MODELING A MODERN MARBLE PROCESSING PLANT BY USING PETRI NET

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖZGE KESER

IN THE PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MINING ENGINEERING

DECEMBER 2003

Approval of The Graduate School of Natural and Applied Sciences of The Middle East Technical University.

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ümit ATALAY
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Tevfik GÜYAGÜLER
Co-Supervisor

Prof. Dr. Neş'e ÇELEBİ
Supervisor

Examining Committee Members

Prof. Dr. Neş'e ÇELEBİ

Prof. Dr. Tevfik GÜYAGÜLER

Prof. Dr. Abdurrahim ÖZGENOĞLU

Prof. Dr. Naci BÖLÜKBAŞI

Doç. Dr. Hadi GÖKÇEN

ABSTRACT

MODELING A MODERN MARBLE PROCESSING PLANT BY USING PETRI NET

Keser, Özge
M. Sc., Department of Mining Engineering
Supervisor: Prof Dr. Neş'e Çelebi
Co-Supervisor: Prof Dr. Tevfik Güyagüler

December 2003, 110 pages

All developing countries need sufficient raw material resources to develop and to guarantee their future. Considering Turkish natural resources, marble has a great importance because of its demand on the market, reserve amount and quality. However, some effort is required to improve the existing marble production and processing efficiency.

Petri nets (PNs) are the information models that control the flow for concurrent and synchronous systems. In this regard, PN application can be useful. However, its application is limited to the complex systems and no application of PN is available in mining sector.

In this sense, this study aims to examine the applicability of PN to mining. This study examines the production system in order to optimize the process in case of two different types of marble product orders. Three case studies are applied to examine benefits and difficulties in implementation of PN to a marble processing plant. The study shows that PN can successfully be used as a tool for the optimization of total production time, simulation and modeling of the system. It provides to see the sequence of the processes, their time, remaining time of each transition and optimum total production times. The difficulties of PN implementation are found out as the determination of each path in the reachability graph, matrix representation with large quantity of place, etc.

Keywords : Petri net(s), flexible manufacturing, manufacturing systems.

ÖZ

MODERN BİR MERMER İŞLEME FABRİKASININ PETRİ AĞLARI KULLANILARAK MODELLENMESİ

Keser, Özge
Yüksek Lisans, Maden Mühendisliği Bölümü
Tez Yöneticisi: Prof Dr. Neş'e Çelebi
Ortak Tez Yöneticisi: Prof Dr. Tevfik Güyagüler

Aralık 2003, 110 Sayfa

Gelişmekte olan ülkelerin geleceklerini garanti altına almak ve ilerlemek için hammadde kaynaklarına ihtiyaç duydukları açıktır. Türk doğal kaynakları düşünüldüğünde mermerin rezervi, kalitesi ve pazardaki talebi nedeniyle büyük bir öneme sahip olduğu görülür. Fakat mevcut üretim metotlarının verimliliğin arttırılması için geliştirilmesi gerekmektedir.

Petri ağları, eş zamanlı ve senkronize sistemlere akış kontrolü sağlayan bilgi verici modellerdir. Bu bağlamda Petri net uygulaması faydalı olabilir. Ancak, karmaşık sistemlerdeki uygulamaları sınırlıdır ve literatürde madencilik sektöründe uygulamasına rastlanmamıştır.

Bu bağlamda bu çalışma, Petri Ağlarının madencilikte uygulanabilirliğinin araştırılmasını amaçlamaktadır. Bu çalışma, üretim sistemini, iki farklı tipte mermer ürünü siparişi durumunda işlemlerin optimizasyonunu incelemektedir. Bir mermer fabrikasına üç farklı örnek çalışma uygulanarak, uygulamadaki yaralar ve zorluklar incelenmiştir. Bu çalışma, Petri Ağlarının toplam üretim sürelerinin optimizasyonu, simülasyon ve modelleme aracı olarak başarıyla uygulanabileceğini göstermektedir. Petri ağları, işlem sıralarını, işlem sürelerini, her işlemin kalan süresini ve optimum toplam üretim süresini görmeyi sağlamıştır. Petri ağlarının uygulanmasındaki zorluklar, yolların çizilmesinde her alternatifin tespiti ve çok sayıda kaynak için matris gösterimi olarak gözlenmiştir.

Anahtar Kelimeler : Petri ağları, esnek üretim, üretim sistemleri.

To the memory of :

NİLGÜN ÖZKAN

ACKNOWLEDGEMENTS

I want to acknowledge and express my graduate to the following people…

To my advisor Prof. Dr. Neş'e Çelebi for her invaluable guidance, moral support throughout this research, and suggestions;

To my co-advisor Prof. Dr. Tevfik Güyagüler for his mentorship, constructive comments and, his tolerance;

To Assoc. Prof. Dr. Hadi Gökçen for his encouragement and moral support;

To the most special person in my life Mesut for always supporting and encouraging me onwards;

To my mother for her love and support to me throughout the years;

To an invaluable person, Nilgün Özkan, for her love, encouragement, support and suggestion for selecting this profession.

I am grateful to all of them…

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

CHAPTER 1

INTRODUCTION

It is obvious that all developing countries need sufficient raw material resources to develop and to guarantee their future. Today all the conflicting and carried out ideas of economy accept one thing, necessity of raw materials in production. In recent economy the basic factors of production includes qualified workforce, raw material and capital. This means that among the countries with rich natural resources ones using this superiority will be formers in the development race in the future.

Among the natural resources, mines have a special place. Since they are unrecoverable, known as limited resource, and they generally have a function of preventing migration and improve the socio-geographical structure. If these properties are understood and evaluated well, the contribution of mining to the country development will increase. On the other hand increase in the life standards increases the demand of minerals and consumption.

If the investments for mining in a country are examined, it is seen that mining has more contribution to the other industries and it is a motivating force. Gross National product increases in parallel to the amount of activated underground resources.

Natural stones widely are used in construction, covering, flooring, sculpture, porcelain and glass industry, optical industry, and decoration. Among the natural stones, marble has an important role in the establishment of civilizations and development of cultures that makes it different from the others. Another difference is the type of production. Unlike the others with continuous production, the production of marble is based on orders.

According to Bilgin and Çakır (1998) marble has two definitions; in scientific definition, marble is defined as a crystallized form of limestone with the effect of metamorphism. Other definition is commercial and as follows: marble is any kind of stone that can be cut and polished.

Turkey has rich marble reserves. Today, Turkey has at least in forty-three countries approximately hundred kinds of marble apparent reserves. Turkish marbles can compete with Italian marbles regarding the quality and amount of reserves. Turkish firms deal with both the marketing of the marble products in domestic market and exporting them. The exportation is mainly carried out to U.S.A., Israel, Saudi Arabia, Russia, Germany, Libya and Italia.

Turkey has total reserve of 14 billion tons (approximately 5 billion m^3). Turkey's proven reserve, which has the amount of just about 1.6 billion tons, can supply for 80 years of world's consumption. With evaluation of reserves, it can have incredible amount of foreign exchange. As a result, miners have a great opportunity in marble industry and evaluating this opportunity an important contribution to the Turkish economy can be provided.

As described above, marble sector is based on orders. Therefore, the production according to the order has a great importance in this sector. This means that increasing the adaptability of the production system to the changes in order specifications obeying delivery time is very beneficial and necessary. This can be considered as the optimization of the marble processing plant for each order.

Optimization of marble production system, before all else, entails analyzing the system in detail. These analyses may cover both examining the existing systems, and also investigating the alternatives. In this sense, optimization, modeling and simulation techniques are used. Petri Net, as a new technique, includes all these three elements.

Since the introduction of Petri nets, there has been a growing interest in the theory and applications of nets for the modeling and analysis of asynchronous concurrent systems. Included among the areas of application are computer systems with concurrent programming and multiprocessor systems, protocol design and verification in computer networks, and the modeling and control of flexible manufacturing systems. However, no mining application of Petri net is found in the literature. Petri net is a graphical and mathematical tool. It provides a uniform environment for modeling, formal analysis, and design of discrete systems. These utilities provide ease of design, evaluation and optimization of any stage of mining activities including mining and processing. Computer - integrated engineering systems are event - driven and often asynchronous, exhibiting concurrent, sequential, competitive and coordinated activities among their components. They are often complex and large in scale. These systems include integrated manufacturing systems, concurrent distributed systems, computer operating systems, communication networks, and intelligent machines. They fall into a class of systems called *discrete event systems (DESs)*. The relations of events, their evolutions over time and their order of appearances are of main interest. Compared with systems modeled as differential or difference equations, e.g., a manufacturing process, one has to deal with qualitative or abrupt changes characterized as events, instead of with quantitative changes in conventional continuous or discrete time systems. An investigation into the properties of and design methods for a DES has to start with an appropriate mathematical representation.

The aim of this thesis is to investigate the applicability of Petri Net in mining sector, to expose the difficulties faced with during the application, and to determine the benefits of Petri Net. As an application area, a marble plant, using advanced technology, is selected.

In the second Chapter, definition of Petri Net and its components, history of Petri Nets, properties of Petri Nets, characteristics of Petri Nets, usage of Petri Nets, needs for Petri Nets, problems of applications, and other modeling techniques used in mining and marble production are given.

In Chapter 3, it is explained that how a system can be modeled by using Petri Nets, Petri Net drawing rules, matrix representation of Petri Nets, and time representation of Petri Nets.

In the fourth Chapter, three different case studies are carried out and the application is presented in details.

In the last chapter the whole study is evaluated, results of the study is argued and recommendations for next studies are given.

CHAPTER 2

PETRI NETS

2.1 Definition of Petri Nets

Petri net is a simple tool for modeling workpiece flow through a manufacturing system and for representing the orderly execution of the individual operations (Rembold, et all, 1996).

According to Zhou and Robbi (1994), Petri nets are graphical and mathematical models for representing information and control flow in an event–driven system. They have two nodes, transitions and places. Direct arcs link places to transitions and transitions to places. Tokens are used to describe the state of discrete event systems.

Petri nets are one of the most rigorous and powerful modeling tools for event – driven systems. Various extensions have greatly enhanced their applicability to various types of DES, particularly automated manufacturing systems. It is proposed here that the use of a graphical, Petri net – based software tools to form the basis of manufacturing system design. The tool provides for design specifications, analysis, simulation and in some cases the actual control codes for system operation (Zhou and Robbi, 1994).

As a Graphical Tool :

- ✰ Petri nets as graphical tools provide a powerful communication medium between the user and the customer.

- ✰ Complex requirement specifications can be presented graphically.

- ✰ Allowing for interactive graphical simulation of Petri nets.

As a Mathematical Tool :

- ✰ Petri nets as mathematical tools reflect the behavior of the system.

- ✰ Allowing for the formal analysis of the model, e.g., precedence relations amongst events, concurrent operations, appropriate synchronization, freedom from deadlock, repetitive activities, and mutual exclusion of shared resources.

- ✰ Important for real - time safety - critical systems.

2.2 Definition of Petri Net Components

Petri net is a graphical and mathematical modeling tool, which is able to model concurrent, asynchronous, distributed and parallel systems. Petri nets are models representing information and control flow in an event–driven system. The basic Petri net consists of four different components: Places, Transitions, Arcs, and Tokens.

Place is a basic Petri net component, which represents a condition. If a place is member of the input function of a transition, it is a pre – condition for that transition, or event, to occur. If a place is a member of the output function of a transition, it is a post – condition of the event or transition firing. A place is generally represented by a circle in a Petri net design.

A place is an input place to a transition if there is a directed arc connecting this place to a transition.

A place is an output place of a transition if there is a directed arc connecting the transition to the place. Input places may represent the availability of resources, the transition may represent their utilization, and output places may represent the release of the resources.

Transition is a basic Petri net component, which represents an event. The event may occur when all of the input places to the transition contain enough tokens to enable the input arcs to the transition. The occurrence of the event, also known as a firing of the transition, will result in tokens being deposited through each of the output arcs of the transition into the respective output places (Kimbler, 1997).

For modeling manufacturing systems with Petri nets, the following interpretations for places, transitions, and tokens are employed by many researchers :

1. A place represents a resource status or an operation; when it represents the former, the initial number of tokens can be either a constant, e.g., the number of machines assuming the plant is fixed, or a variable, e.g., the number of jobs or pallets within the system.

2. If a place represents a resource status, one or more tokens in the place indicate that the resource is available and no token indicates that it

Is not available. A token in it shows that an operation is being executed and no token shows that it is not being performed.

3. A transition represents either start or completion of an event or operation process (Zhou and DiCesare, 1993).

Arc is a Petri net component, which relates places to transitions and transitions to places, thus forming an input function and an output function for each transition. Arcs are generally represented as a directed line in a Petri net design. An arc also has an associated weight, representing the number of tokens needed to enable the arc.

Arcs are labeled with their weights (positive integers), where a k-weighted arc can be interpreted as the set of k parallel arcs. Labels for unity weight are usually omitted (Murata, 1989). In the application of this thesis, all of the arcs have a unit weight, i.e. they have 1 as an arc weight, so, they are omitted.

In order to represent dynamic behavior of the modeled system, each place may hold either none or a positive number of tokens. An object is represented in a Petri net by a token (dot), which is located in a place (circle) (Rembold, et all, 1996). A token signifies that a particular place, or condition, is true. Multiple tokens residing in a place are usually representative of the existence of multiple resources, rather than boolean condition.

The distribution of tokens on places, called Petri net marking, defines the current state of the modeled system. A Petri net containing tokens is called a marked Petri net.

A transition without any input place is called a source transition, and one without any output place is called a sink transition. Note that a source transition is unconditionally enabled, and that the firing of a sink transition consumes tokens, but does not produce any.

A pair of a place p and a transition t is called a self-loop if p is both an input and output place of t. a Petri net is said to be pure if it has no self-loops. A Petri net is said to be ordinary if all of its weights are 1's (Murata, 1989). In the application of this thesis, all of the arcs have 1 as an arc weight, so, the Petri net is ordinary.

After the static structure of a Petri Net is defined, the flow of tokens regulated by transition firings describes its dynamics. The event sequence can be studied by executing two rules.

The enabling rule states that transition t is enabled when all its input places have enough tokens i.e. the number of tokens in each place equals or is greater than that of the arcs from the place to transition t. Transition t being enabled implies that the conditions for its associated event to occur are satisfied. Therefore, it can occur.

The firing rule states that an enabled transition t can fire, or an event can occur. Its firing can be regarded as two separate stages. First, remove the required number of tokens from each of its input places, the quantity determined by the number of arcs between a place and t. Second, deposit tokens into each of it's output places; the number of tokens equals the number of arcs from t to the corresponding output place (Murata, 1989, Zhou and DiCesare, 1993).

Example 1 : The above transition rule is illustrated in Figure 1 using the well-known chemical reaction : 2 H_2 + O_2 $\rightarrow$ 2 H_2O. Two tokens in each input place in Figure 1(a) shows that two units of H_2 and O_2 are available, and the transition t is enabled. After firing t, the marking will change to the one shown in Figure 1(b), where the transition t is no longer enabled.

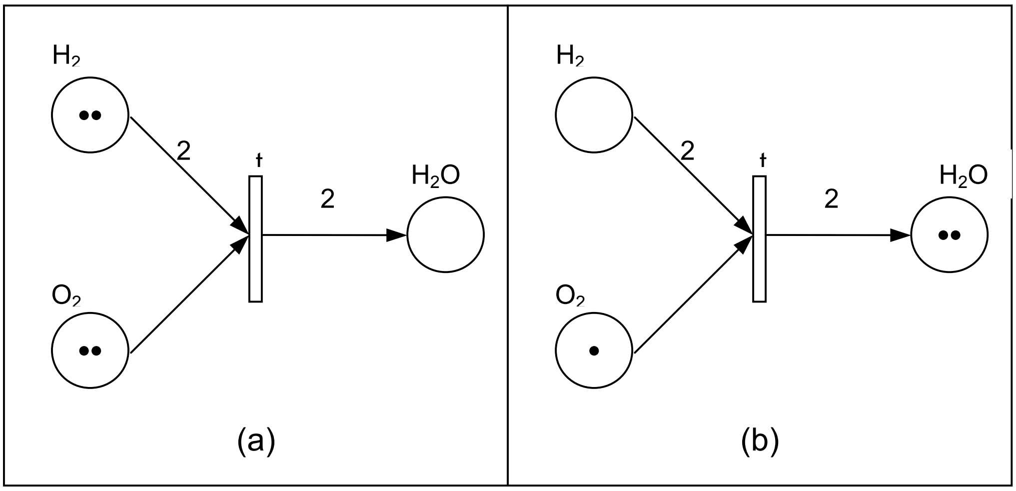


Figure 1 : Example 1 : An illustration of a transition (firing) rule; (a) The marking before firing the enabled transition t. (b) The marking after firing t, where t is disabled (Murata, 1989).

For the above rule of transition enabling, it is assumed that each place can accommodate an unlimited number of tokens. Such a Petri net is referred to as an infinite capacity net. For modeling many physical systems, it is natural to consider an upper limit to the number of tokens that each place can hold. Such a Petri net is referred to as a finite capacity net (Murata, 1989).

2.3 History of Petri Nets

When Petri net was first constructed by Carl A. Petri in 1962, the approach in Petri net was to model and then to analyze. However, Petri net researchers have made remarkable strides in the modeling, analysis, and control of DES.

While many activities are currently engaged in Petri net applications to the manufacturing automation area, the following foci could be observed since the late 70s :

1. Early interest in Petri nets arises from the need to specify and model manufacturing systems. The activities in this area start with the Petri net representation of simple production lines with buffers, machine shops and automotives production systems, and proceed with modeling of flexible manufacturing systems, automated assembly lines, resource – sharing systems and recently just – in – time and kanban – based manufacturing systems.

2. Early research focused on qualitative analysis of PN models of manufacturing systems. Reachability analysis shows whether a system can reach a certain state. Desired sequences of event are validated according to the system requirements. Other Petri Net properties are used to derive the DEDs stability, cycle behavior and freedom from deadlocks.

3. As temporal or quantitative properties become an important consideration, timed PNs are used to derive the cyclic time of repetitive and concurrent manufacturing systems. To deal with the stochastic nature of many production operations, stochastic PNs are used to derive the system production rates or throughputs, critical resource utilization and reliability measures. Their underlying models are Markow or semi-Markow processes. The direct construction of Markow chains is avoided thanks to conversion algorithms for stochastic PNs.

4. When a state explosion problem arises or the underlying stochastic models are not amenable to tractable mathematical analysis, simulation must be conducted for analysis of both qualitative and quantitative properties. Fortunately, PN models can be easily utilized to drive a complex discrete event simulation. Several packages based on PNs exist.

5. Programmable logic controllers (PLCs) are commonly used in industrial sequence control of automated systems. They are designed through ladder logic diagrams, which are known to be very difficult to debug and modify. It is observed that a PLC can be converted into a PN and vice versa for a subclass of PNs. Early work includes the conversion of a PN into a PLC for implementation. Direct PN controllers without the help of PLCs can also be implemented though either a Petri net interpreter or its corresponding control codes. For most cases, additional information to represent the real time signals and status needs to be incorporated into such PN models.

6. The advantages of PNs include their relative ease to represent and modify the control logic and their potential for mathematical analysis and graphical simulation to validate a design. It can be proved that the graphical complexity of PNs grows with systems complexity less than that of ladder logic diagrams.

7. With a mathematical representation available, designers are able to use PNs for rapid prototyping of a process control system or discrete event control. Virtual factories can be realized though computer graphics using PNs. Stepwise testing and implementation can be achieved by connecting the actual equipment into a PN – based design system reducing design and development time.

8. Petri nets have been combined with other approaches to achieve various purposes in process planning and scheduling, intelligent control, expert system construction, knowledge representation and uncertainty reasoning. For example, the correspondence between a PN and an expert system can be established. This can greatly aid in consistency checking of an expert system (Zhou and Robbi, 1994).

2.4 Properties of Petri Nets

The property analysis of a Petri net includes boundedness, safeness, liveness (absence of system deadlock), reversibility and reachability. Their significance to manufacturing systems is stated as follows:

Boundedness or safeness implies the absence of capacity overflows. These Petri net properties help to identify in the modeled system the existence of overflows is the concept of boundedness (Tunçel, 1999). For instance, there may be storage buffers or queues, which have a finite capacity. The boundedness of a place such as a buffer or queue insures that there will be no overflow. Safeness is the special case of boundedness. As an example, safeness of an operation place guarantees that there is no attempt to request execution of an ongoing process. Safeness of a resource place indicates the availability of only a single resource and is often used to guarantee the safeness of some related operation places (Zhou and DiCesare, 1993).

Liveness implies the absence of deadlocks. This property guarantees that a system can successfully produce. Moreover, it insures that all modeled processes can occur (Zhou and DiCesare, 1993). A transition is dead in a marking if there is no sequence of transition firings that can enable it. A transition *is* potentially fireable if there exist some sequence of transition firings that enables *it.* A transition of a PN is live if it is potentially fireable in all reachable markings (Mahmutoğulları,1992).

Reversibility has implications for error recovery in the manufacturing context. It means the system can be initialized from any reachable state (Zhou and DiCesare, 1993). Reversibility implies the cyclic behavior of a system (Tunçel, 1999).

One of the fundamental properties of a net is reachability. A marking is reachable from another marking if the firing of one or more transitions changes one marking into the other (Kimbler, 1997).

A desirable Petri net model should include the following properties :

1. The model should be bounded. In other words, it should have no capacity overflows. This provides no exceeded capacity on buffers and machines.

2. The architecture should be such that whenever a token in a place node which represents an available machine resource leaves that node, at least one value adding processing step must be guaranteed to occur before the token returns to that node. This property is beneficial in that it assures no

unnecessary state sequences are generated during which no useful work is performed.

3. For any realistic initial marking the Petri net model should be globally acyclic. It implies that the model will monotonically progress from its initial state to its target (Terry, 1996).

2.5 Characteristics of Petri Nets

To help ground the discussion of event-driven systems, a simple manufacturing example is presented by Zhou and DiCesare, as follows. The system consists of two different machines, a robot, and a buffer, and is sketched in Figure 2. Every part from the input storage must be processed by Machine 1 first and then by Machine 2 to produce a final product. The robot is used for unloading both machines, and the buffer is used to store intermediate parts : machines 1 and 2 load themselves from the input part storage and the buffer, respectively. Once a machine starts work on a part, it cannot be interrupted until the work is complete. The following characteristics are embedded in this system: it is event-driven, asynchronous, and sequential; it exhibits concurrency, conflict, mutual exclusion, and non-determinism. More importantly, such a simple system can contain a system deadlock, which may result from improperly triggering a particular sequence of events.

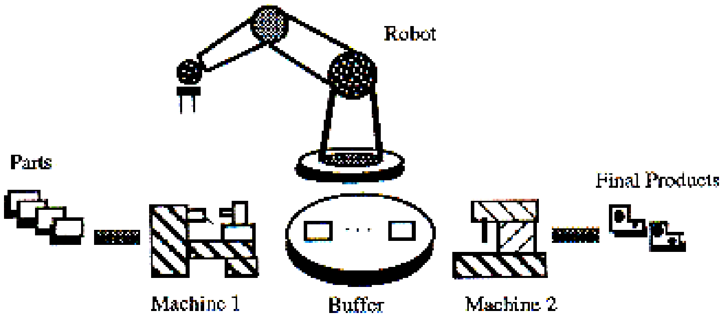


Figure 2 : A discrete event system example

Event – driven : A discrete event system is characterized by a discrete state space where changes in state are triggered by event occurrences. Precedence is a key relation between events, that is, any event may be dependent on the occurrences of other events. In the example shown in Figure 2, one event is the start of Machine 1 loading, fixturing and processing a part. The next event might be the end of this processing and the start of unloading of a part from Machine 1. Note that for a particular part the second event is dependent on the occurrence of the first. It may also be dependent on other event s such as the release of the robot to make it available and the release of a buffer space.

Asynchronous : The asynchronous characteristic of discrete event systems is one of the most important properties by which they differ from traditional systems described by differential or difference equations. In time discretization of sampled systems, each change or step is synchronized by a global clock. In continuous systems, parameters vary continuously with time. However, in discrete event systems the events of ten occur asynchronously. For example, in the preceding discussion, the second event might be dependent on the occurrence of three preceding events. The release of a buffer space will probably occur at a different time from the release of the robot or the end of processing on Machine 1. The start of unloading cannot occur until all of the preceding asynchronous events have occurred. Traditional methods fail to describe asynchronous systems.

Sequential Relation : Given a set of events, there may exist some sequential relationships among them. It is said that there is a sequential relation between two events if one event can occur only after the occurrence of the other. Such a sequential relation can be described as a partial ordering. For the given example, *start-a-job* will occur first, followed by *complete-a-job* for the same part. Mathematically, there is a partial order between *start-b-job* and *complete-a-job.* A part can also be turned into a final product by Machines 1 and 2 in the system, but it has to be processed by M_1 first and then by M_2. This is another example of a sequential relation. In fact, given the same part, one set of events, start-a-part-on-M_2 and complete-a-part-on-M_2, has to follow another set of events, start-a-part-on-M_1 and complete-a-part-on-M_1.

Concurrency : By concurrency it is meant that there are no sequential relationships among the concerned events. For example, physically, two events complete-a-part-on-M_1 and complete-a-part-on-M_2 are concurrent in the sense that either event may occur before the other. After the two machines start to work in the above system, they can operate concurrently since both can process their own parts at the same time after both are loaded by the robot. If machining processes are programmed into two independent programs, which have been implemented on different computer processors, then they are executed concurrently or in parallel after they have been initiated at possibly different time moments.

Also, either machine may break down at any time. One cannot tell exactly when such an unexpected event will occur. The events, M_1-breaks-down and M_2-breaks-down, have an uncertain sequential relationship. Thus, they are said to be concurrent.

Conflict : In the above system, there is a conflict when M_1 and M_2 complete their own parts and are ready to be unloaded, a buffer slot is, ready for an intermediate part, and the robot is available. The robot can either unload M_1 or M_2, but not both of them simultaneously. When a resource is shared by different processes, such conflicts are often inevitable.

Mutual Exclusion : When the above conflict occurs, one can easily find that the two events unload-M_1 and unload-M_2 are mutually exclusive in the sense that they cannot occur at the same time, whereas after one is complete, the other can occur. Often in the literature, mutual exclusion among independent distributed processes is investigated, while mutual exclusion among sequentially related distributed processes has been ignored. Our example here is a simple case of the latter mutual exclusion. Since parts have to be processed by M_1 first and then M_2, unload-M_1 and unload-M_2 can be mutually exclusive, and a sequential relation exists between two events.

Non-determinism : Two kinds of non-determinism may be observed. The first kinds result from uncertain events' occurrence. For example, when there is a conflict as discussed above, either of two events can occur randomly, i.e., the robot can unload M_1 or unload M_2 without any certainty as to which will occur. Also, any two events, which are ready to occur, can take place in a random order. For example, when M_1 and M_2 are idle and there are parts available for their processing, either machine can be loaded, and the order is unpredictable. The second kind of non-determinism results from small changes in process parameters. For example, processing times of a machine differ from time to time due to randomness. Therefore, it cannot be accurately predicted when an event, e.g., complete-a-job*,* will occur. In order to fully explore all the non-determinism in such systems, stochastic timed models need to be used.

System Deadlock : Suppose that, in the above system, the buffer capacity is 1. If the system initially has 3 parts, M_1 processes the first part, and the robot unloads this intermediate part to the buffer. Then M_1*,* and M_2 loads and begins to process this intermediate part. While this is occurring, M_1 loads a second part and processes it, and the robot unloads this intermediate part to the buffer. M_1 proceeds to load and process the third part. Loads a second part and processes. The robot then grasps this part and tries to unload M_1, the system is deadlocked. This is because M_1 cannot be unloaded because the buffer is already full. M_2 cannot be unloaded either because the robot is not available. To avoid such a catastrophic failure, one can adopt either of two approaches. The first approach is that designing a supervisory controller, which may lead to a deadlock. When it is implemented in a real-time environment, if a deadlock results, the system detects and resolves it. It is obvious that using such a method may be very costly since it may be expensive to detect and resolve deadlocks in a fully automated factory or a distributed system. For the system above, if the deadlock is detected, the recovery methods can be, for example, to use another robot to unload M_2 so the buffer becomes idle again. An alternative recovery method is to find a spare place, which can be used for the temporary placement of the part grasped by the robot, and to direct the robot to place the part so that it becomes available to load M_2*.* It can be observed that such an approach may lead a system to chaos if, for example, no good deadlock resolution methods are available.

The second approach is to design a supervisory controller with desirable system properties, for example, freedom from deadlock. In the above example, such a supervisory controller can be designed such that when the buffer is full and Machine 2 is busy, the robot is not allowed to unload Machine 1. Several mechanisms exist to implement such a controller. One of the simple methods is to limit the number of jobs, which is allowed to enter this system to two in a particular time interval. Thus, when buffer is full and Machine 2 is busy, no job can be allowed to enter the system. Therefore, the robot cannot unload Machine 1.

The preceding example is simple and the solutions are intuitive. However, for complex systems, systematic methods are necessary to design supervisory controllers, which have the desirable system properties.

Summarizing, the increasing use of computer-based controllers to execute supervisory tasks in systems makes the study of event-driven system theory very important (Zhou and DiCesare, 1993).

2.6 Usage of Petri Nets

Petri nets are used in the following areas.

- ☆ Modeling and analysis of communication protocols
- ☆ Modeling and analyzing manufacturing systems
- ☆ Modeling sequence controllers
- ☆ Modeling and analysis of software development

☆ Performance evaluation of the modeled systems, e.g., multiprocessor systems

☆ Communication networks

☆ Performance of production systems

☆ Modeling scheduling problems involving manufacturing systems

☆ Modeling dynamics of continuous chemical processes

2.7 Why need Petri Net ?

Compared with other models, they have the following advantages:

☆ Ease of modeling DES (Discrete Event System) characteristics : concurrency, asynchronous and synchronous features, conflict, mutual exclusion, precedence relations, non-determinism, and system deadlocks.

☆ Ability to generate supervisory control code directly from the graphical Petri net representation.

☆ Ability to check the system for undesirable properties such as deadlock and instability and to validate code by mathematically based computer analysis-no time consuming simulation for many cases.

☆ Performance analysis without simulation is impossible for many systems. Production rates, resource utilization, reliability, and performability can be evaluated.

☆ Discrete event simulation can be driven from the model.

☆ Status information that allows for real-time monitoring.

☆ Usefulness for scheduling because the Petri net model contains the system precedence relations as well as constraints on discrete event performance.

☆ The design and operation of modern industrial systems require modeling and analysis in order to select optimal design alternative and operational policy.

☆ To minimize flaws and maximize efficiency.

☆ The same model is used for the analysis of behavioral properties and performance evaluation, as well as for systematic construction of discrete-event simulators and controllers.

☆ It provides a uniform environment for modeling, formal analysis, and design of discrete systems (Keseroğlu, 2000).

☆ It is proposed here the use of a graphical, Petri net – based software tools to form the basis of manufacturing system design.

☆ The tool provides for design specifications, analysis, simulation and in some cases the actual control codes for system operation.

☆ Petri nets are used for modeling, and simulation of automated manufacturing systems.

- ☆ The advantages of PNs include their relative ease to represent and modify the control logic and their potential for mathematical analysis and graphical simulation to validate a design. It can be proved that the graphical complexity of PNs grows with systems complexity less than that of ladder logic diagrams.

- ☆ One of the major advantages of Petri nets is the ability to analyze them for properties related to manufacturing control (Zhou and DiCesare, 1993).

2.8 Problems of Application

Up to this point only a theoretical or conceptual framework for describing the behavior of searchers are presented. It is found that it philosophically appealing because it shifts emphasis from random jumping from one state to another, to fact-driven behavior consisting of transitions among various kinds of persistent search activity.

In order to confront this theory with reality it is needed to be much more specific about the map between these abstractions and behavior of real people as they search. In doing this, it had to be operationalized some fundamental kinds transitions. It is believed that in the course of searching in a networked environment there are several kinds of transitions that are apparent to the user. One is to give up. A second is to change the way in which on think of the question. A third is to persist with the same question but to use some kind of a search engine and describe your same quest in new terms or key word (reformulation). A fourth is to find a particular item to be useful for navigation. A fifth is to find a particular item to provide some useful information. A sixth is to find a particular item to not provide any useful information.

While these alternatives seem conceptually clear, it is not obvious that information can be elicited from real searchers about which of the events or transitions has just occurred to them. In fact, it is not apparent that searchers can be enabled to recount their search experiences in a language of persistence and transitions (Kantor, Nordlie, 1999).

2.9 Types of Manufacturing Operations

Manufacturing companies and plants are referred by a variety of terms that describe something about their manner of operation. The facility, equipment, and operating methods (sometimes called the production system) that a company uses depend on the type of product that it offers and the strategy that it employs to serve its customers. Some of the terms refer to the stage at which the company holds inventory.

Some companies are make-to-stock producers; that is, they make items that are completed and placed in stock before the customer's order is received. The end item is shipped "off the shelf" from finished goods inventory after receipt of a customer's order. In contrast, some companies make to order. A make-to-order producer completed the end item after receipt of the customer's order for the item. If the item is unique, custom designed item, the customer will probably have to wait for many of the materials to be purchased and for the production to be performed, because the producer can not anticipate what each customer might want and have the necessary row materials and component on hand to shorten the production lead time. If components or materials are frequently used by the business, however, the producer may keep some of them in stock-particularly if the lead time to purchase or produce those items is long. When the company produces standard design, optional modules ahead of time and assembles a particular combination of these modules after the customer orders it, the business is said to be an assemble-to-order producer. An example of an assemble-to-order producer is an automobile factory that, in response to a dealer's order, provides an automatic or manuel transmission, air conditioner, sound system, interior options and specific engine options as well as the specified body style and color. Many of the components would already have been purchased and started into production when the dealer placed in order. Otherwise the lead time to deliver the automobile would be much longer.

Other terms are used to distinguish the characteristics of factories and the ways that products are manufactured. Extend to which a factory (or a smaller production area within a facility) has the flexibility to produce a variety of products is one characteristics often used to distinguish among types of factories. Factories or production areas encompass a wide range in regard to these characteristics. At one extreme is a factory that makes custom products in low volume or in single units and probably will never repeat production of one of them later. At the other extreme is a factory that makes only one standard production in very high volume. Although several types of factories could be described within this range, primary only three are used in the further study. The job shop, batch and repetitive production factories are described and used to present some of the general characteristics and management methods that will be studied regard to manufacturing.

Job Shops : A *job shop* manufacturing business contracts to make to order custom products in accordance with designs supplied by the customer. Some of these businesses engineer and build items on performance requirements specified by the customer. Typically the volume of each product is low, so these companies must contract to make a wide variety of products in order to achieve a sufficient level of sales. To increase the changes of making sales and to maintain a volume of business, these factories need general-purpose production equipment that can perform a broad range of operations and employees who have a broad range of skills. Job shops are generally classified as high-variety, low-volume manufacturers.

Each job may be unique, requiring a special set of production steps performed in a particular sequence (a "routing") to convert the raw material into the desired finished item. The next item may require a totally different sequence of production steps from the previous one. Consequently, there is no standard path of material flow though this type of facility. The production equipment must be general-purpose, and materials-handling equipment should be flexible, with the capability to move various sizes and shapes of objects along widely varying paths. Flexibility is important in this business. These companies face a big challenge in planning, scheduling, and coordinating the production of numerous components of a wide variety of unfamiliar products. Job shops may carry an inventory of some raw materials that they use frequently, but often the largest percentage of their inventory is work in process (WIP) that accumulates between process stages because of the sporadic material flow.

Examples of job shops might be a drapery shop that makes custom draperies, a woodworking shop that makes custom kitchen cabinets, or a metal fabrication shop makes special machinery.

Repetitive Manufacturing : Repetitive manufacturing, repetitive production, and production lines are terms used for mass production facilities that produce a high volume of the same or similar units of product that follow the same path through the production steps. An example would be an automobile assembly line. There may be two-door and four-door models in different colors; some get automatic transmissions, and others get manual-shift transmissions; but all have fixed flow path (routing) through the same sequence of work stations to convert the raw material into the finished product. This is also an illustration of mixed-model assembly on an assembly line.

Materials typically are moved along the path in small lots, often one part at a time. The products are normally made to stock, and the items are not identified as belonging to a particular order during their production. The production equipment in this type of facility performs the same operation repeatedly and with little variation in its results from cycle. Materials-handling equipment can also be specifically designed to transport a specific shape of product through a fixed path. Frequently, automated equipment is used. Usually, schedules are expressed as rates, such as a specified number of units per hour or per day. Inventories of raw materials may be held to ensure a supply, so the factory can operate and finished goods may be accumulated to smooth fluctuations in demand. Work in process is low since item move quickly through production. Typical products of this type of production might be televisions, telephones, refrigerators, microwave ovens, and roller skates.

Batch Manufacturing : A vast number of possible combinations of product variety and average volume per products are depicted by Table 1. the general region at the upper left (high variety and low volume) generally describes job shops. The region at the lower right (low variety and low volume) generally describes repetitive manufacturing. Many manufacturing operations fall somewhere between these two regions on the figure and are often called batch manufacturers. A *batch manufacturing*, facility makes some intermediate variety of products and produces an intermediate volume of each. The volume of any one item is not sufficient to justify dedicating a set of equipment to its production, so a few or several products share the production resources. The company will make a batch (a production run), may be less than a hundred or up to a few thousand of one product, then switch over the equipment and make a batch of another item. Eventually it will repeat production of the items.

Production equipments in batch manufacturing must be capable of performing some variety of tasks, but the range of possible operations is much narrower than in a job shop. Attachments and tooling may be installed (that is, the equipment is “set up”) to run one type of item. After a batch is completed, the equipment may be set up a new to run some other item. The ability to change back and forth quickly is important in this type of production system. A company may achieve a high degree of focus (that is, reduced ranges it must make) if it runs families of items that require the same or nearly the same processing steps. The paths of material flow may vary if the company does not have a high degree of focus. If there is focus, however, the flow paths may be identical and the materials-handling equipment can be more specialized, perhaps automated.

Table 1 : Matching the process with product variety, equipment flexibility, and volume requirements.

Product Variety	**High**	**Moderate**	**Low**	**Very Low**
Equipment Flexibility	**High**	**Moderate**	**Low**	**Very Low**
Low Volume	Job Shop			
Moderate Volume		Batch		
High Volume			Repetitive Assembly	
Very High Volume				Continuous Flow

An example of a batch manufacturer might be a company that makes small hand tools. It might make ¼ inch drills, then electric drivers, hand mixers, ½ inch drills, then 1/8 inch drills, etc.

The three major types of manufacturing introduced above differ greatly in the degree of flexibility they have to produce a variety of products. They can be thought of as falling along points on a continuum, with the very flexible job shop at the left and the specific-product, repetitive factory a the right. Figure 3 shows such a continuum.

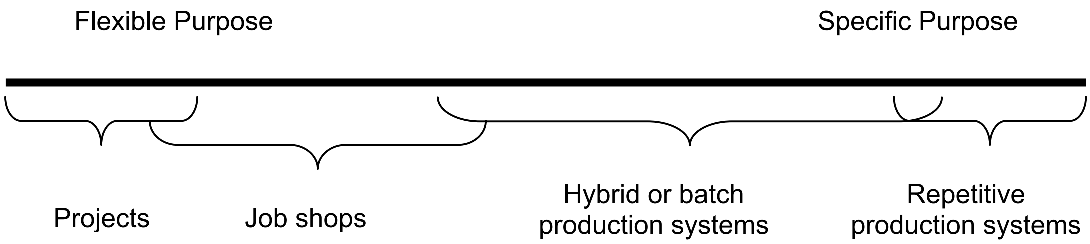


Figure 3 : Degree of flexibility of production systems

Process and project are two other terms that are sometimes used to describe types of manufacturing operations. Some repetitive production operations produce a product that blends together in bulk rather being in discrete units. The industries that produce these types of products are sometimes called *process industries*, particularly if some physical or chemical reactions are used. Examples of process industries include petroleum refineries, flour mills, cement factories and chemical processing plants. (Chemical processing can also take place with batches of material, in what is sometimes called “batch-process” production).

Lying at the high-flexibility end of continuum is the low-volume type of operation often referred to as a *project*. Most projects are of relatively long duration, and the same personnel are often assigned to a project for a significant part of this time. In the manufacturing category, projects include such items as ships, bridges, buildings, and large, special machines. Project teams can be formed for a variety of undertaking in manufacturing operations and in services (Dilworth, 1993).

Match the processes and the product : A key concept in process selection is the need to match product requirements with process capabilities. The difference between success and failure in production can sometimes be traced to choice of process. Product range from highly customized to highly standardized. Generally, volume requirements tend to increase as standardization increases; customized products tend to be low volume, and standardized products tend to be high volume. These factors should be considered in determining which process to use.

Certain processes are more amenable to low volume; customized products, while others are more suited to moderate variety products, and still others to higher volume, highly standardized products. By matching product requirements with process choices, producers can achieve the greatest degree of efficiency in their operations. Table 1 illustrates this important concept.

It can be noticed that the examples all line up along the diagonal of the table. This is the most efficient alignment. If a producer chooses some other combination (e.g., assembly line for a customized product or service), he or she would find the highly customized requirements of the various products are in direct conflict with the more uniform requirements needed to effectively operate in the assembly line mode. Similarly a job shop arrangement (machines and personnel are capable of handling a wide variety of processing requirements) would be rested on a highly standardized product; equipment and personnel need to be highly specialized.

Table 1 can provide insights to managers selecting processes and managing existing operations. For new products, decision makers should make every attempt to achieve a matching of product and process requirements. For an ongoing operation, a manager should examine existing processes in light of the table to see how well processes and products are matched. Poor matches suggest the potential for improvement, perhaps with a substantial increase in efficiency and lowering of cost.

Another consideration is that products and services often go through life cycles that begin with low volume but which increase as products or services become better known. When that happens, a manager must know when to shift from one type of process (e.g., job shop) to the next (e.g., batch). Of course, some operations remain at a certain level (e.g., magazine publishing), while others increase (or decrease as markets become saturated) over- time. Again, it is important for a manager to assess his or her products and services and make a judgment on whether to plan for changes in processing over time (Stevenson, 1996).

2.10 Other Modeling Methods Used in Mining and Marble Production Instead of Petri Nets

Petri net modeling system is mainly used to the complex and highly automated systems because of its structure. However, in marble production, especially in Turkey, un-automated old production techniques are applied. These techniques don’t include production lines. Therefore most of the system based modeling techniques cannot effectively applied to the marble processing plants.

Furthermore, insufficient knowledge level of management and technical staff in marble sector makes the application of modeling and analysis techniques used for the optimization and efficiency difficult.

In recent years the application of techniques is limited to the stock control, project management and some production management techniques. Having inhomogeneity in both raw materials and end products to a certain extent restricts the applicability of most of the methods. However, considering not the product properties but quantities Petri net are convenient for use in mining enterprises. However, no application of Petri net has been observed in mining literature.

CHAPTER 3

MODELLING WITH PETRI NET

3.1 Petri Net Drawing Rules

Rules can be explained by considering the operation of a pick – place robot for component insertion (these robots are widely applied to assembly of components to printed circuit boards). The robot picks up a component and then place it in a desirable position. Two events are *picking-up* and *placing*. The first one can take place if a component is available and the robot is ready. As shown in the Figure 4, two circles called places are used to represent these two conditions, labeled p_1 and p_2. Putting at least one dot, a token, into each circle represents that each condition is true. Draw a bar called transition t_1 to represent event picking-up. Two arcs link p_1 and p_2 to t_1. After the robot picks up a component, a new condition results, i.e.; the robot holds the component represented by another circle called p_3. Then event placing depicted by another bar called transition t_2 can occur. Arcs from t_1 to p_3 and from p_3 to t_2 show the relationships. Once the robot has placed the component, it is ready for the next pick-up operation. Thus the arc from t_2 to p_2 is created. Since the component is already inserted, no arc is formed from t_2 to p_1.

Tokens (dots) in a PN appear identical although they may carry different interpretations in different places. For example, a token in p_1 means one available component and one in p_2 the ready robot. The number of tokens in a place defines a local condition. For example, one token in p_1 implies one component available while two tokens in it means component available. The distribution of tokens in all places determines the status of the entire system, called a system state. A state or marking in PN is formally defined as a vector whose components represent the number of tokens in the corresponding places. The state is changes when a transition fires, i.e. an event occurs. This results in a new marking.

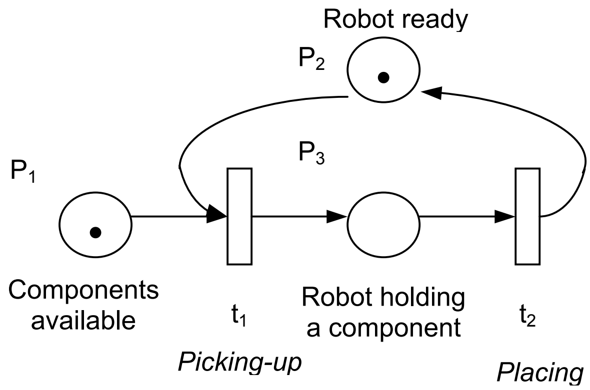


Figure 4 : A simple PN example: a pick - place robot (Zhou and Robbi, 1994).

The following primitives as illustrated in Figure 4 can be modeled with places, transitions and arcs.

Synchronization : The synchronization between two conditions can be realized though two places linking to a common transition. Even though one place obtains a token, the transition has to wait for the other to get a token. Then the event can take place. Therefore, a transition in a net may have multiple input places. In the context of manufacturing, a raw workpiece's arrival and a machine's availability are two conditions that may lead to occurrence of an event machine – *a - workpiece* or start-*machining*. In assembly systems, several parts must be present before a final product is assembled and in this case several places may be designed as the inputs to the transition representing an assembly operation. Synchronizing various activities to reduce the waiting time and thus to maximize production rate is a challenging problem for production engineers in a flexible manufacturing environment.

Choice : When there are two or more machines ready for a coming raw piece, one needs to make a choice as to which machine the piece should be dispatched. This is a dispatching problem. In the context of PNs, a place has two or more outgoing arcs, representing the different possible events to follow, as shown in Figure 5(b). One must be chosen probabilistically, by priority or some other criterion.

Fork : The fork structure is used to model an event whose occurrence leads to more than one information and / or control signal. For example, when a machine completes its operation on a raw piece, a machined part is sent out for the next operation and the machine itself becomes available. In other words, two pieces of information are generated due to this event *complete-operation*. Another example is the disassembly process, which may lead to different parts for different post-processes.

Join / merge : A place may have more than one incoming arc, signifying that several information/control sources lead to a common condition/status represented by a place. The first example is based on manufacturing processes, which produce identical parts and send them to a common station for further processing. The second example is a resource shared by several processes. Release from any process resumes the availability of the resource by depositing a token in the common place.

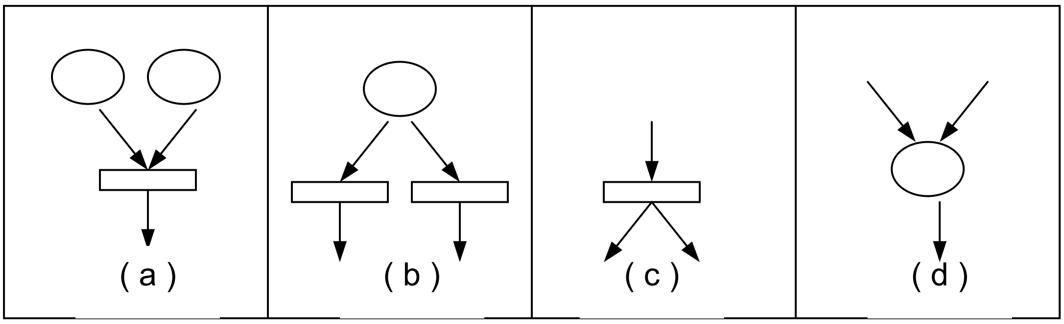


Figure 5: Representation of four primitives: (a) synchronization, (b) choice, (c) fork and (d) join/merge (Zhou and Robbi, 1994).

In addition to above primitives, Petri nets can model the following six structures that are often used in practice, as show in Figure 6.

Sequence : One common structure in event-driven systems is a strict sequence of events. For example, *load-a-part-to-Machine-A*, *process-a-part-on-Machine-A*, *unload-a-part-to-Machine-A* can be such a sequence. In assembly, several parts may have to follow a strict order to finish an assembly job. With PNs, a series of transitions and places can be used to model these sequences of events as shown in Figure 6(a).

Concurrency : In modern manufacturing, many operations have to occur during the same time period to achieve the maximum efficiency and least inventory, which in turn reduces the overall production cost. Pure concurrency of two events can be represented by two non-related transitions. One common case of concurrency, though, can be synchronizing transition to start and a fork transition to end as shown in Figure 6(b). If several tokens are allowed in a sequence structure to mark those places, pipeline concurrency is widely used in manufacturing production lines.

Parallel choice : One commonly used choice structure is shown in Figure 6(c). At one point, there exist choices resulting in different paths, which merge at a later point. Such a structure can represent the case that the order of installing two parts has no effect on the final product in an assembly process. In flexible manufacturing, a part may be either processed by machine A or B and C. this case can also lead to such a parallel choice structure.

Circuit and self-loop : The above three basic structures do not require any initially marked place. They are especially suitable for modeling manufacturing operations. In practice, many systems or their components exhibit cyclic behavior. Using PNs, a circuit may represent the adopted as shown in Figure 6(d). A simple circuit may represent the cycle for a robot to perform repetitive activities, e.g. *load-a-part* and *unload-a-part*. A self-loop may represent the availability of a machine and its working status. It is clear that such a simple circuit is suitable for specifying a resource, especially a non-consumable one like a robot or a machine, and its associated repetitive activities.

Choice-synchronization : In many instances of manufacturing control it is necessary to enforce particular sequences for a choice structure, particularly when assembly processes are involved. Those choice structures required to enforce sequences are called choice-synchronization structures. A typical one is shown in Figure 6(e). Without the enforcement mechanism, places p_4 and p_5 and the dotted arcs, a token residing in place p_1 can reach either p_2 through t_1 or p_3 through t_2. When tokens are present in both places p_2 and p_3, the event associated with transition t_3 may occur. If there is no restriction on the number of tokens passing through t_1 or t_2, then the net may be partially deadlocked since either p_2 or p_3 may never hold a token blocking the event associated with t_3. With p_4, p_5 and related arcs, a strict order can be implemented. If the initial marking is given as shown in Figure 6(e) p_2 and p_3 will obtain a token alternately with p_2 getting it first. Another method to limit the number of tokens passing through sequentially related transitions, e.g. t_1 and t_3, can be to add an initially marked place to form a circuit structure.

Mutually exclusion : A useful structure deals with the resource shared by several processes or machines. For example, a robot may be shared by two machines for loading and unloading. A parallel mutually exclusion example is given in Figure 6(f) in which two parallel processes share the resource in place p_1. A sequential mutual exclusion can be obtained if the two processes are sequentially related.

Combinations of the above modeling primitives enable designers to model numerous manufacturing systems with PNs. This is the first step toward the goal to design, synchronize and coordinate production facilities more efficiency.

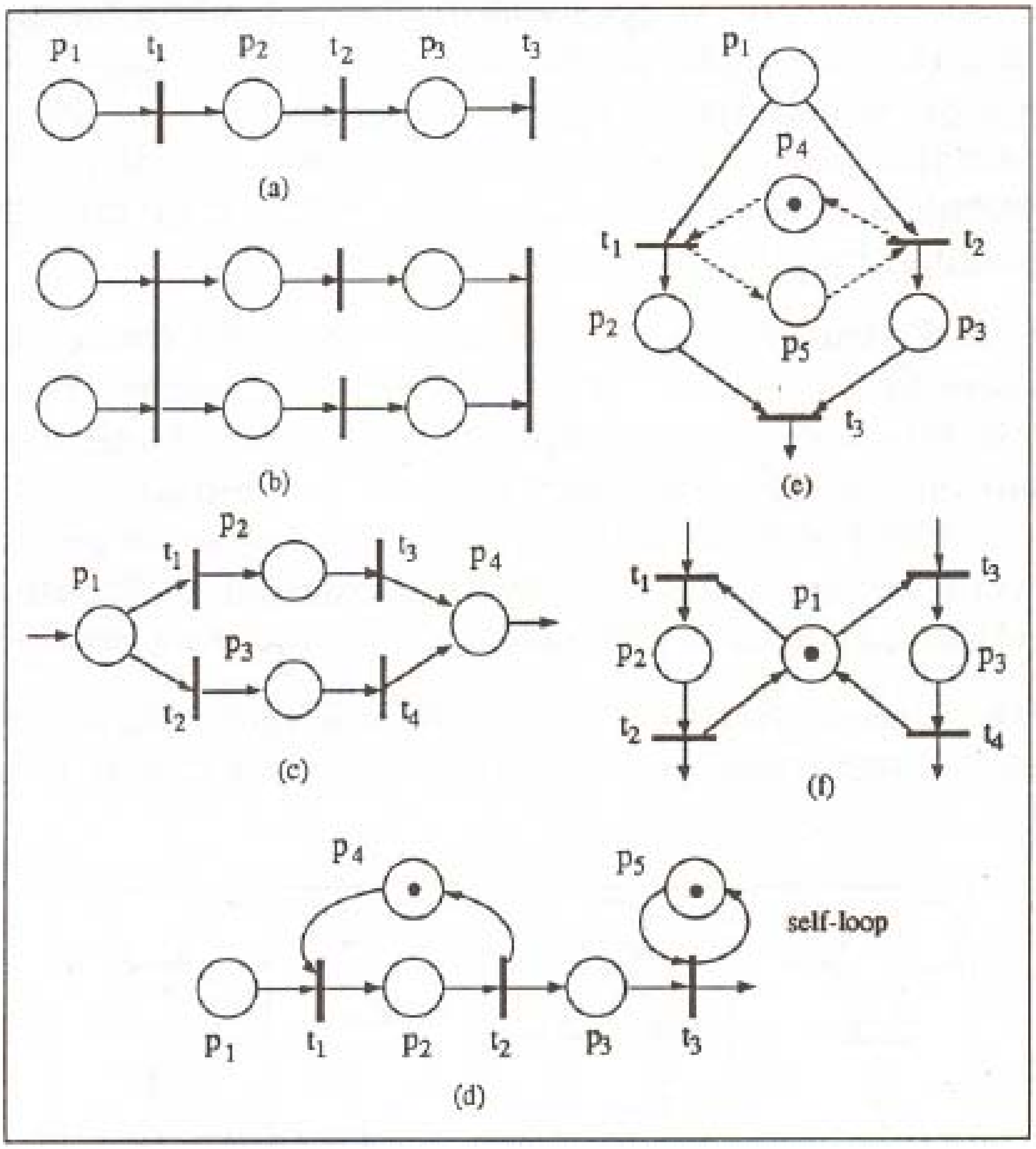


Figure 6: Representation of six structures: (a) sequence, (b) concurrency, (c) parallel choice, (d) circuit and self-loop, (e) choice-synchronization and (f) mutual exclusion (Zhou and Robbi, 1994).

3.2 Matrix Representation of Petri Nets

Any Petri net can be represented as an incidence matrix. For represent a Petri net as an incidence matrix, the following rules should be provided :

1. Find the D^- matrix. It is an m x n (m rows, n columns) matrix, where m is the number of transitions and n is the number of place in the Petri net. For each position [i , j] in the matrix, place a 1 in the position if transition i has input from position j. A 0 is placed in the position if position i does not have input from position j.

2. Find the D^+ matrix. It is an m x n (m rows, n columns) matrix, where m is the number of transitions and n is the number of place in the Petri net. For each position [i , j] in the matrix, place a 1 in the position if transition i has output from position j. A 0 is placed in the position if position i does not have output from position j.

3. Find the D matrix (the composite change matrix). It is computed by subtracting D^- matrix from D^+ matrix.

A 1 x m matrix should be constructed to represent the firing of the Petri net. In each position [1 , j] place the number of times transition j is to fire.

A 1 x n matrix should be constructed to represent the current marking of the Petri net. In each position [1 , j], place the number of tokens in position j.

To determine the marking of the Petri net after the transition(s) specified in the transition matrix, compute : (Kimbler, 1997)

([Transition Matrix] [D]) + [Marking Matrix] = [Next Marking]

3.3 Time Representation of Petri Nets

Because the state reachability graph for a Petri Net of even a very small but *realistic* model of a manufacturing system become large, the following discussion centers upon a very simple Petri net which does not model a manufacturing process per se. The purpose of this discussion is to demonstrate a method for keeping track of elapsing time within all enabled transitions of a Petri net, and subsequently generating a schedule of transition firing times.

Figure 7 below shows a Petri net in its initial state on the left and its (marking) reachability graph on the right. To illustrate how an algorithm for searching the Reachability graph keeps track of the time required each marking, consider the following simple example. The left of the Reachability graph has been indicated with a dashed line corresponds to the transition firing order t2, t0, and t1. for this example, all three transitions have unity time delay. In the initial state of the Petri net, transitions t0 and t2 are enabled.

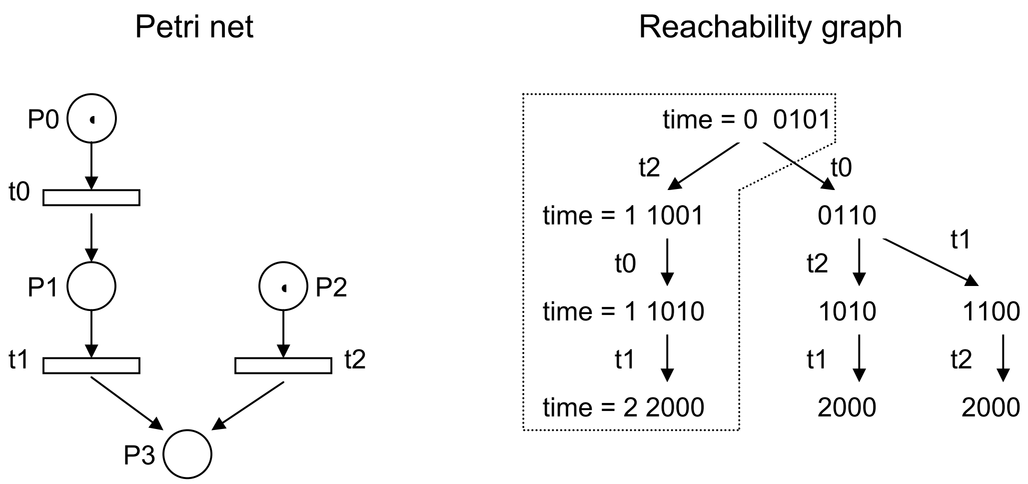


Figure 7 : Petri net and Reachability graph

When the algorithm elects to fire transition t2, the algorithm must also start (or update) timers on all other transitions, which are enabled when t2 is fired – only t0 in this case. Advancing from the marking 0101 to 1001 by firing t2 takes 1 time unit, but subsequently advancing from marking 1001 to 1010 by firing t0 takes zero time units because the timer for t0 was started and left running at the instant the timer for t2 was started. Therefore, even though transitions t2 and t0 were fired in sequence, it takes only one time unit to reach markings 1001 and 1010. Firing transition t1 to advance to marking 2000 from 1010 takes one additional time unit, and corresponding total path cost is two time units. The depth of the final marking is three.

The above example illustrates the idea behind keeping track of timers for all enabled transitions at every node of the Reachability graph. Furthermore, once the target marking has been reached, a schedule of transition firings can be obtained by tracing the parentship pointers of the nodes from the target back to the initial marking (Terry, 1996).

3.4 Reachability Graph

When using Petri nets, one constructs a Reachability tree. One begins by searching the state space, or markings of the Petri net, beginning with the initial state. From there, every possible succession of state transitions of the model is generated. This creates a set of reachable markings. There may be several possible from a given state. Eventually, the search must terminate, since the state space is assumed to be finite given a finite state machine. Every possible execution path is represented by a sequence of consecutive states. This is the *reachability graph* or *reachability tree*. A Reachability graph satisfies the properties that constrain the Petri net to reduce number of feasible state and path. Each state corresponds to a set of conditions and each path to the sequence of events creating those conditions (Moghaddam, 2000).

CHAPTER 4

MODELING STUDIES OF THE PLANT

4.1 General

A marble processing plant in Ankara is selected in order to carry out the case studies. The company owning the plant is one of the leading firms in Turkey. It utilizes high Italian technology with semi-automated production lines. The plant is established near to the Esenboğa Airport. In the plant, different kinds of marble, travertine, granite, diabase and andesite are processed and tile and plate products are presented. The plant produces both plates and tiles. The plant mostly produces length free marble plates and dimensioned marble tiles.

The yearly average production of the plant is 30000 m^2. Company's production is based on orders. Specifications of products are defined by orders. Product dimensions; such as, 30x30x1 cm, 45x45x3 cm, 60x60x2 cm, are determined by customers. Therefore, working shifts of the equipment are not systematic but depend on the requirements. Generally, cutting line works for three shifts and polishing line works for one shift. However, when an urgent production is required, the number of disc are increased and polishing line is adjusted so that cutting and polishing sections work with synchronized three shifts a day.

4.2. Plant Layout and Production Line

The plant has a stock side where the blocks are accumulated. By using a crane, blocks are carried from the stock site to a railway car, which is used for the transportation of the block to block cutters. Two identical block cutters are used to cut the marble blocks into plates. It has two steel saws having different diameters. The bigger saw is located in vertical dimension. It cuts the block from the top to the bottom. The smaller one located in horizontal direction. It cuts the block from one side to the other side. Almost all machines in the plant use water to prevent the formation of the dust and it protect the saw against wearing.

Each block cutter has an unloader to unload the cut plate from the block cutter to a palette. Than the palette is transported to the polishing line. A loader carries the plates from the palette to the polishing line. Head cutter cuts the two sides of each plate. Than, calibrator smoothes the plate's top face roughly. Pre-polishing machine smoothes more than the calibrator. Polishing machine apply the actual polishing process. Therefore, the plate is ready to sizing. Sizing Machine cuts the plates into demanded sizes.

There are two identical Side Calibrating & Bevel Cutting Machines. First one process two opposite sides, then the Rotating Unit turns the plate at 90 degree. Thus, the second Side Calibrating & Bevel Cutting Machine can process the other two opposite sides. After that, the plates are Dried and Cleaned. Next, Quality Control of the plates is done. Then, the plates are packaged and finally, they are carried to the stock side for dispatching.

Flow sheet of the marble processing plant is given below :

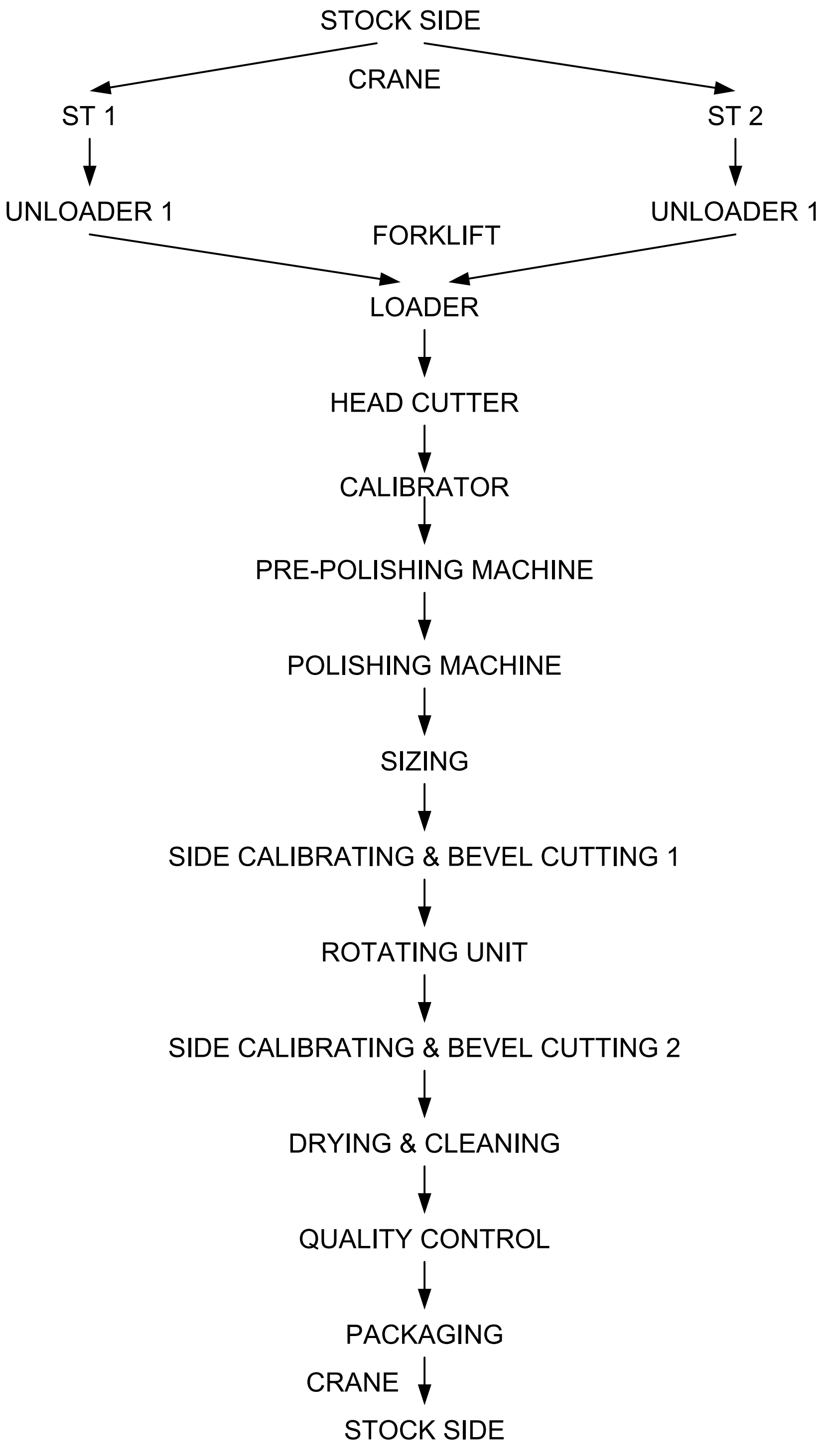


Figure 8 : Flow Sheet of a Modern Marble Processing Plant

The marble tile production line can be shown on a simple sketch in Figure 9.

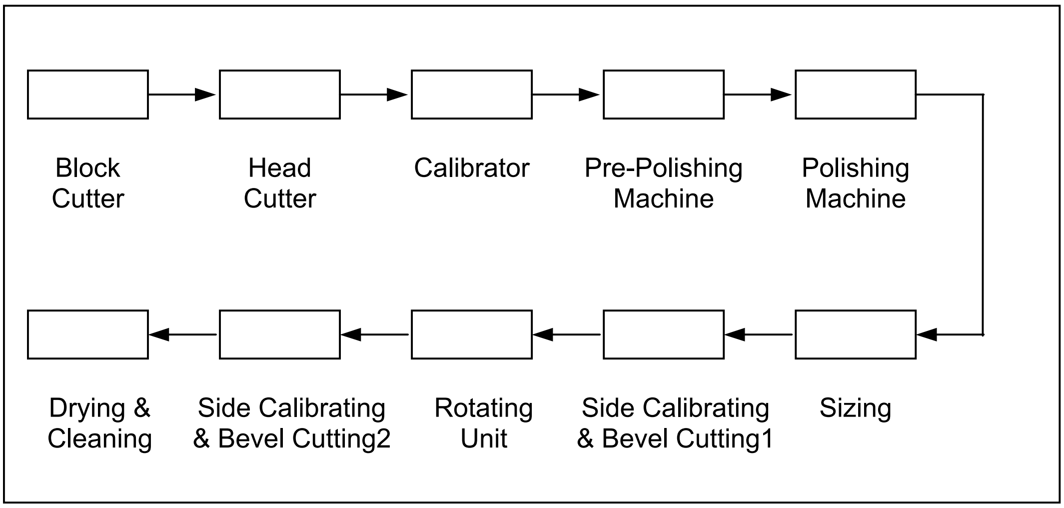


Figure 9 : Marble Tile Production Line

4.3 System Definition and Assumptions

Because of the nature of mining, miners always have to deal with changing conditions. Therefore empirical considerations and assumptions are very important and useful in scientific researches and studies. In this study some assumptions are obligatory to simulation and analysis of the system. These assumptions are given below :

- ☆ Process period of each machine is determined for the production of the same amount of product.
- ☆ 1 palette has a 10 m^2 plate carrying capacity.
- ☆ The distance between the location of blocks in stock side and each block cutters are assumed to be equal. Therefore, carrying a block from the stock side to any of the block cutters has the same time duration.
- ☆ All block are assumed to be uniform; in other words, they are not rubble. Therefore, the trimming time is ignored. The blocks are ready to cut, so when a block is carried to a block cutter, it can be cut at once.
- ☆ It is supposed that all the blocks have the same size.
- ☆ If two different products are processed in the plant at the same time, it is assumed that each block cutter cuts the same product all the time.
- ☆ Adjusting times of the block cutters are included in the process time of them.

- ✰ When the polishing line starts to process any type of a product, the process must be completed for 10 m^2 products. Therefore, two different product types cannot be processed in the polishing line mixed together. However, when a 10 m^2 product is processed for one type of product, the line is adjusted immediately, and the second type of product is fed to the polishing line.

- ✰ Adjusting time of the machines through the polishing line is relatively so small that it is ignored.

- ✰ When a plate is cut by the block cutter, it should be unloaded from the block cutter by an unloader right away.

- ✰ Both of the two block cutters are identical.

4.4 Case Studies

4.4.1 General

In this thesis, three different case studies are performed. Brief information about all case studies is given below :

Case Study 1 ⇒ Two different tile product types with dimensions of 30x30x1 cm and 60x60x2 cm are produced using two block cutters.

Case Study 2 ⇒ One product type is produced using two block cutters.

Case Study 3 ⇒ One product type is produced using only one block cutter.

In the first case study, two different products, which have the dimensions of 30x30x1 cm and 60x60x2 cm, are produced. There are two identical block cutters in the plant. It is assumed that one of these block cutters always cuts the first product and the other one always produces the second product. After cutting by the block cutters, both products are carried to the polishing line and the same polishing line processes both of them in sequence.

In the second case study, both of the block cutters' products have the same dimensions. In sub-case studies, both of the products with 30x30x1 cm and 60x60x2 cm dimensions are examined separately. In this case study, products of both block cutters are also polished by the same polishing line. However, because of having the same dimensions, which the block cutter's product is polished first is not so important.

In the last case study, also only one type of product is produced. However, different from the previous case study, in the third case study, there is only one block cutter and one polishing line. As in the previous case study, both of the products with 30x30x1 cm and 60x60x2 cm dimensions are examined one by one in the sub-case studies.

In the following sections, each case and sub-case studies are studied in detail.

4.4.2 Case Study 1

In this case study, both of the products having the dimensions of 30x30x1 cm and 60x60x2 cm are produced simultaneously. Two identical block cutters are used to cut each product. It is assumed that the same block cutter always cut the same type of product. Although cutting by different block cutters, both products are processed by the same polishing line sequentially.

4.4.2.1 Petri Net

As explained in Chapter 2, a place represents a resource status or an operation, and a transition represents either start or completion of an event or operation process. If a place represents a resource, a token in the place indicates that the resource is available and no token shows that it is not available. If a place has a token, this denotes that an operation is being practiced and no token denotes that it is not being performed.

In the light of these explanations, the Petri nets of each case and sub-case studies are drawn. Petri net drawing for case study 1 is explained for carrying one each block from the stock site to the block cutters by using crane as follows :

For carrying a block from the stock site to the block cutters by using crane, a block, the crane, and the worker are needed. Block cutter also must be available. Therefore, block, crane, block cutter and worker are the resources for the block carriage process. The resources are represented by places, and the processes are represented by transitions in Petri net applications. So, block (P1), crane (P2), block cutter (P4) and worker (P8) are represented by places and the block carriage process (t1) is represented by a transition.

By using this approach, Petri nets of each case and sub-case studies are constructed. Figure 10 shows the Petri net of case study 1 and the following tables give the definitions of each place and transitions of the Petri net.

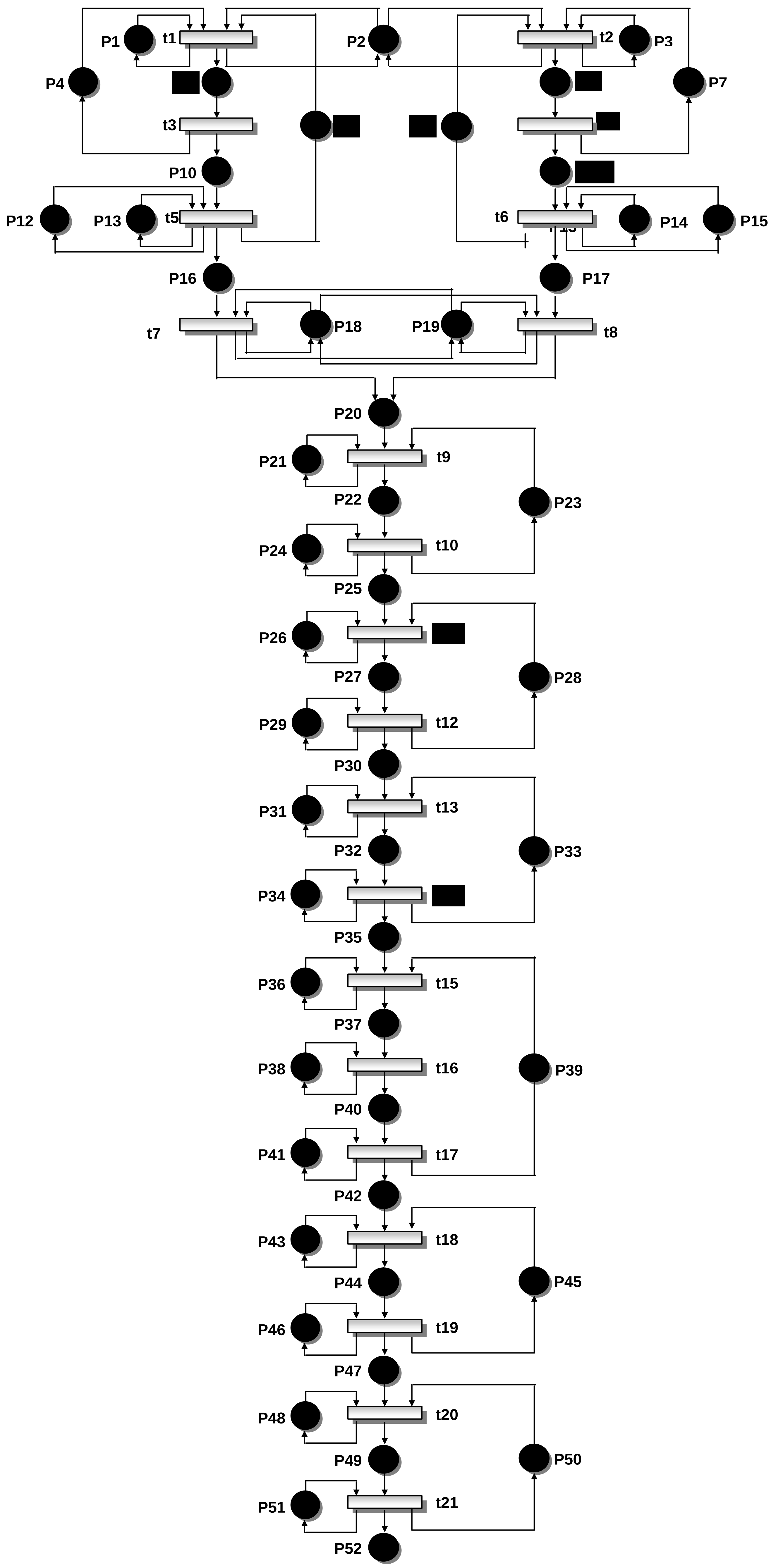


Figure 10 : Petri Net of Case Study 1

Following tables give the definitions of places and transitions of the above Petri net.

Table 2 : Place Definitions for Case Study 1

P1	Block1 is ready to be carried
P2	Crane is ready to carry a block
P3	Block2 is ready to be carried
P4	Block Cutter1 is ready to be loaded / cut
P5	Block1 is ready to be cut
P6	Block2 is ready to be cut
P7	Block Cutter2 is ready to be loaded / cut
P8	Worker is ready
P9	Worker is ready
P10	Plate is ready to be carried
P11	Plate is ready to be carried
P12	Unloader1 is ready to carry a plate
P13	Palette is ready to be loaded
P14	Palette is ready to be loaded
P15	Unloader2 is ready to carry a plate
P16	Palette is ready to be carried
P17	Palette is ready to be carried
P18	Worker is ready
P19	Forklift is ready to carry the palette
P20	Plate is ready to be loaded on the polishing line by loader
P21	Loader is to carry a plate
P22	Plate is ready to cut
P23	Worker is ready
P24	Head cutter is ready
P25	Plate is ready to be calibrated
P26	Calibrator is ready
P27	Plate is ready to pre-polishing
P28	Worker is ready
P29	Pre-polishing machine is ready
P30	Plate is ready to polishing
P31	Polishing machine is ready
P32	Plate is ready to sizing
P33	Worker is ready
P34	Sizing machine is ready
P35	Plate is ready to side calibrating & bevel cutting
P36	Side calibrating & bevel cutting machine is ready
P37	Plate is ready to rotating
P38	Rotating unit is ready
P39	Worker is ready
P40	Plate is ready to side calibrating & bevel cutting

Table 2 : (continued)

P41	Side calibrating & bevel cutting machine is ready
P42	Plate is ready to drying and cleaning
P43	Drying and cleaning unit is ready
P44	Plate is ready to quality control
P45	Worker is ready
P46	Quality control unit is ready
P47	Plate is ready to packaging
P48	Palette is ready to be loaded
P49	Palette is ready to be carried to the stock side
P50	Worker is ready
P51	Crane is ready to carry a palette
P52	Palette is ready to be dispatched

Table 3 : Transition Definitions for Case Study 1

t1	Carrying a block by the crane from the stock side to Block Cutter 1
t2	Carrying a block by the crane from the stock side to Block Cutter 2
t3	Cutting the block into plates with the dimensions of 30 x Length Free x 1 cm
t4	Cutting the block into plates with the dimensions of 60 x Length Free x 2 cm
t5	Loading the cut plates on to a palette by unloader 1
t6	Loading the cut plates on to a palette by unloader 2
t7	Carrying the palette by a forklift from unloader 1 to the polishing line
t8	Carrying the palette by a forklift from unloader 2 to the polishing line
t9	Loading the plates (30 x LF x 1 cm) by loader from palette to the polishing line
t10	Cutting the heads of the plates (30 x LF x 1 cm) by the head cutter
t11	Calibrating the plates (30 x LF x 1 cm) by the calibrator
t12	Pre-polishing the plates (30 x LF x 1 cm) by the pre-polishing machine
t13	Polishing the plates (30 x LF x 1 cm) by the polishing machine
t14	Sizing the plates (30 x LF x 1 cm) by the by the sizing machine
t15	Side calibrating and bevel cutting of the first opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t16	Rotating the plates (30 x 30 x 1 cm) by the rotating unit
t17	Side calibrating and bevel cutting of the second opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t18	Drying and cleaning the plates (30 x 30 x 1 cm) by the drying and cleaning unit
t19	Quality control of the plates (30 x 30 x 1 cm)
t20	Packaging the plates (30 x 30 x 1 cm)
t21	Carrying the palette with the plates (30 x 30 x 1 cm) by the crane
t9’	Loading the plates (60 x LF x 2 cm) by loader from palette to the polishing line
t10’	Cutting the heads of the plates (60 x LF x 2 cm) by the head cutter
t11’	Calibrating the plates (60 x LF x 2 cm) by the calibrator
t12’	Pre-polishing the plates (60 x LF x 2 cm) by the pre-polishing machine

Table 3 : (continued)

t13’	Polishing the plates (60 x LF x 2 cm) by the polishing machine
t14’	Sizing the plates (60 x LF x 2 cm) by the by the sizing machine
t15’	Side calibrating and bevel cutting of the first opposite sides of the plates (60 x 30 x 2 cm) by the side calibrating and bevel cutting machine
t16’	Rotating the plates (60 x 30 x 2 cm) by the rotating unit
t17’	Side calibrating and bevel cutting of the second opposite sides of the plates (60 x 30 x 2 cm) by the side calibrating and bevel cutting machine
t18’	Drying and cleaning the plates (60 x 30 x 2 cm) by the drying and cleaning unit
t19’	Quality control of the plates (60 x 30 x 2 cm)
t20’	Packaging the plates (60 x 30 x 2 cm)
t21’	Carrying the palette with the plates (60 x 30 x 2 cm) by the crane

4.4.2.2 Matrix Representation

In this section, matrix representation of case study is given according to the instructions mentioned in Chapter 3.2. Matrix calculations are given in the following sections.

4.4.2.2.1 D- Matrix

As mentioned in Chapter 3.2, D- matrix is an m x n matrix. m is the number of transitions and n is the number of places. In Case Study 1, there are 21 transitions and 52 places. This means that m is 21 and n is 52. Therefore, D- matrix will be a matrix, which has 21 rows and 52 columns.

When constructing D- matrix, 1 is put if i^{th} place is an input place of j^{th} transition; and 0 is put if i^{th} place is not an input place of j^{th} transition.

P1, P2, P4, and P8 are the input places of transition t1. Thus, 1 is put in $P1^{th}$, $P2^{th}$, $P4^{th}$, and $P8^{th}$ columns of $t1^{th}$ row. For other places, 0 is put in $t1^{th}$ row. By using the same logic, D- matrix is formed. D- matrix is given in Figure 11.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47	P48	P49	P50	P51	P52
t1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t5	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t6	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
t16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
t18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0
t19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
t20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0
t21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Figure 11 : D- Matrix for Case Study 1

4.4.2.2.2 D+ Matrix

Like D- matrix, D+ matrix is also an m x n matrix. m and n show the number of rows and transitions respectively. Petri net is same, so, m is 21 and n is 52. However, different from D- matrix, D+ matrix shows the output places of each transition. In other words, during constructing D+ matrix, 1 is placed if i^{th} place is an output place of j^{th} transition. If i^{th} place is not an output of j^{th} transition, then 0 is placed.

P1, P2 and P5 are the output places of transition t1. Therefore, 1 is placed in $P1^{th}$, $P2^{th}$ and $P5^{th}$ columns of $t1^{th}$ row; and , 0 is placed for other columns of $t1^{th}$ row. Then, D+ matrix is formed, by using the same approach. D+ matrix is given in Figure 12.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47
t1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t4	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t5	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t6	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
t15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
t16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
t17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0
t18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
t19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
t20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12 : D+ Matrix for Case Study 1

4.4.2.2.3 D matrix

After constructing D- and D+ matrices, D matrix can be calculated by subtracting D- matrix from D+ matrix. Thus, D = (D+) – (D-). D matrix also has 21 rows and 52 columns.

D matrix is given in Figure 13.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26
t1	0	0	0	-1	1	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	0	0	0	0	1	-1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	1	-1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t4	0	0	0	0	0	-1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t5	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
t6	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
t7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	1	0	0	0	0	0	0
t8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	1	0	0	0	0	0	0
t9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	-1	0	0	0
t10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	1	0
t11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0
t12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47	P48	P49	P50	P51	P52
t1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t11	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t12	-1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t13	0	0	0	-1	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t14	0	0	0	0	0	-1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t15	0	0	0	0	0	0	0	0	-1	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
t16	0	0	0	0	0	0	0	0	0	0	-1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
t17	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	1	0	0	0	0	0	0	0	0	0	0
t18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	-1	0	0	0	0	0	0	0
t19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	1	0	0	0	0	0
t20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	-1	0	0
t21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	1

Figure 13 : D Matrix for Case Study 1

4.4.2.2.4 Marking Matrices

Marking Matrices are computed for Case Study1. Marking Matrices of the first alternative path for Case Study 1 is given as follows :

M0 = [1111001110011110011010110101101011010110101011010110]
M1 = [1110101010011110011010110101101011010110101011010110]
M2 = [1111001011011110011010110101101011010110101011010110]
M3 = [1111001110011111011010110101101011010110101011010110]
M4 = [1111001110011110011110110101101011010110101011010110]
M5 = [1111001110011110011011010101101011010110101011010110]
M6 = [1111001110011110011010111101101011010110101011010110]
M7 = [1111001110011110011010110110101011010110101011010110]
M8 = [1111001110011110011010110101111011010110101011010110]
M9 = [1111001110011110011010110101101101010110101011010110]
M10 = [1111001110011110011010110101101011110110101011010110]
M11 = [1111001110011110011010110101101011011100101011010110]
M12 = [1111001110011110011010110101101011010101101011010110]
M13 = [1111001110011110011010110101101011010110111011010110]
M14 = [1111001110011110011010110101101011010110101101010110]
M15 = [1111001110011110011010110101101011010110101011110110]
M16 = [1111001110011110011010110101101011010110101011011010]
M17 = [1111001110011110011010110101101011010110101011010111]
M18 = [1111010100011110011010110101101011010110101011010111]
M19 = [1111001100111110011010110101101011010110101011010111]
M20 = [1111001110011110111010110101101011010110101011010111]
M21 = [1111001110011110011110110101101011010110101011010111]
M22 = [1111001110011110011011010101101011010110101011010111]
M23 = [1111001110011110011010111101101011010110101011010111]

M24 = [1111001110011110011010110110101011010110101011010111]

M25 = [1111001110011110011010110101110110101101010110101 11]

M26 = [1111001110011110011010110101101101010110101011010111]

M27 = [1111001110011110011010110101101011101101010110101 11]

M28 = [1111001110011110011010110101101011011100101011010111]

M29 = [1111001110011110011010110101101011010101101011010111]

M30 = [1111001110011110011010110101101011010110111011010111]

M31 = [1111001110011110011010110101101011010110101101010111]

M32 = [1111001110011110011010110101101011010110101011110111]

M33 = [1111001110011110011010110101101011010110101011011011]

M34 = [1111001110011110011010110101101011010110101011010112]

4.4.2.2.5 Matrix Calculations

Main purpose of the matrix calculations is to get next markings of each transition firing; i.e. process occurring sequence. For this goal, the marking calculations are performed for Case Study 1 as an example.

If the first line; i.e., t1, t3, t5, t7, t9, t10, t11, t12, t13, t14, t15, t16, t17, t18, t19, t20, t21, t2, t4, t6, t8, t9', t10', t11', t12', t13', t14', t15', t16', t17', t18', t19', t20', t21', is considered, the following calculations are obtained :

If t1 fires;

t1 fires means that t1 is performed. For calculating next marking when t1 fires, transition matrix, marking matrix and D matrix should be constituted.

When transition matrix is formed, 1 is put for firing transitions and 0 is put for others. Therefore, if t1 fires, put 1 for t1 and put 0 for other transitions. So; the transition matrix for t1 firing is as follows :

Transition Matrix = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

When constructing marking matrix, 1's represent the places having a token and 0's represent the places having no token. Marking matrix indicated the available resources before firing. Therefore, initial marking (M0) should be constructed before t1 fires. P1, P2, P3, P4, P7, P8, P9, P12, P13, P14, P15, P18, P19, P21, P23, P24, P26, P28, P29, P31, P33, P34, P36, P38, P39, P41, P43, P45, P46, P48, P50, and P51 are the available resources. Thus, marking matrix is found as :

Marking Matrix (M0) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

The next marking, after t1 fires, is calculated by using the following formula.

([Transition Matrix] [D]) + [Marking Matrix] = [Next Marking]

As a result, next marking (M1) is calculated as;

Next Marking (M1) = [1 1 1 0 1 0 1 0 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

All next markings for each transition firings can be computed by the same way.

If t3 fires;

Transition Matrix = [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Marking Matrix (M1) = [1 1 1 0 1 0 1 0 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]
([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]
Next Marking (M2) = [1 1 1 1 0 0 1 0 1 1 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

If t5 fires;

Transition Matrix = [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Marking Matrix (M2) = [1 1 1 1 0 0 1 0 1 1 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M3) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

If t7 fires;

Transition Matrix = [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Marking Matrix (M3) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M4) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

If t9 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
Marking Matrix (M4) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M5) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0

If t10 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
Marking Matrix (M5) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M6) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

If t11 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
Marking Matrix (M6) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M7) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

If t12 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
Marking Matrix (M7) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M8) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

If t13 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
Marking Matrix (M8) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M9) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

If t14 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
Marking Matrix (M9) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M10) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]

If t15 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
Marking Matrix (M10) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M11) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0]

If t16 fires;
Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
Marking Matrix (M11) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M12) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0]

If t17 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

Marking Matrix (M12) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M13) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 1 1 0]

If t18 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]

Marking Matrix (M13) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 1 1 0]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M14) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0]

If t19 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]

Marking Matrix (M14) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M15) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 1 1 0 1 1 0]

If t20 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]

Marking Matrix (M15) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 1 1 0 1 1 0]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M16) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 1 0]

If t21 fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]

Marking Matrix (M16) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 1 0]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M17) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t2 fires;

Transition Matrix = [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Marking Matrix (M17) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M18) = [1 1 1 1 0 1 0 1 0 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t4 fires;

Transition Matrix = [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Marking Matrix (M18) = [1 1 1 1 0 1 0 1 0 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M19) = [1 1 1 1 0 0 1 1 0 0 1 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t6 fires;

Transition Matrix = [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Marking Matrix (M19) = [1 1 1 1 0 0 1 1 0 0 1 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M20) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t8 fires;

Transition Matrix = [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Marking Matrix (M20) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]
([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*
Next Marking (M21) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t9' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0]

Marking Matrix (M21) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M22) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t10' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]

Marking Matrix (M22) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M23) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t11' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]

Marking Matrix (M23) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M24) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t12' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]

Marking Matrix (M24) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M25) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t13' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]

Marking Matrix (M25) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M26) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t14' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]

Marking Matrix (M26) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M27) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t15' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]

Marking Matrix (M27) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*

Next Marking (M28) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 1]

If t16' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]

Marking Matrix (M28) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*

Next Marking (M29) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 1]

If t17' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

Marking Matrix (M29) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + *(Marking Matrix] = [Next Marking]*

Next Marking (M30) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 1 1 1]

If t18' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]

Marking Matrix (M30) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M31) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 1]

If t19' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]

Marking Matrix (M31) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M32) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 1 1 0 1 1 1]

If t20' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]

Marking Matrix (M32) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 1 1 0 1 1 1]

([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]

Next Marking (M33) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 1 1]

If t21' fires;

Transition Matrix = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
Marking Matrix (M33) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 1 1]
([Transition Matrix] [D]) + (Marking Matrix] = [Next Marking]
Next Marking (M34) = [1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 2]

4.4.2.3 Reachability Graph

Reachability Graph or Reachability Tree represents all of the obtainable markings for a Petri net. The tree begins with the initial marking and continues towards a new marking for each enabled transition. The process is repeated for each new marking, until all possible markings are found.

Therefore, before constructing the reachability graph, all possible markings should be determined. Next marking calculations of the first 34 markings for Case Study 1 can be found in the previous section.

After calculating the marking matrices, reachability graph can be constructed. Reachability Graphs of the Case Studies are drawn in the light of the assumptions given in Chapter 4.3. Reachability Graphs of each case and sub-case studies are depicted in Appendices. Appendix A shows the entire reachability graph of Case Study 1.

In the following figure, a small part of reachability graph of Case Study 1 is shown. As explained above, markings are calculated to determine the enabled transitions. This means that the markings are used to understand the processes, which can be performed.

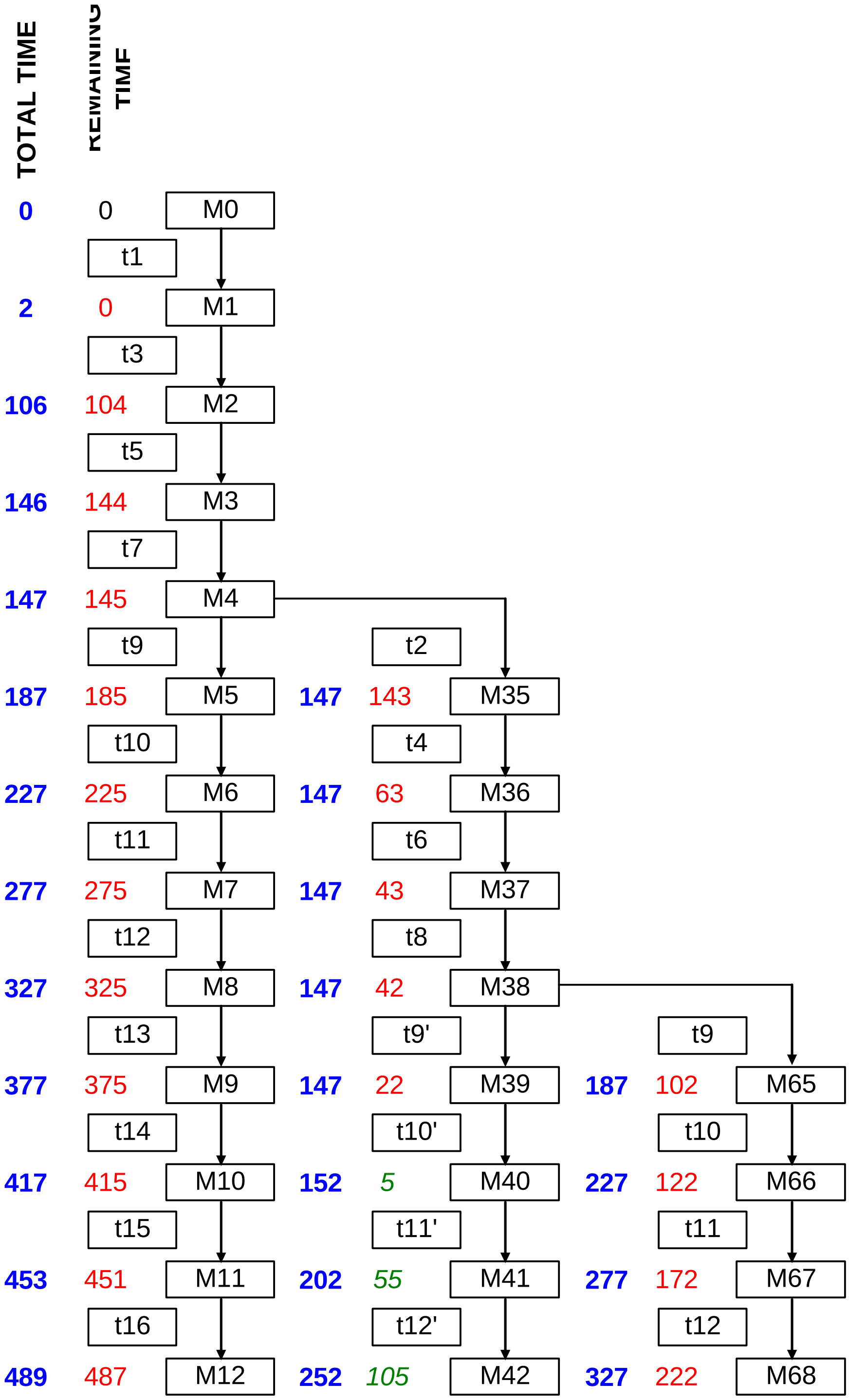


Figure 14 : A part of the Reachability Graph of Case Study 1

Therefore, first of all, initial marking (M0) is determined. Initial marking shows that the transitions t1 and t2 are the enabled transitions. So, t1 or t2 can be fired; in other word, t1 or t2 can be executed. However, having only one crane, both of them cannot be applied at the same time. Therefore, t1 is carried out first and t2 is applied second or visa versa. If t1 is fired, the next marking, which is M1, is obtained. M1 shows that t2 and t3 are the enabled transitions. If t3 is fired, M2 shows the enabled transition is only t5. as assuming that when a plate is cut by the block cutter, it should be unloaded from the block cutter by unloader immediately.

If t5 is fired, M3 is obtained and the new enabled transitions are found as t2 and t7. When t7 is fired, t2 and t3 become ready to be fired. If t9 is fired, only t10 can be fired. Because, it is assumed that when polishing line starts to process any type of product, the process must be completed for 10 m^2 products. Therefore, the sequence of transition firing is from t9 to t21 should be completed.

If t2 is fired after calculating M4, M35 is obtained. Then, t4 and t9 are the new enabled transitions. But, if t4 is fired, only t6 can be performed because of the assumption of when a plate is cut by the block cutter, it should be unloaded from the block cutter by unloader immediately. After getting M37, there are two alternatives for firing, t8 and t9. If t8 fires, t9 and t9' are found as the enabled transitions. If t9' is performed, the sequence should be followed up to t21'. Because, when the polishing line starts to process any type of product, the process must be completed for 10 m^2 products.

However, if t9' is fired after the sequence from t9 to t21 should be covered regarding the assumption described above.

The same process is repeated up to get all possible paths are covered. Examining all the possibilities 70 different process sequences are found for Case Study 1.

4.4.2.4 Time Representation

In the previous sections the assumptions required for the proper examination of production system were described. This section focuses on the reasons for these assumptions and how these assumptions shape the marble production system. Firstly inputs and outputs are assumed to be homogeneous within in each level of production. In other words, input (or output) of a process is identical to each other. On the other hand, blocks as the first input of the production lines are considered as uniform. Therefore, the production times of cutting line is taken under control. In the reachability graph the production times of each process are determined relative to each other. So any mistake caused by the change in the product and raw material characteristics is prevented. This relative timing is constructed for the production of 10 m^2 of each type of product. Contributing to find the optimum time in reachability graph it is assumed that before finishing all the material in a palette no material of the other palette is processed in the polishing line. Thus, the time spent for the adjustment of machinery is recovered to a certain extent.

Looking at all these descriptions, it is seen that assumptions do not cause any change in the sequence of operations and they do not increase the total production time. The most important thing brought by assumptions is to have same level of production in two cutting lines with polishing line. This is provided by increasing the speed of production in cutting line and decreasing the speed of polishing line. This adjustment is required as an obligation of flexible manufacturing systems.

In this section, total completion times for each alternative path, which can be followed, is aimed to be computed. Before calculating total completion time for each path, reachability graph should be constructed.

Two different time values are used for the time representation of Petri net. These time values are namely, total time and remaining time.

First of all, the completion times of each process for producing $10m^2$ product should be known. These times are given in the following table for Case Study 1. these time values are determined relative to each other. So, the unit times are used.

Table 4 : Time Durations for Production of Product 1 and Product 2 together by 2 block cutters.

t1	2t	t18	20t
t2	2t	t19	20t
t3	104t	t20	10t
t4	80t	t21	5t
t5	40t	t9'	20t
t6	20t	t10’	27t
t7	t	t11’	50t
t8	t	t12’	50t
t9	40t	t13’	40t
t10	40t	t14’	25t
t11	50t	t15’	24t
t12	50t	t16’	24t
t13	50t	t17’	24t
t14	40t	t18’	20t
t15	36t	t19’	20t
t16	36t	t20’	10t
t17	36t	t21’	5t

Total time represents the total production time. Remaining time has a meaning that the time in which the following process can occur at the same time with other branch's processes. These time values can be seen on the previous figure.

If the first path is followed, both total and remaining times are zero at the initial condition. If t1 is fired, the completion time of t1 is 2t, so the total time will be 2t, but remaining time is still 0. Because, there isn't any process that can be performed before applying t1.

Then, t3, which has an occurrence time of 104t, is fired. This means that 104t is already passed, so, it should be added to the previous total time. In other words, the new total time is 2t + 104t = 106t. On the other hand, remaining time will be 0 + 104t = 104t.

t5 can be fired after t3 is fired. Therefore, the occurrence time of t5 should be added both total time and remaining time. Thus, the new total time is 106t + 40t = 146t, and remaining time is 104t + 40t = 106t.

t7 cannot be fired before t5 is fired. So, time duration of t7 should be added to both total and remaining times. Total time and remaining time will be 147t and 145t, respectively.

From t9 to t21, the same logic should be worked, and time durations should be added to both total time and remaining time.

If t2 is fired after t7, total time will remains the same. But, the occurrence time of t2 should be subtracted from the remaining time for calculating the new remaining time. Because, t2 can be performed at the same time with its preceding processes.

From t2 to t8, the processing sequence should be followed. Therefore, up to t8, total time will be the same but remaining time will be decreased by occurrence time.

If t9’ is fired after t8 is fired, total time will also be same, but, remaining time will be decreased by the occurrence time of t9’; i.e., 20t. In other word, the total time will be 147t, but, remaining time will be 42t – 20t = 22t.

Remaining time is 22t after t9’ is fired. However, occurrence time of t10’ is 27t. so, 22t can be performed at the same time with the beforehand done process on the first branch; and 5t will be added to the previous remaining time. 5t will also be added to the total time. As a result, the new remaining time is 5t, and the new total time is 147t + 5t = 152t.

After t10’ is fired, the sequence should be followed up to t21’. Therefore, total and remaining times are calculated by adding the time durations to the previous total and remaining times.

If t9 is fired after t8, the processing sequence should be followed up to t21. So, total and remaining times are calculated by adding the time durations.

As a result, the total times and the remaining times of each transition firings are computed for 70 alternative paths, and they are given on Appendix A.

4.4.2.5 Results of Case Study 1

In Case Study 1, two different products with different dimensions, 30x30x1 cm and 60x60x2 cm, are produced by using two identical block cutters. The aim of this case study is to produce 10 m^2 apiece of each product in minimum possible production time duration.

To get this purpose, firstly, Petri net of the production system is drawn. Petri net provides to be seen all processes, all available and non-available processes, clearly. Then, marking matrices are computed, that helps to draw the reachability graph. During drawing the reachability graph, some assumptions are considered. When, the reachability graph is constructed, it is seen that there are 70 alternative ways that can be followed.

Then, the time analysis of the system is done. Total times and remaining times are calculated for all transition firings. Thus, minimum total time of the first case study is calculated as 580t.

4.4.3 Case Study 2

In Case Study 2, both of the products having the dimensions of 30x30x1 cm and 60x60x2 cm are produced separately. In the first sub-case study, only the product with 30x30x1 cm dimensions is produced, and in the second sub-case study, only the product with 60x60x2 cm dimensions is produced. Two identical block cutters are used in both of the two sub-case studies. However, in sub-case study 1, both of the block cutters cut the plate with 30 x length free x 1 cm; on the other hand, in sub-case study 1, both of them cut the plate with 60 x length free x 2 cm. Like Case Study 1, there is only one polishing line. But, both of the block cutters produce the same product, so, which of the palette is came to the polishing line first is not so important.

4.4.3.1 Petri Net

In Case Study2, both of the two products mentioned in the previous section are investigated separately. In other words, it is assumed that plant produces only one product in both block cutters. This case is examined for each product separately. In the following figures, the Petri Nets of such systems are given.

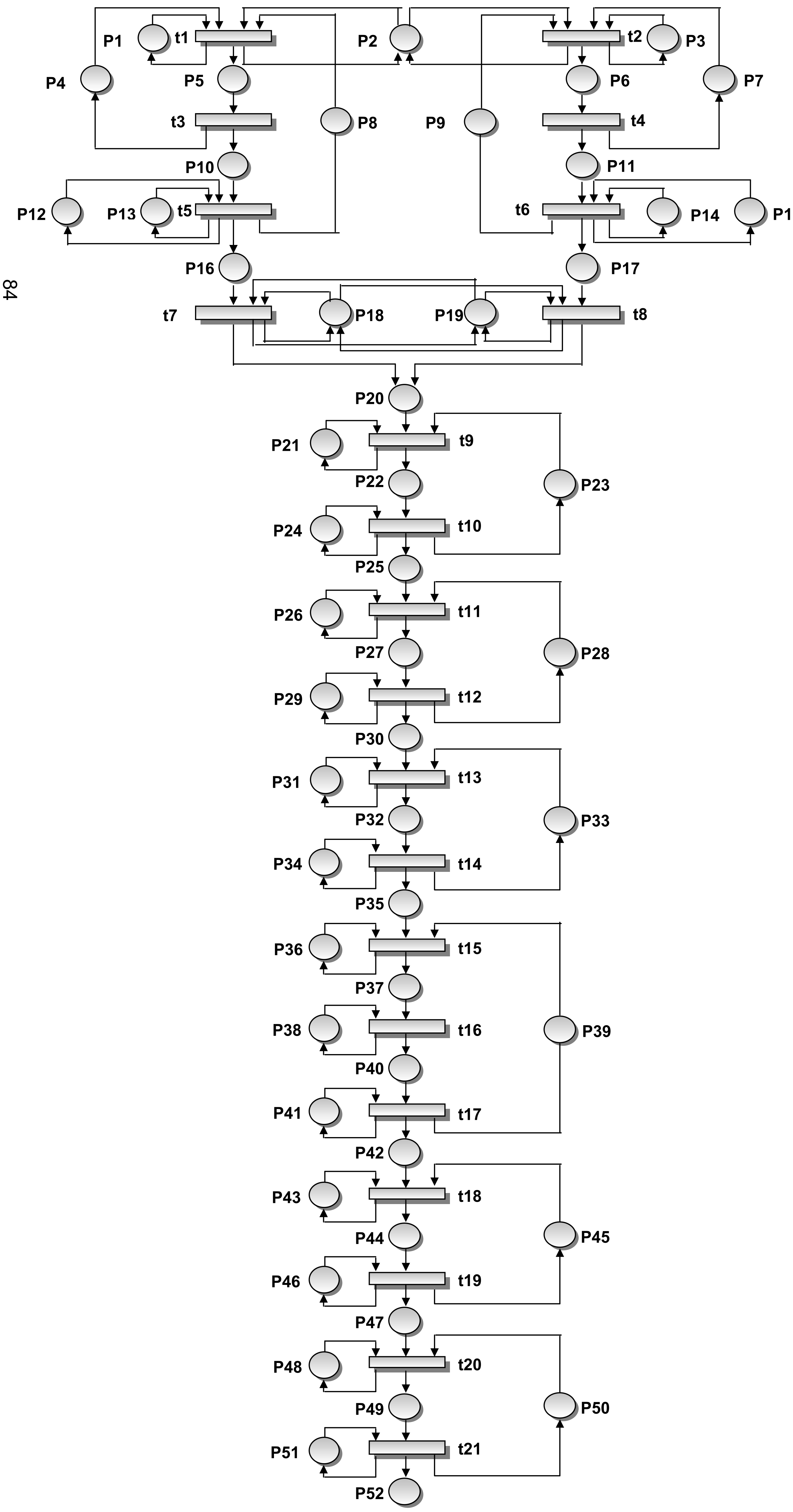


Figure 15 : Petri Net for Sub-case Study 1 of Case Study 2

Table 5 : Place Definitions for Sub-case Study 1 of Case Study 2

P1	Block1 is ready to be carried
P2	Crane is ready to carry a block
P3	Block2 is ready to be carried
P4	Block Cutter1 is ready to be loaded / cut
P5	Block1 is ready to be cut
P6	Block2 is ready to be cut
P7	Block Cutter2 is ready to be loaded / cut
P8	Worker is ready
P9	Worker is ready
P10	Plate is ready to be carried
P11	Plate is ready to be carried
P12	Unloader1 is ready to carry a plate
P13	Palette is ready to be loaded
P14	Palette is ready to be loaded
P15	Unloader2 is ready to carry a plate
P16	Palette is ready to be carried
P17	Palette is ready to be carried
P18	Worker is ready
P19	Forklift is ready to carry the palette
P20	Plate is ready to be loaded on the polishing line by loader
P21	Loader is to carry a plate
P22	Plate is ready to cut
P23	Worker is ready
P24	Head cutter is ready
P25	Plate is ready to be calibrated
P26	Calibrator is ready
P27	Plate is ready to pre-polishing
P28	Worker is ready
P29	Pre-polishing machine is ready
P30	Plate is ready to polishing
P31	Polishing machine is ready
P32	Plate is ready to sizing
P33	Worker is ready
P34	Sizing machine is ready
P35	Plate is ready to side calibrating & bevel cutting
P36	Side calibrating & bevel cutting machine is ready
P37	Plate is ready to rotating
P38	Rotating unit is ready
P39	Worker is ready
P40	Plate is ready to side calibrating & bevel cutting
P41	Side calibrating & bevel cutting machine is ready
P42	Plate is ready to drying and cleaning
P43	Drying and cleaning unit is ready
P44	Plate is ready to quality control
P45	Worker is ready
P46	Quality control unit is ready
P47	Plate is ready to packaging
P48	Palette is ready to be loaded
P49	Palette is ready to be carried to the stock side
P50	Worker is ready
P51	Crane is ready to carry a palette
P52	Palette is ready to be dispatched

Table 6 : Transition Definitions for Sub-case Study 1 of Case Study 2

t1	Carrying a block by the crane from the stock side to Block Cutter 1
t2	Carrying a block by the crane from the stock side to Block Cutter 2
t3	Cutting the block into plates with the dimensions of 30 x Length Free x 1 cm
t4	Cutting the block into plates with the dimensions of 30 x Length Free x 1 cm
t5	Loading the cut plates on to a palette by unloader 1
t6	Loading the cut plates on to a palette by unloader 2
t7	Carrying the palette by a forklift from unloader 1 to the polishing line
t8	Carrying the palette by a forklift from unloader 2 to the polishing line
t9	Loading the plates (30xLFx1 cm) by the loader from palette to the polishing line
t10	Cutting the heads of the plates (30 x LF x 1 cm) by the head cutter
t11	Calibrating the plates (30 x LF x 1 cm) by the calibrator
t12	Pre-polishing the plates (30 x LF x 1 cm) by the pre-polishing machine
t13	Polishing the plates (30 x LF x 1 cm) by the polishing machine
t14	Sizing the plates (30 x LF x 1 cm) by the by the sizing machine
t15	Side calibrating and bevel cutting of the first opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t16	Rotating the plates (30 x 30 x 1 cm) by the rotating unit
t17	Side calibrating and bevel cutting of the second opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t18	Drying and cleaning the plates (30 x 30 x 1 cm) by the drying and cleaning unit
t19	Quality control of the plates (30 x 30 x 1 cm)
t20	Packaging the plates (30 x 30 x 1 cm)
t21	Carrying the palette with the plates (30 x 30 x 1 cm) by the crane
t9’	Loading the plates (30xLFx1 cm) by the loader from palette to the polishing line
t10’	Cutting the heads of the plates (30 x LF x 1 cm) by the head cutter
t11’	Calibrating the plates (30 x LF x 1 cm) by the calibrator
t12’	Pre-polishing the plates (30 x LF x 1 cm) by the pre-polishing machine
t13’	Polishing the plates (30 x LF x 1 cm) by the polishing machine
t14’	Sizing the plates (30 x LF x 1 cm) by the by the sizing machine
t15’	Side calibrating and bevel cutting of the first opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t16’	Rotating the plates (30 x 30 x 1 cm) by the rotating unit
t17’	Side calibrating and bevel cutting of the second opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t18’	Drying and cleaning the plates (30 x 30 x 1 cm) by the drying and cleaning unit
t19’	Quality control of the plates (30 x 30 x 1 cm)
t20’	Packaging the plates (30 x 30 x 1 cm)
t21’	Carrying the palette with the plates (30 x 30 x 1 cm) by the crane

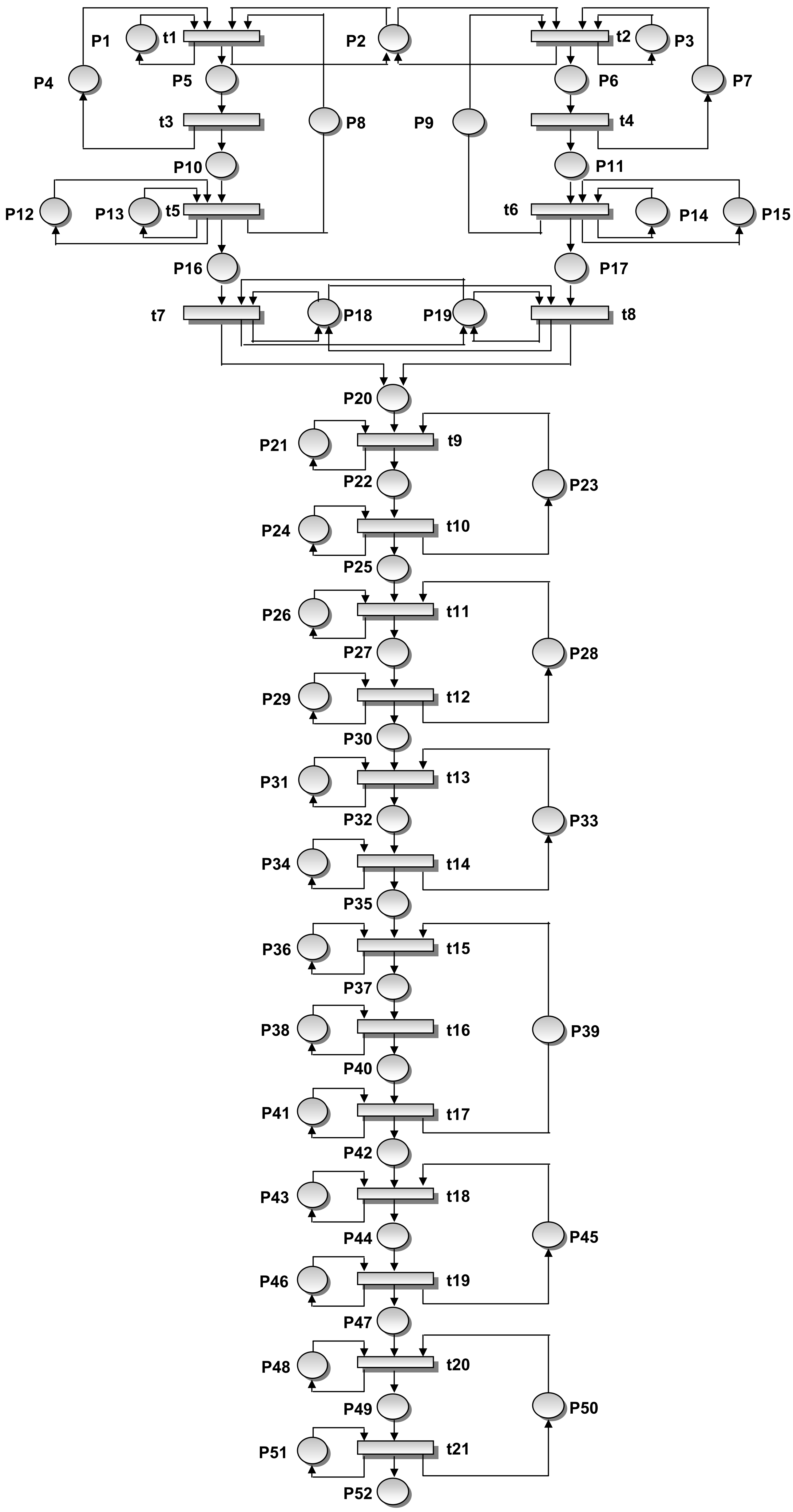


Figure 16 : Petri Net for Sub-case Study 2 of Case Study 2

Table 7 : Place Definitions for Sub-case Study 2 of Case Study 2

P1	Block1 is ready to be carried
P2	Crane is ready to carry a block
P3	Block2 is ready to be carried
P4	Block Cutter1 is ready to be loaded / cut
P5	Block1 is ready to be cut
P6	Block2 is ready to be cut
P7	Block Cutter2 is ready to be loaded / cut
P8	Worker is ready
P9	Worker is ready
P10	Plate is ready to be carried
P11	Plate is ready to be carried
P12	Unloader1 is ready to carry a plate
P13	Palette is ready to be loaded
P14	Palette is ready to be loaded
P15	Unloader2 is ready to carry a plate
P16	Palette is ready to be carried
P17	Palette is ready to be carried
P18	Worker is ready
P19	Forklift is ready to carry the palette
P20	Plate is ready to be loaded on the polishing line by loader
P21	Loader is to carry a plate
P22	Plate is ready to cut
P23	Worker is ready
P24	Head cutter is ready
P25	Plate is ready to be calibrated
P26	Calibrator is ready
P27	Plate is ready to pre-polishing
P28	Worker is ready
P29	Pre-polishing machine is ready
P30	Plate is ready to polishing
P31	Polishing machine is ready
P32	Plate is ready to sizing
P33	Worker is ready
P34	Sizing machine is ready
P35	Plate is ready to side calibrating & bevel cutting
P36	Side calibrating & bevel cutting machine is ready
P37	Plate is ready to rotating
P38	Rotating unit is ready
P39	Worker is ready
P40	Plate is ready to side calibrating & bevel cutting
P41	Side calibrating & bevel cutting machine is ready
P42	Plate is ready to drying and cleaning
P43	Drying and cleaning unit is ready
P44	Plate is ready to quality control
P45	Worker is ready
P46	Quality control unit is ready
P47	Plate is ready to packaging
P48	Palette is ready to be loaded
P49	Palette is ready to be carried to the stock side
P50	Worker is ready
P51	Crane is ready to carry a palette
P52	Palette is ready to be dispatched

Table 8 : Transition Definitions for Sub-case Study 2 of Case Study 2

t1	Carrying a block by the crane from the stock side to Block Cutter 1
t2	Carrying a block by the crane from the stock side to Block Cutter 2
t3	Cutting the block into plates with the dimensions of 60 x Length Free x 2 cm
t4	Cutting the block into plates with the dimensions of 60 x Length Free x 2 cm
t5	Loading the cut plates on to a palette by unloader 1
t6	Loading the cut plates on to a palette by unloader 2
t7	Carrying the palette by a forklift from unloader 1 to the polishing line
t8	Carrying the palette by a forklift from unloader 2 to the polishing line
t9	Loading the plates (60 x LF x 2 cm) by the loader from palette to the polishing line
t10	Cutting the heads of the plates (60 x LF x 2 cm) by the head cutter
t11	Calibrating the plates (60 x LF x 2 cm) by the calibrator
t12	Pre-polishing the plates (60 x LF x 2 cm) by the pre-polishing machine
t13	Polishing the plates (60 x LF x 2 cm) by the polishing machine
t14	Sizing the plates (60 x LF x 2 cm) by the by the sizing machine
t15	Side calibrating and bevel cutting of the first opposite sides of the plates (60 x 30 x 2 cm) by the side calibrating and bevel cutting machine
t16	Rotating the plates (60 x 30 x 2 cm) by the rotating unit
t17	Side calibrating and bevel cutting of the second opposite sides of the plates (60 x 30 x 2 cm) by the side calibrating and bevel cutting machine
t18	Drying and cleaning the plates (60 x 30 x 2 cm) by the drying and cleaning unit
t19	Quality control of the plates (60 x 30 x 2 cm)
t20	Packaging the plates (60 x 30 x 2 cm)
t21	Carrying the palette with the plates (60 x 30 x 2 cm) by the crane
t9’	Loading the plates (60 x LF x 2 cm) by the loader from palette to the polishing line
t10’	Cutting the heads of the plates (60 x LF x 2 cm) by the head cutter
t11’	Calibrating the plates (60 x LF x 2 cm) by the calibrator
t12’	Pre-polishing the plates (60 x LF x 2 cm) by the pre-polishing machine
t13’	Polishing the plates (60 x LF x 2 cm) by the polishing machine
t14’	Sizing the plates (60 x LF x 2 cm) by the by the sizing machine
t15’	Side calibrating and bevel cutting of the first opposite sides of the plates (60 x 30 x 2 cm) by the side calibrating and bevel cutting machine
t16’	Rotating the plates (60 x 30 x 2 cm) by the rotating unit
t17’	Side calibrating and bevel cutting of the second opposite sides of the plates (60 x 30 x 2 cm) by the side calibrating and bevel cutting machine
t18’	Drying and cleaning the plates (60 x 30 x 2 cm) by the drying and cleaning unit
t19’	Quality control of the plates (60 x 30 x 2 cm)
t20’	Packaging the plates (60 x 30 x 2 cm)
t21’	Carrying the palette with the plates (60 x 30 x 2 cm) by the crane

4.4.3.2 Reachability Graph

Reachability Graphs are drawn for each sub-case study. Each of them is examined separately. Both of the sub case studies have 25 different alternative ways. When computing time analyze of each case, it is seen that each branch have the equal total times. It is valid for both cases. The main reason for this result is that both block cutter s are identical and they process the same products. Therefore, process times are equal to each other.

Minimum total time is of the first sub-case study is get as 582t by the following process occurring sequences of all alternatives are the same. But minimum total time is of the second sub-case study is get as 454t by following the process occurring sequences of all alternatives are the same.

On account of having different processing times, the two sub-case studies have different minimum total time durations. The reachability graphs of both sub-cases are similar; therefore, reachability graph of the first one in Appendix B.

4.4.3.3 Time Representation

In Case Study2, for both of the two products are evaluated separately. It means that, the condition is estimated for each product one by one. Consequently, following two tables indicate the time durations for each sub-case study.

Table 9 : Time Durations for Production of Product 1 by 2 block cutters.

t1	2t	t12	50t
t2	2t	t13	50t
t3	104t	t14	40t
t4	104t	t15	36t
t5	40t	t16	36t
t6	40t	t17	36t
t7	t	t18	20t
t8	t	t19	20t
t9	40t	t20	10t
t10	40t	t21	5t
t11	50t		

Table 10 : Time Durations for Production of Product 2 by 2 block cutters.

t1	2t	t12	50t
t2	2t	t13	40t
t3	80t	t14	25t
t4	80t	t15	24t
t5	20t	t16	24t
t6	20t	t17	24t
t7	t	t18	20t
t8	t	t19	20t
t9	20t	t20	10t
t10	27t	t21	5t
t11	50t		

4.4.3.4 Results of Case Study 2

In Case Study 2, both of the products produced in the previous case study are examined in different sub-case studies. Like the first case study, two identical block cutters are also used in this case study.

To obtain the minimum total production times for each product, Petri nets of each sub-case study are drawn separately. Then, the reachability graphs are depicted. When the reachability graphs of each sub-case study are drawn, 25 alternative paths are obtained for each of them.

Then, minimum total production times are found as 582t and 454t, respectively.

4.4.4 Case Study 3

In the last Case Study, both of the two products, with the dimensions of 30x30x1 cm and 60x60x2 cm, are processed separately, in the sub-case studies. In sub-case study 1, the first product having 30x30x1 cm dimensions is processed. In sub-case study 2, only the product with 60x60x2 cm dimensions is processed. Different from the previous case studies, only one block cutter is used to cut the blocks into plates in both of the sub-case studies of Case Study 3. There is also only one polishing line in this case study.

4.4.4.1 Petri Net

In Case Study3, both of the two products mentioned in the Chapter 4.2 are investigated separately like in previous section. However, in this case, it is assumed that there is only one ST. This case, is also examined for each product separately. In the following figure, the Petri Nets of such systems are given.

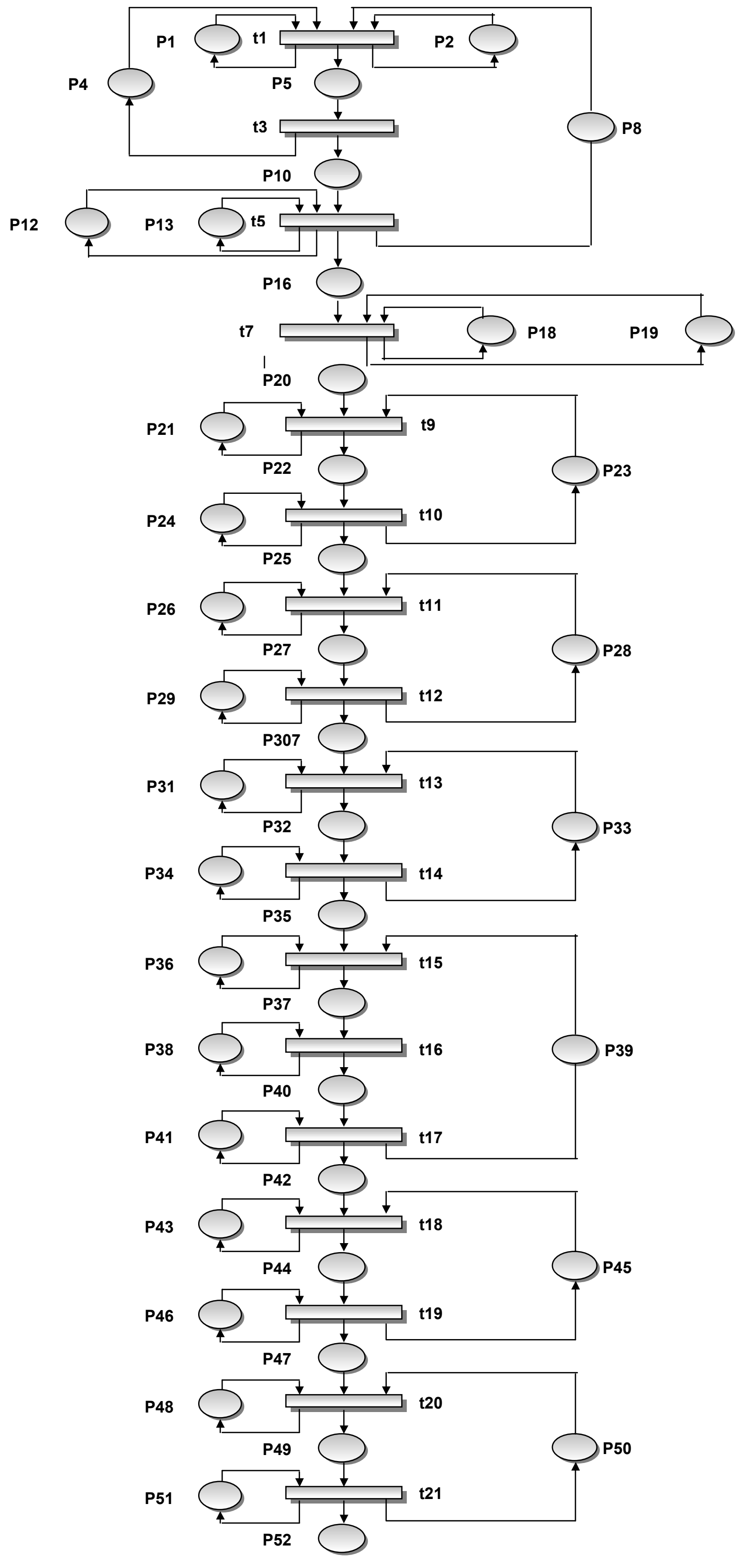


Figure 17 : Petri Net for Sub-case Study 1 of Case Study 3

Table 11 : Place Definitions for Sub-case Study 1 of Case Study 3

P1	Block is ready to be carried
P2	Crane is ready to carry a block
P4	Block Cutter is ready to be loaded / cut
P5	Block is ready to be cut
P8	Worker is ready
P10	Plate is ready to be carried
P12	Unloader is ready to carry a plate
P13	Palette is ready to be loaded
P16	Palette is ready to be carried
P18	Worker is ready
P19	Forklift is ready to carry the palette
P20	Plate is ready to be loaded on the polishing line by loader
P21	Loader is to carry a plate
P22	Plate is ready to cut
P23	Worker is ready
P24	Head cutter is ready
P25	Plate is ready to be calibrated
P26	Calibrator is ready
P27	Plate is ready to pre-polishing
P28	Worker is ready
P29	Pre-polishing machine is ready
P30	Plate is ready to polishing
P31	Polishing machine is ready
P32	Plate is ready to sizing
P33	Worker is ready
P34	Sizing machine is ready
P35	Plate is ready to side calibrating & bevel cutting
P36	Side calibrating & bevel cutting machine is ready
P37	Plate is ready to rotating
P38	Rotating unit is ready
P39	Worker is ready
P40	Plate is ready to side calibrating & bevel cutting
P41	Side calibrating & bevel cutting machine is ready
P42	Plate is ready to drying and cleaning
P43	Drying and cleaning unit is ready
P44	Plate is ready to quality control
P45	Worker is ready
P46	Quality control unit is ready
P47	Plate is ready to packaging
P48	Palette is ready to be loaded
P49	Palette is ready to be carried to the stock side
P50	Worker is ready
P51	Crane is ready to carry a palette
P52	Palette is ready to be dispatched

Table 12 : Transition Definitions for Sub-case Study 1 of Case Study 3

t1	Carrying a block by the crane from the stock side to Block Cutter
t3	Cutting the block into plates with the dimensions of 30 x Length Free x 1 cm
t5	Loading the cut plates on to a palette by unloader
t7	Carrying the palette by a forklift from unloader 1 to the polishing line
t9	Loading the plates (30xLFx1 cm) by the loader from palette to the polishing line
t10	Cutting the heads of the plates (30 x LF x 1 cm) by the head cutter
t11	Calibrating the plates (30 x LF x 1 cm) by the calibrator
t12	Pre-polishing the plates (30 x LF x 1 cm) by the pre-polishing machine
t13	Polishing the plates (30 x LF x 1 cm) by the polishing machine
t14	Sizing the plates (30 x LF x 1 cm) by the by the sizing machine
t15	Side calibrating and bevel cutting of the first opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t16	Rotating the plates (30 x 30 x 1 cm) by the rotating unit
t17	Side calibrating and bevel cutting of the second opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t18	Drying and cleaning the plates (30 x 30 x 1 cm) by the drying and cleaning unit
t19	Quality control of the plates (30 x 30 x 1 cm)
t20	Packaging the plates (30 x 30 x 1 cm)
t21	Carrying the palette with the plates (30 x 30 x 1 cm) by the crane
t9’	Loading the plates (30xLFx1 cm) by the loader from palette to the polishing line
t10’	Cutting the heads of the plates (30 x LF x 1 cm) by the head cutter
t11’	Calibrating the plates (30 x LF x 1 cm) by the calibrator
t12’	Pre-polishing the plates (30 x LF x 1 cm) by the pre-polishing machine
t13’	Polishing the plates (30 x LF x 1 cm) by the polishing machine
t14’	Sizing the plates (30 x LF x 1 cm) by the by the sizing machine
t15’	Side calibrating and bevel cutting of the first opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t16’	Rotating the plates (30 x 30 x 1 cm) by the rotating unit
t17’	Side calibrating and bevel cutting of the second opposite sides of the plates (30 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t18’	Drying and cleaning the plates (30 x 30 x 1 cm) by the drying and cleaning unit
t19’	Quality control of the plates (30 x 30 x 1 cm)
t20’	Packaging the plates (30 x 30 x 1 cm)
t21’	Carrying the palette with the plates (30 x 30 x 1 cm) by the crane

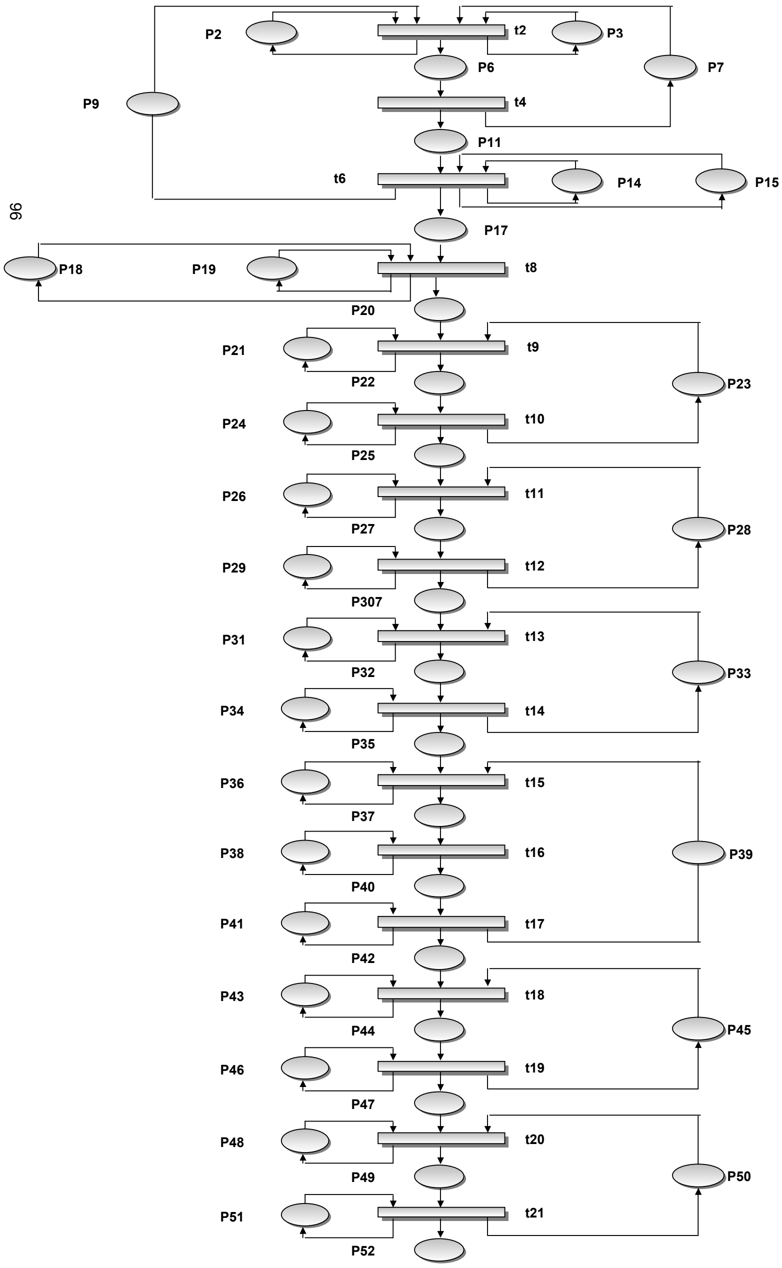


Figure 18 : Petri Net for Sub-case Study 2 of Case Study 3

Table 13 : Place Definitions for Sub-case Study 2 of Case Study 3

P2	Crane is ready to carry a block
P3	Block is ready to be carried
P6	Block is ready to be cut
P7	Block Cutter is ready to be loaded / cut
P9	Worker is ready
P11	Plate is ready to be carried
P14	Palette is ready to be loaded
P15	Unloader is ready to carry a plate
P17	Palette is ready to be carried
P18	Worker is ready
P19	Forklift is ready to carry the palette
P20	Plate is ready to be loaded on the polishing line by loader
P21	Loader is to carry a plate
P22	Plate is ready to cut
P23	Worker is ready
P24	Head cutter is ready
P25	Plate is ready to be calibrated
P26	Calibrator is ready
P27	Plate is ready to pre-polishing
P28	Worker is ready
P29	Pre-polishing machine is ready
P30	Plate is ready to polishing
P31	Polishing machine is ready
P32	Plate is ready to sizing
P33	Worker is ready
P34	Sizing machine is ready
P35	Plate is ready to side calibrating & bevel cutting
P36	Side calibrating & bevel cutting machine is ready
P37	Plate is ready to rotating
P38	Rotating unit is ready
P39	Worker is ready
P40	Plate is ready to side calibrating & bevel cutting
P41	Side calibrating & bevel cutting machine is ready
P42	Plate is ready to drying and cleaning
P43	Drying and cleaning unit is ready
P44	Plate is ready to quality control
P45	Worker is ready
P46	Quality control unit is ready
P47	Plate is ready to packaging
P48	Palette is ready to be loaded
P49	Palette is ready to be carried to the stock side
P50	Worker is ready
P51	Crane is ready to carry a palette
P52	Palette is ready to be dispatched

Table 14 : Transition Definitions for Sub-case Study 2 of Case Study 3

t2	Carrying a block by the crane from the stock side to Block Cutter
t4	Cutting the block into plates with the dimensions of 60 x Length Free x 2 cm
t6	Loading the cut plates on to a palette by unloader
t8	Carrying the palette by a forklift from unloader to the polishing line
t9	Loading the plates (60 x LF x 2 cm) by the loader from palette to the polishing line
t10	Cutting the heads of the plates (60 x LF x 2 cm) by the head cutter
t11	Calibrating the plates (60 x LF x 2 cm) by the calibrator
t12	Pre-polishing the plates (60 x LF x 2 cm) by the pre-polishing machine
t13	Polishing the plates (60 x LF x 2 cm) by the polishing machine
t14	Sizing the plates (60 x LF x 2 cm) by the by the sizing machine
t15	Side calibrating and bevel cutting of the first opposite sides of the plates (60 x 30 x 2 cm) by the side calibrating and bevel cutting machine
t16	Rotating the plates (60 x 30 x 2 cm) by the rotating unit
t17	Side calibrating and bevel cutting of the second opposite sides of the plates (60 x 30 x 1 cm) by the side calibrating and bevel cutting machine
t18	Drying and cleaning the plates (60 x 30 x 2 cm) by the drying and cleaning unit
t19	Quality control of the plates (60 x 30 x 2 cm)
t20	Packaging the plates (60 x 30 x 2 cm)
t21	Carrying the palette with the plates (60 x 30 x 2 cm) by the crane
t9’	Loading the plates (60 x LF x 2 cm) by the loader from palette to the polishing line
t10’	Cutting the heads of the plates (60 x LF x 2 cm) by the head cutter
t11’	Calibrating the plates (60 x LF x 2 cm) by the calibrator
t12’	Pre-polishing the plates (60 x LF x 2 cm) by the pre-polishing machine
t13’	Polishing the plates (60 x LF x 2 cm) by the polishing machine
t14’	Sizing the plates (60 x LF x 2 cm) by the by the sizing machine
t15’	Side calibrating and bevel cutting of the first opposite sides of the plates (60 x 30 x 2 cm) by the side calibrating and bevel cutting machine
t16’	Rotating the plates (60 x 30 x 2 cm) by the rotating unit
t17’	Side calibrating and bevel cutting of the second opposite sides of the plates (60 x 30 x 2 cm) by the side calibrating and bevel cutting machine
t18’	Drying and cleaning the plates (60 x 30 x 2 cm) by the drying and cleaning unit
t19’	Quality control of the plates (60 x 30 x 2 cm)
t20’	Packaging the plates (60 x 30 x 2 cm)
t21’	Carrying the palette with the plates (60 x 30 x 2 cm) by the crane

4.4.4.2 Reachability Graph

While depicting the Reachability Graphs of sub-case studies, there is only one apiece way to follow for each case. For the first sub-case study, minimum total time is of is found as 728t. And minimum total time of the second sub-case is computed as 556t. The reachability graphs and the time calculations can be seen in Appendix C.

4.4.4.3 Time Representation

In Case Study 3, both products are produced by only one block cutter. Hence, when a cycle is completed, 10 m^2 products will be produced. However, in the previous Case Studies, totally 20 m^2 production was considered. So, for making better comparison, the cycle should be completed twice.

In Case Study3, like Case Study 2, both products are evaluated separately. Accordingly, following two tables shows the time durations for each sub-case study.

Table 15 : Time Durations for Production of Product 1 by one block cutter

t1	2t	t14	40t
t3	104t	t15	36t
t5	40t	t16	36t
t7	t	t17	36t
t9	40t	t18	20t
t10	40t	t19	20t
t11	50t	t20	10t
t12	50t	t21	5t
t13	50t		

Table 16 : Time Durations for Production of Product 2 by one block cutter

t2	2t	t14’	25t
t4	80t	t15’	24t
t6	20t	t16’	24t
t8	t	t17’	24t
t9’	20t	t18’	20t
t10’	27t	t19’	20t
t11’	50t	t20’	10t
t12’	50t	t21’	5t
t13’	50t		

4.4.4.4 Results of Case Study 3

In the last case study, the same products are processed with Case Study 2. However, in this case, only one block cutter is used.

After drawing Petri nets, reachability graphs are constructed. In this case study, each reachability graph has only one branch for each sub-case study.

Finally, minimum total production times are calculated as 728t for producing only the first product, and 556t for only producing the second product.

4.5 Discussions

Different techniques can be applied to examine the production systems. However, Petri Net is a new one and it had not been applied to the marble industry up to now. This study aimed to investigate the applicability of Petri net to marble sector.

To this goal, three different case studies are applied. First of all, Petri nets of each case study are drawn. Then, marking matrices are calculated. Next, in the light of some assumption, reachability graphs of each case and sub-case studies are constructed.

Applying all the three case studies the minimum production times and the path (i.e. the sequence of the processes with their start and finish times) have been found. These time values are determined for different production opportunities such as yield of two different products at simultaneously or production of the same type of product with two cutting line system etc. Actually the results of these case studies should be evaluated according to the order requirements and dead lines for the delivery times. However, these results show that the production system applied in the plant is convenient for the flexible manufacturing. Since the preparation time (the time required to prepare the system to the production of another type of product) is negligible in relative to the production times. In this regard, it is beneficial to examine the reachability graph of two-product system for the optimization of production schedule.

First, the time durations of producing the related product of each machine is determined for 10 m^2 products. Two different times namely, the total production time; and the remaining time, are used for the time representation of the Petri net. Remaining time means that the time in which the following process can occur at the same time with its preceding processes. These time calculations can be seen on the related Reachability Graphs.

At the end of each alternative, minimum completion time is obtained. The minimum total production time for Case Study 1 is 580 minute. On the other hand, total production time is 582 minute for the first product in Case Study 2, and 454 minute for the second product of Case Study 2. Finally, in Case Study 3, total production times are 728 minute and 556 minute for each product respectively.

This study examines the applicability of PN to mining and benefits and difficulties of implementations, especially in marble production sector. The results of the study shows that Petri Nets can be easily and successfully applied to the marble sector. And this implementation is very useful to see the deadlocks, time loses, process sequence etc. Some difficulties arisen in this study such as drawing reachability graph etc. can also be met in the implementation of PN to the other industries and these difficulties do not affect the reliability and success of studies. Petri net provides to get optimal design alternatives and operational policy.

These case studies and the thesis study as a whole shows that application of Petri net to the mining industry brings the optimization, simulation and modeling of the production systems. These three factors are specially important for the mining sector as most of the mining companies, especially Turkish Governmental mines, work with low efficiency and high work power and time losses. Moreover, having great variability in the raw materials in any stages of mining, Petri nets can be used successfully not only taking the time as variant but also other characteristics.

Beside these, the study reveals that the previous studies on Petri net are limited with the theoretical works and not supported with the industrial applications. In this regard, this study is useful for the further applications as a reference.

However, some difficulties are faced with during application of Petri net technique to the selected marble processing plant. There aren’t any software programs drawing Petri net and reachability graphs and making time analysis of the existing production system. Therefore, especially large and complex system’s graphical mathematical analysis become more difficult or even impossible to provide it will be time consuming. Because of this difficulty, the method can only be examined for 10 m^2 products. But, if the desired production quantity is more than10 m^2 product, an advanced computer program is needed.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

Turkey has one of the biggest marble reserves in the world. These resources include high quality marbles. If these reserves are evaluated correctly, it will bring a great contribution to Turkish economy. To provide these new techniques should be applied in order to analyze and improve the production stages. Petri Nets are one of the most popular formal models of concurrent systems, used by both theoreticians and practitioners. They are very important and useful tools for modeling and analysis of the systems. However, they have not been used in the area of manufacturing yet in Turkey. As described in the preceding chapter this study examines the applicability of PN to mining and benefits and difficulties of implementations. The results of the study shows that Petri Nets can be easily and successfully applied to the marble sector. And this implementation is very useful to see the deadlocks, time loses, process sequence etc.

In this study, three different case studies are applied to check, whether, Petri net can or cannot be used on marble sector.

In the first case study, two different products with 30x30x1 cm and 60x60x2 cm dimensions are processed by using two identical block cutters.

In sub-case study 1 of Case Study 2, two identical block cutters process only 30x30x1 cm product. Besides, in sub-case study 2 of Case Study 2, the same block cutters, process only 60x60x2 cm product.

In the last case study, like Case Study 2, products are examined separately in the sub-case studies. However, different from the previous case studies, only one block cutter is used to cut the blocks into plates, in the last case study.

After drawing the Petri nets of all of the three case studies, marking matrices are computed. Then, the reachability graphs are depicted. When examining the first case study, 70 different alternative paths, which can be followed, are obtained. But, in Case Study 2, the numbers of different alternative paths are decrease to 25 for each sub-case study. On the other hand, if one block cutter is used instead of two, only one way to be followed is get for each sub-case studies.

When applying the time analysis for each case study, different total production times are obtained. In the first case study, minimum total production time is found as 580t. But, in the second case study, minimum total production time is 582t for the first product and 454t for the second one. On the other hand, if only one block cutter is used for the same products' processing, these times are computed as 728t and 556t, respectively.

As mentioned above, for the first case study, minimum total production time is found as 580t, by using Petri Net technique, for 70 different alternative paths. As a conclusion, Petri Net provides the mining engineer the path that gives the optimum total production time.

However, in the real applications, the engineer selects the sequence, randomly. The optimum sequence may or may not be selected by chance. But, when the system is large and complex, then this chance is very low, so that a method, like Petri Net, will be needed for selecting the optimum total production time.

The products processed are exactly the same in the second and the third studies. The only the difference between them is the number of block cutters used. In the former one, minimum total production time is 582t for the first product and 454t for the second product for 25 alternative paths by using two block cutters. However, in the later one, the times are computed as 728t and 556t, respectively and for each product, one path is obtained using one block cutter. This means that, if one block cutter is used instead of two, the total minimum time increases by 146t for the first product, and by 102t for the second product. Therefore, the number of block cutters that will be used can be determined depending on the quantity and the urgency of the order.

Some difficulties arisen in this study such as drawing reachability graph etc. can also be met in the implementation of PN to the other industries and these difficulties do not affect the reliability and success of studies.

Briefly, Petri net implementation is very useful to see the deadlocks, the losses, process sequence, available and non-available resources, etc. Optimum design alternatives, operational policy, and minimum total production time can be determined by using Petri net, easily. Petri net provides optimization, simulation, and modeling of the system.

The next study on this subject can be recommended as the examination of the production in quarries or the improvement of this study to the production of more than one block in a cutting line.

In this study, only one iteration can be applied for each case and sub-case studies because it is applied manually and computations will be more complex when the iterations are continued. For further iterations software programs are required. Although some software programs have been designed, they can only be used to draw Petri nets manually and they can make simulations for token flows. However, none of these programs can draw Petri nets automatically, can draw reachability graphs or make time analysis etc. either. For making all of these computations and drawings, advanced software programs should be developed.

REFERENCES

Bilgin, M., Çakır, E., 1998, Mermer Araştırması, **İstanbul Ticaret Odası Yayınları**, İstanbul.

Desel, J., Javier, E., 1995, Free Choice Petri Nets, **Cambridge University Press**, Australia.

Dilworth, J., B., 1993, Production and Operations Management : Manufacturing and Services, **McGrow-Hill, Inc.**, Fifth Edition, New York.

Kantor, P.B., Nordlie, R., 1999, Models of the Behavior of People Searching the Internet: a Petri Net Approach, **Rutgers University**, New Brunswick.

Keseroğlu, N.A., 2000, Modelling and Performance Analysis of an Electronic Assembly System Using Petri Nets, **Dokuz Eylül University**, İzmir, Master of Science Thesis.

Kimbler, D.L., **Clemson University**, IE340 Lecture Notes, 1997. (http://taylor.ces.clemson.edu/ie340)

Mahmutoğulları, R., 1992, Development of an Interactive Too for Stochastic Petri Net Simulation, **Boğaziçi University**, İstanbul, Master of Science Thesis.

Moghaddam, A., S., 2000, Network Simulator Design with Extended Object Model and Generalized Stochastic Petri Net, **University of Missouri**, Columbia, Doctor of Philosophy.

Murata, T.,1989, *Petri nets : Properties, Analysis and Application*s, **Proceedings of the IEEE**, Vol. 77, No 4, April 1989, pp : 541-580.

Pittsburgh University, Coarse 365 Lecture Notes, (http://www.cs.pitt.edu.tr/~ chang/365/petri)

Rembold, U., Nnaji, B.O. and Storr, A., (1996), Computer Integrated Manufacturing and Engineering, **Addison Wesley Publishing Company**.

Stevenson, W., J., 1996, Production / Operations Management, **Irwin**, Fifth Edition, Chicago.

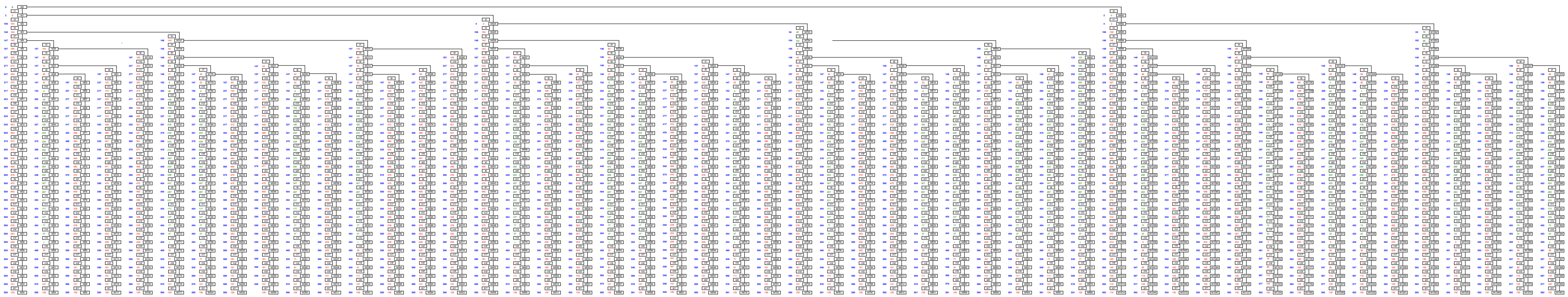
Tempelmeier, H., Kuhn, H., 1993, Flexible Manufacturing Systems, **John Wiley & Sons, Inc.**

Terry, D., 1996, A Petri Net Based On-line Scheduling System for a General Manufacturing Job Shop, **Rensselaer Polytechnic Institute Troy**, New York, Doctor of Philosophy.

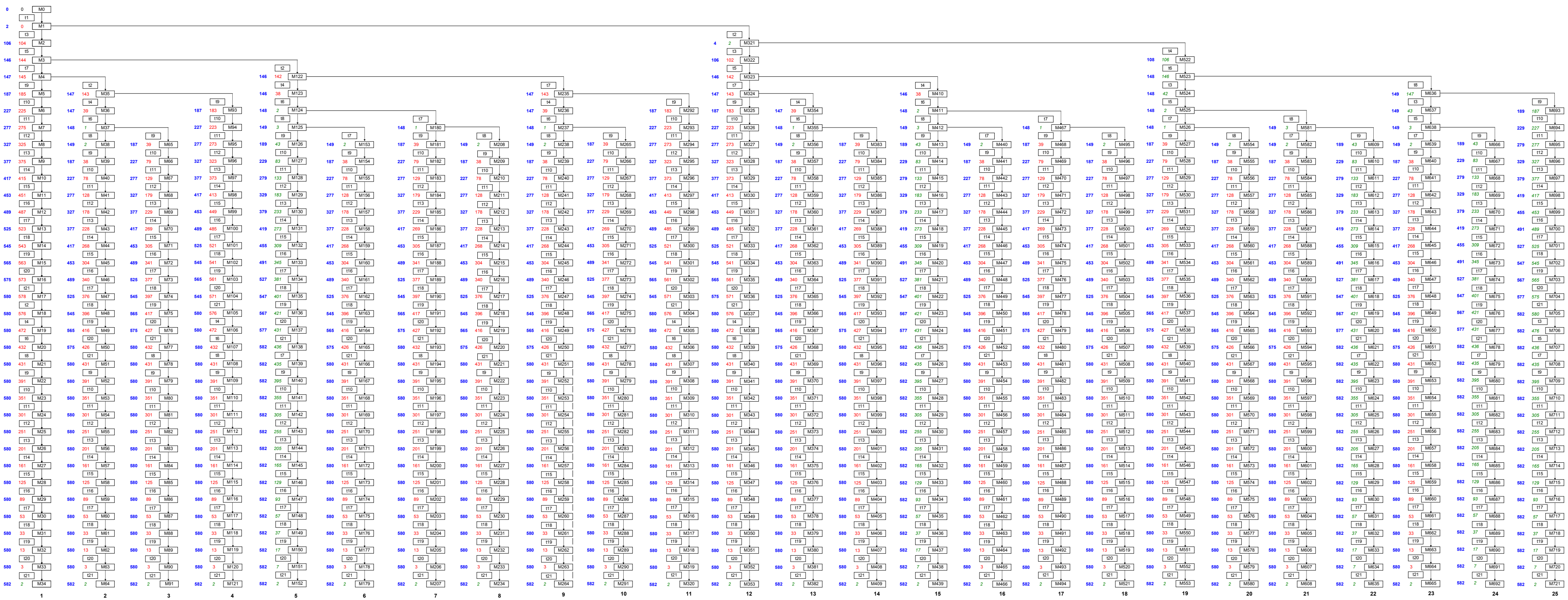
Tunçel, G., 1999, Modeling, Behavioral Analysis and Performance Evaluation of an Automotive Assembly Plant by Using Petri Net, **Dokuz Eylül Universty**, İzmir, Master of Science Thesis.

Zhou, M.C. and DiCesare, F., 1993, Petri Net Synthesis for Discrete Event Control of Manufacturing Systems, **Kluwer Academic Publishers**, Boston.

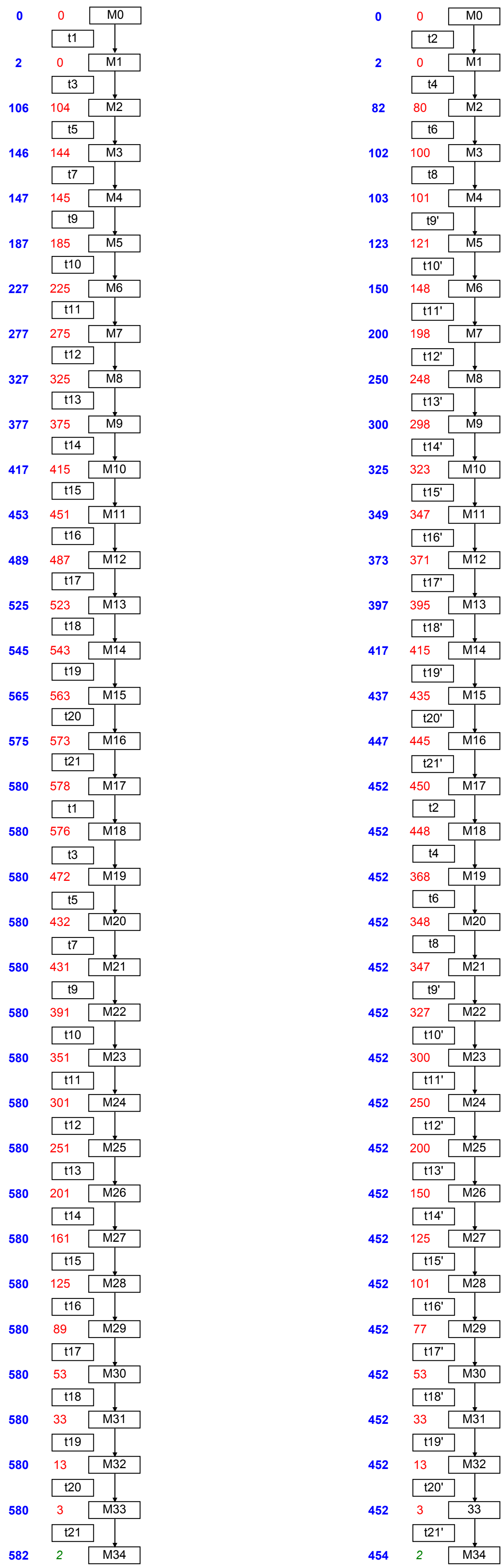
Zhou, M.C. and Robbi, A.D., 1994, “Application of Petri net methodology to manufacturing systems”, Computer Control of Flexible Manufacturing Systems : Research and Development (Edited by : Joshi, S.B. and Smith, J.S.), **Chapman & Hall**, Hong Hong.



APPENDIX A : REACHABILITY GRAPH OF CASE STUDY 1



APPENDIX B : REACHABILITY GRAPH OF CASE STUDY 2



APPENDIX C : REACHABILITY GRAPH OF CASE STUDY 3