*International Journal of Computer Applications (0975 – 8887)*
*Volume 119 – No.22, June 2015*

# Flexible Manufacturing System Design and Optimization using Petri Net-based Elementary Siphons

Mowafak Hassan Abdul-Hussin
Central Technical Information's & Communications,
University of Technology, Baghdad. Iraq.

## ABSTRACT

A computer simulation system, which is based-siphon Petri net for control of concurrent processes of the reusable resources, that is mainly aimed to provide guidelines for control and management engineering of flexible manufacturing systems (FMSs) is presented in this work. We present an important technique for the analysis of siphon Petri net (PN) based on structural analysis, control, and simulation of the reachability tree used PN-tool with MATLAB. We are representing the deadlock prevention problem, which can be solved using the concept of Petri nets based siphons. Structural deadlock prevention is presenting for supervisors a class of FMS in simulating the system behaviour and elementary siphons control rule's inference. We are recovering the satiation of the liveness system from the consequence of analysis net by adding control place (monitor). Our experiment, a few monitors is added to the net for every minimal siphon taken from the consequence of the siphon in order to be liveness (i.e. deadlock-freeness), proposing extensions of the structural analysis of Petri nets, where deadlocks are regarding to the unmarked siphons. The siphon is recognized in the structural analysis of PN and can be used to control deadlocks in resource systems modelled.

## Keywords

Petri net, Structural analysis, Minimal siphons, $S^3PR$, Reachability analysis, FMSs, Petri Net-Toolbox V. 2.3

## 1. INTRODUCTION

Recently a great amount of work has been devoted to the control and performance evaluation of flexible manufacturing systems (FMSs) is a perfectly automated multi-product running system that consists of a finite number of shared resources of each type (machines, pallets, automated loading and unloading of parts), which achieves optimal configuration. An automated guided vehicle system (AGVs) is moving parts between machines, robots, buffers, fixtures, and other automated elements allow unattended production of parts, which are linked together with a central computer system. An FMS allow a manufacturer to change quickly processes or operations to produce any product, at any time. Raw parts in an FMS can be processed concurrently, synchronization in the system with respect to time. A dynamic modelling tool should represent these aspects to analyze the conflicts and deadlock during the system execution. In addition to all these requirements, a dynamic modelling tool should support the system designer for system performance evaluation and assist control engineers to control and monitor the FMS. Petri nets has all these capabilities and hence be suitable for dynamic modelling tools for the various methods used for object models.

The deadlock problem can occur to the system when competitive processes lead to shared resources can cause deadlocks, where two or more processes need to access a resource during the course of their execution or more jobs keeps waiting indefinitely for the other jobs in the set to release resources [1, 2]. A deadlock occurs to an FMS when raw parts are blocked waiting for shared resources held by others that will never be granted. Petri Nets (PNs) constituted a good tool for the design and operation of many systems is very useful in the deadlock solution to an FMS. Deadlock prevention is considered to be one of the most effective methods of deadlock control, which is usually implemented by adding monitors for a net model to ensure that deadlocks never occur. A Special class of Petri net models siphon is defined that allows to capture resource allocation conditions used to synchronize processes that have to share a set of reusable system resources [8-13].

Petri nets have been excellent tool as one of the most powerful tools to describe and analyze the behavior of discrete event systems, including (FMS) [1, 3], because it can describe resource sharing, conflict, mutual exclusion, concurrency, synchronization among objects and performance analysis of FMS, and uncertainty successfully. Besides, due to its brief and normative presentation, Petri net has applied more broadly and developed furthermore, in modelling, analyzing, simulation and control of the manufacturing systems. In this paper, a deadlock prevention policy is proposed to a class of Petri nets called $S^3PR$ control of FMS. In addition, the structural analysis and reachability graph analysis is used for analysis and control of Petri nets.

While the new production requirement concentrated, is an encouraging approach to improve significantly the competitiveness of the manufacturing industry. The advanced manufacturing lead to an increasingly automated world that will continue to rely less on labor-intensive mechanical processes and more on sophisticated information-technology intensive processes that enable flexibility to developers. Thus, optimal control strategies are used a central computer can be controlled all machines, robots, transportation, and communication system. Petri net is an excellent tool to describe structural analysis, and control of the behavior of discrete event systems (DES), including FMS, such as concurrency, synchronization competitive processes, conflict, deadlock, and shared resources. For the structural object of Petri nets calls siphon is a mainly used in the development of Petri net theory of the control of FMS modelling to design of operation, control can be representation it.

The liveness of a Petri net is closely related to the existence of some special structural so-called siphons [3]. Also, liveness in Flexible Manufacturing Systems (FMS) modelled by ordinary Petri nets is closely related to emptiable siphons. Deadlock analysis and control techniques that are based on the structural theory of Petri nets aim at finding a relationship between liveness of the net and its structure. Deadlock analysis of Petri nets has been extensively investigated in the context of FMSs, and other technological applications involving a resource system [1-4]. Petri nets have been widely used to model a variety of resources including FMS.

Abdul-Hussin (2015) [1, 2] presents a structural analysis of Petri nets, where siphon is a main utility used in the development of Petri net theory to control of flexible manufacturing systems (FMSs) modelling, control and simulation, which has been exploited successfully for the design of supervisors of some supervisory control problems of FMS. In addition, the effective deadlock prevention policy of a special class of Petri nets namely $S^3PR$ and be shown the discriminating siphon can be solved deadlock prevention policy. In addition, Petri net models in the efficiency structure analysis and utilization of the FMSs when different policy can be implemented for the prevention of deadlock.

Abdul-Hussin (2014) [3, 4] proposed two techniques when analysis Petri nets models for deadlock prevention in FMSs, structure analysis to find the elementary siphons, and reachability graph analysis was used Petri net with MATLAB. Siphons and traps are analysis structures which allow for some implications on the net's can be well controlled by adding control places and related arcs for strict minimal siphons (SMS) of its Petri net model for each uncontrolled siphon in the net in order to become deadlock-free stations in the system. The structured analysis techniques and behavior in Petri nets is investigating the relationship between the behavior and structures of the net. In particular structural, a computer simulation system, and the Petri Net Toolbox in MATLAB [5] environment, which is based on analytical models of concurrent processes of the reusable resources that are aimed mainly at providing for control of a system is representative of this work.

Banaszak and Abdul-Hussin (1988) [6] proposed the deadlocks avoidance methods with a Petri net approach to the automatic design of control programs which are aimed at supervision of concurrent, pipeline-like flowing processes of FMS. They formulated investigation of the sufficient condition for the design of a class of deadlock-free Petri net. Petri nets have been proven to be convenient and multi-level applicable tools for the specification as well as for the verification of complex distributed industrial control systems. They provide not only a language for the design process, but also a theory backing this process. They had applied for the course of the formal investigation into the conditions necessary for the design of a net class respecting such basic dynamic properties as the buffer overflows and deadlock avoidance. The main results obtained allow for automatic conversion to a process specification, via a Petri net model of a control flow, into the relevant control program.

Banaszak and Abdul-Hussin (1999) [7] presents an approach towards constructing a class of Place/ Transition Petri nets for modelling, simulation and control of processes occurring in the Flexible Assembly Systems (FASs). The assembly and robots can perform their tasks asynchronously, some conflicts may occur. They present modelling of FASs by means of Petri nets has allowed the programmer to determine an algorithm transforming a given process specification into their simulation program. This program reflecting the structure of admissible controls involved in the processes accomplishment, can be able to serve as a control program for a system controller as well as a task-oriented package for the computer-assisted process planning of FASs.

The system of simple sequential processes with resources ($S^3PR$) has attracted many scholars' attention since it was proposed by Ezpeleta et al. [8]. It can model a class of FMS in which a set of different types of products can be manufactured concurrently and each step in one manufacturing process only needs one resource such as a machine or robot. While the competition among manufacturing processes of the limited resources, deadlocks can occur. One way policy, which is based on strict minimal siphons (SMSs), is proposed in [8] to prevent deadlocks.

Li and Zhou [9, 10] divide siphon into two kind's elementary and dependent siphons. They can control all SMS by controlling elementary SMS only, leading to a simple supervisor, and this is needed control place smaller than [8]. In addition, they are used a linear integer programming (LIP) to test must be carried out to decide the liveness of the controlled system. For all the deadlock prevention policies afterwards monitor added to SMS to enforce liveness, which is resulting in a deadlock-free Petri Net. This new method requires a much smaller number of control places. Although this paper explores the ways to minimize the new additions of places, while achieving the same controlling purpose, the control policy is similar to [8]. In this paper, an elementary siphon concept is used to reduce the number of control places. This paper further presents a new siphon-based policy of deadlock prevention for improving the complex Petri nets and their controlled nets permissive problems. Moreover, they are proposing a method to compute some SMS in a $S^3PR$ based on resource circuits. In their approach, for each resource circuit in the net, they compute its related strict minimal siphon. Their method can reduce the computational complexity, but it only computes the SMS generated by resource circuits and hence we can't compute all the SMS in a $S^3PR$ by their method.

Huang et al. (2002) [11] present an algorithm of deadlock prevention for the class Petri nets, where a new class of Petri nets called **extended** $S^3PR$ to ($ES^3PR$) for modeling, manufacturing systems where only parts can interact with resources, and resources alone cannot interact with one another. They proposed method is an iterative approach by adding two kinds of control places called the ordinary control place and weighted control place to the original model can be to prevent siphoning from being unmarked.

**Organization.** In section 2, briefly review preliminaries to Petri nets that are used in this paper. A method of computing all, the concept of elementary siphons in $S^3PR$ is developed in Section 3. Section 4, introduces the practical application to present an FMS example. Finally, section 5 conclude's this paper.

## 2. PRELIMINARIES [1]

**Definitions 1.**

A Petri net is a four-tuple $\Psi = (P, T, E, W)$, where P and T are a finite nonempty, and disjoint sets. P is the set of places, and T is the set of transitions. The set of $E \subseteq (P \times T) \cup (T \times P)$ is the flow relation or (a set of directed arcs). W: $(P \times T) \cup (T \times P) \longrightarrow Z^+$ is a weight function attached to the arcs, where $Z^+ \longrightarrow (1, 2, 3, \ldots, Z^+)$. A net $\Psi = (P, T, E, W)$ is ordinary, when weights W, of the arcs (W) = l, the net $\Psi$ is called ordinary Petri net. The weights W: $(P \times T) \cup (T \times P) \longrightarrow Z$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ if $(x, y) \in E$, and $W(x, y) = 0$ otherwise, where $x, y \in P \cup T$ and $Z = \{0, 1, 2, \ldots\}$ is a set of non-negative integers. A net is said to be ordinary if $\forall (x, y) \in E$, $W(x, y) = 1$.

**Definition 2.** A marking Petri net is $\Psi = (\Psi, M_0)$ where, $\Psi$ is a Petri net, and $M_0$: $P \longrightarrow \{1, 2, \ldots, Z^+\}$ is the net initial marking of $\Psi$, assigned to each place $p \in P$, $M_0(p)$ tokens, and where ($Z^+$ is a set of non-negative integers). Every P/T-net is provided with an initial marking $M_0$ that may change from results of the firing of a transition (or a marking Petri net is 5 – tuple: $\Psi = (P, T, E, W, M_0)$. Moreover, PN will be described either by the pair $\Psi = (\Psi, M_0)$, where $\Psi$ is a Petri net and $M_0$ is initial marking, or by 5 – tuple: $\Psi = (P, T, E, W, M_0)$. P is marked by M iff $M(p) > 0$. Tokens reside in the places of a Petri net. The number and position of tokens may change during the execution of a Petri net.

**Definition 3.** A net is pure (self-loop free) if $\forall (x, y) \in (P \times T) \cup (T \times P)$, $W(x, y) > 0$ implies $W(y, x) = 0$. Incidence matrix $[\Psi]$ of pure net $\Psi$ is a $|P| \times |T|$ integer matrix with $[\Psi](p, t) = W(t, p) - W(p, t)$. The pre-set (post-set) of a transition t is the set of all input (output) places of t, ${}^\bullet t = \{p \mid Pre(p, t) > 0\}$ and ($t^\bullet = \{p \mid Post(p, t) > 0\}$). The preset (post-set) of a place p is the set of all input (output) transitions of p, ${}^\bullet p = \{\forall t \in T, \mid E(t, p) \neq 0\}$, and ($p^\bullet = \{\forall t \in T, \mid E(p, t) \neq 0\}$). Suppose $x \in X$ is arbitrary elements of the net $\Psi$, ${}^\bullet x = \{y \mid (y, x) \in E\}$ is called pre-set of x, and $x^\bullet = \{y \mid (x, y) \in E\}$ is called post-set of x. The usefulness of the symbols is a pre-set and a post-set of a place $p \in P$ or a transition $t \in T$:

${}^\bullet t = \{p \in P \mid W(p, t) > 0\}$ is the set of input places of t. $t^\bullet = \{p \in P \mid W(p, t) > 0\}$is the set of output places of t. ${}^\bullet p = \{p \in P \mid W(p, t) > 0\}$is the set of input transitions of p. $p^\bullet = \{p \in P \mid W(p, t) > 0\}$ is the set of output transitions of p.

**Definition 4.** At the marking M, a transition t is enabled if $(\forall p \in {}^\bullet t)$, $M(p) \geq E(p, t)$. This fact is denoted as $M[t\rangle$. Firing an enabled transition t results in a new marking M1, which is obtained by removing E(p, t) tokens from each place $p \in {}^\bullet t)$, and placing E(t, p' ) tokens in each place $p' \in {}^\bullet t$ moving the system state from $M_0$ to M1. Repeating this process, it reaches M' by firing a sequence $\sigma = \{t_1, t_2, \dots, t_n\}$ of transitions. M' is said to be reachable from $M_0$; i. e, $[M_0[\sigma\rangle M']$. The firing sequence is a marking $(M_1, M_2, M_3, \dots, M_{n+1})$ such that: $(\forall i, 1 \leq i \leq n)$, and $(M_i\ [t_i\ \rangle M_{i+1})$, We can also write its by $[M_1[\sigma\rangle M_{n+1}]$. The set of all markings reachable from $M_0$ is denoted by Reachability set $R(M_0)$. The function $\sigma': T \longrightarrow Z^+$ is the firing count vector of the firable sequence $\sigma$, i.e. $\sigma'[t]$, presents the algebraic sum of all the occurrences of $t \in T$ in $\sigma$. If $M_0[\sigma\rangle M'$, then we can write in vector form $M' = M_0 + C.\ \sigma'$, which is referred to as the linear state equation of the Petri net.

**Definition 5.**
A transition $t \in T$ is enabled at M if and only if (iff): $(\forall p \in {}^\bullet t)\ (M(p) = 1)$ and $(\forall p \in t^\bullet)\ (M(p) = 0)$.

**Definition 6.** P-invariant (resp. T-invariant) of a net $\Psi = (\Psi, M_0)$ is a non-negative row integer $|P|$-vector x (resp., $|T|$-vector y) satisfying the equation $x^T.\ C = 0$, (resp., $C.\ y^T = 0$), where C is the incidence matrix of $\Psi$. A non-zero integer vector $y \neq 0$, (resp. $x \neq 0$).

**Definition 7.** Let S is a non-empty sub-set of places. $S \subseteq P$ is a siphon (trap) iff ${}^\bullet S \subseteq S^\bullet$ and trap ($Q^\bullet \subseteq {}^\bullet Q$). A marked trap can never be emptied. A siphon is said to be minimal iff contain no other siphons as its proper sub-sets. A minimal siphon is strict if it contains no marked trap. A siphon is said to be controlled in an ordinary net system iff it can never be emptied. A siphon S is said to be invariant- controlled by P-invariant I if $I^T.\ M_0 > 0$, and $\| I \|^+ \subseteq S$.

Example 1: Consider the Petri net of Fig. 1, show an example of a Petri net. The Petri net consists of the five places, and four transitions is shown in Fig. 1, which has the strict minimal siphons as: $S_1 = \{p_1, p_4\}$, $S_2 = \{p_2, p_5\}$, and the two minimal trap as $Q_1 = \{p_1, p_2, p_3\}$, $Q_2 = \{p_2, p_5\}$. Siphons are very usefulness in the analysis structure and control of deadlocks in a Petri net for FMS. The Petri net model in Fig. 1 is not live.

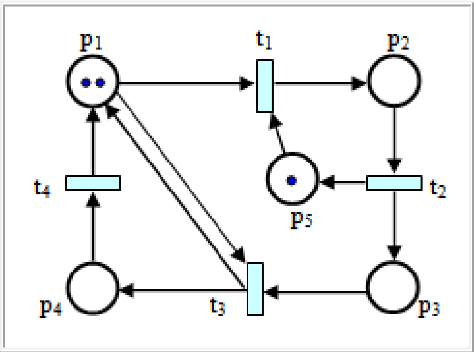


Fig. 1 shows a simple example of a Petri net.

**Definition 8.** Let Petri net $PN = (P, T, E, M_0)$, $n = |P|$ row , $m = |T|$ columns, be a PT-net. A matrix A of size (m x n) is said to be the incidence matrix of PN if:

$$C^- = \begin{cases} 1 & \text{if } p_j \in {}^\bullet t_i \\ 0 & \text{otherwise} \end{cases} \text{, and } C^+ = \begin{cases} 1 & \text{if } p_j \in t_i^\bullet \\ 0 & \text{otherwise} \end{cases}$$

A pair $C = (C^+ - C^-)$, $Pre = C^+ = [C^+_{ij}]_{m \times n}$, and $post = C^- = [C^-_{ij}]_{m \times n}$. We further define the pre incidence matrix:, and the post incidence matrix : where (${}^\bullet t = \{p|(p, t) \in E\}\{t^\bullet = \{p \mid (t, p) \in E\}$) is called a set of the input (respectively output) places of the transition t. We can be writing; $C[i] = (C^+[i], C^-[i])$ for the i-th row of matrix C as well as $C^+[i, j]$ for the j-th column in the i-th row of matrix $C^+$ (resp. $C^-$). The incidence matrix be $C = (C_{ij})$ which is defined by:
$C(p_i, t_j) = post(p_i, t_j) - pre(p_i, t_j)$, is the change in number of tokens in $p_i$ after firing $t_j$ once, for i = (1, 2, ... , n) and j = (1, 2, ..., m). We can be calculated T-vector as follows:

incidence matrix Example 1

| | | $t_1$ | $t_2$ | $t_3$ | $t_4$ |
|---|---|---|---|---|---|
| | $P_1$ | -1 | 0 | 0 | 1 |
| | $P_2$ | 1 | -1 | 0 | 0 |
| Cij = | $P_3$ | 0 | 1 | -1 | 0 |
| | $P_4$ | 0 | 0 | 1 | -1 |
| | $P_5$ | -1 | 1 | 0 | 0 |

Figure 2. The incidence matrix of Fig. 1

Let us comeback to the definition (4) (linear state equation) of an ordinary Petri net a transition t of an ordinary. Petri net is enabled if and only if each of its input places contains at least one token. In the example represented in Fig. 1, the initial marking is $M_0 = [2, 0, 0, 0, 1]^T$ and the following transitions can be fired starting from $M_0$.

Having applied definition (4), state equation $M' = M_0 + C.\sigma_t$ , and taking the column vector of firing transition from the incidence matrix on Fig. 2. The marking can be evolves as: $M_1 = [1,1,0,0,0]^T$ after the firing of $t_1$. Such that $M_1 = M_0$

+ C. $\sigma_{t1}$ = $M_0$ $[2,0,0,0,1]^T$ + column $t_1$ $[-1,1,0,0,-1]^T$ = [1,1,0,0,0] = $M_1$. Each of these new markings represents a node of the reachability tree and used next-state at firing transition.

In sequel, the next marking can be find by add last marking to the column of transition shown in Fig. 2, where transition $t_2$ is enable. The new marking $M_2 = M_1 + C.\sigma_{t2}$ = [1, 1, 0, 0, 0] + $[0,-1,1, 0, 1]^T$ = [1,0,1,0,1] = $M_2$. Then the firing $t_3$, $M_4 = M_2 + C.\sigma_{t3} = M_2$ $[1,0,1,0,1]^T$ + $t_3$ $[0,0,-1,1,0]^T$ = [1, 0, 0, 1,1] = $M_4$ in the right side in Fig. 3. In the rest, of our mathematical computation of the net of Fig. 1, the reachability tree of Petri net is shown in Fig. 3. The reachability tree can show deadlock in Petri net. In the Fig. 1, the deadlock is occurring at the marking: $M_5(t_2)$ = [0, 0, 2, 0,1] in red colored.

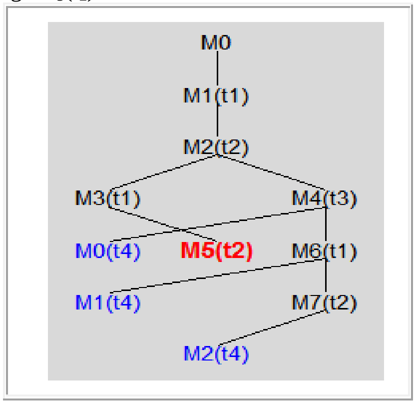


Fig. 3, the reachability tree of Petri net Fig. 1.

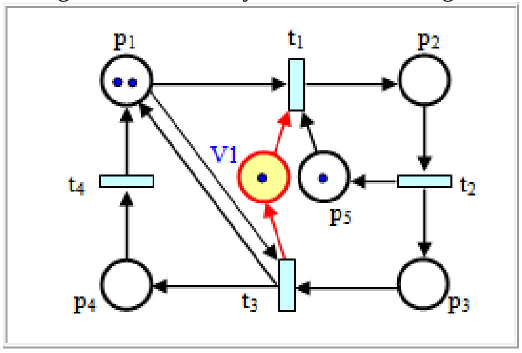


Fig. 4 control Petri net of Fig. 1.

The minimal siphon that is empty marking leads to the deadlock of the net. To control the net, we can prevent forms being unmarked, a place $V_1$ is added with $^\bullet V_1 = \{t_2\}$ and $V_1^\bullet = \{t_1\}$, as shown in Fig. 4, in order to the control is a Petri net. Additional place $V_1$, called control place in order to controlled to siphon (deadlock). We have predicted a deadlock with a look-ahead Petri net controller.

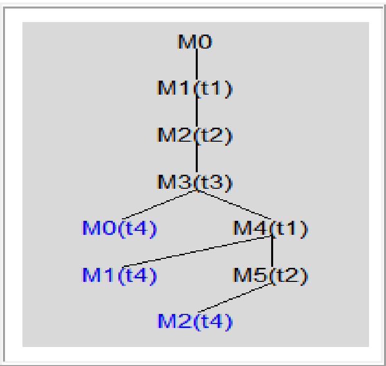


Fig. 5. The coverability tree is live of Fig. 4, simulation of Petri net in Toolbox with MATLAB [5].

After adding control place in the net, we can see that the Petri net liveness, and have three location return to the initial marking can be see in Fig. 5. The effecting controlled place in Petri net is shown in Fig. 3. It can verify that deadlock does not occur to this Petri net.

**Definition 9.** A P-vector is a column vector I: P $\rightarrow$ Z indexed by P, where Z is the set of integers. I is a P-invariant (place invariant) if and only if (iff) $I \neq 0$ and $I^T \bullet [\Psi] = 0^T$ holds. P-invariant I is said to be a P-semiflow if every element of I is non-negative. $\|I\|^+ = \{p \in P \mid I(p) \neq 0\}$ is called the support of I. If I is a P-invariant of $(\Psi, M_0)$ then $\forall M \in R(\Psi, M_0)$: $I^T \bullet M = I^T \bullet M_0$. In an ordinary net, siphon S is controlled by P-invariant I under $M_0$ if and only if $(I^T \bullet M_0 > 0)$ and $\{p \in P \mid I(p) > 0\} \subseteq S\}$. Such a siphon is called invariant-controlled siphon.

# 3. DEADLOCK PREVENT POLICY

Deadlock prevention policy is dealing with a special class of Petri nets, which is a subclass of ordinary and conservative Petri nets called S³PR. In this the section, we introduced some definitions have been needed for our application for a deadlock prevention policy which can be kept the system in liveness for a class of Petri nets, that is called $S^3PR$ nets.

## 3.1 The class of the $S^3PR$ nets

The class of Petri nets investigated in this research is an $S^3PR$ that is first proposed in Ezpeleta et al. [8]. Before the representation of its formal definition is needing for our application. The following results are mainly from [8].

**Definition 10.** A simple sequential process ($S^2P$) is a Petri net $\Psi = (P_A \cup \{p^0\}, T, E)$, where the following statements are true: (1) $P_A \neq \varnothing$ is called a set of operation places;
(2) $p^0 \notin P_A$ is called the process idle place;
(3) A net $\Psi$ is a strongly connected state machine;
(4) Every circuit of $\Psi$ contains place $p^0$.

**Definition 11.** A system of simple sequential processes with resources ($S^3PR$): $\Psi = O_{i-1}^k \Psi_i =$ $\Psi_i = (P_i \cup P_i^0 \cup P_R, T, E)$ is defined in [8] as the union of a set of nets: $\Psi_i = (P_i \cup \{P_i^0\} \cup P_R, T_i, E_i)$, sharing common places, where the following statements are true: (1) $P_i^0$ is called the process idle places of net $\Psi_i$. The elements in $P_A^i$ and $P_R^i$ are called operation places and resource places respectively.

(2) $p_A^i \neq \varnothing$; $p_R^i \neq \varnothing$; $p_0^i \notin p_A^i$; and $(p_A^i \cup \{p_0^i\} \cap p_R^i = \varnothing$; $\forall p \in p_A^i$, $\forall t \in {}^\bullet p$, $\forall t' \in p^\bullet$, $\exists r_p \in p_R^i$, ${}^\bullet t \cap p_R^i = t'^\bullet \cap p_R^i = \{r_p\}$; $\forall r \in p_r^i$, ${}^{\bullet\bullet}r \cap p_A^i = r^{\bullet\bullet} \cap p_A^i \neq \varnothing$, and ${}^\bullet r \cap r^\bullet = \varnothing$, ${}^{\bullet\bullet}(p_i^0) \cap (p_R^i) = (p_i^0)^{\bullet\bullet} \cap p_R^i = \varnothing$;

(3) $\Psi_i'$ is a strongly connected state machine, where $\Psi_i' = (p_A^i \cup \{p_0^i\}, T_i, E_i)$, is the resulting net after the places in $p_R^i$ and related arcs are removed from $\Psi_i$.

(4) Every circuit of $\Psi'_i$ contains place $P_i^0$;

(5) Any two $\Psi'_i$ are composable when they share a set of common places. Every shared place must be a resource.

(6) Transitions in $(p_i^0)^{\bullet}$ *and* ${}^{\bullet}(p_i^0)$ are called source and sink transitions of the net $\Psi$ respectively.

**Definition 12.** Let $\Psi_i = (P_A \cup P^0 \cup P_R, T, E)$, be an $S^3PR$. An initial marking $M_0$ is called an acceptable one if: 1) $\forall p \in P^0$, $M_0(p) \geq 1$; 2) $\forall p \in P_A$, $M_0(p) = 0$; and 3) $\forall p \in P_R$, $M_0(p) \geq 1$.

## 3.2 Elementary Siphons in Petri Nets

The concept of elementary and dependent siphons is original work by Li et al. [9, 10] and [12]. They are developed the Petri nets theory of computation and powerful mathematics. We have introduced the concept of elementary and dependent siphons as well as used in this paper.

**Definition 13.** Let S ⊆ P be a subset of places of Petri net Ψ = (P, T, F, W). P-vector $\lambda_S$ is called the characteristic P-vector of S iff $\forall$ p ∈ S, $\lambda_S$ (p) = 1; otherwise $\lambda_s$ (p) = 0.

**Definition 14.** Let S ⊆ P is a subset of places of Petri net Ψ. $\eta_S$ is characteristic T-Vector of S if and only if $\eta_S = \lambda_S^T \bullet [C]$, where C is incidence matrix of the net Ψ.

**Definition 15.**

Let Ψ = (P, T, F, W) be a net with |P| = m, |T| = n, and k siphons, $S_1, S_2, \ldots, S_k$, m, n, k ∈ $Z^+$. Let $\lambda_{S_i}$ $(\eta_{S_i})$ is be the characteristic P(T)-vector of siphon $S_i$, where i ∈ {1, 2, ... , k}. We define $[\lambda]_{k \times m} = [\lambda_{S1} | \lambda_{S2} | \ldots | \lambda_{Sk}]^T$, and $[\eta]_{k \times n} = [\lambda]_{k \times m} \times [C]_{m \times n} = [\eta_{S1} | \eta_{S2} | \ldots | \eta_{Sk}]^T$. Where $[\lambda]([\eta])$ is called the characteristic P(T)-vector matrix of the siphons in net Ψ.

**Definition 16.**

Let $\eta_{S\alpha}$, $\eta_{S\beta}$, … , and $\eta_{S\gamma}$ ({α, β, …., γ}⊆ $\Psi_k$) a linearly independent maximal set of matrix [η]. Then $\Pi_E = \{S_\alpha, S_\beta, \ldots, S_\gamma\}$ is called a set of elementary siphons in net Ψ.

**Definition 17.** Let S ∉ $\Pi_E$ be a siphon in net Ψ. Then S is called a strongly dependent siphon if $\eta_S = \sum_{S_i \in \Pi_E} a_i \eta_{S_i}$ holds, where $a_i \geq 0$.

# 4. PETRI NET MODELLING OF FMS

A Flexible manufacturing cells have machined station show in Fig. 6 (a), where for machines (M1-M4) are served by three robotics (R1-R3), are used for moving parts between machines by Colom et al. [13]. This manufacturing is producing two product types, i.e. part-1 and part-2 (p1, and p2), is represented to show how to model and control of an FMS using methodologies Petri net presented throughout this paper.

The production routing of J1 is part type P1:

The production routing of J2 is part type P2:

In this cell, two types of parts must be processed according to their own production routings that is depicted in Fig. 6 (b). These machines use different tools for their work (H1-H4). Robot R1 can load parts of conveyor I1 to M1 and M3 and unload parts from M3 to conveyor O2. Machines M1 and M3 can use tools H1 and H2. Robot R2 can load and unload M1-M4. Machines, M2 and M4 can use tools H3 and H4. Finally, robot R3 can unload M2 to load exit conveyor O1 and can load M4 from conveyor I2.

Parts of type 1 enter into the system via conveyor I1 and leave it via conveyor O1. They can be processed first in M1 or M3. M1 needs to take the tool H1 to accomplish its work and M3 needs to take the tools H1 and H2. After that, the parts are to be processed in M2 that needs to use tools H3 and H4 (according to production routing J1). Parts of type 2 arrive to the system via conveyor I2 and leave it via conveyor O2. They first need to be processed in M4. This machine takes tools H3 and H4 to do the processing. Later, they are processed in M3 that needs to use tools H1 and H2 (according to production routing J2). Suppose that M1 and M4 can process two parts concurrently; M2 and M3 can manage three parts each and that the system has two tools H1, one tool H2, three tools H3 and three H4. Each robot can hold a single part at a time.

In this cell, two types of parts must be processed according to their own production routings that are depicted in Fig. 1(b).

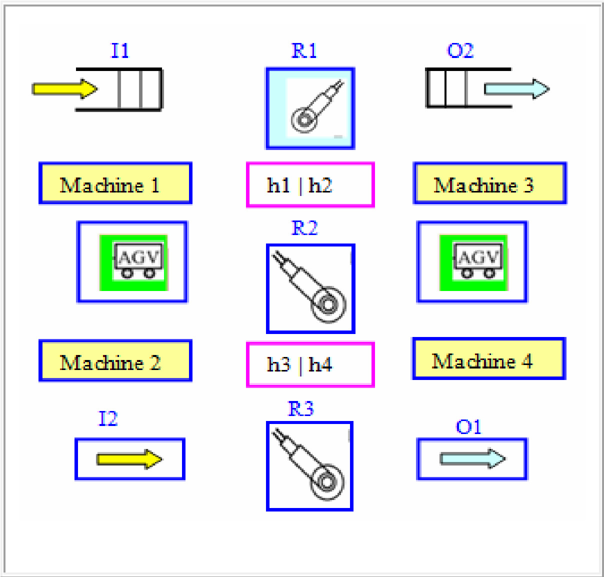


Fig. 6. (a) The layout of an FMS

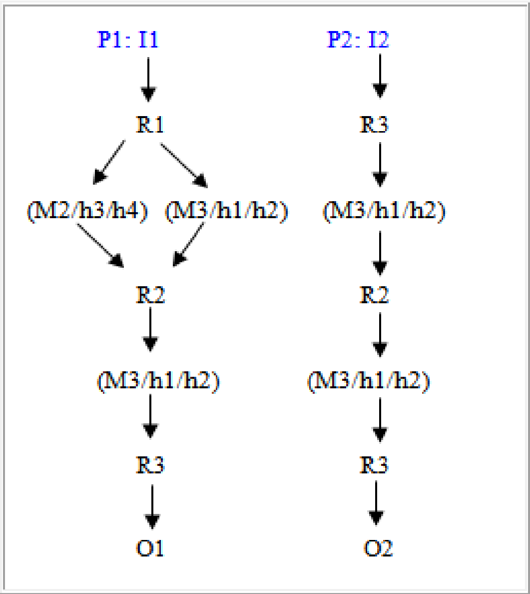


Fig. 6. (b) Production routing.

A parts enter into the FMS through input/output buffers I1/O1 and I2/O2. In the state, we consider that there are two part types J1–J2's, can be produced in this system.

The Petri net model of the FMS is shown in Fig. 7, where are the pleases ($p_{16}$, $p_{21}$, $p_{22}$, $p_{23}$) denoted to (M1-M4) and places ($p_{13}$, $p_{17}$, $p_{20}$) represented R1, R2, and R3 respectively. Places ($p_2 - p_7$) represent the operation of R1, M2, R2 and M3 respectively, for production routing part type-P1, while places ($p_8 - p_{12}$) are represented the operation place of M3, R2 and M2 respectively for production routing part type-P2.

The Petri net is represented an $S^3PR$ by this place specification that uses a multi-set of resource at a production routing. The number of token in Fig. 7, $M_0$ is an acceptable initial marking if $M_0(p_1) = M_0(p_{24}) = 9$, places $M_0(p_{13}) = M_0(p_{17}) = M_0(p_{20}) = 1$, places $M_0(p_{14}) = M_0(p_{15}) = M_0(p_{18}) = M_0(p_{19}) = M_0(p_{21}) = M_0(p_{22}) = M_0(p_{23}) = 2$, places $M_0(p_{16}) = 3$, and others are zero. The elements of this net are defined as the process idle places $P^0 = \{p_1, p_{24}\}$, the resource places: $P_R = \{p_{13}, p_{16}, p_{17}, p_{20} - p_{23}\}$.

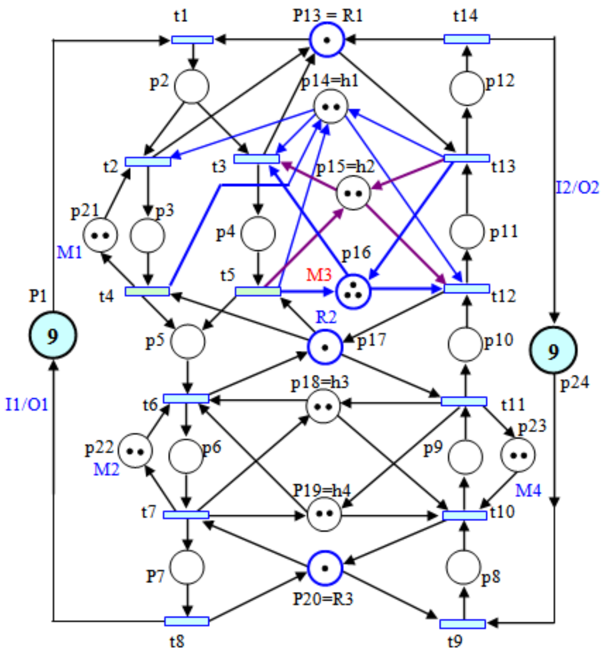


Figure 7, $S^3PR$ net model of the FMS

Simulation and structural analysis of the behavioral properties of Petri net model use the PN-tool with MATLAB [5], starts with coverability tree key. We can see the original net system Ψ has (**1155**) reachable states with initial marking, among which there are **27**- deadlock states. To solve the deadlock problem, one of the most extensively is used the Petri nets tool to design a controller to avoid deadlock. We can find out minimal siphons of Petri net shown in Fig. 7. The net system of FMS is an $S^3PR$, and contains deadlocks. Analysis structured of PN, there are 54 strict minimal siphons as shown below, among the set of siphons from $S_1$–$S_4$ are elementary siphon, and $S_5$–$S_{54}$ are element dependent which are referred to the element dependent ones are marked by *.

$S_1 = \{p_5, p_{11}, p_{14}, p_{17}\}$, $S_2 = \{p_3, p_4, p_{12}, p_{13}, p_{14}\}$,
$S_3 = \{p_7, p_9, p_{19}, p_{20}\}$, $S_4 = \{p_5, p_{11}, p_{16}, p_{17}\}$,
$S_5^* = \{p_6, p_{10}, p_{17}, p_{18}\}$, $S_6^* = \{p_7, p_9, p_{18}, p_{20}\}$,
$S_7^* = \{p_5, p_{11}, p_{15}, p_{17}\}$, $S_8^* = \{p_6, p_{10}, p_{17}, p_{19}\}$,
$S_9^* = \{p_7, p_{12}, p_{13}, p_{16}, p_{17}, p_{18}, p_{20}, p_{21}\}$,
$S_{10}^* = \{p_7, p_{12}, p_{13}, p_{15}, p_{17}, p_{18}, p_{20}, p_{21}\}$,
$S_{11}^* = \{p_7, p_{11}, p_{16}, p_{17}, p_{18}, p_{20}\}$,
$S_{12}^* = \{p_7, p_{11}, p_{16}, p_{17}, p_{19}, p_{20}\}$,
$S_{13}^* = \{p_7, p_{12}, p_{13}, p_{16}, p_{17}, p_{19}, p_{20}, p_{21}\}$,
$S_{14}^* = \{p_7, p_{12}, p_{13}, p_{15}, p_{17}, p_{19}, p_{20}, p_{21}\}$,
$S_{15}^* = \{p_7, p_{11}, p_{15}, p_{17}, p_{18}, p_{20}\}$,
$S_{16}^* = \{p_7, p_{11}, p_{15}, p_{17}, p_{19}, p_{20}\}$,
$S_{17}^* = \{p_7, p_{12}, p_{13}, p_{14}, p_{17}, p_{19}, p_{20}\}$,
$S_{18} = \{p_7, p_{12}, p_{13}, p_{14}, p_{17}, p_{18}, p_{20}\}$,
$S_{19}^* = \{p_7, p_{11}, p_{14}, p_{17}, p_{19}, p_{20}\}$,
$S_{20}^* = \{p_7, p_{11}, p_{14}, p_{17}, p_{18}, p_{20}\}$,
$S_{21}^* = \{p_7, p_{10}, p_{17}, p_{19}, p_{20}\}$,
$S_{22}^* = \{p_7, p_{10}, p_{17}, p_{18}, p_{20}\}$,
$S_{23}^* = \{p_6, p_{12}, p_{13}, p_{15}, p_{17}, p_{18}, p_{21}\}$,
$S_{24}^* = \{p_6, p_{12}, p_{13}, p_{16}, p_{17}, p_{18}, p_{21}\}$,
$S_{25}^* = \{p_7, p_{12}, p_{13}, p_{16}, p_{17}, p_{20}, p_{21}, p_{22}, p_{23}\}$,
$S_{26}^* = \{p_5, p_{12}, p_{13}, p_{15}, p_{17}, p_{21}\}$,
$S_{27}^* = \{p_6, p_{12}, p_{13}, p_{15}, p_{17}, p_{19}, p_{21}\}$,
$S_{28}^* = \{p_6, p_{12}, p_{13}, p_{16}, p_{17}, p_{19}, p_{21}\}$,
$S_{29}^* = \{p_7, p_{12}, p_{13}, p_{15}, p_{17}, p_{20}, p_{21}, p_{22}, p_{23}\}$,
$S_{30}^* = \{p_5, p_{12}, p_{13}, p_{16}, p_{17}, p_{21}\}$,
$S_{31}^* = \{p_6, p_{12}, p_{13}, p_{14}, p_{17}, p_{18}\}$,
$S_{32}^* = \{p_6, p_{12}, p_{13}, p_{14}, p_{17}, p_{19}\}$,
$S_{33}^* = \{p_7, p_{12}, p_{13}, p_{14}, p_{17}, p_{20}, p_{22}, p_{23}\}$,
$S_{34}^* = \{p_5, p_{12}, p_{13}, p_{14}, p_{17}\}$,
$S_{35}^* = \{p_6, p_{11}, p_{14}, p_{17}, p_{19}\}$,
$S_{36}^* = \{p_6, p_{11}, p_{14}, p_{17}, p_{18}\}$,
$S_{37}^* = \{p_7, p_{11}, p_{16}, p_{17}, p_{20}, p_{22}, p_{23}\}$,
$S_{38}^* = \{p_7, p_{11}, p_{15}, p_{17}, p_{20}, p_{22}, p_{23}\}$,
$S_{39}^* = \{p_7, p_{11}, p_{14}, p_{17}, p_{20}, p_{22}, p_{23}\}$,
$S_{40}^* = \{p_7, p_{10}, p_{17}, p_{20}, p_{22}, p_{23}\}$,
$S_{41}^* = \{p_6, p_{11}, p_{15}, p_{17}, p_{19}\}$,
$S_{42}^* = \{p_6, p_{11}, p_{16}, p_{17}, p_{19}\}$,
$S_{43}^* = \{p_6, p_{11}, p_{16}, p_{17}, p_{18}\}$,
$S_{44}^* = \{p_6, p_{11}, p_{15}, p_{17}, p_{18}\}$,
$S_{45}^* = \{p_3, p_4, p_5, p_{11}, p_{15}, p_{21}\}$,
$S_{46}^* = \{p_3, p_4, p_5, p_{11}, p_{14}, p_{15}\}$,
$S_{47}^* = \{p_3, p_4, p_5, p_{11}, p_{16}, p_{21}\}$,
$S_{48}^* = \{p_3, p_4, p_5, p_{11}, p_{14}, p_{16}\}$,
$S_{49}^* = \{p_3, p_4, p_5, p_{11}, p_{12}, p_{15}, p_{21}\}$,
$S_{50}^* = \{p_3, p_4, p_5, p_{11}, p_{12}, p_{14}, p_{15}\}$,
$S_{51}^* = \{p_3, p_4, p_5, p_{11}, p_{12}, p_{16}, p_{21}\}$,
$S_{52}^* = \{p_3, p_4, p_5, p_{11}, p_{12}, p_{14}, p_{16}\}$,
$S_{53}^* = \{p_3, p_4, p_5, p_{12}, p_{13}, p_{14}\}$,
$S_{54}^* = \{p_2, p_3, p_4, p_5, p_{12}, p_{13}, p_{16}\}$.

We have choices four the strict minimal siphons $S_1$, $S_2$, $S_3$, and $S_4$, which are depending on the elementary theorem siphons in [9, 10] as:

$S_1 = \{p_5, p_{11}, p_{14}, p_{17}\}$, $S_2 = \{p_3, p_4, p_{12}, p_{13}, p_{14}\}$,
$S_3 = \{p_7, p_9, p_{19}, p_{20}\}$, $S_4 = \{p_5, p_{11}, p_{16}, p_{17}\}$.

According to elementary theorem [10], we can obtain that the p-vector such as:

$\lambda_{S1} = (0,0,0,0,1,0,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0)^T$,
$\lambda_{S2} = (0,0,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0)^T$,
$\lambda_{S3} = (0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0)^T$,
$\lambda_{S4} = (0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,1,0,0,0,0,0,0,0)^T$.

The T-vector matrix can be constructed in [η] shown as follows:

$\eta_{S1} = -t_1 + t_4 + t_5 - t_9 + t_{13}$,
$\eta_{S2} = -t_1 + t_7 - t_9 + t_{12}$,
$\eta_{S3} = -t_1 + t_2 + t_3 - t_9 + t_{11}$.

The T-vector matrix can be constructed in [η] shown as follows:

| | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ | $t_8$ | $t_9$ | $t_{10}$ | $t_{11}$ | $t_{12}$ | $t_{13}$ | $t_{14}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\eta_{S1}$ = | -1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 1 | 0 | |
| $\eta_{S2}$ = | -1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | 0 | 1 | 0 | 0 | |
| $\eta_{S3}$ = | -1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | 0 | 0 | 0 | |
| $\eta_{S4}$ = | -2 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | -2 | 0 | 0 | 1 | 1 | 0 | + |
| $\eta_{S5}$ = | -2 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | -2 | 0 | 1 | 0 | 1 | 0 | + |

Later, we calculated, the linearly independent vectors can be write as:

$\eta_{S4} = \eta_{S1} + \eta_{S2} = -2t_1 + t_4 + t_5 + t_7 - 2t_9 + t_{12} + t_{13}$, and $\eta_{S5} = \eta_{S1} + \eta_{S3} = -2t_1 + t_2 + t_3 + t_4 + t_5 - 2t_9 + t_{11} + t_{13}$.

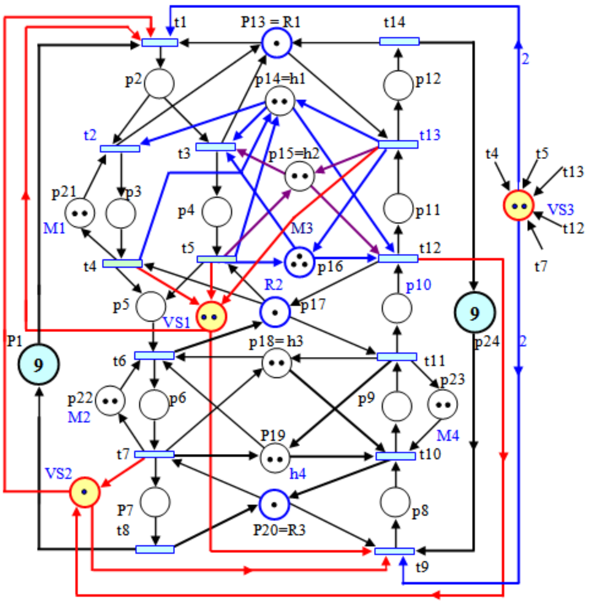


Fig. 8. Additional place $VS_1$, $VS_2$, and $VS_3$, called control place in order to controlled siphon.

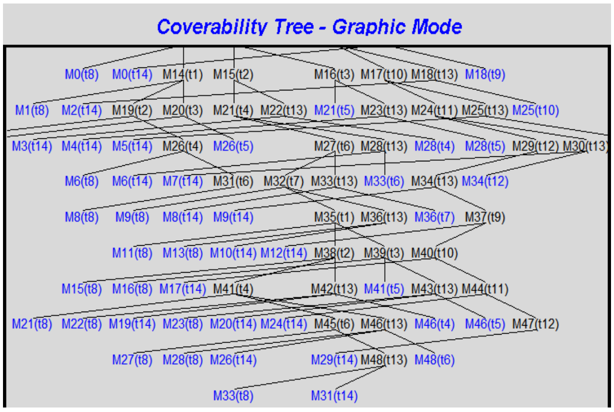


Fig. 9 shows reachability tree of Fig. 7, with two monitors $VS_1$, and $VS_2$.

At the first, we can add two control places $VS_1$ and $VS_2$ to Fig. 8. The reachability tree is live and has **49** reachable states with initial marking shown in Fig. 8, and Fig. 9. For the purposed to reduce the reachability tree more, we are the choice ($\eta_{S4}$) as the third control place (monitor), because when added ($\eta_{S3}$), the net does not affect as shown in Fig. 8. Adding control place ($\eta_{S4}$) such as $VS_3$ the reachability tree is live and has 21 reachable states as shown in Fig. 10.

Having resulting a P-invariant form the incidence matrix of Figure 8. A control place VS1 is added such that: $I_1 = \{p_2 + p_3 + p_4 + p_8 + p_9 + p_{10} + p_{11} + VS_1\}$ is a p- invariant of $(\Psi_1, M_1)$. Therefore, $I_1 \bullet M_1 = 0$ by definition 9, the computation of places control that is:

$[(\Psi_1](VS_1, t) = -t_1 + t_4 + t_5 - t_9 + t_{13}$, Similarly, $I_2 = \{p_2 + p_3 + p_4 + p_5 + p_6 + p_8 + p_9 + p_{10} + VS_2\}$ is a p-invariant of $(\Psi_1, M_1)$. Therefore, $I_2 \bullet M_1 = 0$, by the computation of places control that is: $[(\Psi_1](VS_2, t) = -t_1 + t_7 - t_9 + t_{12}$, Similarly, $I_3 = \{2p_2 + 2p_3 + 2p_4 + p_5 + p_6 + 2p_8 + 2p_9 + 2p_{10} + p_{11} + VS_3\}$ is a p-invariant of $(\Psi_1, M_1)$. Therefore, $I_3 \bullet M_1 = 0$, by the computation of places control that is:

$[(\Psi_1](VS_3, t) = -2t_1 + t_4 + t_5 + t_7 - 2t_9 + t_{12} + t_{13}$.

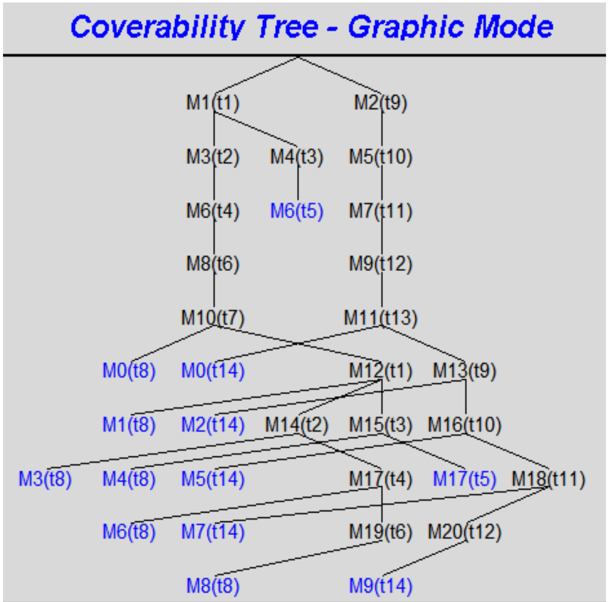


Fig. 10 shows reachability tree of Fig. 8, with three monitors $VS_1$–$VS_3$.

**Definition 18.** Siphon S in a net system $(\Psi, M_0)$ is invariant controlled by P-invariant I under $M_0$ iff $I^T \bullet M_0 > 0$, and $\forall p \in P \backslash S$, $I(p) \leq 0$, or equivalently, $I^T \bullet M_0 > 0$ and that $\| I \|^+ \subseteq S$, [10]. Such a siphon is also called an invariant-controlled one. If S is controlled by P-invariant I under $M_0$, S cannot be emptied, i.e., $\forall M \in R(\Psi, M_0)$, S is marked under M.

Let $I_4$ is a P-invariant can be found from Fig. 8, and implementation manual. For instance siphon $S_1 = \{p_5, p_{11}, p_{14}, p_{17}\}$ can be found on p-invariant from Fig. 8.

$I_4 = \{-1, 0, 1, 0, 0, -1, -1, 0, 0, 1, 1, -1, 0, 1, 0, -1, 1, 0, 0, 0, 0, 0, 0, -1, -1VS_1\}$. We have $I_4 \bullet [\Psi] = 0^T$, $\{p \mid I_4(p) > 0\} \subseteq S_1$, and $I_4 \bullet M_0 = \{M_0(p_5) + M_0(p_{11}) + M_0(p_{14}) + M_0(p_{17}) - M_0(p_1) + M_0(p_3) - M_0(p_5) - M_0(p_6) - M_0(p_7) + M_0(p_{10}) - M_0(p_{12}) + M_0(p_{16}) - M_0(p_{24}) - M_0(VS_1)\} = 1 > 0$. It is mean:

$I_4 = \{p_5 + p_{11} + p_{14} + p_{17} - p_1 + p_3 - p_5 - p_6 - p_7 + p_{10} - p_{12} - p_{16} - p_{24} - VS_1\}$. Then $S_1$ is invariant controlled siphon and it can never be emptied. Similarly, we can compute P-invariant of the siphon $S_2 = \{p_3, p_4, p_{12}, p_{13}, p_{14}\}$ is a siphon of the net. Siphon $S_2$ is controlling by P-invariant: $I_5 = \{-1, 1, 1, 1, 0, 0, -1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, -1, -1VS_2\}$. We have $I_5 \bullet [\Psi] = 0^T$, $\{p \mid I_5(p) > 0\} \subseteq S_2$, and $I_5 \bullet M_0 = \{M_0(p_3) + M_0(p_4) + M_0(p_{12}) + M_0(p_{13}) + M_0(p_{14}) - M_0(p_1) + M_0(p_2) + M_0(p_7) - M_0(p_{12}) - M_0(p_{24}) - M_0(VS_2)\} = 1 > 0$. It is mean: $I_5 = \{(p_3 + p_4 + p_{12} + p_{13} + p_{14}) - p_1 + p_2 + p_7 - p_{12} - p_{24} - VS_2\}$.

Then $S_2$ is invariant controlled siphon and it can never be emptied. Therefore, both $S_1$ and $S_2$ are invariant controlled siphons. This work has been done with a Laptop computer, Intel(R) Core(TM) i5-3337U CPU @ 1.80GHz, with 4Gb of RAM, with Windows 7 Ultimate.

In addition, there are eleven resources in this system leading to eleven minimal P-invariants, we would like mention as follow:

$I_{13} = p_2 + p_{12} + p_{13}$, where $M_0(p_{13}) = 1$,
$I_{14} = p_3 + p_4 + p_{11} + p_{14}$, where $M_0(p_{14}) = 2$,
$I_{15} = p_4 + p_{11} + p_{15}$, where $M_0(p_{15}) = 2$,
$I_{16} = p_4 + p_{11} + p_{16}$, $M_0(p_{16}) = 3$ ,
$I_{17} = p_5 + p_{10} + p_{17}$, $M_0(p_{17}) = 1$ ,
$I_{18} = p_6 + p_9 + p_{18}$, $M_0(p_{18}) = 2$ ,
$I_{19} = p_6 + p_9 + p_{19}$, $M_0(p_{19}) = 2$ ,
$I_{20} = p_7 + p_8 + p_{20}$, $M_0(p_{20}) = 1$ ,
$I_{21} = p_3 + p_{21}$, $M_0(p_{21}) = 2$ ,
$I_{22} = p_6 + p_{22}$, $M_0(p_{22}) = 2$ ,
$I_{23} = p_9 + p_{23}$, $M_0(p_{23}) = 2$ .

# 5. CONCLUSIONS

This paper presents a Petri net-based siphon for designing and implementing the modular supervisor for control FMSs. A deadlock prevention method of a class of FMS, where the unmarked siphons in their Petri net models cause the deadlocks is presented in this work. The FMS are modelled using $S^3PR$, which is a special class of Petri nets. We propose to allocate the tokens in the control places reasonably to guarantee with absence of deadlock states, and monitor added is to each elementary siphon to make ensure that all elementary siphons in the $S^3PR$ net are invariant-controlled. The siphon is successfully controlled and the resultant net system is live (i.e. deadlock-free). An efficient method to compute all the resource system of FMS represented in the Petri net model, which is shown in the experimental results, used to test Petri net toolbox in [5] MATLAB. The minimal siphons in the class of $S^3PR$ nets have become a conceptual and practical central tools to deal with deadlocks caused by the sharing of resources in flexible manufacturing systems. Future work should be included extending this method of more general classes of Petri nets.

# 6. REFERENCES

[1] Abdul-Hussin Mowafak (2015), "A Structural Analysis of Petri Nets-Based Siphons Supervisors of Flexible Manufacturing Systems", Published in: Proc. IEEE-UKSim-AMSS, 17th International Conference on Computer Modelling and Simulation, Cambridge University, March 2015, pp. 235-241.

[2] Abdul-Hussin Mowafak (2015), "Design of a Petri Net Based Deadlock Prevention Policy Supervisor for $S^3PR$", in Proc. IEEE-ISMS2015, 6th Intentional Conference on Intelligent Systems, Modelling and Simulation, Kuala Lumpur, Malaysia, pp. 46-52.

[3] Abdul-Hussin Mowafak (2014)," Siphons and Traps Structural Analysis Techniques Behaviour of a Petri Nets", In: World Symp. on Computer Networks and Information Security, Int. Conference on Advances in Eng. & Tech. (ICAET'2014), Hammamet, Tunisia, pp. 186-193.

[4] Abdul-Hussin Mowafak, (2014) ," Petri Nets approach to simulate and control of Flexible Manufacturing Systems", In: World Congress on E-learning, Education and Computer Science International Conference on Advanced Studies in Computer Science and Engin., (ICASCSE'2014), Hammamet, Tunisia, pages 75-85.

[5] Mahulea, C., M. H. et al. (2003)." Home Page of the Petri Net" PETRI NET Tool-box in MATLAB version 2.3, http://www.ac.tuiasi.ro/pntool

[6] Banaszak, Z., and Mowafak Abdul-Hussin (1988), "Petri net approach to automatic real-time program synthesis", In: Control and Cybernetics, Vol. 17, No. 4, pag. 361-375, Poland

[7] Banaszak, Z., and Mowafak Abdul-Hussin (1999)," Planning of Robots Cooperation: Automatic Modelling and Control", In: Engineering and Technology Journal, Published by University of Technology, Baghdad, Vol. 18, No.1, pages 42- 65.

[8] Ezpeleta. J; J.M. Colom, et al., (1995)," A Petri net based deadlock prevention policy for flexible manufacturing systems," IEEE Trans. on Robotics and Automation, vol. 11, n. 2, pp. 173-184.

[9] Li Z. W., Zhou M. C., (2004), " Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems", IEEE Transactions on System Man.

[10] Li Z. W. and M. C. Zhou, (2008),"Control of elementary and dependent siphons in Petri nets and their application," IEEE Trans. Syst., Man, Cybernetics Part A, System, Humans, vol. 38, no. 1, pp. 133–148.

[11] Huang Y. S. and M. D. Jeng,(2002)"An algorithm of deadlock prevention for flexible manufacturing systems," in Proc. IEEE Int. Conf. Syst., Man, and Cybern., Hammamet, Tunisia, 2002, pp. 168–173.

[12] Wang ShouGuang, MengChu Zhou, et.al. (2015)," Design of a Maximally Permissive Liveness-Enforcing Supervisor with Reduced Complexity For Automated Manufacturing Systems", Asian Journal of Control, Vol. 17, No. 1, pp.190-201.

[13] Colom, J. M., (2003), " The Resource Allocation Problem in Flexible Manufacturing Systems Extended Abstract", In W. van der Aalst and E. Best, editors, Application and Theory of Petri Nets 2003, ICATPN'2003, Lecture Notes in Computer Sci., vol. 2679, pp. 23-35, Springer-Verlag.