order of size.

9. Implementation of the Distributed Supervisor Architecture

The problem of implementation of modular supervisory control schemes in PLCs is tackled in many articles; see [24,25]. In [21,22], the implementation of various modular supervisory control schemes to real-time industrial controllers, e.g., PLCs, has been presented using ladder diagrams following the international standard IEC 61131-3 (2013). The distributed supervisory architecture proposed here can be easily implemented following the directions of [22]. More specifically, each local control unit implements a specific number of supervisors with predefined complexity. The supervisors of each local control unit can be realized either by classic ladder diagrams or by function blocks using ladder diagrams or languages based on structured text. It is important to mention that, as shown in Section 5, many supervisors share a common structure. Hence, they can be easily implemented in the event-driven architecture of the IEC 61499 function blocks [26] with parametric input/output signals [27]. The above characteristic of the supervisory control

architecture, proposed here, facilitates the implementation of the supervisors as they can easily be programmed in function blocks by developing one library for each supervisor. Clearly, for any possible extension of the manufacturing process, the respective control scheme can easily be implemented using these libraries.

In order to reveal the implementability of the present results, the supervisors embedded in U_i are implemented, using the IEC 61131-3 (2013) [28] ladder diagram (LD) framework, shown in Figures 18–22. The implementation is based on the respective state diagrams presented in Section 4. The ease of implementability of the present scheme is based on the explicit design of small-size supervisor automata. This is an advantage of the present results. In Figures 18–22 the numbers at the side of each block, named rung, denote the transitions of each state.

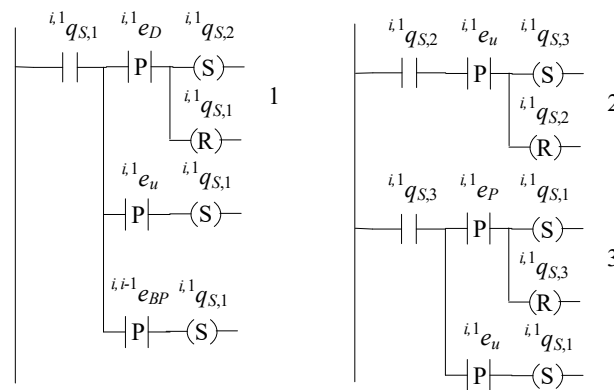


Figure 18. Ladder diagram of $i,1S$.

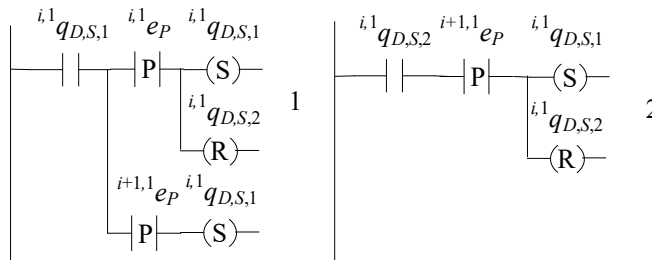


Figure 19. Ladder diagram of $i,1SD$.

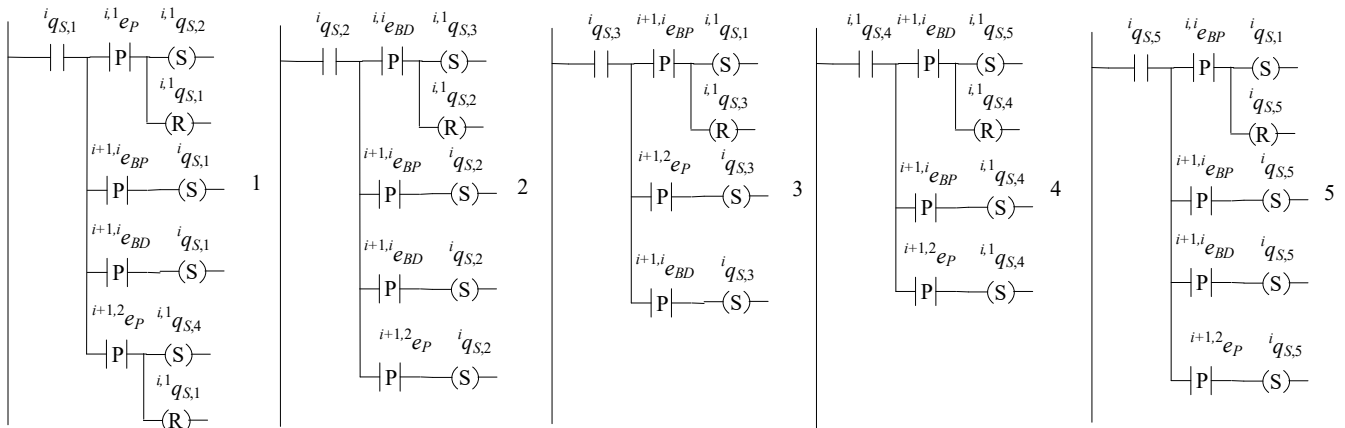
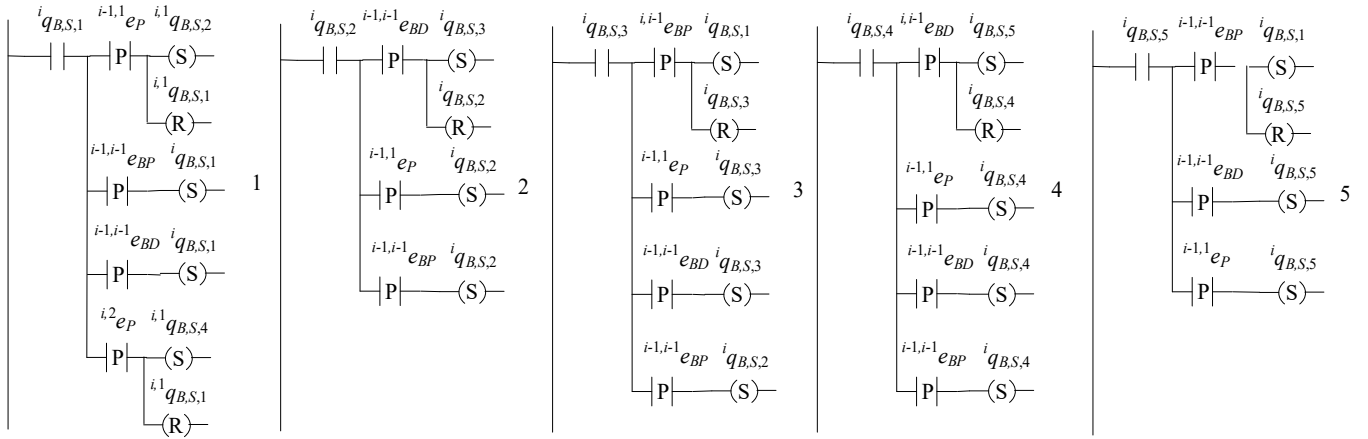
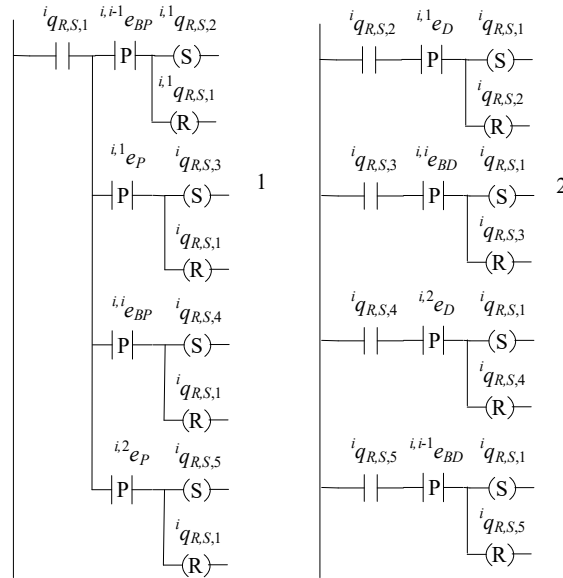


Figure 20. Ladder diagram of iS .

Figure 21. Ladder diagram of iS_B .Figure 22. Ladder diagram of iS_R .

As mentioned in Section 4, the state diagrams of $^{1,1}S$, $^{i,2}S$, and $^{n+1,j}S$, where $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$, have the same structure as the state diagram of $^{i,1}S$. Hence, the LDs of $^{1,1}S$, $^{i,2}S$, and $^{n+1,j}S$, where $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$, have the same structure as the state diagram of $^{i,1}S$ in Figure 18. Similarly, the LD of $^{i,2}S_D$, where $i \in \{2, \dots, n+1\}$, has the same structure as the LD of $^{i,1}S_D$ in Figure 19. Finally, it is mentioned that the LDs of ^1S_R and $^{n+1}S_R$ have the same structure as the LD of iS_R in Figure 22.

According to [29], a distributed control architecture cannot be viewed only in a geographical point of view but also regarding the functionality of the system's components. In order to impose the desired control actions [30], the control units of a distributed architecture communicate with each other and possibly with other external units. In Industry 4.0, the most common network for the interconnection of the distributed controllers is the ethernet network for Supervisory Control and Data Acquisition (SCADA) and Manufacturing Execution System (MES), which connects controllers, machines, and systems ([29–31]). Regarding PLC communication, a widely used network is the OPC UA as well as the Modbus protocol (see [32,33]). In the present control architecture, the resources of the control unit of each robotic manipulator are the signals of the previous and the next robotic manipulator as well as the two uncontrollable events (signals) of the production machines.

All these signals, together with the commands of the respective robotic manipulator, are the signals used by the respective control unit and required from the design specifications of the supervisors implemented in the control unit.

The control implementation directions, proposed above, facilitate the development of a control software tool that is dedicated to wafer manufacturing industries with no limitation to the number of vacuum chambers and consequently the number of robotic manipulators. The software offers itself for newly installed or extended wafer manufacturing plants. This can be accomplished due to the parametric and distributed structure of the control scheme. In simple words, the implementation is accomplished by embedding a copy of the local control software to any additional control unit. Therefore, the work of the programmer is limited mainly to the assignment of the parameters of the local control function blocks.

10. Conclusions

A modular parametric discrete event model with respect to the number of production stations, the number of buffers, and the number of robotic manipulators has been developed for a wafer manufacturing system. The parametric model has been built in a way that provides a clear direction towards the modelling of the continuously expandable wafer production industry. A set of desired specifications has been imposed in the form of five rules. A comparison between the proposed rules and those of the literature has been presented. Next, the desired specifications have been translated and decomposed to a set of local regular languages, one set per robotic manipulator.

A distributed supervisory control architecture has been developed based on the local regular languages. In the proposed architecture, a set of local supervisors is designed for each robotic manipulator with appropriate signal resources shared between adjacent control units.

Here, the PR properties of the developed supervisors have been proved regarding the total automaton of the system. The nonblocking property of the proposed architecture has been guaranteed. Finally, implementation issues have been tackled, and the complexity of the distributed architecture has been determined in a parametric formula.

The extension of the modelling and the distributed supervisory control architecture to wafer manufacturing systems with a parametric number of production stations served by the same robotic manipulator is currently under investigation. In addition, diagnosis and fault tolerance issues of the wafer manufacturing systems are under investigation. First, the fault tolerance characteristics of the proposed control scheme will be investigated in the cases of sensors and actuator faults as well as the resilience of the control scheme in the presence of cyberattacks. Second, extensions of the proposed control scheme towards improving tolerance will be investigated. Third, the enrichment of the proposed control scheme with appropriately designed observers is expected to contribute to a satisfactory solution of the diagnosis problem.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/s23094545/s1>, Table S1: List of Acronyms; Table S2: List of symbols.

Author Contributions: Conceptualization, F.N.K. and D.G.F.; methodology, F.N.K. and D.G.F.; validation, F.N.K.; investigation, D.G.F. and P.G.; resources, D.G.F.; data curation, D.G.F. and P.G.; writing—original draft preparation, F.N.K. and D.G.F.; writing—review and editing, D.G.F. and P.G.; supervision, F.N.K.; project administration, F.N.K. and D.G.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article and Supplementary Material.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Proof of Theorem 3. Any state of ${}^i\tilde{G}_c$ is of the following 12-tuple form:

$$({}^iq_R, {}^{i,1}q, {}^{i,2}q, {}^iq, {}^{i-1}q, {}^iq_{R,S}, {}^{i,1}q_S, {}^{i,2}q_S, {}^iq_S, {}^iq_{B,S}, {}^{i,1}q_{D,S}, {}^{i,2}q_{D,S}) \in {}^iQ_R \times {}^{i,1}Q \times {}^{i,2}Q \times {}^iQ_m \times {}^{i-1}Q_m \times {}^iQ_{R,S} \times {}^{i,1}Q_S \times {}^{i,2}Q_S \times {}^iQ_S \times {}^iQ_{B,S} \times {}^{i,1}Q_{D,S} \times {}^{i,2}Q_{D,S}.$$

It is important to mention that all states of the supervisors in (49a) are marked states. Regarding the automata ${}^{i,1}G_c$, ${}^{i,2}G_c$, ${}^i\tilde{G}$, and ${}^{i-1}\tilde{G}$, it is noted that according to Lemma 1 and Lemma 2, all of their states are marked. Thus, the characteristic of a state of ${}^i\tilde{G}_c$ to be or not be a marked state depends entirely upon the respective element of the 12-tuple state, being a state of ${}^iG_R$. Hence, for ${}^i\tilde{G}_c$ to be a nonblocking automaton, it suffices to prove that

$${}^i\mathbb{H}_c\left({}^iq_R, {}^{i,1}q, {}^{i,2}q, {}^iq, {}^iq_{R,S}, {}^{i,1}q_S, {}^{i,2}q_S, {}^iq_S, {}^{i+1}q_S, {}^iq_{B,S}, {}^{i,1}q_{D,S}, {}^{i,2}q_{D,S}\right) \cap {}^i\mathbb{H}_R({}^iq_{R,2}) \neq \emptyset$$

where ${}^i\mathbb{H}_c(\cdot)$ is the active event set per state of ${}^i\tilde{G}_c$, and ${}^i\mathbb{H}_R({}^iq_{R,2}) = \{{}^{i,1}e_D, {}^{i,2}e_D, {}^{i-1}e_{BD}, {}^ie_{BD}\}$. In other words, it suffices to prove that when the robotic manipulator picks a product, i.e., the automaton ${}^iG_R$ is in the state ${}^iq_{R,2}$, then a drop event is always in the active event set of the state of ${}^i\tilde{G}_c$, having as an element the state ${}^iq_{R,2}$, to guarantee that the robotic manipulator has the ability to drop the product, and consequently the automaton ${}^iG_R$ is able to return to the marked state ${}^iq_{R,1}$.

The alphabet of ${}^iS_R$ is ${}^iE_R$. Hence, it holds that ${}^iG_R || {}^iS_R = {}^iG_R \times {}^iS_R$, where “ $\times$ ” denotes the product of two automata (see [16]). Thus, according to the product definition, it holds that ${}^i\mathbb{H}_{R||S}({}^iq_R, {}^iq_{R,S}) = {}^i\mathbb{H}_R({}^iq_R) \cap {}^i\mathbb{H}_{R,S}({}^iq_{R,S})$, where ${}^i\mathbb{H}_{R||S}$ is the active event set of the automaton ${}^iG_R || {}^iS_R$. Since ${}^i\mathbb{H}_{R,S}({}^iq_{R,S}) \subseteq {}^i\mathbb{H}_R({}^iq_R)$, it is observed that the following equality holds true: ${}^i\mathbb{H}_{R||S}({}^iq_R, {}^iq_{R,S}) = {}^i\mathbb{H}_{R,S}({}^iq_{R,S})$. The transition of ${}^iG_R$ from ${}^iq_{R,1}$ to ${}^iq_{R,2}$ can be accomplished only through a pick event, i.e., an event of the set $\{{}^{i,1}e_P, {}^{i,2}e_P, {}^{i-1}e_{BP}, {}^ie_{BP}\}$. In what follows, the restrictions of the supervisors and the automata ${}^{i,1}G_c$, ${}^{i,2}G_c$, ${}^i\tilde{G}$, and ${}^{i-1}\tilde{G}$ to all possible transitions from ${}^iq_{R,2}$ to ${}^iq_{R,1}$ will be investigated by considering four different cases, for the arrival from ${}^iq_{R,1}$ to ${}^iq_{R,2}$. Taking into account the accessible states of the synchronous product in (49a), it is observed that if ${}^iG_R$ is in its initial state ${}^iq_{R,1}$, then ${}^iS_R$ is in its initial state ${}^iq_{R,S,1}$. Thus, the transitions from ${}^iq_{R,1}$ to ${}^iq_{R,2}$ through a pick event corresponds to the transition of ${}^iS_R$ from ${}^iq_{R,S,1}$ to the respective one of the supervisor’s rest four states. Based on this property, the four cases will be presented.

Case 1: The transition from the initial state ${}^iq_{R,1}$ to ${}^iq_{R,2}$, through the event ${}^{i,1}e_P$ is investigated. Regarding ${}^iG_R || {}^iS_R$, it is observed that if the event ${}^{i,1}e_P$ takes place, then ${}^iG_R$ arrives at ${}^iq_{R,2}$ and ${}^iS_R$ arrives at ${}^iq_{R,S,3}$. Note that ${}^i\mathbb{H}_{R,S}({}^iq_{R,S,3}) = \{{}^{i,i}e_{BD}\}$; thus, it holds that ${}^i\mathbb{H}_{R||S}({}^iq_{R,2}, {}^iq_{R,S,3}) = \{{}^{i,i}e_{BD}\}$. Regarding the rest supervisors, i.e., the supervisors ${}^iS_B$, ${}^{i,1}S_D$, and ${}^{i,2}S_D$, it is observed that their alphabets do not contain the event ${}^{i,i}e_{BD}$. Regarding the automata ${}^{i,1}G_c$, ${}^{i,2}G_c$, and ${}^{i-1}\tilde{G}$, it is observed that their alphabets do not contain the event ${}^{i,i}e_{BD}$. Thus, these supervisors and the automata ${}^{i,1}G_c$, ${}^{i,2}G_c$, and ${}^{i-1}\tilde{G}$ do not restrict the transitions of the synchronous product in (49a) through the event ${}^{i,i}e_{BD}$. From Figure 13, it is observed that if ${}^iS$ is in ${}^iq_{S,1}$, then if the event ${}^{i,1}e_P$ takes place, then the supervisor ${}^iS$ arrives at ${}^iq_{S,2}$. Clearly, it holds that ${}^i\mathbb{H}_S({}^iq_{S,2}) = \{{}^{i,i}e_{BD}\}$. Regarding ${}^i\tilde{G}$, it is mentioned that its alphabet contains the event ${}^{i,i}e_{BD}$. From Figure 3, it is observed that if ${}^i\tilde{G}$ is in ${}^iq_1$, then if the event ${}^{i,1}e_P$ takes place, then ${}^i\tilde{G}$ stays in ${}^iq_1$. Clearly, it holds that ${}^i\mathbb{H}({}^iq_1) = \{{}^{i,i}e_{BD}\}$. Regarding ${}^iG_R || {}^i\tilde{G} || {}^iS_R || {}^iS$, it holds that

$${}^i\mathbb{H}_c({}^iq_{R,2}, {}^iq_{R,S,3}, {}^iq_1, {}^iq_{S,2}) = {}^i\mathbb{H}_{R||S}({}^iq_{R,2}, {}^iq_{R,S,3}) \cap {}^i\mathbb{H}({}^iq_1) \cap {}^i\mathbb{H}_S({}^iq_{S,2}) = \{{}^{i,i}e_{BD}\}$$

Thus, the transition of ${}^iG_R$ from ${}^iq_{R,2}$ to ${}^iq_{R,1}$ is feasible through ${}^{i,i}e_{BD}$. If ${}^i\tilde{G}$ or ${}^iS$ are not in ${}^iq_1$ or ${}^iq_{S,1}$, respectively, i.e., it is in a state where the event ${}^{i,1}e_P$ is not an active event,

then no transition of the synchronous product ${}^i\mathbf{G}_R||{}^{i-1}\tilde{\mathbf{G}}||{}^i\mathbf{S}_R||{}^i\mathbf{S}$ will take place through the event ${}^{i,1}e_P$, and consequently ${}^i\mathbf{G}_R$ will be in the marked state ${}^iq_{R,1}$.

Case 2: The transition from the initial state ${}^iq_{R,1}$ to ${}^iq_{R,2}$ through the event ${}^{i,2}e_P$ is investigated. Regarding ${}^i\mathbf{G}_R||{}^i\mathbf{S}_R$, it is observed that if the event ${}^{i,2}e_P$ takes place, then ${}^i\mathbf{G}_R$ arrives at ${}^iq_{R,2}$ and ${}^i\mathbf{S}_R$ arrives at ${}^iq_{R,S,5}$. Note that ${}^i\mathbb{H}_{R,S}({}^iq_{R,S,5}) = \{{}^{i,i-1}e_{BD}\}$; thus, it holds that ${}^i\mathbb{H}_{R||S}({}^iq_{R,2}, {}^iq_{R,S,5}) = \{{}^{i,i-1}e_{BD}\}$. Regarding the rest supervisors, i.e., the supervisors ${}^i\mathbf{S}$, ${}^{i,1}\mathbf{S}_D$, and ${}^{i,2}\mathbf{S}_D$, it is observed that their alphabets do not contain the event ${}^{i,i-1}e_{BD}$. Regarding the automata ${}^{i,1}\mathbf{G}_c$, ${}^{i,2}\mathbf{G}_c$, and ${}^{i-1}\tilde{\mathbf{G}}$, it is observed that their alphabets do not contain the event ${}^{i,i-1}e_{BD}$. Thus, these supervisors and the automata ${}^{i,1}\mathbf{G}_c$, ${}^{i,2}\mathbf{G}_c$, and ${}^{i-1}\tilde{\mathbf{G}}$ do not restrict the transitions of the synchronous product in (49a) through the event ${}^{i,i-1}e_{BD}$. From Figure 13, it is observed that if ${}^i\mathbf{S}_B$ is in ${}^iq_{S,B,1}$, then if the event ${}^{i,i-1}e_{BD}$ takes place, then the supervisor ${}^i\mathbf{S}_B$ arrives at ${}^iq_{B,S,4}$. Clearly, it holds that ${}^i\mathbb{H}_{B,S}({}^iq_{B,S,4}) = \{{}^{i,i-1}e_{BD}\}$. Regarding ${}^{i-1}\tilde{\mathbf{G}}$, it is mentioned that its alphabet contains the event ${}^{i,i-1}e_{BD}$. From Figure 3, it is observed that if ${}^{i-1}\tilde{\mathbf{G}}$ is in ${}^{i-1}q_1$, then if the event ${}^{i,2}e_P$ takes place, then ${}^{i-1}\tilde{\mathbf{G}}$ stays in ${}^{i-1}q_1$. Clearly, it holds that ${}^{i-1}\mathbb{H}({}^{i-1}q_1) = \{{}^{i,i-1}e_{BD}\}$. Regarding ${}^i\mathbf{G}_R||{}^{i-1}\tilde{\mathbf{G}}||{}^i\mathbf{S}_R||{}^i\mathbf{S}_B$, it holds that

$${}^i\mathbb{H}_{i\mathbf{G}_R||{}^{i-1}\tilde{\mathbf{G}}||{}^i\mathbf{S}_R||{}^i\mathbf{S}_B}({}^iq_{R,2}, {}^iq_{R,S,5}, {}^{i-1}q_1, {}^iq_{B,S,4}) = {}^i\mathbb{H}_{R||S}({}^iq_{R,2}, {}^iq_{R,S,5}) \cap {}^{i-1}\mathbb{H}({}^{i-1}q_1) \cap {}^i\mathbb{H}_B({}^iq_1, {}^iq_{B,S,4}) = \{{}^{i,i-1}e_{BD}\}$$

where ${}^i\mathbb{H}_{i\mathbf{G}_R||{}^{i-1}\tilde{\mathbf{G}}||{}^i\mathbf{S}_R||{}^i\mathbf{S}_B}(\cdot, \cdot, \cdot, \cdot)$ denotes the active event sets of the states of ${}^i\mathbf{G}_R||{}^{i-1}\tilde{\mathbf{G}}||{}^i\mathbf{S}_R||{}^i\mathbf{S}_B$. Thus, the transition of ${}^i\mathbf{G}_R$ from ${}^iq_{R,2}$ to ${}^iq_{R,1}$, under the influence of ${}^{i-1}\tilde{\mathbf{G}}$, ${}^i\mathbf{S}_R$ and ${}^i\mathbf{S}_B$, is feasible through ${}^{i,i-1}e_{BD}$. If ${}^{i-1}\tilde{\mathbf{G}}$ is not in ${}^{i-1}q_1$ or ${}^i\mathbf{S}_B$ is not in ${}^iq_{B,S,4}$, the automaton ${}^i\mathbf{G}_R||{}^{i-1}\tilde{\mathbf{G}}||{}^i\mathbf{S}_R||{}^i\mathbf{S}_B$ is in a state where the event ${}^{i,2}e_P$ is not an active event and no transition of ${}^i\mathbf{G}_R||{}^{i-1}\tilde{\mathbf{G}}||{}^i\mathbf{S}_R||{}^i\mathbf{S}_B$ will take place, through the event ${}^{i,2}e_P$. Consequently ${}^i\mathbf{G}_R$ will be in the marked state ${}^iq_{R,1}$.

Case 3: The transition from the initial state ${}^iq_{R,1}$ to ${}^iq_{R,2}$ through the event ${}^{i,i-1}e_{BP}$ is investigated. Regarding ${}^i\mathbf{G}_R||{}^i\mathbf{S}_R$, it is observed that if the event ${}^{i,i-1}e_{BP}$ takes place, then ${}^i\mathbf{G}_R$ arrives at ${}^iq_{R,2}$ and ${}^i\mathbf{S}_R$ arrives at ${}^iq_{R,S,2}$. Note that ${}^i\mathbb{H}_{R,S}({}^iq_{R,S,2}) = \{{}^{i,1}e_D\}$; thus, it holds that ${}^i\mathbb{H}_{R||S}({}^iq_{R,2}, {}^iq_{R,S,2}) = \{{}^{i,1}e_D\}$. Regarding the rest supervisors, i.e., the supervisors ${}^i\mathbf{S}$, ${}^i\mathbf{S}_B$, ${}^{i,1}\mathbf{S}_D$, and ${}^{i,2}\mathbf{S}_D$, it is observed that their alphabets do not contain the event ${}^{i,1}e_D$. Regarding the automata ${}^{i,2}\mathbf{G}_c$, ${}^{i-1}\tilde{\mathbf{G}}$, and ${}^{i-1}\tilde{\mathbf{G}}$, it is observed that their alphabets do not contain the event ${}^{i,1}e_D$. Thus, these supervisors and the automata ${}^{i,2}\mathbf{G}_c$, ${}^{i-1}\tilde{\mathbf{G}}$, and ${}^{i-1}\tilde{\mathbf{G}}$ do not restrict the transitions of the synchronous product in (49a) through the event ${}^{i,1}e_D$. Regarding ${}^{i,1}\mathbf{G}_c$, it is mentioned that its alphabet contains the event ${}^{i,1}e_D$. From Figure 2, it is observed that if ${}^{i,1}\mathbf{G}_c$ is in $({}^{i,1}q_1, {}^{i,1}q_{S,1})$, then if the event ${}^{i,i-1}e_{BP}$ takes place, then ${}^{i,1}\mathbf{G}_c$ stays at $({}^{i,1}q_1, {}^{i,1}q_{S,1})$. Clearly, it holds that ${}^{i,1}\mathbb{H}_c({}^{i,1}q_1, {}^{i,1}q_{S,1}) = \{{}^{i,1}e_D\}$. Regarding ${}^i\mathbf{G}_R||{}^{i,1}\mathbf{G}_c||{}^i\mathbf{S}_R$, it holds that

$${}^i\mathbb{H}_c({}^iq_{R,2}, {}^iq_{R,S,2}, ({}^{i,1}q_1, {}^{i,1}q_{S,1})) = {}^i\mathbb{H}_{R||S}({}^iq_{R,2}, {}^iq_{R,S,2}) \cap {}^{i,1}\mathbb{H}_c({}^{i,1}q_1, {}^{i,1}q_{S,1}) = \{{}^{i,1}e_D\}$$

Thus, the transition of ${}^i\mathbf{G}_R$ from ${}^iq_{R,2}$ to ${}^iq_{R,1}$ is feasible through ${}^{i,1}e_D$. If ${}^{i,1}\mathbf{G}_c$ is not in $({}^{i,1}q_1, {}^{i,1}q_{S,1})$, i.e., it is in a state where the event ${}^{i,i-1}e_{BP}$ is not an active event, then no transition of the synchronous product ${}^i\mathbf{G}_R||{}^{i,1}\mathbf{G}_c||{}^i\mathbf{S}_R$ will take place through the event ${}^{i,i-1}e_{BP}$, and consequently ${}^i\mathbf{G}_R$ will be in the marked state ${}^iq_{R,1}$.

Case 4: the transition from the initial state ${}^iq_{R,1}$ to ${}^iq_{R,2}$ through the event ${}^{i,i}e_{BP}$ is investigated. Regarding ${}^i\mathbf{G}_R||{}^i\mathbf{S}_R$, it is observed that if the event ${}^{i,i}e_{BP}$ takes place, then ${}^i\mathbf{G}_R$ arrives at ${}^iq_{R,2}$ and ${}^i\mathbf{S}_R$ arrives at ${}^iq_{R,S,4}$. Note that ${}^i\mathbb{H}_{R,S}({}^iq_{R,S,4}) = \{{}^{i,2}e_D\}$; thus, it holds that ${}^i\mathbb{H}_{R||S}({}^iq_{R,2}, {}^iq_{R,S,4}) = \{{}^{i,2}e_D\}$. Regarding the rest supervisors, i.e., the supervisors ${}^i\mathbf{S}$, ${}^i\mathbf{S}_B$, ${}^{i,1}\mathbf{S}_D$, and ${}^{i,2}\mathbf{S}_D$, it is observed that their alphabets do not contain the event ${}^{i,2}e_D$. Regarding the automata ${}^{i,1}\mathbf{G}_c$, ${}^{i-1}\tilde{\mathbf{G}}$, and ${}^{i-1}\tilde{\mathbf{G}}$, it is observed that their alphabets do not contain the event ${}^{i,2}e_D$. Thus, these supervisors and the automata ${}^{i,1}\mathbf{G}_c$, ${}^{i-1}\tilde{\mathbf{G}}$, and ${}^{i-1}\tilde{\mathbf{G}}$ do not restrict the transitions of the synchronous product in (49a) through the event ${}^{i,2}e_D$. Regarding ${}^{i,2}\mathbf{G}_c$, it is mentioned that its alphabet contains the event ${}^{i,2}e_D$. From Figure 2,

it is observed that if ${}^{i,2}\mathbf{G}_c$ is in $({}^{i,2}q_1, {}^{i,2}q_{S,1})$, then if the event ${}^{i,i}e_{BP}$ takes place, then ${}^{i,2}\mathbf{G}_c$ stays at $({}^{i,2}q_1, {}^{i,2}q_{S,1})$. Clearly, it holds that ${}^{i,2}\mathbb{H}_c(({}^{i,2}q_1, {}^{i,2}q_{S,1})) = \{{}^{i,2}e_D\}$. Regarding ${}^i\mathbf{G}_R || {}^{i,2}\mathbf{G}_c || {}^i\mathbf{S}_R$, it holds that

$${}^i\mathbb{H}_c\left({}^iq_{R,2}, {}^iq_{R,S,4}, ({}^{i,2}q_1, {}^{i,2}q_{S,1})\right) = {}^i\mathbb{H}_{R||S}\left({}^iq_{R,2}, {}^iq_{R,S,4}\right) \cap {}^{i,2}\mathbb{H}_c\left({}^{i,2}q_1, {}^{i,2}q_{S,1}\right) = \{{}^{i,2}e_D\}$$

Thus, the transition of ${}^i\mathbf{G}_R$ from ${}^iq_{R,2}$ to ${}^iq_{R,1}$ is feasible through ${}^{i,2}e_D$. If ${}^{i,2}\mathbf{G}_c$ is not in $({}^{i,2}q_1, {}^{i,2}q_{S,1})$, respectively, i.e., it is in a state where the event ${}^{i,i}e_{BP}$ is not an active event, then no transition of the synchronous product ${}^i\mathbf{G}_R || {}^{i,2}\mathbf{G}_c || {}^i\mathbf{S}_R$ will take place through the event ${}^{i,i}e_{BP}$, and consequently ${}^i\mathbf{G}_R$ will be in the marked state ${}^iq_{R,1}$.

From the above four cases, it is concluded that if a pick event takes place, then if the transition through a pick event is allowable by the synchronous product in (49a), then the automaton ${}^i\mathbf{G}_R$ will arrive at the non-marked state ${}^iq_{R,2}$. The return from ${}^iq_{R,2}$ to ${}^iq_{R,1}$ is always allowable through a drop event. Hence, ${}^i\tilde{\mathbf{G}}_c$ is a nonblocking automaton. $\square$

Appendix B

Proof of Theorem 4. Any state of ${}^{n+1}\tilde{\mathbf{G}}_c$ is of the following $2m + 7$ -tuple form

$$({}^{n+1}q_R, ({}^{n+1,1}q, {}^{n+1,1}q_S), \dots, ({}^{n+1,m}q, {}^{n+1,m}q_S), {}^nq, {}^{n+1}q_{R,S}, {}^{n+1}q_{B,S}, {}^{n+1,2}q_{D,S}) \in {}^{n+1}\mathbb{Q}_R \times ({}^{n+1,1}\mathbb{Q} \times {}^{n+1,1}\mathbb{Q}_S) \times \dots \times ({}^{n+1,m}\mathbb{Q} \times {}^{n+1,m}\mathbb{Q}_S) \times {}^n\mathbb{Q}_m \times {}^{n+1}\mathbb{Q}_{R,S} \times {}^{n+1}\mathbb{Q}_{B,S} \times {}^{n+1,2}\mathbb{Q}_{D,S}$$

It is important to mention that all states of the supervisors are marked states. According to Lemma 1 and Lemma 2, it is mentioned that all states of the automata ${}^{n+1,1}\mathbf{G}_c$ to ${}^{n+1,m}\mathbf{G}_c$ and ${}^n\tilde{\mathbf{G}}$ are marked. Thus, the characteristic of a state of ${}^{n+1}\tilde{\mathbf{G}}_c$ to be or not be a marked state depends entirely upon the state of ${}^{n+1}\mathbf{G}_R$. Hence, for ${}^{n+1}\tilde{\mathbf{G}}_c$ to be a nonblocking automaton, it suffices to prove that

$${}^{n+1}\mathbb{H}_c\left({}^{n+1}q_R, ({}^{n+1,1}q, {}^{n+1,1}q_S), \dots, ({}^{n+1,m}q, {}^{n+1,m}q_S), {}^nq, {}^{n+1}q_{R,S}, {}^{n+1}q_{B,S}, {}^{n+1,2}q_{D,S}\right) \cap {}^{n+1}\mathbb{H}_R({}^{n+1}q_{R,2}) \neq \emptyset$$

where ${}^{n+1}\mathbb{H}_c(\cdot)$ is the active event set per state of ${}^{n+1}\tilde{\mathbf{G}}_c$, and ${}^{n+1}\mathbb{H}_R({}^{n+1}q_{R,2}) = \{{}^{n+1,1}e_D, {}^{n+1,2}e_D, \dots, {}^{n+1,m}e_D, {}^{n+1,n}e_{BD}\}$. In other words, it suffices to prove that when the robotic manipulator picks a product, i.e., the automaton ${}^{n+1}\mathbf{G}_R$ is in the state ${}^{n+1}q_{R,2}$, then a drop event is always in the active event set of the respective state of ${}^{n+1}\tilde{\mathbf{G}}_c$, thus allowing the robotic manipulator to drop the product and consequently the automaton ${}^{n+1}\mathbf{G}_R$ to return to the marked state ${}^iq_{R,1}$. The alphabet of ${}^{n+1}\mathbf{S}_R$ is ${}^{n+1}\mathbb{E}_R$. Thus, ${}^{n+1}\mathbb{H}_{R||S}(({}^{n+1}q_R, {}^{n+1}q_{R,S})) = {}^{n+1}\mathbb{H}_R({}^{n+1}q_R) \cap {}^{n+1}\mathbb{H}_{R,S}({}^{n+1}q_{R,S})$, where ${}^{n+1}\mathbb{H}_{R||S}$ is the active event set per state of ${}^{n+1}\mathbf{G}_R || {}^{n+1}\mathbf{S}_R$. Since ${}^{n+1}\mathbb{H}_{R,S}({}^{n+1}q_{R,S}) \subseteq {}^{n+1}\mathbb{H}_R({}^{n+1}q_R)$, it is observed that ${}^{n+1}\mathbb{H}_{R||S}(({}^{n+1}q_R, {}^{n+1}q_{R,S})) = {}^{n+1}\mathbb{H}_{R,S}({}^{n+1}q_{R,S})$. The transition of ${}^{n+1}\mathbf{G}_R$ from ${}^{n+1}q_{R,1}$ to ${}^{n+1}q_{R,2}$ can be accomplished only through a pick event, i.e., an event of the set $\{{}^{n+1,n}e_{BP}, {}^{n+1,1}e_P, {}^{n+1,2}e_P, \dots, {}^{n+1,m}e_P\}$. In what follows, the restrictions of the supervisors in (49b) to all possible transitions from ${}^{n+1}q_{R,2}$ to ${}^{n+1}q_{R,1}$ in the synchronous product (49b) will be investigated by considering all possible cases for the arrival from ${}^{n+1}q_{R,1}$ to ${}^{n+1}q_{R,2}$. The number of the cases is five. Taking into account the accessible states of the synchronous product in (49b), it is observed that if ${}^{n+1}\mathbf{G}_R$ is in its initial state ${}^{n+1}q_{R,1}$, then ${}^{n+1}\mathbf{S}_R$ is also in its initial state ${}^{n+1}q_{R,S,1}$. Thus, the transition from ${}^{n+1}q_{R,1}$ to ${}^{n+1}q_{R,2}$, through a pick event, corresponds to the transition of ${}^{n+1}\mathbf{S}_R$ from ${}^{n+1}q_{R,S,1}$ to one of each m other states.

Case 1: The transition from the initial state ${}^{n+1}q_{R,1}$ to ${}^{n+1}q_{R,2}$ through the event ${}^{n+1,1}e_P$ is investigated. Regarding ${}^{n+1}\mathbf{G}_R || {}^{n+1}\mathbf{S}_R$, it is observed that if the event ${}^{n+1,1}e_P$ takes place, then ${}^{n+1}\mathbf{G}_R$ arrives at ${}^{n+1}q_{R,2}$ and ${}^{n+1}\mathbf{S}_R$ arrives at ${}^{n+1}q_{R,S,3}$. Note that ${}^{n+1}\mathbb{H}_{R,S}({}^{n+1}q_{R,S,3}) = \{{}^{n+1,3}e_D\}$; thus, it holds that ${}^{n+1}\mathbb{H}_{R||S}(({}^{n+1}q_{R,2}, {}^{n+1}q_{R,S,3})) = \{{}^{n+1,3}e_D\}$. Regarding the supervisors ${}^{n+1}\mathbf{S}_B$ and ${}^{n+1,2}\mathbf{S}_D$, it is observed that their alphabets do not contain the event ${}^{n+1,3}e_D$. Regarding the automata ${}^{n+1,1}\mathbf{G}_c$, ${}^{n+1,3}\mathbf{G}_c$ to ${}^{n+1,m}\mathbf{G}_c$, and ${}^n\tilde{\mathbf{G}}$, it is observed that

their alphabets do not contain the event $^{n+1,3}e_D$. Thus, the supervisor $^{n+1}\mathbf{S}_B$ and $^{n+1,2}\mathbf{S}_D$, as well as the automata $^{n+1,1}\mathbf{G}_c, \dots, ^{n+1,m}\mathbf{G}_c$, and $^n\tilde{\mathbf{G}}$, do not restrict the transitions of the synchronous product in (49b) through the event $^{n+1,3}e_D$. Regarding $^{n+1,2}\mathbf{G}_c$, it is observed that its alphabet contains the event $^{n+1,3}e_D$. From Figure 2, it is observed that if $^{n+1,2}\mathbf{G}_c$ is in $(^{n+1,2}q_1, ^{n+1,2}q_{S,1})$ and the event $^{n+1,1}e_P$ takes place, then $^{n+1,2}\mathbf{G}_c$ stays at $(^{n+1,2}q_1, ^{n+1,2}q_{S,1})$. Clearly, it holds that $^{n+1,2}\mathbb{H}_c((^{n+1,2}q_1, ^{n+1,2}q_{S,1})) = \{^{n+1,3}e_D\}$. Regarding $^{n+1}\mathbf{G}_R || ^{n+1}\mathbf{S}_R || ^{n+1,2}\mathbf{G}_c$, it holds that

$$^{n+1}\mathbb{H}_c\left(^{n+1}q_{R,2}, ^{n+1}q_{R,S,3}, (^{n+1,2}q_1, ^{n+1,2}q_{S,1})\right) = ^{n+1}\mathbb{H}_{R||S}\left((^{n+1}q_{R,2}, ^{n+1}q_{R,S,3})\right) \cap ^{n+1,2}\mathbb{H}_c\left((^{n+1,2}q_1, ^{n+1,2}q_{S,1})\right) = \{^{n+1,3}e_D\}$$

Hence, the transition of $^{n+1}\mathbf{G}_R$ from $^{n+1}q_{R,2}$ to $^{n+1}q_{R,1}$ is feasible through $^{n+1,3}e_D$. If $^{n+1,2}\mathbf{G}_c$ is not in $(^{n+1,2}q_1, ^{n+1,2}q_{S,1})$, then it will be in a state where the event $^{n+1,1}e_P$ is not an active event and consequently no transition of the synchronous product $^{n+1}\mathbf{G}_R || ^{n+1}\mathbf{S}_R || ^{n+1,2}\mathbf{G}_c$ will take place through the event $^{n+1,1}e_P$. Subsequently the automaton $^{n+1}\mathbf{G}_R$ will be in the marked state $^{n+1}q_{R,1}$.

Case 2: The transition from the initial state $^{n+1}q_{R,1}$ to $^{n+1}q_{R,2}$ through the event $^{n+1,2}e_P$ is investigated. Regarding $^{n+1}\mathbf{G}_R || ^{n+1}\mathbf{S}_R$, it is observed that if the event $^{n+1,2}e_P$ takes place, then $^{n+1}\mathbf{G}_R$ arrives at $^{n+1}q_{R,2}$ and $^{n+1}\mathbf{S}_R$ arrives at $^{n+1}q_{R,S,m+2}$. Note that $^{n+1}\mathbb{H}_{R,S}(^{n+1}q_{R,S,m+2}) = \{^{n+1,n}e_{BD}\}$; thus, $^{n+1}\mathbb{H}_{R||S}((^{n+1}q_{R,2}, ^{n+1}q_{R,S,m+2})) = \{^{n+1,n}e_{BD}\}$. Regarding supervisor $^{n+1,2}\mathbf{S}_D$, it is observed that its alphabet does not contain the event $^{n+1,n}e_{BD}$. Regarding the automata $^{n+1,1}\mathbf{G}_c$ to $^{n+1,m}\mathbf{G}_c$, it is observed that their alphabets do not contain the event $^{n+1,n}e_{BD}$. Thus, supervisor $^{n+1,2}\mathbf{S}_D$ and the automata $^{n+1,1}\mathbf{G}_c, \dots, ^{n+1,m}\mathbf{G}_c$ do not restrict the transitions of the synchronous product in (49b) through the event $^{n+1,n}e_{BD}$. From Figure 13, it is observed that if $^{n+1}\mathbf{S}_B$ is in $^nq_{S,B,1}$ and the event $^{n+1,2}e_P$ takes place, then the supervisor $^{n+1}\mathbf{S}_B$ arrives at $^{n+1}q_{B,S,2}$. Clearly, it holds that $^{n+1}\mathbb{H}_{B,S}(^{n+1}q_{B,S,2}) = \{^{n+1,n}e_{BD}\}$. Regarding $^n\tilde{\mathbf{G}}$, it is observed that its alphabet contains the event $^{n+1,n}e_{BD}$. From Figure 3, it is observed that if $^n\tilde{\mathbf{G}}$ is in $^nq_1$ and the event $^{n+1,2}e_P$ takes place, then $^n\tilde{\mathbf{G}}$ will stay in $^nq_1$. Clearly, it holds that $^n\mathbb{H}(^nq_1) = \{^{n+1,n}e_{BD}\}$. Regarding $^{n+1}\mathbf{G}_R || ^n\tilde{\mathbf{G}} || ^{n+1}\mathbf{S}_R || ^{n+1}\mathbf{S}_B$, it holds that

$$^{n+1}\mathbb{H}_c\left(^{n+1}q_{R,2}, ^{n+1}q_{R,S,m+2}, ^nq_1, ^{n+1}q_{B,S,2}\right) = ^{n+1}\mathbb{H}_{R||S}\left(^{n+1}q_{R,2}, ^{n+1}q_{R,S,m+2}\right) \cap ^n\mathbb{H}(^nq_1) \cap ^{n+1}\mathbb{H}_B(^{n+1}q_{B,S,2}) = \{^{n+1,n}e_{BD}\}$$

Hence, the transition of $^{n+1}\mathbf{G}_R$ from $^{n+1}q_{R,2}$ to $^{n+1}q_{R,1}$ is feasible through $^{n+1,n}e_{BD}$. If $^n\tilde{\mathbf{G}}$ is not in $^nq_1$ or $^{n+1}\mathbf{S}_B$ is not in $^{n+1}q_{B,S,1}$, both automata are in states where the event $^{n+1,2}e_P$ is not active and consequently no transition of the synchronous product $^{n+1}\mathbf{G}_R || ^n\tilde{\mathbf{G}} || ^{n+1}\mathbf{S}_R || ^{n+1}\mathbf{S}_B$ will take place through $^{n+1,2}e_P$. Subsequently $^{n+1}\mathbf{G}_R$ will be in the marked state $^{n+1}q_{R,1}$.

Case 3: The transition from the initial state $^{n+1}q_{R,1}$ to $^{n+1}q_{R,2}$ through $^{n+1,j}e_P$, where $j \in \{3, \dots, m-1\}$, is investigated. Regarding $^{n+1}\mathbf{G}_R || ^{n+1}\mathbf{S}_R$, it is observed that if $^{n+1,j}e_P$ takes place, then $^{n+1}\mathbf{G}_R$ arrives at $^{n+1}q_{R,2}$ and $^{n+1}\mathbf{S}_R$ arrives at $^{n+1}q_{R,S,j+1}$. Note that $^{n+1}\mathbb{H}_{R,S}(^{n+1}q_{R,S,j+1}) = \{^{n+1,j+1}e_D\}$. Hence, $^{n+1}\mathbb{H}_{R,R}(^{n+1}q_{R,2}, ^{n+1}q_{R,S,j+1}) = \{^{n+1,j+1}e_D\}$. Regarding the supervisors $^{n+1}\mathbf{S}_B$ and $^{n+1,2}\mathbf{S}_D$, it is observed that their alphabets do not contain the event $^{n+1,j+1}e_D$. Regarding the automata $^{n+1,1}\mathbf{G}_c$ to $^{n+1,j-1}\mathbf{G}_c$, $^{n+1,j+1}\mathbf{G}_c$ to $^{n+1,m}\mathbf{G}_c$, and $^n\tilde{\mathbf{G}}$, it is observed that their alphabets do not contain the event $^{n+1,j+1}e_D$. Thus, the supervisors $^{n+1}\mathbf{S}_B$ and $^{n+1,2}\mathbf{S}_D$, as well as the automata $^{n+1,1}\mathbf{G}_c, \dots, ^{n+1,j-1}\mathbf{G}_c, ^{n+1,j+1}\mathbf{G}_c, \dots, ^{n+1,m}\mathbf{G}_c$, and $^n\tilde{\mathbf{G}}$, do not restrict the transitions of the synchronous product in (49b) through the event $^{n+1,j+1}e_D$. Regarding $^{n+1,j}\mathbf{G}_c$, it is mentioned that its alphabet contains the event $^{n+1,j+1}e_D$. From Figure 2, it is observed that if $^{n+1,j}\mathbf{G}_c$ is in $(^{n+1,j}q_1, ^{n+1,j}q_{S,1})$ and the event $^{n+1,j}e_P$ takes place, then $^{n+1,j}\mathbf{G}_c$ will stay at $(^{n+1,j}q_1, ^{n+1,j}q_{S,1})$. Clearly, it holds that $^{n+1,j}\mathbb{H}_c((^{n+1,j}q_1, ^{n+1,j}q_{S,1})) = \{^{n+1,j+1}e_D\}$. Regarding $^{n+1}\mathbf{G}_R || ^{n+1}\mathbf{S}_R || ^{n+1,j}\mathbf{G}_c$, it holds that

$$^{n+1}\mathbb{H}_c\left(^{n+1}q_{R,2}, ^{n+1}q_{R,S,j+1}, (^{n+1,j}q_1, ^{n+1,j}q_{S,1})\right) = ^{n+1}\mathbb{H}_{R||S}\left(^{n+1}q_{R,2}, ^{n+1}q_{R,S,j+1}\right) \cap ^{n+1,j}\mathbb{H}_c\left((^{n+1,j}q_1, ^{n+1,j}q_{S,1})\right) = \{^{n+1,j+1}e_D\}$$

Thus, the transition of $^{n+1,j}G_c$ from $^{n+1}q_{R,2}$ to $^{n+1}q_{R,1}$ is feasible through $^{n+1,j+1}e_D$. If $^{n+1,j}G_c$ is not in $(^{n+1,j}q_1, ^{n+1,j}q_{S,1})$, then $^{n+1,j}G_c$ is in a state where the event $^{n+1,j}e_P$ is not active and no transition of the synchronous product $^{n+1}G_R || ^{n+1}S_R || ^{n+1,j}S$ will take place through the event $^{n+1,j}e_P$, and consequently $^{n+1}G_R$ will be in the marked state $^{n+1}q_{R,1}$.

Case 4: The transition from the initial state $^{n+1}q_{R,1}$ to $^{n+1}q_{R,2}$ through the event $^{n+1,m}e_P$ is investigated. Regarding $^{n+1}G_R || ^{n+1}S_R$, it is observed that if the event $^{n+1,m}e_P$ takes place, then $^{n+1}G_R$ arrives at $^{n+1}q_{R,2}$ and $^{n+1}S_R$ arrives at $^{n+1}q_{R,S,m+1}$. Note that $^{n+1}\mathbb{H}_{R,S}(^{n+1}q_{R,S,m+1}) = \{^{n+1,2}e_D\}$. Hence, $^{n+1}\mathbb{H}_{R,R}(^{n+1}q_{R,2}, ^{n+1}q_{R,S,m+1}) = \{^{n+1,2}e_D\}$. Regarding the supervisors $^{n+1}S_B$ and $^{n+1,2}S_D$, it is observed that their alphabets do not contain the event $^{n+1,2}e_D$. Regarding the automata $^{n+1,1}G_c, \dots, ^{n+1,m-1}G_c$, and $^n\tilde{G}$, it is observed that their alphabets do not contain the event $^{n+1,2}e_D$. Thus, the supervisors $^{n+1}S_B$ and $^{n+1,2}S_D$, as well as the automata $^{n+1,1}G_c, \dots, ^{n+1,m-1}G_c$, and $^n\tilde{G}$, do not restrict the transitions of the synchronous product in (49b) through the event $^{n+1,2}e_D$. Regarding $^{n+1,m}G_c$, it is mentioned that its alphabet contains $^{n+1,2}e_D$. From Figure 2, it is observed that if $^{n+1,m}G_c$ is in $(^{n+1,m}q_1, ^{n+1,m}q_{S,1})$ and the event $^{n+1,m}e_P$ takes place, then the supervisor $^{n+1,m}G_c$ will stay at $(^{n+1,m}q_1, ^{n+1,m}q_{S,1})$. Clearly, it holds that $^{n+1,m}\mathbb{H}_c((^{n+1,m}q_1, ^{n+1,m}q_{S,1})) = \{^{n+1,2}e_D\}$. Regarding $^{n+1}G_R || ^{n+1}S_R || ^{n+1,m}G_c$, it holds that

$$^{n+1}\mathbb{H}_c(^{n+1}q_{R,2}, ^{n+1}q_{R,S,m+1}, (^{n+1,m}q_1, ^{n+1,m}q_{S,1})) = ^{n+1}\mathbb{H}_{R||S}((^{n+1}q_{R,2}, ^{n+1}q_{R,S,m+1})) \cap ^{n+1,m}\mathbb{H}_c((^{n+1,m}q_1, ^{n+1,m}q_{S,1})) = \{^{n+1,2}e_D\}$$

Hence, the transition of $^{n+1}G_R$ from $^{n+1}q_{R,2}$ to $^{n+1}q_{R,1}$ is feasible through $^{n+1,2}e_D$. If $^{n+1,m}G_c$ is not in $(^{n+1,m}q_1, ^{n+1,m}q_{S,1})$, then it is in a state where the event $^{n+1,m}e_P$ is not an active event and consequently no transition of $^{n+1}G_R || ^{n+1}S_R || ^{n+1,m}G_c$ will take place through the event $^{n+1,m}e_P$. Subsequently, $^{n+1}G_R$ will be in the marked state $^{n+1}q_{R,1}$.

Case 5: The transition from the initial state $^{n+1}q_{R,1}$ to $^{n+1}q_{R,2}$ through $^{n+1,n}e_{BP}$ is investigated. Regarding $^{n+1}G_R || ^{n+1}S_R$, it is observed that if $^{n+1,n}e_{BP}$ takes place, then $^{n+1}G_R$ arrives at $^{n+1}q_{R,2}$ and $^{n+1}S_R$ arrives at $^{n+1}q_{R,S,2}$. Note that $^{n+1}\mathbb{H}_{R,S}(^{n+1}q_{R,S,2}) = \{^{n+1,1}e_D\}$. Hence, $^{n+1}\mathbb{H}_{R||S}((^{n+1}q_{R,2}, ^{n+1}q_{R,S,2})) = \{^{n+1,1}e_D\}$. Regarding the supervisors $^{n+1}S_B$ and $^{n+1,2}S_D$, it is observed that their alphabets do not contain the event $^{n+1,1}e_D$. Regarding the automata $^{n+1,2}G_c$ to $^{n+1,m}G_c$ and $^n\tilde{G}$, it is observed that their alphabets do not contain the event $^{n+1,1}e_D$. Thus, the supervisors $^{n+1}S_B$ and $^{n+1,2}S_D$, as well as the automata $^{n+1,1}G_c$ to $^{n+1,m}G_c$ and $^n\tilde{G}$, do not restrict the transitions of the synchronous product in (49b), through the event $^{n+1,1}e_D$. Regarding $^{n+1,1}G_c$, it is mentioned that its alphabet contains the event $^{n+1,1}e_D$. From Figure 2, it is observed that if $^{n+1,1}G_c$ is in $(^{n+1,1}q_1, ^{n+1,1}q_{S,1})$ and the event $^{n+1,n}e_{BP}$ takes place, then $^{n+1,1}G_c$ will stay at $(^{n+1,1}q_1, ^{n+1,1}q_{S,1})$. Clearly, it holds that $^{n+1,1}\mathbb{H}_c((^{n+1,1}q_1, ^{n+1,1}q_{S,1})) = \{^{n+1,1}e_D\}$. Regarding $^{n+1}G_R || ^{n+1}S_R || ^{n+1,1}G_c$, it holds that

$$^{n+1}\mathbb{H}_c(^{n+1}q_{R,2}, ^{n+1}q_{R,S,2}, (^{n+1,1}q_1, ^{n+1,1}q_{S,1})) = ^{n+1}\mathbb{H}_{R||S}((^{n+1}q_{R,2}, ^{n+1}q_{R,S,2})) \cap ^{n+1,1}\mathbb{H}_c((^{n+1,1}q_1, ^{n+1,1}q_{S,1})) = \{^{n+1,1}e_D\}$$

Hence, the transition of $^{n+1}G_R$ from $^{n+1}q_{R,2}$ to $^{n+1}q_{R,1}$ is feasible through $^{n+1,1}e_D$. If $^{n+1,1}G_c$ is not in $(^{n+1,1}q_1, ^{n+1,1}q_{S,1})$, then it is in a state where the event $^{n+1,n}e_{BP}$ is not an active event and consequently no transition of the synchronous product $^{n+1}G_R || ^{n+1}S_R || ^{n+1,1}G_c$ will take place through the event $^{n+1,n}e_{BP}$. Subsequently, $^{n+1}G_R$ will be in the marked state $^{n+1}q_{R,1}$.

From the above five cases it is concluded that if a pick event takes place, then if the transition through a pick event is allowable by the synchronous product in (49b), then the automaton $^{n+1}G_R$ will arrive at the non-marked state $^{n+1}q_{R,2}$. The return from $^{n+1}q_{R,2}$ to $^{n+1}q_{R,1}$ is always allowable through a drop event. Hence, $^{n+1}\tilde{G}_c$ is a nonblocking automaton. $\square$

References

1. Scheiber, M.; Nartey, C.; Zernig, A.; Kaestner, A. Improving wafer map classification in Industry 4.0. In Proceedings of the IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS), Coventry, UK, 24–26 May 2022. [\[CrossRef\]](#)
2. Yi, J.; Ding, S.; Zhang, M.T.; van der Meulen, P. Throughput analysis of linear cluster tools. In Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering (CASE), Scottsdale, AZ, USA, 22–25 September 2007; pp. 1063–1068. [\[CrossRef\]](#)
3. Su, R.; van Schuppen, J.H.; Rooda, J.E. Aggregative Synthesis of Distributed Supervisors Based on Automaton Abstraction. *IEEE Trans. Autom. Control* **2010**, *55*, 1627–1640. [\[CrossRef\]](#)
4. Su, R.; van Schuppen, J.H.; Rooda, J.E. Maximally permissive coordinated distributed supervisory control of nondeterministic discrete-event systems. *Automatica* **2012**, *48*, 1237–1247. [\[CrossRef\]](#)
5. Su, R.; van Schuppen, J.H.; Rooda, J.E. The Synthesis of Time Optimal Supervisors by Using Heaps-of-Pieces. *IEEE Trans. Autom. Control* **2012**, *57*, 105–118. [\[CrossRef\]](#)
6. Su, R. Coordinated distributed time optimal supervisory control. In Proceedings of the American Control Conference (ACC), Washington, DC, USA, 17–19 June 2013; pp. 905–910. [\[CrossRef\]](#)
7. Cai, K.; Wonham, W.M. New results on supervisor localization, with case studies. *Discret. Event Dyn. Syst.* **2015**, *25*, 203–226. [\[CrossRef\]](#)
8. Ware, S.; Su, R. Time Optimal Synthesis Based Upon Sequential Abstraction and Its Application to Cluster Tools. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 772–784. [\[CrossRef\]](#)
9. Ware, S.; Su, R. Time optimal synthesis based upon sequential abstraction and maximizing parallelism. In Proceedings of the 13th IEEE Conference on Automation Science and Engineering (CASE), Xi'an, China, 20–23 August 2017; pp. 926–931. [\[CrossRef\]](#)
10. Yi, J.; Ding, S.; Song, D.; Zhang, M.T. Steady-State Throughput and Scheduling Analysis of Multicluster Tools: A Decomposition Approach. *IEEE Trans. Autom. Sci. Eng.* **2008**, *5*, 321–336. [\[CrossRef\]](#)
11. Wu, N.; Yang, F.; Bai, L.; Zhou, M.; Li, Z. Multi Cluster Tool System and a Method of Controlling a Multi Tool Cluster System. U.S. Patent 10,520,914 B2, 31 December 2019.
12. Zhu, Q.; Wang, G.; Hou, Y.; Wu, N.; Qiao, Y. Optimally Scheduling Dual-Arm Multi-Cluster Tools to Process Two Wafer Types. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5920–5927. [\[CrossRef\]](#)
13. Tu, Y.-M. Short-Term Scheduling Model of Cluster Tool in Wafer Fabrication. *Mathematics* **2021**, *9*, 1029. [\[CrossRef\]](#)
14. Li, J.; Qiao, Y.; Zhang, S.; Li, Z.; Wu, N.; Song, T. Scheduling of Single-Arm Cluster Tools with Residency Time Constraints and Chamber Cleaning Operations. *Appl. Sci.* **2021**, *11*, 9193. [\[CrossRef\]](#)
15. Wonham, W.M.; Kai, C. *Supervisory Control of Discrete-Event Systems*; Springer: Cham, Switzerland, 2019. [\[CrossRef\]](#)
16. Cassandras, C.G.; Lafortune, S. *Introduction to Discrete Event Systems*, 3rd ed.; Springer: Cham, Switzerland, 2021. [\[CrossRef\]](#)
17. Koumboulis, F.N.; Fragkoulis, D.G.; Arapakis, S. Supervisor design for an assembly line in the presence of faults. In Proceedings of the 27th IEEE International Conference on Emerging Technology and Factory Automation, Stuttgart, Germany, 6–9 September 2022. [\[CrossRef\]](#)
18. Koumboulis, F.N.; Fragkoulis, D.G.; Menexis, A.N. Supervisory Control for Flexibility of Production Manufacturing Processes. In Proceedings of the IEEE 21st International Conference on Intelligent Engineering Systems (INES), Larnaca, Cyprus, 20–23 October 2017. [\[CrossRef\]](#)
19. Koumboulis, F.N.; Fragkoulis, D.G.; Ioannou, K.A. Control of Router Nodes in Production Manufacturing Processes. In Proceedings of the 2018 7th International Conference on Systems and Control (ICSC), Valencia, Spain, 24–26 October 2018. [\[CrossRef\]](#)
20. Koumboulis, F.N.; Fragkoulis, D.G.; Diveris, G.K. Function Supervisors for Storage Systems. In Proceedings of the International Conference on Modern Circuits and Systems Technologies (MOCAS), Thessaloniki, Greece, 7–9 May 2018. [\[CrossRef\]](#)
21. Koumboulis, F.N.; Fragkoulis, D.G.; Kalkanis, I.; Fragulis, G.F. Supervisor Design for a Pressurized Reactor Unit in the Presence of Sensor and Actuator Faults. *Electronics* **2022**, *11*, 2534. [\[CrossRef\]](#)
22. Kouvakas, N.D.; Koumboulis, F.N.; Fragkoulis, D.G.; Souliotis, A. Modular Supervisory Control for the Coordination of a Manufacturing Cell with Observable Faults. *Sensors* **2023**, *23*, 163. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Guo, L.; Vincentelli, A.S.; Pinto, A. A complexity metric for concurrent finite state machine based embedded software. In Proceedings of the 8th IEEE International Symposium on Industrial Embedded Systems (SIES), Porto, Portugal, 19–21 June 2013. [\[CrossRef\]](#)
24. Vieira, A.D.; Santos, E.A.P.; de Queiroz, M.H.; Leal, A.B.; de Paula Neto, A.D.; Cury, J.E.R. A Method for PLC Implementation of Supervisory Control of Discrete Event Systems. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 175–191. [\[CrossRef\]](#)
25. Szpak, R.; de Queiroz, M.H.; Cury, J.E.R. Synthesis and implementation of supervisory control for manufacturing systems under processing uncertainties and time constraints. *IFAC Pap.* **2020**, *53*, 229–234. [\[CrossRef\]](#)
26. Function Blocks IEC 61499 ed: 2, International Electrotechnical Commission. 2012. Available online: <https://webstore.iec.ch/publication/5507> (accessed on 4 March 2023).
27. Leitão, H.A.S.; Rosso, R.S.U.; Leal, A.B.; Zoitl, A. Fault Handling in Discrete Event Systems Applied to IEC 61499. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020. [\[CrossRef\]](#)

28. Programmable Controllers IEC 61131-Part 3: Programming Languages, International Electrotechnical Commission. 2013. Available online: <https://webstore.iec.ch/publication/4552> (accessed on 4 March 2023).
29. Morgan, J.; Halton, M.; Qiao, Y.; Breslin, J.G. Industry 4.0 smart reconfigurable manufacturing machines. *J. Manuf. Syst.* **2021**, *59*, 481–506. [[CrossRef](#)]
30. Trentesaux, D. Distributed control of production systems. *Eng. Appl. Artif. Intell.* **2009**, *22*, 971–978. [[CrossRef](#)]
31. Tang, H.; Li, D.; Wang, S.; Dong, Z. CASOA: An Architecture for Agent-Based Manufacturing System in the Context of Industry 4.0. *IEEE Access* **2018**, *6*, 12746–12754. [[CrossRef](#)]
32. Ioana, A.; Korodi, A. DDS and OPC UA Protocol Coexistence Solution in Real-Time and Industry 4.0 Context Using Non-Ideal Infrastructure. *Sensors* **2021**, *21*, 7760. [[CrossRef](#)] [[PubMed](#)]
33. Kučera, E.; Haffner, O.; Drahoš, P.; Cigánek, J. Educational Case Studies for Pilot Engineer 4.0 Programme: Monitoring and Control of Discrete-Event Systems Using OPC UA and Cloud Applications. *Appl. Sci.* **2022**, *12*, 8802. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.