

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**DEMİRYOLU ANKLAŞMAN SİSTEMLERİNİN PETRİ AĞLARI İLE
TASARIMI VE GERÇEKLENMESİ**

DOKTORA TEZİ

Cem BAŞKOCAGİL

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

NİSAN 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**DEMİRYOLU ANKLAŞMAN SİSTEMLERİNİN PETRİ AĞLARI İLE
TASARIMI VE GERÇEKLENMESİ**

DOKTORA TEZİ

**Cem BAŞKOCAGİL
(504042102)**

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Salman KURTULAN

NİSAN 2014

İTÜ, Fen Bilimleri Enstitüsü'nün **504042102** numaralı Doktora Öğrencisi **Cem BAŞKOCAGİL**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**DEMİRYOLU ANKLAŞMAN SİSTEMLERİNİN PETRİ AĞLARI İLE TASARIMI VE GERÇEKLENMESİ**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Salman KURTULAN**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Leyla GÖREN SÜMER**
İstanbul Teknik Üniversitesi

Doç. Dr. Mürvet KIRCI
İstanbul Teknik Üniversitesi

Prof. Dr. Galip CANSEVER
Yıldız Teknik Üniversitesi

Dr. İbrahim TOLGA HASDEMİR
Siemens San. ve Tic. A.Ş.

Teslim Tarihi : **27 Ocak 2014**
Savunma Tarihi : **10 Nisan 2014**

Babamın anısına,

ÖNSÖZ

Çalışmalarımın tüm aşamalarında tecrübelerinden faydalandığım, birlikte çalışmaktan onur duyduğum, benden yardımlarını esirgemeyen değerli hocam ve aynı zamanda danışmanım Prof. Dr. Salman KURTULAN'a, tez çalışmamın olgunlaşmasındaki katkılarından dolayı değerli hocalarım Prof. Dr. Leyla GÖREN SÜMER'e ve Doç. Dr. Mürvet KIRCI'ya sonsuz teşekkürü bir borç bilirim.

Ayrıca halen çalışmakta olduğum TÜBİTAK-BİLGEM'in üst yönetimine tez çalışmamı yürütmemde sağladığı imkânlar için teşekkürlerimi sunarım.

Son olarak tez çalışmam boyunca gösterdiği anlayış ve verdiği manevi destek için eşime ve bu günlere gelmemde büyük pay sahibi olan annem ve babama sonsuz teşekkürlerimi sunarım.

Nisan 2014

Cem BAŞKOCAGİL
(Kontrol ve Otomasyon Yüksek Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR.....	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY	xix
1 GİRİŞ	1
2 PETRİ AĞLARININ GENEL TANIMI.....	5
2.1 Temel Bileşenler	5
2.2 Durum Geçiş Mekanizması.....	7
2.3 Durum Denklemleri	8
2.4 Sınırlılık Kavramı.....	9
3 PETRİ AĞLARI İLE TASARIM YÖNTEMLERİ.....	11
3.1 Primitif Yapılar	11
3.1.1 Paralellik yapısı.....	11
3.1.2 Senkronizasyon yapısı.....	12
3.1.3 Çakışma yapısı	12
3.1.4 Sıralama yapısı	13
3.1.5 Kaynak sınırlandırma yapısı	13
3.2 Petri Ağları'nın Türleri	14
3.2.1 Modüler Petri Ağları	15
3.2.2 Renkli Petri Ağları	16
3.2.3 Zaman Etiketli Petri Ağları	18
4 GENİŞLETİLMİŞ PETRİ AĞLARI	19
4.1 Klasik Petri Ağlarına Eklentiler	19
4.1.1 Yasaklayıcı yay	19
4.1.2 Yetkilendirici yay	20
4.1.3 Yutucu yay	22
4.2 Yaylar için Durum Denklemleri.....	23
4.2.1 Yasaklayıcı yay	23
4.2.2 Yetkilendirici yay	26
4.2.3 Yutucu yay	28
4.3 Genelleştirilmiş Durum Denklemi	30
4.3.1 Örnek bir uygulama.....	32
4.4 Otomasyon Petri Ağları.....	39
5 PETRİ AĞLARINDAN PLC KODUNA DÖNÜŞÜM	43
5.1 PLC Koduna Dönüşüm için Mevcut Yöntemler.....	43
5.1.1 Kurma ve silme komutları ile kodlama yöntemi.....	43
5.1.1.1 Çığ etkisinin giderilmesi	45
5.1.2 Mantıksal ifadeler ile kodlama yöntemi.....	47

5.2	Mantıksal İfadeler Yönteminin Petri Ağları için Uyarlanması	51
5.3	Petri Ağlarının Sistem Matrisinden PLC Koduna Geçiş.....	56
5.4	Petri Ağları Yapılarının PLC Koduna Dönüşümü	57
5.4.1	Senkronizasyon yapısının PLC kodu karşılığı	57
5.4.2	Çakışma yapısının PLC kodu karşılığı	59
5.4.3	Yasaklayıcı yay içeren Petri Ağları'nın PLC kodu karşılığı.....	62
5.4.4	Yetkilendirici yay içeren Petri Ağları'nın PLC kodu karşılığı	64
5.4.5	Zaman Etiketli Petri Ağları'nın PLC kodu karşılığı	66
5.4.5.1	Zamanlama öncelikli Petri Ağları	67
5.4.5.2	Olay öncelikli Petri Ağları	69
5.5	Petri Ağlarından PLC Koduna Otomatik Dönüşüm Yapan Yazılım	71
5.5.1	Yazılıma genel bakış	71
5.5.2	Örnek sistemin tanımı	73
5.5.3	Sistemin Petri Ağları modelinin oluşturulması	75
5.5.4	Petri Ağlarının PLC koduna otomatik dönüşümü	76
6	DEMİRYOLU SİNYALİZASYON SİSTEMİ.....	81
6.1	Trafik Kumanda Merkezi	82
6.2	Saha Elemanları.....	83
6.2.1	Makaslar	83
6.2.2	Sinyaller	84
6.2.2.1	Dörtlü yüksek sinyaller	85
6.2.2.2	Üçlü yüksek sinyaller	86
6.2.2.3	Üçlü cüce sinyaller	86
6.2.3	Ray devreleri	87
6.3	Anklaşman Sistemi.....	88
6.3.1	Örnek anklaşman tablosu	93
7	UYGULAMA.....	95
7.1	Mithatpaşa Tren İstasyonu	95
7.2	Anklaşman Sisteminin Petri Ağları ile Modellenmesi	97
7.2.1	Tanzim bloğunun Petri Ağları modeli.....	101
7.2.2	Makas bloğunun Petri Ağları modeli	102
7.2.3	Tren takip bloğunun Petri Ağları modeli	106
7.2.4	Cüce sinyal bloğunun Petri Ağları modeli	109
7.2.5	Yüksek sinyal bloğunun Petri Ağları modeli	111
7.3	SCADA Sistemi Tasarımı	114
8	SONUÇLAR VE ÖNERİLER.....	131
	KAYNAKLAR.....	133
	ÖZGEÇMİŞ.....	137

KISALTMALAR

CENELEC	: Comité Européen de Normalisation Électrotechnique
IEC	: International Electrotechnical Commission
OPA	: Otomasyon Petri Ağları
PA	: Petri Ağları
PLC	: Programmable Logic Controller
SCADA	: Supervisory Control And Data Acquisition
SCL	: Structured Control Language
STL	: Statement List
TKM	: Trafik Kumanda Merkezi

ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1 : Sistemin PA modelindeki yerlerin tanımı.	33
Çizelge 5.1 : Örnek sisteme ait PA'daki yerler ve geçişler.	75
Çizelge 6.1 : Makasların konum bilgisini belirleyen mantıksal çıkışlar.	83
Çizelge 6.2 : Dörtlü yüksek sinyal renk bildirim açıklamaları.	85
Çizelge 6.3 : Üçlü yüksek sinyal renk bildirim açıklamaları.	86
Çizelge 6.4 : Üçlü cüce sinyal renk bildirim açıklamaları.	87
Çizelge 6.5 : Örnek anlaşıman tablosu.	93
Çizelge 7.1 : Mithatpaşa tren istasyonuna ait anlaşıman tablosu (1 – 14 no'lu rotalar).	99
Çizelge 7.2 : Mithatpaşa tren istasyonuna ait anlaşıman tablosu (15 – 20 no'lu rotalar).	100
Çizelge 7.3 : Tanzim bloğuna ait PA'daki yerlerin tanımı.	101
Çizelge 7.4 : Makas bloğuna ait PA'daki yerlerin tanımı.	102
Çizelge 7.5 : Tren takip bloğuna ait PA'daki yerlerin tanımı.	106
Çizelge 7.6 : Cüce sinyal bloğuna ait PA'daki yerlerin tanımı.	109
Çizelge 7.7 : Yüksek sinyal bloğuna ait PA'daki yerlerin tanımı.	111

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : PA'nın temel bileşenleri.	5
Şekil 2.2 : 1-sınırlı PA'ya örnek.	9
Şekil 3.1 : Paralellik yapısına örnek.	11
Şekil 3.2 : Senkronizasyon yapısına örnek.	12
Şekil 3.3 : Çakışma yapısına örnek.	12
Şekil 3.4 : Sıralama yapısına örnek.	13
Şekil 3.5 : Kaynak sınırlandırma yapısına örnek.	14
Şekil 3.6 : (a) Klasik PA'ya örnek. (b) Modüler PA karşılığı.	15
Şekil 3.7 : Renkli PA'ya örnek.	17
Şekil 3.8 : Zaman Etiketli PA'ya örnek.	18
Şekil 4.1 : (a) t_1 geçişi aktif değil. (b) t_1 geçişi aktif. (c) t_1 geçişi tetiklendikten sonra.	20
Şekil 4.2 : (a) t_1 geçişi aktif. (b) t_1 geçişi aktif değil.	20
Şekil 4.3 : (a) Yetkilendirici yay. (b) Yetkilendirici yayın karşılığı.	21
Şekil 4.4 : Yasaklayıcı yay içeren bir PA.	23
Şekil 4.5 : Yetkilendirici yay içeren bir PA.	27
Şekil 4.6 : (a) Yutucu yay. (b) Yutucu yayın karşılığı.	29
Şekil 4.7 : Sistemin giriş ve çıkışları.	32
Şekil 4.8 : Sistemin PA modeli.	33
Şekil 4.9 : Sisteme ait PA'nın yazılım tarafından hesaplanmış son durum bilgisi (1).	36
Şekil 4.10 : Sisteme ait PA'nın yazılım tarafından hesaplanmış son durum bilgisi (2).	38
Şekil 4.11 : Örnek bir OPA.	41
Şekil 5.1 : (a) Örnek bir PA. (b) PA'nın merdiven diyagramı karşılığı.	44
Şekil 5.2 : (a) Örnek bir PA. (b) PA'nın PLC kodu karşılığı (Çığ etkisi var).	45
Şekil 5.3 : (a) Örnek bir PA. (b) PA'nın PLC kodu karşılığı (Çığ etkisi giderilmiş).	47
Şekil 5.4 : Örnek bir otomat diyagramı.	49
Şekil 5.5 : Geçişlere ilişkin mantıksal ifadelerin elde edilmesi.	51
Şekil 5.6 : Yerlere ilişkin mantıksal ifadelerin elde edilmesi.	52
Şekil 5.7 : PA'nın PLC koduna dönüştürülmesi.	54
Şekil 5.8 : (a) Senkronizasyon yapısı. (b) Yapının PLC kodu karşılığı.	58
Şekil 5.9 : (a) Çakışma yapısı. (b) Yapının PLC kodu karşılığı.	60
Şekil 5.10 : (a) Yasaklayıcı yay içeren PA. (b) PA'nın PLC kodu karşılığı.	62
Şekil 5.11 : (a) Yetkilendirici yay içeren PA. (b) PA'nın PLC kodu karşılığı.	64
Şekil 5.12 : (a) Zamanlama öncelikli PA yapısı. (b) PA'nın PLC kodu karşılığı.	67
Şekil 5.13 : (a) Olay öncelikli PA yapısı. (b) PA'nın PLC kodu karşılığı.	69
Şekil 5.14 : Yazılımın ana penceresi.	71
Şekil 5.15 : Örnek sistemin otomat diyagramı.	73
Şekil 5.16 : Örnek sistemin PA modeli.	76
Şekil 5.17 : PLC kodunun üretilmesi için kullanılacak dosya.	77

Şekil 5.18 : Örnek sisteme ait PA'nın PLC koduna dönüştürülmesi.	78
Şekil 5.19 : Örnek sistemin yazılım tarafından üretilen PLC kodu.	79
Şekil 6.1 : Sinyalizasyon sistemi blok diyagramı.	82
Şekil 6.2 : DC ray devresi.	88
Şekil 6.3 : Tanzim isteğinin değerlendirilmesine ilişkin akış diyagramı.	90
Şekil 6.4 : Tanzim iptal isteğinin değerlendirilmesine ilişkin akış diyagramı.	92
Şekil 6.5 : Örnek demiryolu bölgesi.	93
Şekil 7.1 : Mithatpaşa tren istasyonu.	98
Şekil 7.2 : Tanzim bloğunun PA modeli.	104
Şekil 7.3 : Makas bloğunun PA modeli.	105
Şekil 7.4 : Tren takip bloğunun PA modeli.	108
Şekil 7.5 : Cüce sinyal bloğunun PA modeli.	110
Şekil 7.6 : Yüksek sinyal bloğunun PA modeli.	112
Şekil 7.7 : Fonksiyon bloklarının PA modelleri arasındaki bağlantılar.	113
Şekil 7.8 : Tren istasyonunun ilk durumu.	115
Şekil 7.9 : “001BT-3ST” güzergâhı için tanzim isteğinin reddedilmesi (ray devresi meşgul).	116
Şekil 7.10 : “001BT-2ST” güzergâhı için tanzim isteğinin reddedilmesi (çakışan güzergâh).	117
Şekil 7.11 : “(002B)-1ST” güzergâhı için tanzim isteğinin reddedilmesi (makas hatası).	118
Şekil 7.12 : “001BT-1ST” güzergâhı için tanzim isteğinin kabul edilmesi.	120
Şekil 7.13 : “001BT-1ST” güzergâhında trenin ilerlemesi.	121
Şekil 7.14 : “001BT-1ST” güzergâhı için yapılan tanzimin sonlanması.	122
Şekil 7.15 : “001BT-1ST” ve “1ST-001DT” ardışık güzergâhları için yapılan tanzim isteği.	123
Şekil 7.16 : “001BT-1ST” güzergâhı için yapılmış tanzimin iptal edilmesi (1).	125
Şekil 7.17 : “001BT-1ST” güzergâhı için yapılmış tanzimin iptal edilmesi (2).	126
Şekil 7.18 : “001BT-1ST” güzergâhı için zorunlu güzergâh iptalinin yapılması (1). ...	127
Şekil 7.19 : “001BT-1ST” güzergâhı için zorunlu güzergâh iptalinin yapılması (2). ...	128
Şekil 7.20 : “001BT-1ST” güzergâhı için “ray devrelerinde beklenmedik meşguliyet” hatasının giderilmesi.	129

DEMİRYOLU ANKLAŞMAN SİSTEMLERİNİN PETRİ AĞLARI İLE TASARIMI VE GERÇEKLENMESİ

ÖZET

Raylı ulaşım sistemlerinde güvenliğin ve sürekliliğin sağlanması esastır. Bu anlamda tren güzergâhlarında meydana gelebilecek çakışmaların engellenmesi büyük önem taşımaktadır. Trenlerin demiryolu üzerinde güvenli bir şekilde hareket edebilmesi için anlaşıman sistemleri kullanılmaktadır. Çakışma probleminin giderilmesi için anlaşıman sistemlerinin etkin bir şekilde tasarlanmasına ihtiyaç duyulmaktadır. Aksi takdirde anlaşıman sistemlerinin tasarımında yapılacak en ufak hata bile raylı ulaşım sistemlerinin güvenliğini tehlikeye atacaktır. Anlaşıman sistemlerinin etkin bir şekilde tasarlanıp gerçekleştirilmesi bu tezin ana konusunu oluşturmaktadır.

Bu tez çalışmasında anlaşıman sistemlerinin etkin bir şekilde tasarlanması ve gerçekleştirilmesi için Petri Ağları'nın kullanımı üzerinde durulmaktadır. Bunun için öncelikle Petri Ağları'nın tasarımsal amaçlı kullanımı ile ilgili bilgi altyapısı oluşturulmaktadır. Ardından Petri Ağları'nın modelleme yeteneğini arttırmak için tanımlanmış olan yay türleri ele alınmakta ve bu yay türlerinin hepsini içeren Petri Ağları'nın matematiksel olarak ifade edilebilmesi için “genelleştirilmiş durum denklemi” adı altında yeni bir durum denklemi önerisi ortaya atılmaktadır.

Anlaşıman sisteminin gerçekleştirilmesi için PLC kullanılacaktır. Dolayısıyla anlaşıman sistemi için oluşturulacak Petri Ağları modelinin PLC koduna dönüştürülmesi gerekecektir. Yapılan literatür taraması sonucunda PLC koduna dönüşüm için Petri Ağları'nın sadece grafiksel özelliklerinin kullanıldığı görülmüştür. Programlamaya uygun bir altyapı sağlanabilmesi amacıyla PLC koduna dönüşüm işlemi için Petri Ağları'nın matematiksel özelliklerinin kullanılması gerekmektedir. Tezde bu amaca hizmet eden bir yöntem önerisinde bulunmaktadır. Bu yöntem ile Petri Ağları'nın sistem matrisinden PLC koduna doğrudan dönüşüm sağlanabilmektedir. Aynı zamanda önerilen yöntem ile sistemlerde sıklıkla karşılaşılan çığ etkisinin giderilmesi de sağlanmaktadır. Bununla birlikte yöntemin hızlı ve kolay şekilde uygulanabilmesi için tez kapsamında geliştirilen yazılım tanıtılmakta ve bu yazılımın kullanımı gösterilmektedir. Bu yazılım sayesinde Petri Ağları'nın PLC koduna dönüşümü otomatik olarak gerçekleştirilebilmektedir.

Bu tez çalışmasında önerilen yöntemin uygulanabilmesi için gerçek bir tren istasyonu ele alınmaktadır. Öncelikle bu tren istasyonuna ait anlaşıman sistemi Petri Ağları ile modüler bir yaklaşım kullanılarak modellenmektedir. Bu amaçla anlaşıman sistemi bir takım alt sistemlere bölünmekte ve bu alt sistemlerin her biri ayrı ayrı Petri Ağları ile modellenmektedir. Alt sistemlere ait her bir Petri Ağları modelinin PLC koduna dönüştürülmesi için önerilen yöntem kullanılmaktadır. Böylelikle anlaşıman sisteminin Petri Ağları kullanılarak tasarlanması ve gerçekleştirilmesi sağlanmaktadır. Son olarak gerçekleştirilen anlaşıman sisteminin test edilmesi amacıyla geliştirilen SCADA uygulaması üzerinde sistemin davranışı gözlemlenerek yapılan gerçekleştirme doğrulanmaktadır.

RAILWAY INTERLOCKING SYSTEM DESIGN AND IMPLEMENTATION USING PETRI NETS

SUMMARY

It is vital to provide safety and availability of the railway transportation systems. For serving such a purpose, route conflicts should be prevented in the first place. However, railway transportation systems are complex in nature and this makes it hard to accomplish such a task. That is where railway interlocking systems come into play. Interlocking systems are utilized so that trains can safely move on the railway. This can only be achieved by an efficiently designed interlocking system. Otherwise, any simple flaw in the design would lead to route conflicts, thus rendering the railway transportation system unsafe. Efficient design and implementation of railway interlocking systems constitutes the main subject of this study.

In this thesis, Petri Nets would be used for designing and implementation of interlocking systems. Petri Nets is a modeling tool that is able to represent systems both visually and mathematically. Visual aspects of Petri Nets are utilized to model the general structure of a system. Nevertheless, mathematical aspects of Petri Nets are used to realize the system dynamics. Petri Nets makes it possible to represent the system behavior mathematically via state equation. One can exploit mathematical features of Petri Nets to perform modeling, simulation, testing and verification of systems.

In order to model large-scale systems using Petri Nets, particular enhancements have become necessary to overcome arising shortfalls and to improve modeling capability. Modular Petri Nets is introduced to model complex systems in modular way. This is useful for simplifying the investigation of the system. Colored Petri Nets is proposed to make it easy to keep track of specific events taking place within the system. Timed Petri Nets is created to represent time-dependent events in the system. Together with these enhancements, Petri Nets has become more powerful modeling tool.

Specific arc types are defined to further enhance the modeling capability of Petri Nets. Inhibitor arcs are defined to enable a certain transition to take precedence over the other transitions. Using inhibitor arcs, it becomes possible to visually represent prioritization mechanism in Petri Nets. Enabling arcs are introduced to keep tokens in the input places while triggering the transition. An enabling arc is very similar to an ordinary arc. The only difference is the fact that when a transition is triggered, there is no token removal from the input place(s) connected to enabling arc(s). Nullifier arcs are proposed to remove all of the tokens from an input place at once when a transition is triggered. This functionality is used to implement the resetting mechanism in Petri Nets.

In the literature, the state equation for Petri Nets is used to model and analyze systems mathematically. However, there are not any formalized state equations defined in order to cover inhibitor, enabling and nullifier arcs. Therefore, we do not have the opportunity to represent the systems mathematically when these types of

arcs are involved. In this thesis, the state equation for Petri Nets is revisited and modified to obtain a novel state equation called “generalized state equation”. This state equation can be used to mathematically represent the behavior of inhibitor, enabling and nullifier arcs. Besides, software is developed and utilized to demonstrate the application of generalized state equation on an industrial automation system.

Railway signalization systems play a major role in providing safety and reliability for the railway transportation systems. The main purpose of railway signalization systems is to prevent train collisions from happening. A railway signalization system is composed of three main elements such as traffic control center, interlocking system and the field equipments.

Traffic control center is responsible for management and observation of the entire railway traffic. Route requests are created and train positions, switch positions, signal states are observed by traffic control center. There is a two-way communication between the traffic control center and the interlocking system. Route requests originated by the traffic control center is sent to the interlocking system and at the same time the status information coming from the field equipments is sent by the interlocking system to the traffic control center. As a result of this information exchange, the interlocking software processes the route requests and these requests are either accepted or rejected regarding the status of the field equipments. Interlocking software only accepts route requests that do not lead to train collisions.

Interlocking system both gathers information from and sends commands to the field equipments. The field equipments form the infrastructure of the railway signalization system. The field equipments are composed of switches, track circuits and signals. Switches are mechanical devices that enable trains to change direction. Generally, switches have two positions such as “normal” and “reverse”. When a switch is in “normal” position, train crossing over this switch does not change its direction, whereas in “reverse” position switch causes train crossing over it to change its direction. Track circuits inform the interlocking system about the existence of a train on the railway. Signals are used to inform trains about whether the railway is occupied or not. “Red” light means the railway is occupied, whereas “green” light means the railway is free.

The interlocking system is the decision maker in a railway signalization system. The main task of the interlocking system is to evaluate route requests coming from the traffic control center. During this evaluation, existing route requests and status of the field equipments are considered by the interlocking system. As a result, reasonable requests are accepted and the field equipments are locked accordingly to apply these requests. Unsuitable requests that cause collisions are rejected and are not applied. Interlocking system uses interlocking table to make these decisions. Interlocking table has entries for all of the possible routes and each of these entries identifies the conditions under which the corresponding route request should be accepted.

It is clear that the reliability of a railway signalization system is directly related to the interlocking system. Therefore, safety standards called EN50126, EN50128 and EN50129 are established by CENELEC to define general safety requirements for the railway interlocking systems. Both the hardware and the software components of an interlocking system must fulfill these standards. It is essential to use formal methods for the design of the software in order to meet these security standards.

In this thesis, the interlocking system would be implemented using PLC. In order to accomplish this task, at first, the interlocking system would be modeled using Petri Nets. Then, this Petri Nets model would have to be converted into PLC code. During literature research, it is observed that only visual aspects of Petri Nets are used for such a conversion process. And also, in the literature, there are not any available studies that exploit mathematical aspects of Petri Nets in order to realize interlocking systems using PLC code that is defined in IEC 61131-3 standards. In order to maintain an infrastructure that is favorable for programming, mathematical aspects of Petri Nets should be used for the conversion process. In this thesis, an efficient method is proposed to make the conversion from mathematical model of Petri Nets into PLC code that is defined in IEC 61131-3 standards. Then, this method is used to implement the interlocking system with PLC. In this method, conversion of Petri Nets into PLC code is directly accomplished by making use of the incidence matrix. At the same time, the proposed method also provides elimination of avalanche effect from which most of the systems suffer. In addition, software is developed to implement this methodology and usage of this software is shown on a case study. One can use this software to perform conversion of Petri Nets into PLC code automatically.

In this study, in order to apply aforementioned method, a real-life railway station is considered. At first, the interlocking system of this station is designed modularly using Petri Nets. Modularity is provided by means of partitioning the interlocking system into subsystems. Then each of these subsystems is modeled using Petri Nets, which is further converted into PLC code using proposed methodology. This concludes design and implementation of the interlocking system using Petri Nets. Finally, SCADA application is developed to test the interlocking system and on this application, the implementation of this interlocking system is verified.

1 GİRİŞ

Günümüzde raylı taşıtlar ulaşımın vazgeçilmez bir parçası haline gelmiştir. Dolayısıyla yoğun bir şekilde kullanılan modern raylı ulaşım sistemlerinde güvenliğin ve sürekliliğin sağlanması esastır. Ancak bu tür sistemler genellikle karmaşık yapıya sahip olduklarından söz konusu gereksinimlerin karşılanması güç olmaktadır. En çok göze çarpan zorluklardan biri de tren güzergâhlarında meydana gelebilecek çakışmaların engellenmesidir. Bu amaçla etkin bir sinyalizasyon altyapısı oluşturulmalıdır [1]. Bu sinyalizasyon sistemiyle çakışmalar önceden gerçek-zamanlı olarak tespit edilebilmelidir. Ancak raylı ulaşım sistemleri karmaşık ve büyük bir yapıya sahiptir. Tasarımda unutulacak en küçük ayrıntı ya da yapılacak en ufak hata sistemi işlevsiz kılabilir. Bu nedenle tasarım için çok etkin yöntemlerin kullanılması gerekmektedir. Bu tez çalışmasında demiryolu sinyalizasyon ve anlaşılan sistemlerinin etkin bir şekilde tasarlanıp gerçekleştirilmesi için Petri Ağları'ndan (PA) yararlanma konusu irdelenmiş, PA ile gerçekleştirme için bir yöntem önerilmiş ve uygulanmıştır.

PA kavramı ilk defa 1962 yılında Carl Adam Petri tarafından ortaya atılmıştır. Sistemlerin modellenmesini oldukça kolaylaştıran PA büyük ilgi görmeye başlamış ve sistemlerin analiz ve tasarımında giderek daha sık kullanılmaya başlamıştır. Bu anlamda endüstriyel otomasyon sistemlerinin modellenmesinde PA'nın kullanımı artmıştır.

PA hem grafiksel hem de matematiksel bir modelleme aracıdır. Grafiksel yönü ile PA bir sistemin genel yapısı ve işleyişi hakkında bilgi verir. PA ile sistem dinamikleri görsel olarak kolaylıkla ifade edilebilir. PA'nın matematiksel yönü kullanılarak da durum denklemleri ve sistem matrisi gibi araçlarla sistem davranışının ifade edilmesi sağlanır. Bu matematiksel ifadeler bilgisayar ortamında işlenerek sistemin davranışı için benzetim yapılabilir. Bu yönleriyle PA sistemlerin tasarlanmasını, incelenmesini, analiz edilmesini ve sistemdeki sorunların giderilmesini oldukça kolaylaştırmaktadır.

PA'nın geniş ölçekli sistemlerin modellenmesinde de etkin olarak kullanılabilmesi için PA'ya bir takım özellikler kazandırılmıştır. Modüler PA ile sistemlerin modüler olarak modellenebilmesi sağlanmış ve böylelikle elde edilen modellerin daha kolay incelenmesi sağlanmıştır. Renkli PA kavramı ortaya atılarak sistemdeki süreçlerin birbirinden ayırt edilebilmesi sağlanmıştır. PA'ya zaman etiketleri eklenerek zamana dayalı süreçlerin modellenebilmesi sağlanmıştır. Bu eklentilerle birlikte PA'nın modelleme için daha güçlü bir araç haline gelmesi sağlanmıştır.

Yapılan literatür taraması sonucunda raylı ulaşım sistemlerinde yaya/araç trafiğinin modellenmesi [2,3], trafik akışının analizinin yapılması [4], çakışmaların giderilmesi [5] ve trenlerin zaman çizelgelerinin optimizasyonu [6,7] gibi amaçlar için PA'nın kullanıldığı, sinyalizasyon sistemlerinin gerçekleştirilmesi ile ilgili çalışmalarda ise PA'nın sadece grafiksel özelliklerinden yararlanıldığı görülmüştür. PA'nın matematiksel modelinden yararlanılarak, doğrudan sistem matrisinden türetilen, IEC 61131-3 standartlarında tanımlı herhangi bir programlama dili ile gerçekleştirilen anlaşılan sisteme ilişkin bir çalışmaya rastlanmamıştır. Bu çalışmada raylı ulaşım sistemlerinde taşıtların düzenli hareket etmesini sağlayacak, çakışmaları engelleyecek bir sinyalizasyon sisteminin PA kullanılarak tasarlanması ve PA'nın matematiksel modelinden IEC 61131-3 standartlarında tanımlı PLC programlama diline (STL veya SCL) dönüşüm için yeni bir yöntem önerilecektir. Bu yöntem kullanılarak bir tren istasyonuna ait sinyalizasyon sisteminin tasarlanıp PLC ile gerçekleştirilmesi sağlanacaktır.

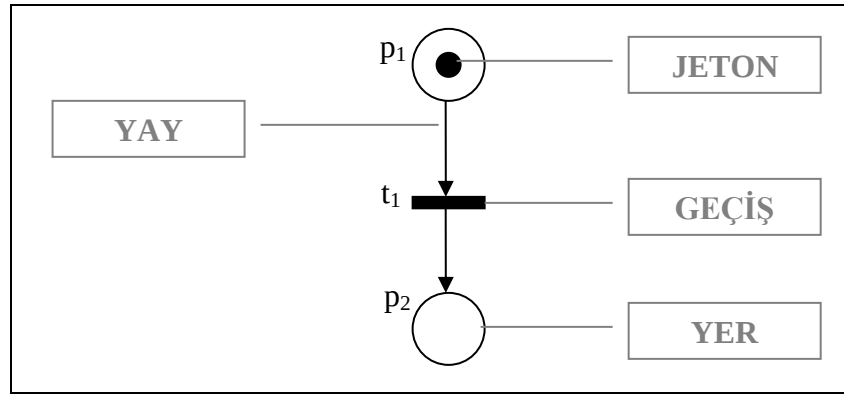
Bu tez çalışmasının ikinci bölümünde ön bilgi olarak PA ile ilgili temel kavramlar, PA'nın işlevselliği hakkında bilgi ve PA'nın durum denklemlerine ilişkin tanımlar verilecektir. Üçüncü bölümde literatürde var olan modelleme amaçlı olarak sıklıkla kullanılan temel PA yapıları hakkında bilgi verilecek ve PA'nın türleri ele alınacaktır. Dördüncü bölümde PA'nın modelleme yeteneğini arttırmak amacıyla tanımlanmış yay türlerinden bahsedilecek ve bu yay türlerinin her birinin matematiksel olarak ifade edilebilmesi için öneriler sunulacaktır. Ardından tüm bu yay türlerini içeren PA'nın matematiksel olarak ifade edilebilmesi için “genelleştirilmiş durum denklemi” adı altında genel bir yapı önerisinde bulunulacak ve bu durum denklemlerinin elde edilmesi için bu tez kapsamında geliştirilen bir yazılım tanıtılacaktır. Bu yazılımın kullanımı örnek bir uygulama üzerinde gösterilecektir. Beşinci bölümde PA'nın PLC koduna dönüştürülmesi için sistem

matrisinin kullanıldığı yeni bir yöntem önerilecek, bu dönüşümün otomatik olarak yapılması için geliştirilen bir yazılım tanıtılacak ve örnek bir uygulama üzerinde bu yazılımın kullanımı gösterilecektir. Altıncı bölümde demiryolu sinyalizasyon sisteminin elemanları olan trafik kumanda merkezi, saha elemanları, anlaşıman sistemi ele alınacak ve örnek bir tren istasyonu için anlaşıman tablosu oluşturulacaktır. Yedinci bölümde ise Adapazarı Mithatpaşa tren istasyonu ele alınacak ve bu tren istasyonuna ait anlaşıman sistemi PA ile modellenenecektir. Ardından bu PA modelinin Bölüm 5’te önerilen yöntem kullanılarak PLC koduna dönüşümü sağlanacaktır. Böylelikle Mithatpaşa tren istasyonuna ait anlaşıman sisteminin PA ile tasarlanması ve PLC üzerinde gereklenmesi sağlanacaktır. Son olarak gereklenen anlaşıman sisteminin test edilmesi amacıyla geliştirilen bir SCADA benzetim ve izleme uygulaması tanıtılacak ve bu uygulama üzerinde sistemin test süreçleri (geerlilik ve dođrulama) gerekleştirilip gözlemlenecektir.

2 PETRİ AĞLARININ GENEL TANIMI

2.1 Temel Bileşenler

Petri Ağları yerlerden ve geçişlerden oluşan iki bölgeli yönlü bir graf olarak tanımlanmaktadır. Yerler ve geçişler birbirine yönlü yaylar ile bağlanmakta ve sistem dinamiği jetonlar ile temsil edilmektedir. Bu kavramlar Şekil 2.1’de örnek bir PA üzerinde gösterilmektedir [8]:



Şekil 2.1 : PA’nın temel bileşenleri.

Jetonlar, PA’daki yerlerin içinde bulunur. Eğer bir yer jeton içeriyorsa “işaretli” olarak anılır ve bu yere ilişkin koşul sağlanmış olur. Aksi takdirde bir yerin jeton içermemesi söz konusu yere ilişkin koşulun henüz sağlanmamış olduğu anlamına gelir. Bütün sistemin durum bilgisi jetonların PA’daki yerlerde anlık dağılımıyla ifade edilir [8]. PA’daki bir yer herhangi bir zaman diliminde hiçbir jeton içermeyebilir ya da birden fazla jeton içerebilir. Bu tamamen modellenen sistemin davranışıyla ilgilidir.

Yerler sistemle ilgili koşulları, geçişler ise sistem içinde gerçekleşen olayları ifade eder. Yaylar ise yerler ile geçişleri birbirine bağlar. Yaylar ile iki geçiş ya da iki yer birbirine bağlanamaz. Sadece geçişleri yerlere veya yerleri geçişlere bağlamaya izin verilmiştir [9,10]. PA’nın temel bileşenleri için kullanılan notasyonlar bir sonraki sayfada verilmiştir.

Tanım 2.1: PA'da yerler kümesi $P = \{p_1, p_2, \dots, p_n\}$ ve geçişler kümesi $T = \{t_1, t_2, \dots, t_m\}$ ile ifade edilir. i . yerden j . geçişe çizilen bir yay (p_i, t_j) ve j . geçişten i . yere çizilen bir yay ise (t_j, p_i) olarak ifade edilir. Yaylar kümesi A ile gösterilir [9].

Tanım 2.2: PA'da aynı tipte olmayan herhangi iki düğümü birleştiren birden fazla yay bulunabilir. Bu yayların sayısını ağırlık fonksiyonu belirler. Yönlü yaylara ilişkin ağırlık değerleri w ile ifade edilir ve pozitif tamsayılarla gösterilir [9].

Tanım 2.3: PA tanımlanırken bir t_j geçişine ait olan giriş yerleri kümesi $I(t_j)$ ve benzer bir şekilde aynı t_j geçişine ait olan çıkış yerleri kümesi de $O(t_j)$ ile ifade edilir. Bu notasyonlar aşağıdaki gibi tanımlanır [9]:

$$I(t_j) = \{p_i \in P : (p_i, t_j) \in A\} \quad (2.1)$$

$$O(t_j) = \{p_i \in P : (t_j, p_i) \in A\} \quad (2.2)$$

Benzer şekilde bir p_i yerine ait olan giriş geçişleri kümesi $I(p_i)$ ve aynı p_i yerine ait olan çıkış geçişleri kümesi ise $O(p_i)$ ile ifade edilir.

Tanım 2.4: Yerleri ve geçişleri birbirine bağlayan yönlü yaylar, yaylar kümesini (A) oluşturur. Bir p_i yerinden t_j geçişine çizilmiş olan yay, $p_i \in I(t_j)$ olduğunu göstermektedir. Eğer $w(p_i, t_j) = k$ ise o zaman p_i yerinden t_j geçişine giden k adet yay var demektir. Başka bir deyişle p_i yerinden t_j yerine giden k ağırlıklı tek bir yay mevcuttur. Benzer bir şekilde eğer bir t_j geçişinden p_i yerine giden k adet yay varsa, $p_i \in O(t_j)$ demektir ve $w(t_j, p_i) = k$ değerini alır [9].

Tanım 2.5: Eğer bir yay üzerinde ağırlık değeri belirtiliyorsa söz konusu yer ve geçiş arasında o kadar yay var demektir. Eğer yay üzerinde ağırlık değeri belirtilmiyorsa bu ağırlık değeri 1 (bir) olarak kabul edilir. Ağırlık fonksiyonuna ilişkin aşağıdaki denklemler yazılabilir [9]:

$$p_i \notin I(t_j) \text{ ise } w(p_i, t_j) = 0 \quad (2.3)$$

$$p_i \notin O(t_j) \text{ ise } w(t_j, p_i) = 0 \quad (2.4)$$

Tanım 2.6: Koşulların sağlanma durumunun jetonlarla belirlendiği PA'ya, "İşaretli PA" denir. İşaretli PA (P, T, A, w, x) beşlisiyle tanımlanır. Burada (P, T, A, w) dörtlüsü PA'ya ilişkin grafi, x ise yerlerdeki jeton sayısını belirtir [9]:

$$x = [x(p_1), x(p_2), \dots, x(p_n)] \in N^n \quad (2.5)$$

Bu satır vektöründe n , PA'daki yerlerin sayısını belirtir. Bu vektörün i . elemanı, p_i yerindeki jetonların sayısına karşılık gelir. Özetle x vektörü PA'nın anlık durumunu ifade eder.

2.2 Durum Geçiş Mekanizması

PA'daki durum geçiş mekanizması jetonların bir yerden başka bir yere transfer edilmesiyle sağlanır. Böylelikle PA'nın durumu değiştirilmiş olur ve PA bulunduğu durumdan başka bir duruma geçer. Bu da ancak aktifleşmiş geçişlerin tetiklenmesiyle mümkün olabilir. PA'ya ilişkin durum geçiş fonksiyonu, aktifleşmiş geçişlerin tetiklenmesiyle PA'nın bulunduğu durumunu değiştirmesi olarak tanımlanır [9,11]. Durum geçiş mekanizmasına değinilmeden önce "aktifleşmiş geçiş" kavramının tanımı yapılmalıdır:

Tanım 2.7: PA'daki bir $t_j \in T$ geçişinin aktif hale gelmesi için aşağıdaki koşul sağlanmalıdır [9]:

$$x(p_i) \geq w(p_i, t_j) \quad , \quad \forall p_i \in I(t_j) \quad (2.6)$$

Sözlü olarak ifade etmek gerekirse; PA'daki bir t_j geçişi, her p_i giriş yerindeki jeton sayısı en az söz konusu p_i yerini t_j geçişine bağlayan yayın ağırlık değeri kadarsa aktifleşmiş demektir [9].

Tanım 2.8: Bir t_j geçişi tetiklendikten sonra bu geçişin her bir giriş yerindeki jeton sayısı $x'(p_i)$ aşağıdaki gibi tanımlanır:

$$x'(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i) \quad , \quad i = 1, \dots, n \quad (2.7)$$

(2.7)'de verilen $x'(p_i)$ ifadesinin t_j geçişinin p_i giriş yerindeki mevcut jeton sayısına ve p_i geçişi ile t_j geçişi arasındaki yayların ağırlık değerlerine bağlı olduğu görülmektedir [9].

(2.7)'ye göre t_j geiři tetiklendikten sonra, eęer bir p_i yeri t_j geiřinin giriř yeri ise bu yerdeki jeton sayısı p_i 'yi t_j 'ye baęlayan yayın aęırlık deęeri kadar azaltılır; eęer p_i yeri t_j 'nin ıkıř yeri ise bu yerdeki jeton sayısı t_j 'yi p_i 'ye baęlayan yayın aęırlık deęeri kadar arttırılır. Bir p_i yerinin bir t_j geiřine ait hem giriř hem de ıkıř yeri olması mmkndr. Bu durumda p_i yerinden $w(p_i, t_j)$ kadar jeton alınır ve hemen ardından yine aynı p_i yerine $w(t_j, p_i)$ kadar jeton konulur [9].

Yapılan tanımlar ıřıęında PA'da bir geiřin tetiklenmesine iliřkin olarak ařaęıdakiler sylenebilir:

- Eęer bir geiře iliřkin btn giriř yerleri en az bu yerleri ilgili geiře baęlayan yayların aęırlık deęeri kadar jeton ieriyorsa sz konusu geiř aktifleřmiř demektir.
- Aktifleřmiř bir geiř tetiklenebilir. Bylelikle PA'nın bir durumdan bařka bir duruma gemesi saęlanabilir.
- Aktifleřmiř bir geiř tetiklendięinde her bir giriř yerinden o giriř yerini sz konusu geiře baęlayan yayın aęırlık deęeri kadar jeton alınır.
- Aktifleřmiř bir geiř tetiklendięinde her bir ıkıř yerine sz konusu geiři o ıkıř yerine baęlayan yayın aęırlık deęeri kadar jeton konulur.

2.3 Durum Denklemleri

Bir PA'nın herhangi bir tetikleme gerekleřtikten sonraki durum bilgisi matematiksel olarak elde edilebilir. Bu amala ařaęıdaki tanımlar yapılmıřtır:

Tanım 2.9: $m -$ boyutlu bir u tetikleme vektr ařaęıdaki gibi tanımlanır [9]:

$$u = [0, \dots, 0, 1, 0, \dots, 0] \quad (2.8)$$

Burada j . geiřin tetikleneceęinin belirtilmesi iin u vektrnn sadece j . elemanı 1 (bir) deęerini alır. Dięer elemanlar 0 (sıfır) olarak kalır.

Tanım 2.10: $m \times n$ boyutlu D sistem matrisi (*incidence matrix*) ařaęıdaki gibi tanımlanır [9]:

$$d_{ji} = w(t_j, p_i) - w(p_i, t_j) \quad (2.9)$$

Tanım 2.11: x başlangıç durum vektörü ve x' bir sonraki durum vektörü olmak üzere (2.8) ve (2.9) tanımları da kullanılarak PA'ya ilişkin bir sonraki durum vektörü aşağıdaki gibi ifade edilir [9]:

$$x' = x + uD \quad (2.10)$$

Aşağıda bu tanımın genişletilmiş hali verilmiştir:

Tanım 2.12: x_0 ilk durumu verilmiş bir PA'ya ilişkin tüm durumlar aşağıdaki denklem ile elde edilebilir [9]:

$$x_{k+1} = f(x_k, t_k) = x_k + u_k D \quad (2.11)$$

Burada u_k , k .geçişin tetikleneceği bilgisini taşımaktadır.

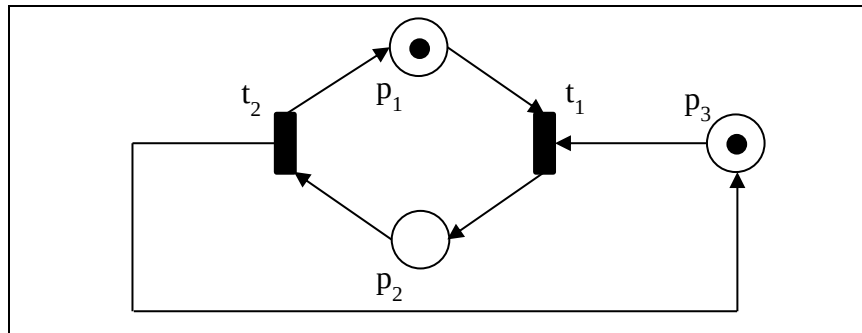
2.4 Sınırlılık Kavramı

PA'daki yerlerde bulunan jetonlar genellikle sistemdeki kaynakları temsil etmektedir. Bu kaynakların sınırsız olması sistemin kararsız olmasına neden olacağından istenen bir durum değildir. Sistemin kararlı olması için gerekli koşullardan biri de PA'daki her bir yerdeki jeton sayısının belirli bir sayıdan fazla olmamasıdır. Bu duruma ilişkin tanım aşağıda verilmiştir [12]:

Tanım 2.13: PA'daki bir $p_i \in P$ yerinin k -sınırlı veya k -güvenli olabilmesi için PA'daki olası bütün erişilebilir durumlar için aşağıdaki koşul geçerli olmalıdır [12]:

$$x(p_i) \leq k \quad (2.12)$$

(2.12)'ye göre PA'daki bir p_i yeri 1-sınırlı ise, bu yer “güvenli yer” olarak adlandırılır. Tüm yerlerin güvenli olduğu PA ise “1-sınırlı PA” olarak tanımlanır [13]. Şekil 2.2'de 1-sınırlı PA'ya örnek verilmiştir:



Şekil 2.2 : 1-sınırlı PA'ya örnek.

3 PETRİ AĞLARI İLE TASARIM YÖNTEMLERİ

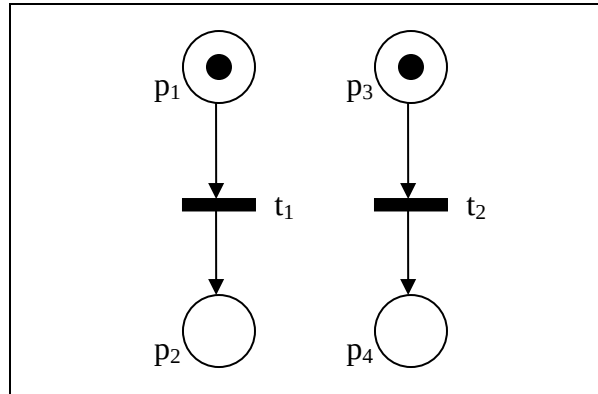
3.1 Primitif Yapılar

Gerçek sistemler modellenirken bazı sistem davranışlarının ifade edilmesi için bir takım primitif PA yapılarından yararlanılır. Bu yapılar aşağıda verilmiştir [14]:

- Paralellik (*Concurrency*)
- Senkronizasyon (*Synchronization*)
- Çakışma (*Conflict*)
- Sıralama (*Sequentiality*)
- Kaynak Sınırlandırma (*Resource Limitation*)

3.1.1 Paralellik yapısı

Bağımsız olarak gerçekleşebilen iki ya da daha fazla olayın modellenmesinde kullanılan yapıdır. Bu yapıda olaylar birbirinden bağımsız olarak aynı anda gerçekleşebilir [14].

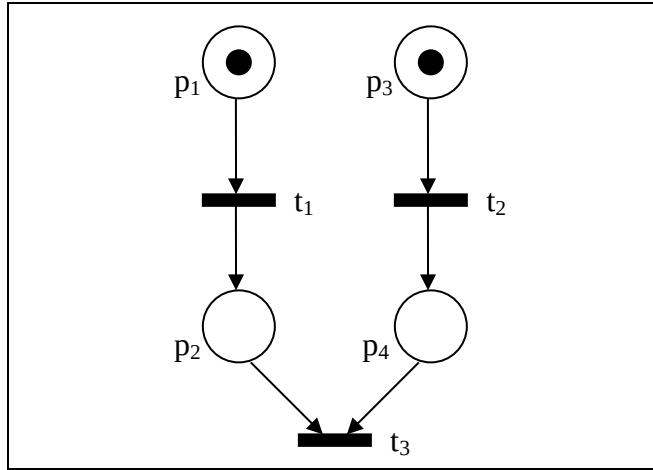


Şekil 3.1 : Paralellik yapısına örnek.

Şekil 3.1’de verilen PA’da t_1 ve t_2 geçişleri aynı anda tetiklenebilmektedir [14].

3.1.2 Senkronizasyon yapısı

Bir geçişin tetiklenebilmesi için iki ya da daha fazla koşulun sağlanmış olmasını gerektiren bir yapıdır. Bu yapıda herhangi bir geçiş ancak tüm koşullar sağlandığı takdirde tetiklenebilir. Eğer koşullardan en az biri bile sağlanmıyorsa ilgili geçiş tetiklenmez [14].

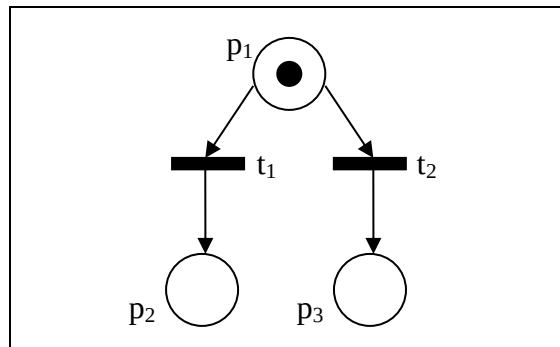


Şekil 3.2 : Senkronizasyon yapısına örnek.

Şekil 3.2’de verilen PA’da t_3 geçişinin tetiklenebilmesi için hem t_1 hem de t_2 geçişinin tetiklenmiş olması gerekmektedir [14].

3.1.3 Çakışma yapısı

Bir sistemde aynı kaynağı paylaşan iki ya da daha fazla olay olabilir. Bu durumda kaynağa aynı anda sadece bir olayın erişmesine izin verilmeli ve diğer olayların söz konusu kaynağa olan erişimleri engellenmelidir. Bunu sağlayan yapı “çakışma yapısı” olarak adlandırılır [14].

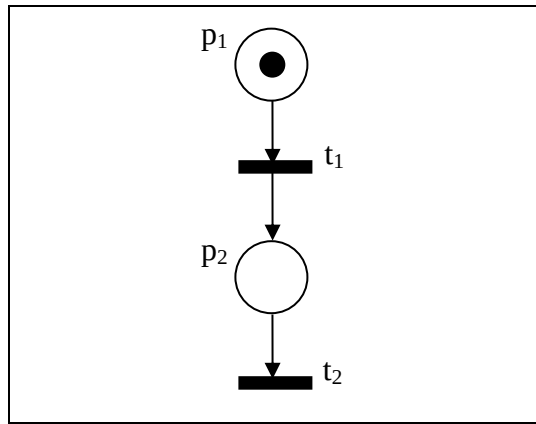


Şekil 3.3 : Çakışma yapısına örnek.

Şekil 3.3'te verilen PA'da p_1 yeri hem t_1 hem de t_2 geçişleri tarafından paylaşılmakta olan ortak bir kaynağı temsil etmektedir. Bu durumda t_1 ve t_2 geçişleri aynı anda tetiklenemez. Başka bir deyişle ya t_1 geçişi ya da t_2 geçişi tetiklenecektir. Bunun seçimi sistemle ilgili koşullara bağlı olacaktır [14].

3.1.4 Sıralama yapısı

Olayların belirli bir sırada gerçekleştiği sistemlerin modellenmesi gerekebilir. Bu durumda “sıralama yapısı” adı verilen yapı kullanılır. Böylelikle olayların gerçekleşme sırası belirlenip modellenenbilir [14].

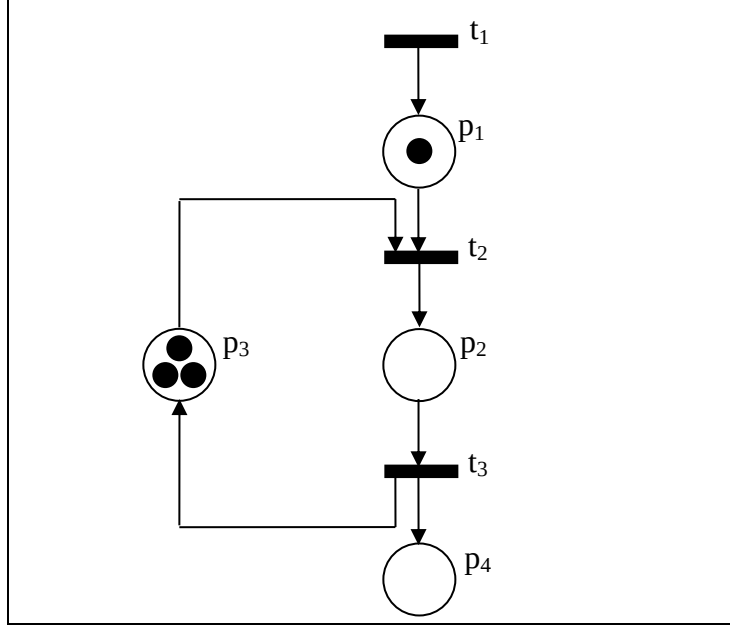


Şekil 3.4 : Sıralama yapısına örnek.

Şekil 3.4'te verilen PA'da t_1 geçişi her zaman t_2 geçişinden önce tetiklenir. Böylelikle olayların gerçekleşme sırası modellenmiş olur [14].

3.1.5 Kaynak sınırlandırma yapısı

Modellenen sistemlerde sıklıkla rastlanan bir durum da kaynakların sınırlı olmasıdır. Genellikle bu duruma üretim sistemlerinde rastlanır. Bu gibi sistemlerde kaynaklar sınırlı olmakla birlikte birçok süreç tarafından da paylaşılabilir. Sınırlı kaynaklara sahip sistemlerin modellenmesinde “kaynak sınırlandırma yapısı” adı verilen yapı kullanılır [14].



Şekil 3.5 : Kaynak sınırlandırma yapısına örnek.

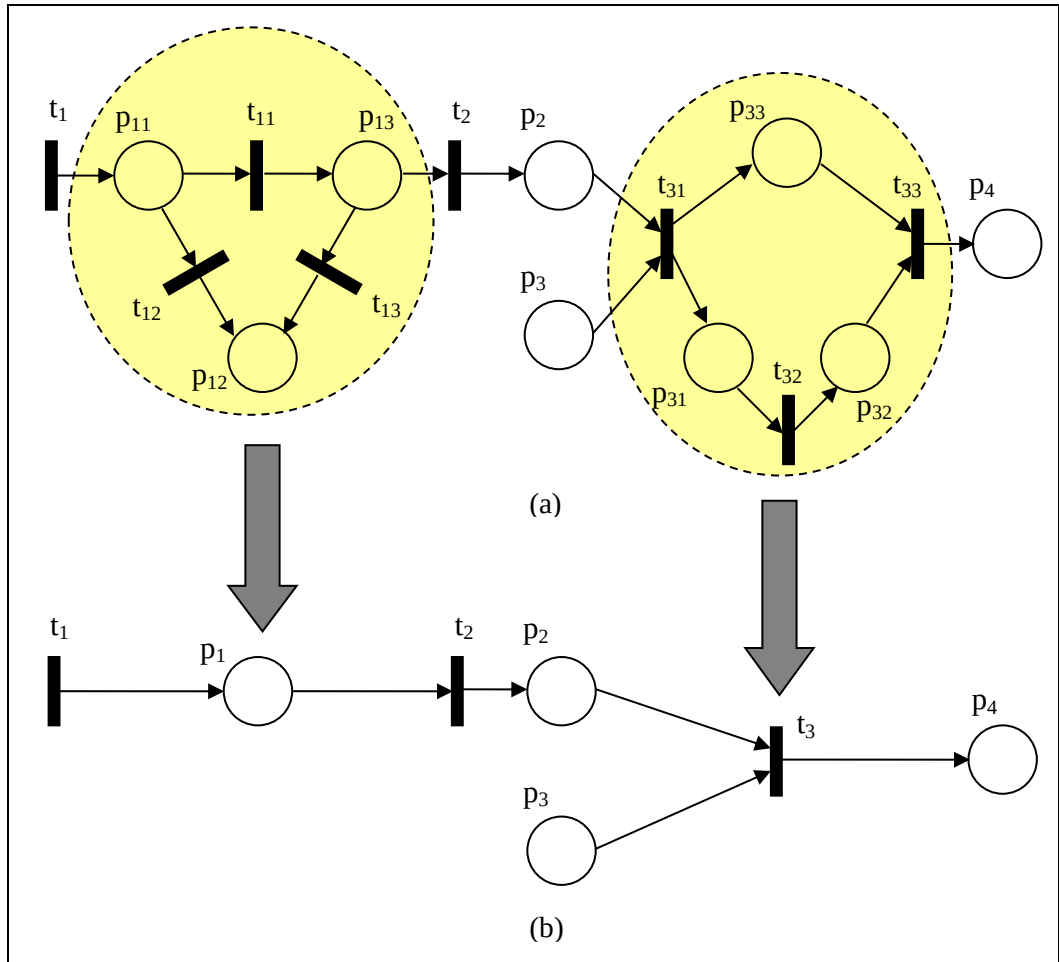
Şekil 3.5’te verilen PA’da kaynakları sınırlı olan bir işlem kuyruğu modellenmiştir. Bu kuyruğun bir bilgisayar işlemcisine ait işlem kuyruğu olduğu düşünülebilir. Bu yapıda p_1 kuyrukta bekleyen işlemleri, p_2 yürütülen işlemleri ve p_4 ise tamamlanan işlemleri göstermektedir. p_3 ise asıl kaynak sınırlandırmayı sağlayan yerdir ve kuyruğun kapasitesini belirlemektedir. O halde bu kuyruğun kapasitesi başlangıçta p_3 yerinde bulunan jeton sayısı kadardır. Bu örnekte kuyruğun toplam kapasitesi üçtür. Dikkat edilecek olursa süreç boyunca p_2 ve p_3 yerlerindeki toplam jeton sayısının sabit kaldığı ve kuyruğun toplam kapasitesini ifade ettiği görülmektedir [14].

3.2 Petri Ağları’nın Türleri

Klasik PA’da zamanla birtakım eksiklikler boy göstermiş ve bu nedenle PA sistemlerin modellenmesinde yetersiz kalmaya başlamıştır. Klasik PA ile karmaşık süreçler kolay bir şekilde modellenememekte, süreçlerin birbirinden ayırt edilmesi sağlanamamakta ve zaman kavramı ifade edilememektedir. Bu eksikliklerin giderilmesi için farklı PA türleri ortaya atılmış ve PA’nın modelleme yeteneği arttırılmıştır. “Modüler PA”, “Renkli PA” ve “Zaman Etiketli PA” ortaya atılan bu PA türlerine örnektir. Modüler PA ile karmaşık yapıdaki sistemlerin modellenmesi ve incelenmesi kolaylaşmış, Renkli PA ile sistemdeki olayların takip edilebilmesi sağlanmış ve Zaman Etiketli PA ile zamana dayalı sistemler modellenebilir hale gelmiştir [14].

3.2.1 Modüler Petri Ağları

Klasik PA ile basit yapıdaki sistemlerin modellenmesi ve ifade edilmesi oldukça kolaydır. Ancak karmaşık sistemler modellenmek istendiğinde ortaya oldukça karmaşık yapıda bir PA çıkmaktadır. Bu nedenle süreçlerin izlenmesi ve hataların bulunması zorlaşmaktadır. Bu sorunun ortadan kaldırılması için Modüler PA ortaya atılmıştır. Bu PA türü ile karmaşık sistemler modüler olarak ifade edilebilmekte ve detaylar bu modüllerin içinde gizlenerek gösterim kolaylığı sağlanabilmektedir (Şekil 3.6):



Şekil 3.6 : (a) Klasik PA'ya örnek. (b) Modüler PA karşılığı.

Şekil 3.6(a)'da verilmiş olan klasik PA modülerlik özelliği kullanılıp sadeleştirilerek Şekil 3.6(b)'deki Modüler PA'ya dönüştürülmüştür. Bu işlem sonucunda p_1 yeri ve t_3 geçişi elde edilmiştir. Böylelikle analiz edilmesi daha kolay bir model oluşturulmuştur.

3.2.2 Renkli Petri Ağları

Klasik PA’da kullanılan jetonlar renksizdir. Başka bir deyişle herhangi bir jetonu diğerlerinden ayırt etmek mümkün değildir. Bu nedenle ortaya çıkan modellerde farklı sistem kaynakları birbirinin aynısıymış gibi gözükür ve süreçlerin anlık olarak hangi kaynakları kullandığının takibi zorlaşır. Jetonların takip edilebilmesi ve dolayısıyla modellenen sistemin daha kolay anlaşılıp analiz edilebilmesi amacıyla Danimarka’daki Aarhus Üniversitesi tarafından Renkli PA kavramı ortaya atılmıştır. Renkli PA’da sistemlerin modellenmesi daha basit ve anlaşılabilir şekilde gerçekleştirilebilir. Bir Renkli PA aşağıdaki gibi tanımlanır [15]:

$$RPA = (\Sigma, P, T, A, N, C, G, E, I) \quad (3.1)$$

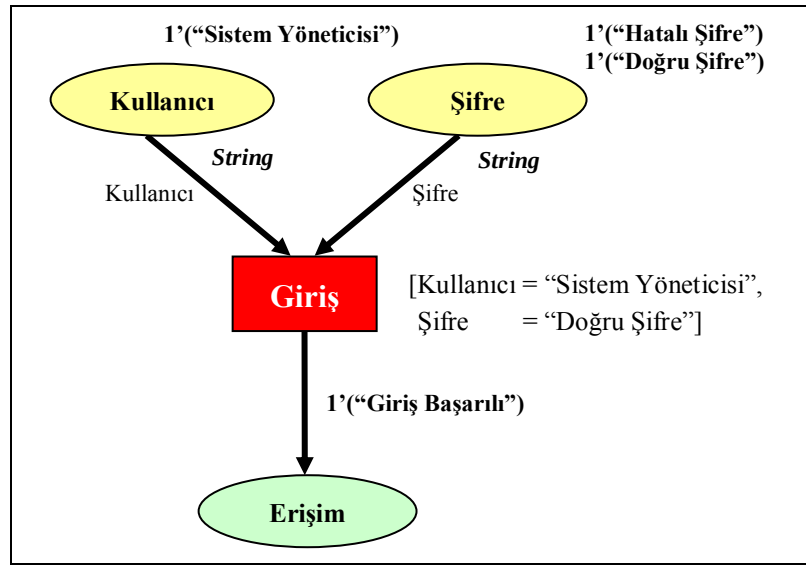
Burada;

- Σ : Sonlu renkler kümesini tanımlar.
- P : Sonlu yerler kümesini ifade eder.
- T : Sonlu geçişler kümesini ifade eder.
- A : Sonlu yaylar kümesini ifade eder.
- N : Düğüm fonksiyonunu tanımlar. A kümesinden $P \times T \cup T \times P$ ’ye tanımlıdır.
- C : Renk fonksiyonunu tanımlar. P kümesinden Σ kümesine tanımlıdır.
- G : Ek koşul ifadesidir (*guard*). Birden fazla olabilir.
- E : Yay ifadesidir (*arc expression*).
- I : İlkendirme fonksiyonudur.

Renkli PA’da her bir yer bir renk kümesi ile ilişkilendirilmiştir. Böylelikle söz konusu yerde sadece belirtilen renklere sahip jetonların bulunması sağlanmıştır. Her bir jetona renk kümesinden renkler/değerler atanmıştır. Bir yerde ancak kendi renk kümesindeki renklere sahip jetonlar bulunabilir. Klasik PA’nın aksine Renkli PA’da jetonlar için siyah renkli daireler kullanılmaz. Renkli PA’da bir yerdeki jetonların sayısı ve bu jetonlara ait renkler söz konusu yerin sağ üst köşesine yazılır [15].

Renkli PA’da bir geçişin tetiklenebilmesi için gerekli koşullar Klasik PA ile aynıdır. Ancak Renkli PA’da geçişlerde ek koşullar bulunabilir. Bu durumda tetiklenmenin gerçekleşebilmesi için gerekli koşulların yanı sıra bu ek koşulların da sağlanması gerekir. Bu koşullar “*guard*” olarak adlandırılır ve ilgili geçişin yanına yazılır.

Tetiklenme sonrasında giriş yerlerinden hangi renkte kaçar tane jeton alınacağı ve çıkış yerlerine hangi renkte kaçar tane jeton konulacağı yay ifadesi (*arc expression*) ile belirlenir. Bu ifadeler yayların üzerine yazılır. Renkli PA’nın daha kolay anlaşılabilmesi bakımından Şekil 3.7’de bir örnek verilmiştir [15]:



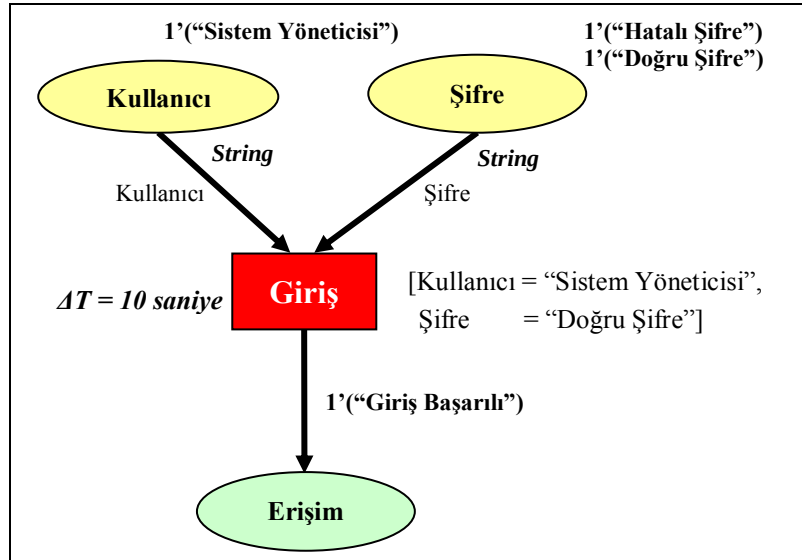
Şekil 3.7 : Renkli PA’ya örnek.

Şekil 3.7’de verilen örnekte bir bilgisayar sistemine girişte kullanıcı kimliğinin doğrulanması süreci Renkli PA ile modellenmiştir. Sistemde üç adet yer bulunmaktadır. Bunlardan “Kullanıcı” yerinde sisteme girişte kullanılan kullanıcı adı, “Şifre” yerinde kullanıcıya ilişkin şifre bulunmaktadır. Sonuç olarak doğru kullanıcı adı ve şifre girildiğinde “Erişim” yerinde girişin başarılı olduğunu belirten “Giriş Başarılı” karakter dizisi bulunacaktır. Modelin işleyişi şu şekilde anlatılabilir: İlk durumda “Kullanıcı” yerinde 1 adet “Sistem Yöneticisi”, “Şifre” yerinde ise 1 adet “Hatalı Şifre” ve 1 adet “Doğru Şifre” bulunmaktadır. Bu durumda “Giriş” geçişi aktif durumdadır. Ancak dikkat edilecek olursa bu geçişin tetiklenmesi için ek bir koşul bulunmaktadır. Bu koşula göre “Kullanıcı” yerindeki renk değeri “Sistem Yöneticisi” ve “Şifre” yerindeki renk değeri ise “Doğru Şifre” olduğunda söz konusu geçiş tetiklenebilecektir. Aksi takdirde bu geçişin tetiklenmesi mümkün olmayacaktır. Sistemde bu koşul sağlanmış durumdadır. Dolayısıyla geçiş

tetiklenecek ve giriş yerlerinde bulunan jetonların uygun olanlarından 1'er adet harcanıp "Erişim" yerine 1 adet "Giriş Başarılı" jetonu konulacaktır. Dikkat edilirse giriş yerlerinden hangi renkte kaç jeton alınacağı ve çıkış yerine hangi renkte kaç jeton konulacağı yayların üzerinde belirtilmiştir [15].

3.2.3 Zaman Etiketli Petri Ağları

Bir sistemdeki olayların modellenmesinde zaman kavramının da belirtilmesi amacıyla Zaman Etiketli PA ortaya atılmıştır. Zaman Etiketli PA'da geçişlerin tetiklenebilmesi için bir zamanlama mekanizması bulunmaktadır. Bu PA türünde ilgili geçişlere bir ΔT gecikme süresi atanmaktadır. Eğer zaman etiketi atanmış bir geçiş aktif hale gelirse ΔT süresi kadar bekler ve ondan sonra tetiklenir. Sadece tetikleme gerçekleştikten sonra ilgili girişe ait jetonlar giriş yerlerinden alınır ve çıkış yerlerine yeni jetonlar konulur. Aktif hale gelmiş zaman etiketli bir geçiş tetikleninceye kadar bu geçişe ilişkin hiçbir jeton transferi gerçekleşmez [15].



Şekil 3.8 : Zaman Etiketli PA'ya örnek.

Şekil 3.7'de verilmiş olan Renkli PA'ya zaman etiketi eklenerek Şekil 3.8'deki Zaman Etiketli Renkli PA elde edilmiştir. Dikkat edilecek olursa "Giriş" geçişine "10 saniye" değerinde bir zaman etiketi eklenmiştir. Bu durumda doğru kullanıcı adı ve şifresi girildikten 10 saniye sonra jetonlar harcanacak ve sisteme başarılı giriş yapılabilecektir [15].

4 GENİŞLETİLMİŞ PETRİ AĞLARI

4.1 Klasik Petri Ağlarına Eklentiler

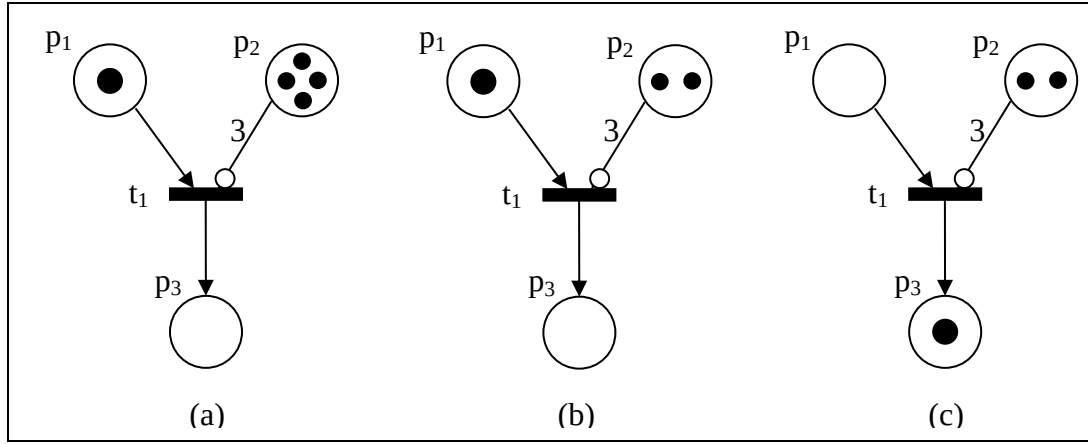
4.1.1 Yasaklayıcı yay

Olayların akışının farklı şekillerde ifade edilmesi amacıyla PA'ya sonradan birçok eklenti yapılmıştır. Bu eklentilerden birisi de “yasaklayıcı yaylar”dır (*inhibitor arc*). Yasaklayıcı yaylar bir geçişin tetiklenmesini engellemek amacıyla kullanılmaktadır [10]. Bu yaylar ile belirli bir geçişin diğer geçişlerden önce tetiklenmesi sağlanabilir.

Bir p_i yerinden t_j geçişine çizilmiş “n” ağırlıklı bir yasaklayıcı yay mevcutsa, p_i yerinde en az “n” adet jeton bulunması durumunda t_j geçişinin aktif halde olup olmamasına bakılmaksızın tetiklenmesi engellenecektir. Bu durum şöyle ifade edilebilir: Söz konusu geçişin tetiklenmesi için uygun koşullar oluşmuştur ancak ortamda bu geçişin tetiklenmesini engelleyen bir takım olumsuz koşullar da mevcuttur. Bu nedenle bu geçiş tetiklenemeyecektir. Yasaklayıcı yaylar aslında bu olumsuz koşulları ifade etmektedir. Olumsuz durumlar ortadan kalkıp yasaklayıcı yaylar etkinliğini yitirdiğinde söz konusu geçiş aktif hale gelip tetiklenebilecektir [16].

Yasaklayıcı yay, ucu boş daire ile biten yay ile ($\text{---}\bigcirc$) gösterilir. Şekil 4.1’de yasaklayıcı yayla ilişkin örnek bir PA verilmiştir. Bu PA’da (p_2, t_1) yayı bir yasaklayıcı yaydır ve bu yayın ağırlık değeri 3’tür. p_2 yerinde en az 3 adet jeton bulunması durumunda t_1 geçişinin aktif hale gelip tetiklenmesi engellenecektir. Ancak p_2 yerindeki jeton sayısının 3’ten az olması durumunda t_1 geçişi aktif hale gelip tetiklenebilecektir. Şekil 4.1(a)’da p_1 yerinde 1, p_2 yerinde 4 adet jeton bulunmaktadır. Eğer yasaklayıcı yay hiç dikkate alınmasaydı, p_1 yerinde 1 jeton bulunduğundan t_1 geçişi aktif hale gelip tetiklenebilecekti. Ancak p_2 yerinde 4 adet jeton olduğundan (p_2, t_1) yasaklayıcı yayı t_1 geçişinin aktif hale gelmesini engellemektedir. Şekil 4.1(b)’de p_1 yerinde 1, p_2 yerinde ise 2 adet jeton bulunmaktadır. Yasaklayıcı yayın t_1 geçişinin aktif hale gelmesini engellemesi için p_2 yerinde en az 3 adet jeton bulunmalıdır. Ancak p_2 yerinde 2 adet jeton

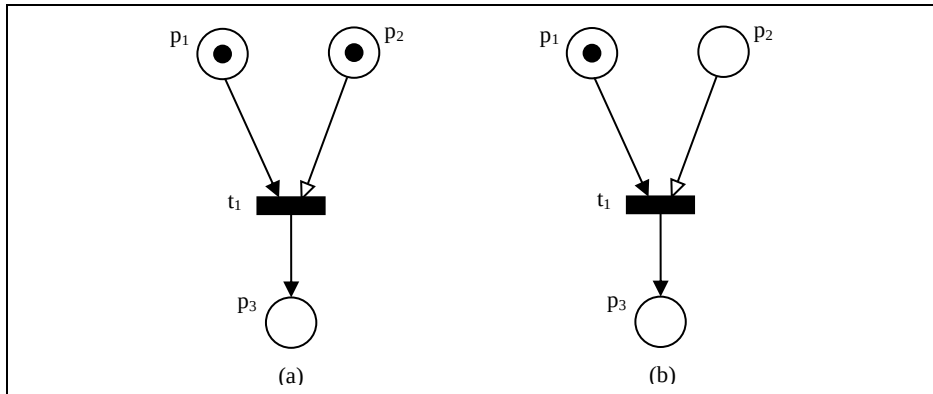
bulunmaktadır. Bu nedenle yasaklayıcı yay t_1 geçişinin tetiklenmesini engellemektedir. Sonuç olarak t_1 geçişi tetiklenmekte ve p_1 yerinden 1 jeton alınıp p_3 yerine 1 jeton konulmaktadır. Bu durum Şekil 4.1(c)'de gösterilmektedir. Dikkat edilecek olursa t_1 geçişi tetiklendikten sonra p_2 yerindeki jeton sayısı değişmemektedir. Bir PA'da tetikleme sonrası yasaklayıcı yay(lar)a bağlı olan yer(ler)de herhangi bir jeton transferi gerçekleşmez. Dolayısıyla bu yer(ler)deki jeton sayısı sabit kalır.



Şekil 4.1 : (a) t_1 geçişi aktif değil. (b) t_1 geçişi aktif. (c) t_1 geçişi tetiklendikten sonra.

4.1.2 Yetkilendirici yay

PA'nın modelleme yeteneği "yetkilendirici yay" (*enabling arc*) kavramı ile daha da artırılabilir. PA'da p_i yerinden t_j geçişine çizilmiş "n" ağırlıklı bir yetkilendirici yay t_j geçişinin tetiklenme mekanizmasını şu şekilde etkiler: t_j geçişi, eğer p_i yerinde en az "n" adet jeton varsa aktif hale gelebilir ve tetikleme gerçekleştiikten sonra yetkilendirici yayın bağlı olduğu giriş yerinden herhangi bir jeton harcanmaz. Yetkilendirici yay, ucu boş okla biten yay ile ($\rightarrow$) gösterilir [17].

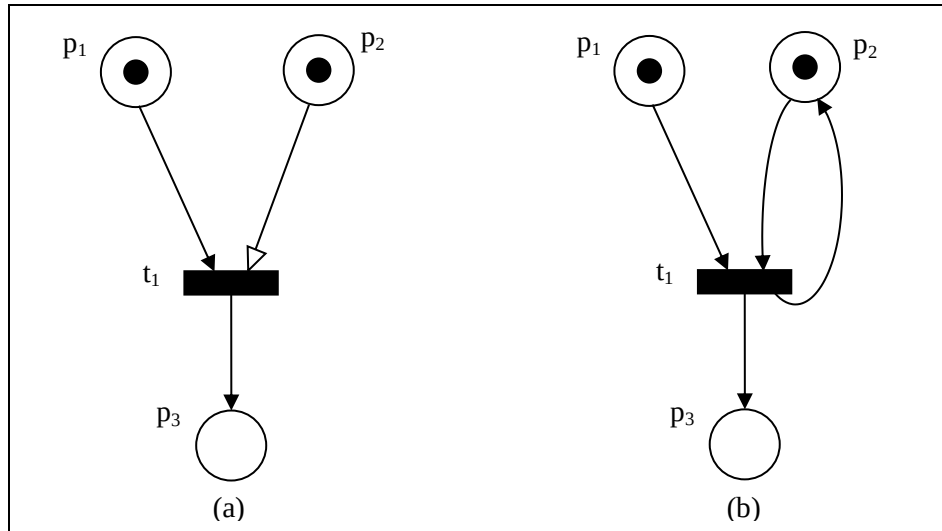


Şekil 4.2 : (a) t_1 geçişi aktif. (b) t_1 geçişi aktif değil.

Şekil 4.2'deki örnekte yetkilendirici yay içeren bir PA'nın iki farklı durumu ele alınmaktadır. Burada p_2 yerinden t_1 geçişine çizilmiş olan bir yetkilendirici yay görülmektedir. t_1 geçişinin aktif hale gelip tetiklenebilmesi için p_1 ve p_2 yerinde en az 1'er adet jeton bulunması gerekmektedir. Şekil 4.2(a)'da p_1 yerinde 1, yetkilendirici yayın giriş yeri olan p_2 yerinde de 1 adet jeton olmasından dolayı t_1 geçişi aktif hale gelip tetiklenebilecektir. Şekil 4.2(b)'de ise p_1 yerinde jeton bulunmasına karşın yetkilendirici yaya ilişkin p_2 yerinde hiçbir jeton bulunmamaktadır. t_1 geçişinin aktif hale gelmesi için p_2 yerinde de jeton bulunması gerekmektedir. Yetkilendirici yaya ilişkin p_2 yerinde herhangi bir jeton bulunmadığı için t_1 geçişi aktif hale gelip tetiklenemeyecektir. Aslında bu durum yetkilendirici yayın yerinde standart yayın bulunması durumu ile aynıdır. Aradaki tek fark, tetikleme gerçekleştiğinde yetkilendirici yayın bağlı olduğu p_2 giriş yerindeki jetonun harcanmamasıdır [17].

Tıpkı diğer yaylar gibi yetkilendirici yaylar için de ağırlık değeri kullanılabilir. Bu durumda örneğin p_i yerinden t_j geçişine çizilmiş "n" ağırlıklı bir yetkilendirici yay varsa, t_j geçişinin aktif hale gelip tetiklenebilmesi için p_i yerinde en az "n" adet jeton bulunması gerekir.

Yetkilendirici yay, standart yaylarla da ifade edilebilmektedir. Yetkilendirici yayın standart yaylar ile karşılığı Şekil 4.3'te verilmiştir:



Şekil 4.3 : (a) Yetkilendirici yay. (b) Yetkilendirici yayın karşılığı.

Şekil 4.3(a)'da yetkilendirici yay içeren bir PA verilmiş, Şekil 4.3(b)'de ise PA'daki bu yetkilendirici yayın standart yaylarla ifade edilmiş hali verilmiştir. Dikkat edilecek olursa yetkilendirici yay, p_2 yerinden t_1 geçişine giden ve t_1 geçişinden

tekrar p_2 yerine giden iki ayrı standart yay ile ifade edilmiştir. Bu gösterim şeklinin yetkilendirici yayın karşılığı olabilmesi için aşağıdaki iki koşulun sağlanması gerekmektedir:

1. t_1 geçişinin aktif hale gelmesi için bu geçişin bağlı olduğu p_2 yerinde en az 1 adet jeton bulunması gerekmektedir.
2. t_1 geçişi aktif hale gelip tetiklendiği zaman bu geçişin bağlı olduğu p_2 yerindeki jeton sayısı sabit kalmalıdır.

Şekil 4.3(b)'deki gösterimde t_1 geçişinin ancak ve ancak p_1 ve p_2 yerlerinde en az 1'er adet jeton bulunması durumunda aktif hale gelip tetiklenebileceği açıktır. Bu durumda 1 no'lu koşul sağlanmış olur. t_1 geçişi aktif hale gelip tetiklendikten sonra p_2 yerinden t_1 geçişine giden yay nedeniyle ilk başta p_2 yerindeki jeton harcanır. Ancak t_1 geçişinden p_2 yerine giden yay aracılığıyla p_2 yerine jeton konulur. Böylelikle p_2 yerindeki jeton korunmuş olur. Sonuçta p_2 yerinden aslında hiçbir jeton harcanmamış gibi bir durum ortaya çıkar. Böylelikle 2 no'lu koşul da sağlanmış olur. Dolayısıyla Şekil 4.3(a)'daki yetkilendirici yayın Şekil 4.3(b)'deki gibi iki ayrı standart yay ile ifade edilebileceği sonucuna varılır.

4.1.3 Yutucu yay

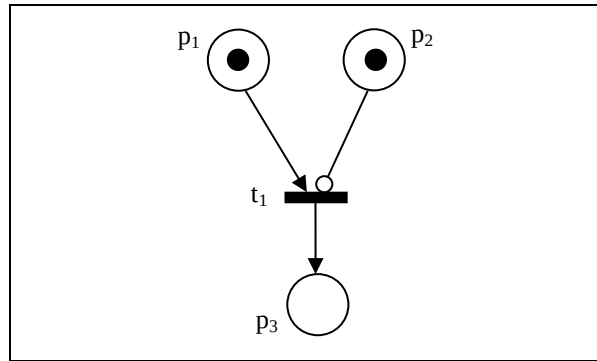
Bazı durumlarda herhangi bir giriş yerindeki jetonların sayısı ne olursa olsun hepsinin bir anda harcanmasına gereksinim duyulabilir. Bunun için söz konusu yerdeki anlık jeton sayısının önceden bilinmesi gerekmektedir. Örneğin herhangi bir yerdeki anlık jeton sayısının 5 olacağı önceden bilinirse o zaman bu yere ilişkin yayın ağırlık değeri 5 olarak ayarlanıp tüm jetonların sıfırlanması sağlanabilir. Ancak gerek yerlerdeki jeton sayısının önceden bilinmemesi gerekse yay ağırlık değerlerinin dinamik olarak ayarlanamamasından dolayı klasik PA'da böyle bir davranışın gerçekleşmesi mümkün değildir. Dolayısıyla klasik PA'da jetonların aynı anda sıfırlanmasını sağlayan bir mekanizma bulunmamaktadır. Bu amaçla “yutucu yay” (*nullifier arc*) kavramı ortaya atılmıştır. Bu yayın bağlı olduğu yerdeki jeton sayısına bakılmaksızın bu yerdeki tüm jetonlar tek seferde harcanıp sıfırlanabilir.

4.2 Yaylar için Durum Denklemleri

Yapılan literatür taramasında yasaklayıcı, yetkilendirici ve yutucu yayların PA'nın durum denklemlerine dâhil edilmesi için kullanılabilir herhangi bir yaklaşıma rastlanmamıştır. Bilindiği gibi bu yapılar PA'nın modelleme yeteneğini arttırmak ve daha karmaşık sistemlerin modellenmesini sağlamak amacıyla ortaya atılmıştır. Bu anlamda söz konusu yapıları içeren PA'nın bilgisayar ortamında işlenebilmesi büyük önem taşımaktadır. Bu amaçla söz konusu yapıların PA'nın durum denklemlerine dâhil edilerek matematiksel olarak ifade edilmesi gerekmektedir. Böylelikle bu yapıları içeren PA'nın bilgisayar ortamında tasarlanması, benzetiminin yapılması, analiz edilmesi, doğrulanması ve test edilmesi gerçekleştirilebilir.

4.2.1 Yasaklayıcı yay

Yasaklayıcı yayların durum denkleminde ifade edilmesi için yeni bir yaklaşım önerilecektir. Bu yaklaşımın vurgulanması ve daha kolay anlaşılması bakımından Şekil 4.4'teki örnek PA ele alınmıştır:



Şekil 4.4 : Yasaklayıcı yay içeren bir PA.

Bu PA'da p_2 yerinden t_1 geçişine çizilmiş olan yasaklayıcı yay görülmektedir. p_1 yerinde jeton olduğu için normalde t_1 geçişinin aktif olması gerekmektedir. Ancak p_2 yerinde en az 1 tane jeton bulunduğundan t_1 geçişi pasif hale gelip tetiklenemeyecektir. Bu PA'nın bir sonraki durumu yasaklayıcı yayın etkisi dikkate alınmadan aşağıdaki gibi hesaplanabilir:

$$x' = x + uD$$

$$x' = [1 \ 1 \ 0] + [1][-1 \ -1 \ 1]$$

$$x' = [0 \ 0 \ 1] \quad (4.1)$$

(4.1)'deki durum denkleminde yasaklayıcı yayın etkisi dikkate alınmamakta ve bu yay, standart bir yay gibi değerlendirilmektedir. Bu nedenle bir sonraki durum bilgisi doğru olarak elde edilememiştir. Normalde bir sonraki durum bilgisinin değişmemiş olması beklenmektedir, çünkü yasaklayıcı yayın bağlı olduğu yerde 1 adet jeton bulunduğundan ilgili geçişin aktif hale gelip tetiklenmesi engellenmekte ve bu nedenle herhangi bir jeton transferi gerçekleşmemektedir. Bir sonraki durum bilgisinin doğru olarak elde edilebilmesi için yasaklayıcı yayların standart durum denkleminde matematiksel olarak ifade edilmesi gerekmektedir. Bu amaçla $m \times m$ boyutlu yeni bir matris sunulacaktır. Bu matrise yasaklayıcı matris (*inhibitor matrix*) adı verilip $\tilde{H}(x)$ ile gösterilecektir. $\tilde{H}(x)$ matrisinin tanımı aşağıda verilmiştir:

$$\tilde{H}(x) = \begin{cases} \tilde{H}_{ij}(x) = 0, & i = 1, \dots, m, \quad j = 1, \dots, m, \quad i \neq j, \\ \tilde{H}_{jj}(x) = 1, & j = 1, \dots, m, \quad t_j \text{ geçişi bir yasaklayıcı yay} \\ & \text{tarafından bloklanmamışsa,} \\ \tilde{H}_{jj}(x) = 0, & j = 1, \dots, m, \quad \text{diğer.} \end{cases} \quad (4.2)$$

(4.2)'de verilen tanımda “m”, PA'daki toplam geçiş sayısını göstermektedir. Yasaklayıcı matris PA'daki geçişlerin bloklanmış olup olmadığı bilgisini taşımaktadır. Bu bilgi sistemin anlık durumuna bağlı olarak değişebileceği için yasaklayıcı matris değişken bir matris olarak tanımlanmıştır.

Yasaklayıcı yayların durum denkleminde ifade edilebilmesi için sistem matrisinde de ufak bir değişiklik yapılması gerekmektedir. Bilindiği gibi bir geçiş tetiklendiğinde yasaklayıcı yayın bağlı olduğu yerden herhangi bir jeton transferi gerçekleşmemekte ve dolayısıyla bu yerdeki jeton sayısında herhangi bir değişim olmamaktadır. Bu davranışın ifade edilebilmesi için D sistem matrisinde aşağıdaki gibi bir değişiklik yapılmalı ve yeni bir D' sistem matrisi tanımlanmalıdır:

$$D' = \begin{cases} d'_{ji} = 0, & i = 1, \dots, n, \quad j = 1, \dots, m, \quad p_i \text{ ile } t_j \text{ arasında} \\ & \text{bir yasaklayıcı yay varsa,} \\ d'_{ji} = d_{ji}, & i = 1, \dots, n, \quad j = 1, \dots, m, \quad \text{diğer.} \end{cases} \quad (4.3)$$

(4.3)'te verilen tanımda “n” ve “m” sırasıyla PA'daki toplam yer ve geçiş sayısını göstermektedir. Önerilen yeni sistem matrisinde jeton transfer mekanizması yasaklayıcı yaylar için uygun hale getirilmektedir. PA'da sadece standart yayların bulunması durumunda yeni sistem matrisi standart sistem matrisi gibi davranacaktır.

(4.2)'de verilen yasaklayıcı matris ve (4.3)'te verilen yeni sistem matrisi kullanılarak yasaklayıcı yay içeren bir PA'nın yeni durum denklemi aşağıdaki gibi elde edilir:

$$x' = x + u \tilde{H}(x) D' \quad (4.4)$$

(4.4)'te önerilen yeni durum denklemini kullanarak Şekil 4.4'teki PA'nın bir sonraki durum bilgisini tekrar elde etmeye çalışalım. Bu amaçla öncelikle $\tilde{H}(x)$ yasaklayıcı matrisini hesaplayalım. p_2 yerinde en az 1 adet jeton olduğu için t_1 geçişi pasif durumda kalacaktır. (4.2)'de verilen tanım kullanılarak yasaklayıcı matris $\tilde{H}(x)$ aşağıdaki gibi elde edilir:

$$\tilde{H}(x) = [0] \quad (4.5)$$

Yeni sistem matrisinin hesaplanmasında ise p_2 yerinden t_1 geçişine çizilmiş olan bir yasaklayıcı yay bulunduğu göz önüne alınarak “ $d'_{12} = 0$ ” olarak elde edilir. PA'da başka bir yasaklayıcı yay bulunmadığı için yeni sistem matrisinin diğer elemanları standart sistem matrisiyle aynı olacaktır. Bu bilgiler ışığında (4.3)'te verilen tanım kullanılarak yeni sistem matrisi ise aşağıdaki gibi elde edilir:

$$D' = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad (4.6)$$

(4.5)'te elde edilen yasaklayıcı matris ile (4.6)'da elde edilen sistem matrisi (4.4)'teki yeni durum denkleminde yerlerine konulduğunda PA'nın bir sonraki durum bilgisi ifadesi aşağıdaki gibi olur:

$$x' = x + u [0] \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad (4.7)$$

(4.7)'de elde edilen yeni durum denkleminde bakıldığında yasaklayıcı matrisin sıfır olduğu görülmektedir. Böylelikle PA'nın bir sonraki durumu ilk durumuna eşit olarak elde edilir. Bu da PA'da herhangi bir durum değişikliğinin olmadığı anlamına gelir. Gerçekten de yasaklayıcı yay nedeniyle t_1 geçişi tetiklenemeyeceği için hiçbir durum değişikliğinin olmaması gerekmektedir. Bu durum (4.7)'de elde edilen sonucu doğrulamaktadır.

Şimdi de aynı PA'nın farklı bir ilk durumu için bir sonraki durum bilgisini elde etmeye çalışalım. Bu sefer p_2 yerinde herhangi bir jetonun bulunmadığı durumu ele alalım. Bu durumda yasaklayıcı yay t_1 geçişini pasif hale getiremeyecek ve t_1 geçişi aktif hale gelip tetiklenebilecektir. (4.2)'de verilen tanım kullanılarak yasaklayıcı matris $\tilde{H}(x)$ aşağıdaki gibi elde edilir:

$$\tilde{H}(x) = [1] \quad (4.8)$$

D' sistem matrisi önceden (4.6)'da elde edildiği için bu matrisin tekrar hesaplanmasına gerek yoktur. (4.8)'deki yasaklayıcı matris ile (4.6)'daki sistem matrisi (4.4)'teki yeni durum denkleminde yerlerine konulduğunda PA'nın bir sonraki durum bilgisi aşağıdaki gibi hesaplanır:

$$\begin{aligned}
 x' &= x + u [1] [-1 \ 0 \ 1] \\
 x' &= [1 \ 0 \ 0] + [1] [1] [-1 \ 0 \ 1] \\
 x' &= [1 \ 0 \ 0] + [-1 \ 0 \ 1] \\
 x' &= [0 \ 0 \ 1]
 \end{aligned} \tag{4.9}$$

(4.9)'daki bir sonraki durum bilgisi incelendiğinde p_1 yerinde hiçbir jeton kalmadığı ve p_3 yerine 1 adet jeton konulduğu görülmektedir. Elde edilen bu durum bilgisinin doğru olup olmadığını anlamak için PA'nın bir sonraki durum bilgisini bir de grafiksel olarak hesaplayalım. İlk durumda p_2 yerinde hiçbir jeton bulunmadığı için t_1 geçişi aktif hale gelecektir. Böylelikle aktif hale gelen t_1 geçişi tetiklenecek ve p_1 yerindeki 1 adet jeton harcanıp p_3 yerine 1 adet jeton konulacaktır. (4.9)'da elde edilen bir sonraki durum vektörüne bakıldığında bu sonuca varıldığı açıkça görülebilmektedir.

Sonuç olarak yasaklayıcı yaylar için (4.4)'te önerilen yeni durum denkleminin, yasaklayıcı yayların davranışını matematiksel olarak ifade etmekte kullanılabildiği görülmektedir. Dolayısıyla bu durum denklemi kullanılarak yasaklayıcı yay içeren PA'nın bilgisayar ortamında işlenmesi sağlanabilir.

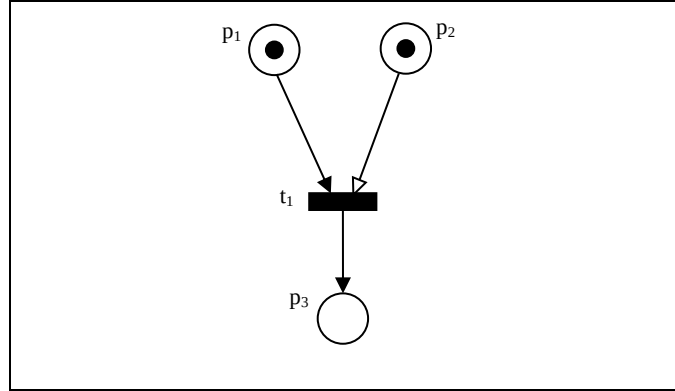
4.2.2 Yetkilendirici yay

Yetkilendirici yay, standart yaylarla ifade edilebilmektedir (bkz. “4.1.2 Yetkilendirici yay”). Dolayısıyla yetkilendirici yay için yeni durum denklemi bilinen yöntemler kullanılarak oluşturulabilir. Bu amaçla yasaklayıcı yay için (4.3)'te verilen sistem matrisinde olduğu gibi her p_i yerinden t_j yerine çizilmiş olan yetkilendirici yay için “ $d'_{ji} = 0$ ” olarak alınır. Sistem matrisinin geri kalan kısmı ise (4.3)'te verilen tanıma göre uygun olarak oluşturulur. Başka bir deyişle (4.3)'teki D' sistem matrisi, yetkilendirici yay bulunduran PA için de benzer şekilde kullanılabilir. Bunun dışında yeni durum denkleminin oluşturulması için başka herhangi bir şeyin yapılmasına

gerek yoktur. Bu bilgiler doğrultusunda yetkilendirici yay içeren bir PA'nın yeni durum denklemi aşağıdaki gibi elde edilir:

$$x' = x + u D' \quad (4.10)$$

(4.10)'da önerilen yeni durum denklemi yetkilendirici yay içeren bir PA'nın bir sonraki durum bilgisinin elde edilmesi için kullanılabilir. Bu amaçla Şekil 4.5'te verilen örnek PA'yı ele alalım:



Şekil 4.5 : Yetkilendirici yay içeren bir PA.

Şekil 4.5'teki PA'da p_2 yerinden t_1 geçişine çizilmiş olan bir yetkilendirici yay bulunduğu için yeni sistem matrisinde " $d'_{12} = 0$ " olarak elde edilir. PA'da başka bir yetkilendirici yay bulunmadığı için yeni sistem matrisinin diğer elemanları standart sistem matrisiyle aynı olacaktır. Bu bilgiler ışığında (4.3)'te verilen tanım kullanılarak yeni sistem matrisi aşağıdaki gibi elde edilir:

$$D' = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad (4.11)$$

(4.11)'de elde edilen sistem matrisi (4.10)'daki yeni durum denkleminde yerine konulduğunda PA'nın bir sonraki durum bilgisi aşağıdaki gibi hesaplanır:

$$\begin{aligned} x' &= x + u \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \\ x' &= \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \\ x' &= \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \end{aligned} \quad (4.12)$$

(4.12)'de elde edilen bir sonraki durum bilgisi incelendiğinde p_1 yerinde hiçbir jeton kalmadığı, p_2 yerindeki jetonun harcanmadığı ve p_3 yerine 1 adet jeton konulduğu görülmektedir. Elde edilen bu durum bilgisinin doğru olup olmadığını anlamak için PA'nın bir sonraki durum bilgisini bir de grafiksel olarak hesaplayalım. İlk durumda p_1 ve p_2 yerlerinde en az 1'er adet jeton bulunduğu için t_1 geçişi aktif hale gelip

tetiklenecektir. Bunun sonucunda p_1 yerindeki 1 adet jeton harcanıp p_3 yerine 1 adet jeton konulacaktır. Ancak, p_2 yeri ile t_1 geçişi arasında yetkilendirici yay bulunduğundan p_2 yerindeki jeton harcanmayacaktır. (4.12)'de elde edilen bir sonraki durum vektörüne bakıldığında bu sonuca varıldığı açıkça görülebilmektedir.

Sonuç olarak yetkilendirici yaylar için (4.10)'da önerilen yeni durum denkleminin, yetkilendirici yayların davranışını matematiksel olarak ifade etmekte kullanılabildiği görülmektedir. Dolayısıyla bu durum denklemi kullanılarak yetkilendirici yay içeren PA'nın bilgisayar ortamında işlenmesi sağlanabilir.

4.2.3 Yutucu yay

Yutucu yay, bağlı olduğu yerdeki jetonların tümünü bir kerede harcayıp sıfırlamak için kullanılmaktadır. Bu sıfırlama işlemi söz konusu yerdeki jetonların sayısından bağımsız olarak gerçekleştirilebilmektedir (bkz. "4.1.3 Yutucu yay"). Yutucu yay içeren bir PA'da bir geçişin aktifleşebilmesi için gerekli koşul aşağıdaki gibi yeniden tanımlanmalıdır:

$$x(p_i) \geq 0, \quad \forall p_i \in I(t_j) \quad (4.13)$$

(4.13)'te verilen tanıma göre PA'da sadece yutucu yayların bağlı olduğu her bir $t_j \in T$ geçişinin aktif hale gelebilmesi için her p_i giriş yerindeki jeton sayısının sıfır veya daha fazla olması gerekmektedir. Başka bir deyişle sadece yutucu yayın bağlı olduğu bir geçiş her zaman tetiklenebilir durumdadır. Dikkat edilecek olursa yutucu yaylar için ağırlık değeri tanımlı değildir.

Yutucu yayların bağlı olduğu bir $t_j \in T$ geçişi tetiklendiğinde, PA'nın bir sonraki durum bilgisi $x'(p_i)$ aşağıdaki gibi yeniden tanımlanmalıdır:

$$x'(p_i) = 0, \quad i = 1, \dots, n \quad (4.14)$$

(4.14)'te verilen tanım gereğince, eğer bir p_i yerinden t_j geçişine olan bir yutucu yay mevcutsa bu t_j geçişi tetiklendiği zaman p_i yerindeki tüm jetonlar harcanıp sıfırlanmaktadır. Bu sıfırlama işleminde p_i yerindeki jetonların sayısına herhangi bir bağımlılık bulunmamaktadır. Görüldüğü gibi (4.13) ve (4.14)'te verilen tanımlar ile yutucu yay içeren PA için durum geçiş mekanizması ifade edilebilmektedir.

$$D''(x) = \begin{cases} d''_{ji} = -x(p_i), & i = 1, \dots, n, \quad j = 1, \dots, m, \text{ p}_i \text{ ve t}_j \text{ arasında} \\ & \text{bir yutucu yay varsa,} \\ d''_{ji} = d'_{ji}, & i = 1, \dots, n, \quad j = 1, \dots, m, \text{ diğ.} \end{cases} \quad (4.15)$$
$$x' = x + uD''(x) \quad (4.16)$$

Figure 1 consists of two Petri nets, (a) and (b).
 (a) A simple Petri net with one place p_1 (top circle), one transition t_1 (bottom rectangle), and one place (bottom circle). The transition t_1 is represented by a black dot on a vertical line connecting p_1 to the bottom place.
 (b) A more complex Petri net with places p_1 (top circle), p_2 (middle circle), and two transitions t_1' (top rectangle) and t_1'' (bottom rectangle). There is a feedback loop from p_2 to t_1' . A dashed yellow region encloses the entire net. The transition t_1 is represented by a black dot on a vertical line connecting p_1 to the bottom place.

29

yutucu yayın karşılığı aynı işlevselliği p_1 yerindeki jeton sayısının 1 fazlası kadar adımda gerçekleştirebilir. Bununla birlikte yutucu yay, karşılığına göre daha sade bir gösterim sunmaktadır. Bir sonraki bölümde yutucu yay ile karşılığının aynı işlevselliğe sahip olduğu gösterilecektir. Bu amaçla Şekil 4.6(a) ve Şekil 4.6(b)'de verilen PA'nın durum denklemleri karşılaştırılacaktır.

4.3 Genelleştirilmiş Durum Denklemi

Buraya kadar verilen tanımlarda yasaklayıcı, yetkilendirici ve yutucu yaylar için yeni durum denklemleri ayrı ayrı verilmiştir. Ancak bu yayların hepsi ile aynı anda çalışma yapılması gerektiğinde tek bir durum denklemine gereksinim duyulacaktır. Bu amaçla söz konusu yaylar için (4.4), (4.10) ve (4.16)'daki yeni durum denklemleri birleştirilerek aşağıdaki gibi tek bir durum denklemi elde edilebilir:

$$x' = x + u \tilde{H}(x) D''(x) \quad (4.17)$$

(4.17)'de önerilen durum denklemini “genelleştirilmiş durum denklemi” adı verilmiştir. Genelleştirilmiş durum denklemi yasaklayıcı, yetkilendirici ve yutucu yay türlerinden en az birini içeren bir PA'nın matematiksel olarak ifade edilmesinde kullanılabilir.

“4.2.3 Yutucu yay” bölümünde Şekil 4.6(a) ve Şekil 4.6(b)'de verilen PA'nın birbirine denk olduğunu göstermek için genelleştirilmiş durum denklemini kullanabiliriz. Bu amaçla öncelikle Şekil 4.6(b)'deki yutucu yayın karşılığına ait PA için genelleştirilmiş durum denklemini yazalım:

$$x' = [x(p_1) \ x(p_2)] + u \tilde{H}(x) \begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad (4.18)$$

Başlangıçta p_2 yerinde 1 adet jeton bulunmaktadır. Dolayısıyla “ $x(p_2) = 1$ ” olarak elde edilir. p_1 yerindeki jeton sayısına göre iki farklı durum söz konusu olabilir:

1. Eğer ilk başta p_1 yerinde hiç jeton yoksa sadece t_1 geçişi tetiklenebilir hale gelir. Bu geçişin tetiklenmesi için $u = [0 \ 0 \ 1]$ olarak alınır ve bir sonraki durum bilgisi (4.19)'daki gibi elde edilir:

$$x' = [0 \ 1] + [0 \ 0 \ 1] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$x' = [0 \ 0] \quad (4.19)$$

2. Eğer ilk başta p_1 yerindeki jeton sayısı 1 ya da daha fazla ise p_1 yerinden t_1 geçişine çizilmiş olan yasaklayıcı yay t_1 geçişinin tetiklenmesine izin vermeyecektir. Benzer şekilde p_2 yerinde 1 tane jeton bulunduğundan p_2 yerinden t_1' geçişine çizilmiş olan yasaklayıcı yay nedeniyle t_1' geçişi de aktif hale gelip tetiklenemeyecektir. Bu durumda sadece t_1'' geçişi aktif hale gelip tetiklenebilecektir, çünkü p_1 ve p_2 yerlerinde en az 1'er adet jeton bulunmaktadır. t_1'' geçişi tetiklendikten sonraki durum bilgisi aşağıdaki gibi elde edilir:

$$x' = [x(p_1) \ 1] + [0 \ 1 \ 0] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$x' = [x(p_1) - 1 \ 1] \quad (4.20)$$

(4.20)'den de anlaşılabileceği gibi 2 no'lu koşul geçerli olduğu sürece p_1 yerinden her seferinde 1 adet jeton harcanmaktadır. p_1 yerindeki tüm jetonlar harcanıp bitinceye kadar 2 no'lu koşul geçerli olacak ve her seferinde p_1 yerinden 1 adet jeton harcanacaktır. Ancak p_1 yerinde artık hiç bir jeton kalmadığında 1 no'lu koşul geçerli olacak ve p_2 yerindeki jeton da harcanacaktır. Sonuç olarak tüm jetonlar harcanmış olacaktır. Böylelikle yutucu yayın karşılığının sıfırlama işlemini gerçekleştirdiğini göstermiş olduk. Şimdi de Şekil 4.6(a)'da verilen yutucu yay için genelleştirilmiş durum denklemini yazalım:

$$x' = [x(p_1)] + [1][1][-x(p_1)]$$

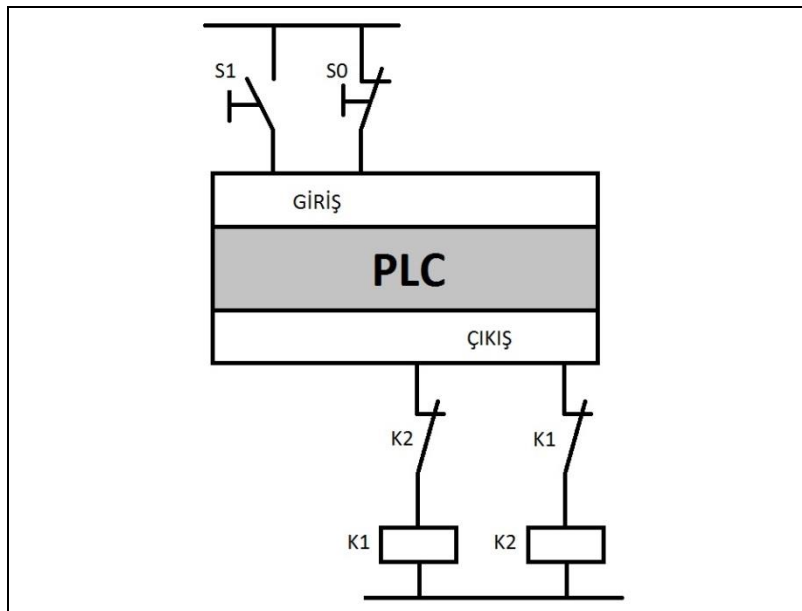
$$x' = 0 \quad (4.21)$$

(4.21)'de elde edilen son durum bilgisine bakıldığında t_1 geçişi tetiklendiğinde p_1 yerindeki tüm jetonların tek seferde harcandığı görülmektedir. Elde edilen bu sonuç yutucu yayın işlevselliğini doğrulamaktadır. Böylelikle önerdiğimiz “genelleştirilmiş durum denklemi”ni kullanarak Şekil 4.6(a)'da verilen yutucu yay ile Şekil 4.6(b)'de verilen yutucu yayın karşılığının birbirine denk olduğunu göstermiş olduk.

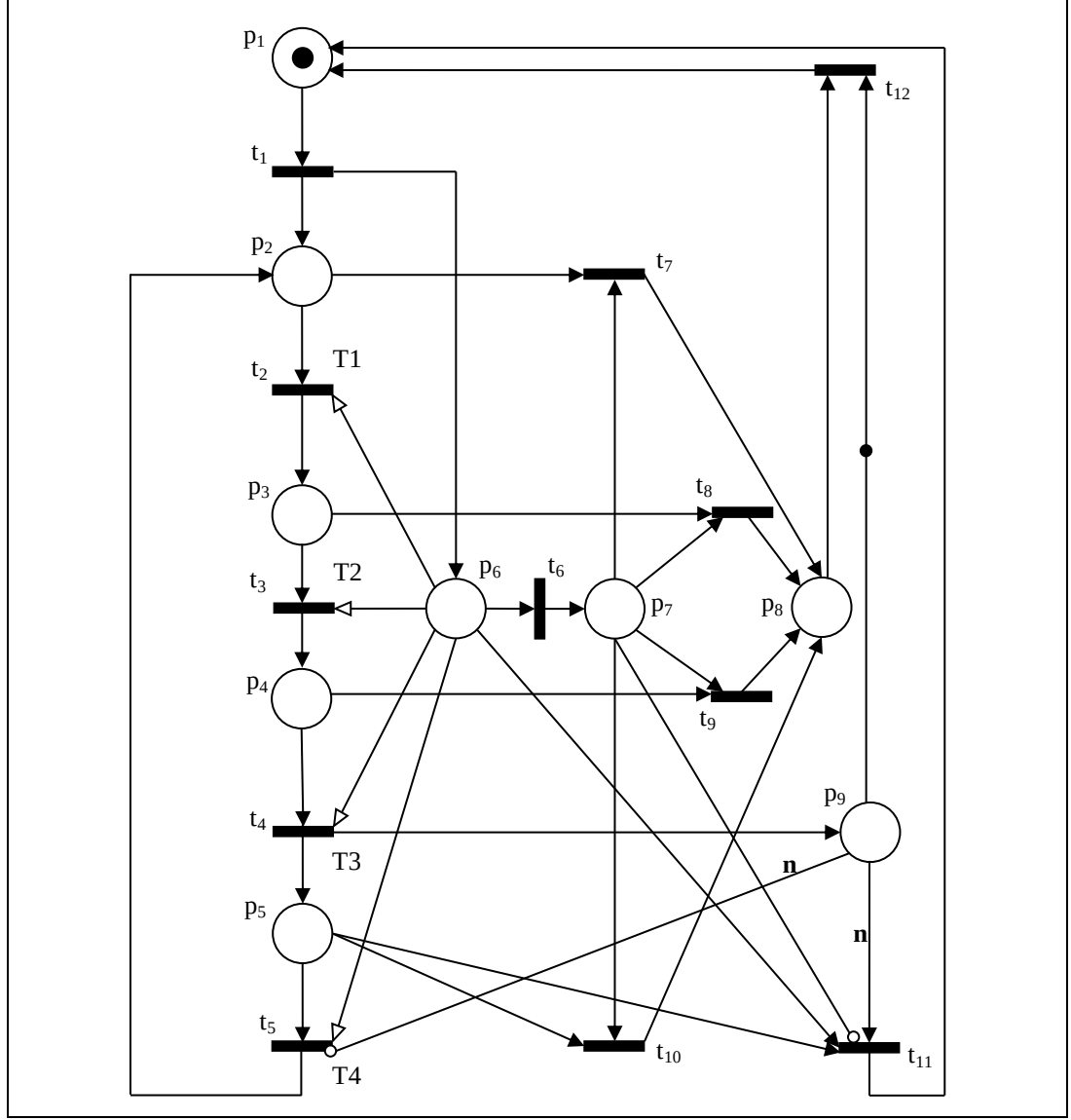
4.3.1 Örnek bir uygulama

Önerilen genelleştirilmiş durum denkleminin yasaklayıcı, yetkilendirici ve yutucu yayların hepsini aynı anda içeren bir PA'yı matematiksel olarak ifade etmekte kullanılabildiğini göstermek amacıyla örnek bir endüstriyel otomasyon sistemi ele alınmıştır. Bu sistemin PA modeli yasaklayıcı, yetkilendirici ve yutucu yayların hepsini içermektedir. Dolayısıyla bu sistemin durumlarının elde edilmesi için genelleştirilmiş durum denklemi kullanılabilir. Sistem durumları öncelikle grafiksel yöntem ve daha sonra genelleştirilmiş durum denklemi kullanılarak hesaplanmıştır. Ardından her iki yöntemle elde edilen sonuçlar karşılaştırılmıştır.

Örnek endüstriyel otomasyon sistemi olarak bir karıştırıcı sistem seçilmiştir. Bu sistemin iki adet girişi ve iki adet çıkışı bulunmaktadır (Şekil 4.7). Girişlerden biri sistemi başlatmak, diğeri ise sistemi durdurmak için kullanılmaktadır. Karıştırma mekanizmasının gerçekleşmesi için hem düz hem de ters yönlerde dönebilen bir motor bulunmaktadır. Sistemde dört adet zamanlayıcı mevcuttur. Bu zamanlayıcılar ile motorun dönüşleri ve dönüşler arasındaki durma süresi ayarlanmaktadır. Motor hem düz hem de ters yönlerde birer kez döndükten sonra bir turunu tamamlamış sayılmaktadır. Motorun tamamladığı tur sayısının saklanması için bir sayaç kullanılmıştır. Motor önceden belirlenmiş sayıdaki kadar turu tamamladığında ilgili sayaç sıfırlanmakta ve sistem otomatik olarak durdurulmaktadır. Bu sayacın sıfırlanması için sistemin PA modelinde yutucu yay kullanılmıştır (Şekil 4.8).



Şekil 4.7 : Sistemin giriş ve çıkışları.



Şekil 4.8 : Sistemin PA modeli.

Şekil 4.8’de verilen PA’da p_9 yeri ile t_{12} geçişi arasında bir yutucu yay bulunmaktadır. Bununla birlikte motorun sistem durmadan önce tamamlaması istenen toplam tur sayısı $w(p_9, t_5) = w(p_9, t_{11}) = n$ ile ifade edilmektedir (Çizelge 4.1).

Çizelge 4.1 : Sistemin PA modelindeki yerlerin tanımı.

Yer	Tanım
p_1	Başlangıç
p_2	Motor düz yönde dönüyor
p_3	Motor hareketsiz
p_4	Motor ters yönde dönüyor
p_5	Motor hareketsiz
p_6	Sistem devrede
p_7	Sistem durdurulacak
p_8	Sistem durduruluyor
p_9	Motorun tamamladığı tur sayısı

Şekil 4.8’de verilen PA’nın aşağıdaki başlangıç durumu için bir sonraki durumunu hem grafiksel olarak hem de genelleştirilmiş durum denklemini kullanarak matematiksel olarak elde edelim:

$$x = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2] \quad (4.22)$$

1. Grafiksel Yöntem:

p_3 ve p_7 yerlerinde birer adet jeton bulunduğundan t_8 geçişi aktif hale gelip tetiklenecektir. Bu geçiş dışında tetiklenebilecek başka bir geçiş bulunmamaktadır. t_8 geçişi tetiklendiğinde p_3 ve p_7 yerleri birer jeton kaybedecek ve p_8 yeri bir jeton kazanacaktır. Böylelikle sistemin bir sonraki durumu aşağıdaki gibi elde edilir:

$$x' = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 2] \quad (4.23)$$

2. Matematiksel Yöntem:

Bir sonraki durumun matematiksel olarak elde edilmesi için genelleştirilmiş durum denklemi kullanılacaktır. Bunun için öncelikle (4.2)’de verilen tanıma uygun olarak yasaklayıcı matrisi elde edelim. Sadece t_{11} geçişi bir yasaklayıcı yay tarafından bloklanmış olup diğer tüm geçişler bloklanmadığı için yasaklayıcı matriste t_{11} geçişine karşılık gelen diyagonal eleman “0” değerini alırken diğer bütün diyagonal elemanlar “1” olacaktır:

$$\tilde{H}(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.24)$$

Yeni sistem matrisi, (4.3) ve (4.15)'teki tanımlara göre aşağıdaki gibi hesaplanır:

$$D''(x) = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -n \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -2 \end{bmatrix} \quad (4.25)$$

(4.24)'te elde edilen yasaklayıcı matris ile (4.25)'te elde edilen sistem matrisi genelleştirilmiş durum denkleminde yerlerine konulduğunda sistemin bir sonraki durumu aşağıdaki gibi elde edilir:

$$\begin{aligned} x' &= x + u\tilde{H}(x)D''(x) \\ x' &= [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2] + [0 \ 0 \ -1 \ 0 \ 0 \ 0 \ -1 \ 1 \ 0] \\ x' &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 2] \end{aligned} \quad (4.26)$$

(4.23) ve (4.26)'da sırasıyla grafiksel ve matematiksel olarak elde edilen bir sonraki durum vektörleri karşılaştırıldığında bunların birbirinin aynısı olduğu görülmektedir. Bu da genelleştirilmiş durum denkleminin yasaklayıcı, yetkilendirici ve yutucu yaylar içeren bir PA'nın matematiksel olarak ifade edilmesinde kullanılabildiğini doğrulamaktadır.

Genelleştirilmiş durum denkleminin elle hesaplamaya gerek duyulmadan bilgisayar ortamında hızlı bir şekilde elde edilmesi amacıyla bu tez kapsamında bir yazılım da geliştirilmiştir. Bu yazılım kullanılarak genelleştirilmiş durum denklemi kolaylıkla hesaplanabilir. Üzerinde çalıştığımız örnek sisteme ait PA'nın bir sonraki durumu söz konusu yazılım kullanılarak Şekil 4.9'daki gibi elde edilmiştir. Bu yazılım bir sonraki durum bilgisi ile birlikte hesaplamış olduğu yasaklayıcı matrisi ve yeni sistem matrisini de kullanıcıya sağlamaktadır. Yazılımın çıktılarına bakıldığında elle yapılan hesaplamalar ile aynı sonuçların elde edildiği görülmektedir ($n = 3$).

Petri Ağları için Genelleştirilmiş Durum Denklemi

Petri Ağı dosyasını seçin:
 Gözet...

İlk durum:
 Hesapla

Aktifleşmiş geçişler:

Son durum:
 İlk duruma kopyala

Geçiş tetikle

Bozucu matris:

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
t1	1	0	0	0	0	0	0	0	0	0	0	0
t2	0	1	0	0	0	0	0	0	0	0	0	0
t3	0	0	1	0	0	0	0	0	0	0	0	0
t4	0	0	0	1	0	0	0	0	0	0	0	0
t5	0	0	0	0	1	0	0	0	0	0	0	0
t6	0	0	0	0	0	1	0	0	0	0	0	0
t7	0	0	0	0	0	0	1	0	0	0	0	0
t8	0	0	0	0	0	0	0	1	0	0	0	0
t9	0	0	0	0	0	0	0	0	1	0	0	0
t10	0	0	0	0	0	0	0	0	0	1	0	0
t11	0	0	0	0	0	0	0	0	0	0	0	0
t12	0	0	0	0	0	0	0	0	0	0	0	1

Yeni sistem matrisi:

	p1	p2	p3	p4	p5	p6	p7	p8	p9
t1	-1	1	0	0	0	1	0	0	0
t2	0	-1	1	0	0	0	0	0	0
t3	0	0	-1	1	0	0	0	0	0
t4	0	0	0	-1	1	0	0	0	1
t5	0	1	0	0	-1	0	0	0	0
t6	0	0	0	0	0	-1	1	0	0
t7	0	-1	0	0	0	0	-1	1	0
t8	0	0	-1	0	0	0	-1	1	0
t9	0	0	0	-1	0	0	-1	1	0
t10	0	0	0	0	-1	0	-1	1	0
t11	1	0	0	0	-1	-1	0	0	-3
t12	1	0	0	0	0	0	0	-1	-2

Şekil 4.9 : Sisteme ait PA'nın yazılım tarafından hesaplanmış son durum bilgisi (1).

Şimdi de (4.26)'da elde ettiğimiz son durum bilgisini PA'nın yeni başlangıç durumu olarak ele alalım ve bu yeni başlangıç durumu için bir sonraki durum bilgisini aynı yöntemlerle elde edelim:

$$x' = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 2] \quad (4.27)$$

1. Grafiksel Yöntem:

p_8 yerinde bir adet ve p_9 yerinde iki adet jeton bulunduğuundan t_{12} geçişi aktif hale gelip tetiklenecektir. Bu geçiş tetiklendiğinde p_8 yeri bir jeton kaybedecek ve p_9 yeri ile t_{12} geçişi arasındaki yutucu yay nedeniyle p_9 yerindeki tüm jetonlar da harcanacaktır. p_1 yeri ise bir jeton kazanacaktır. Böylelikle sistemin bir sonraki durumu aşağıdaki gibi elde edilir:

$$x'' = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (4.28)$$

2. Matematiksel Yöntem:

Bir sonraki durumun matematiksel olarak elde edilmesi için genelleştirilmiş durum denklemi kullanılacaktır. Bunun için öncelikle (4.2)'de verilen tanıma uygun olarak yasaklayıcı matrisi elde edelim. Yasaklayıcı yaylar tarafından hiçbir geçiş bloklanmadığı için yasaklayıcı matrisin tüm diyagonal elemanları “1” değerini alacaktır. Bu durumda yasaklayıcı matris 12x12’lik bir birim matris halini alacaktır:

$$\tilde{H}(x') = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.29)$$

Yeni sistem matrisi, (4.3) ve (4.15)’teki tanımlara göre aşağıdaki gibi hesaplanır:

$$D''(x') = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -n \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -2 \end{bmatrix} \quad (4.30)$$

(4.29)'da elde edilen yasaklayıcı matris ile (4.30)'da elde edilen sistem matrisi genişletilmiş durum denkleminde yerlerine konulduğunda sistemin bir sonraki durumu aşağıdaki gibi elde edilir:

$$x'' = x' + u\tilde{H}(x')D''(x')$$

$$x'' = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 2] + [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ -2]$$

$$x'' = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (4.31)$$

(4.28) ve (4.31)'de sırasıyla grafiksel ve matematiksel olarak elde edilen bir sonraki durum vektörleri karşılaştırıldığında bunların birbirinin aynısı olduğu görülmektedir.

Geliştirdiğimiz yazılım kullanılarak bir sonraki durum bilgisi Şekil 4.10'daki gibi elde edilmiştir. Yazılımın çıktılarına bakıldığında elle yapılan hesaplamalar ile aynı sonuçların elde edildiği görülmektedir ($n = 3$).

Petri Ağları için Genelleştirilmiş Durum Denklemi

Petri Ağı dosyasını seçin:
C:\karistirici_sistem.xml Gözet...

İlk durum:
0 0 0 0 0 0 1 2 Hesapla

Aktifleşmiş geçişler:
Geçiş tetikle

Son durum:
1 0 0 0 0 0 0 0 İlk duruma kopyala

Bozucu matris:

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
t1	1	0	0	0	0	0	0	0	0	0	0	0
t2	0	1	0	0	0	0	0	0	0	0	0	0
t3	0	0	1	0	0	0	0	0	0	0	0	0
t4	0	0	0	1	0	0	0	0	0	0	0	0
t5	0	0	0	0	1	0	0	0	0	0	0	0
t6	0	0	0	0	0	1	0	0	0	0	0	0
t7	0	0	0	0	0	0	1	0	0	0	0	0
t8	0	0	0	0	0	0	0	1	0	0	0	0
t9	0	0	0	0	0	0	0	0	1	0	0	0
t10	0	0	0	0	0	0	0	0	0	1	0	0
t11	0	0	0	0	0	0	0	0	0	0	1	0
t12	0	0	0	0	0	0	0	0	0	0	0	1

Yeni sistem matrisi:

	p1	p2	p3	p4	p5	p6	p7	p8	p9
t1	-1	1	0	0	0	1	0	0	0
t2	0	-1	1	0	0	0	0	0	0
t3	0	0	-1	1	0	0	0	0	0
t4	0	0	0	-1	1	0	0	0	1
t5	0	1	0	0	-1	0	0	0	0
t6	0	0	0	0	0	-1	1	0	0
t7	0	-1	0	0	0	0	-1	1	0
t8	0	0	-1	0	0	0	-1	1	0
t9	0	0	0	-1	0	0	-1	1	0
t10	0	0	0	0	-1	0	-1	1	0
t11	1	0	0	0	-1	-1	0	0	-3
t12	1	0	0	0	0	0	0	-1	-2

Şekil 4.10 : Sisteme ait PA'nın yazılım tarafından hesaplanmış son durum bilgisi (2).

4.4 Otomasyon Petri Ağları

Endüstriyel otomasyon sistemleri zamanla daha karmaşık hale geldikçe bu sistemlerin modellenmesi için klasik PA yetersiz kalmaya başlamıştır. Klasik PA'nın modelleme yeteneklerinin artırılması ve karmaşık sistemlerin daha etkin ve basit bir şekilde ifade edilebilmesi için "Otomasyon Petri Ağları" kavramı ortaya atılmıştır. OPA, klasik PA'nın temel özelliklerini baz almakta ve klasik PA'ya bir takım yeni özellikler getirmektedir. OPA, aşağıdaki gibi tanımlanır [18]:

$$OPA = (P, T, Pre, Post, In, En, \chi, Q, M_0) \quad (4.32)$$

(4.32)'deki sembollere ait tanımlar aşağıda verilmiştir:

- $P = \{p_1, p_2, \dots, p_n\}$ boş olmamakla birlikte, sonlu yerler kümesidir.
- $T = \{t_1, t_2, \dots, t_n\}$ boş olmamakla birlikte, sonlu geçişler kümesidir. P ve T kümesi $P \cup T \neq \emptyset$ ve $P \cap T = \emptyset$ koşullarını sağlar.
- $Pre: (P \times T) \rightarrow N$ yerlerden geçişlere olan standart yönlü yayları tanımlayan giriş fonksiyonudur. N, sıfırdan büyük tamsayılardan oluşan bir kümeyi ifade etmektedir.
- $Post: (T \times P) \rightarrow N$ geçişlerden yerlere olan standart yönlü yayları tanımlayan çıkış fonksiyonudur. N, sıfırdan büyük tamsayılardan oluşan bir kümeyi ifade etmektedir.
- $In: (P \times T) \rightarrow N$ yerlerden geçişlere olan yasaklayıcı yayları tanımlayan bir yasaklayıcı giriş fonksiyonudur.
- $En: (P \times T) \rightarrow N$ yerlerden geçişlere olan yetkilendirici yayları tanımlayan bir yetkilendirici giriş fonksiyonudur.
- $\chi = \{\chi_1, \chi_2, \dots, \chi_m\}$ boş küme olmamakla birlikte, geçişlerin tetiklenebilmesi için gerekli koşulları ifade etmektedir.
- $Q = \{q_1, q_2, \dots, q_n\}$ yerlere atanmış sonlu sayıdaki faaliyetleri ifade etmektedir.
- $M_0: P \rightarrow N$ başlangıçtaki işaretlemeyi ifade etmektedir.

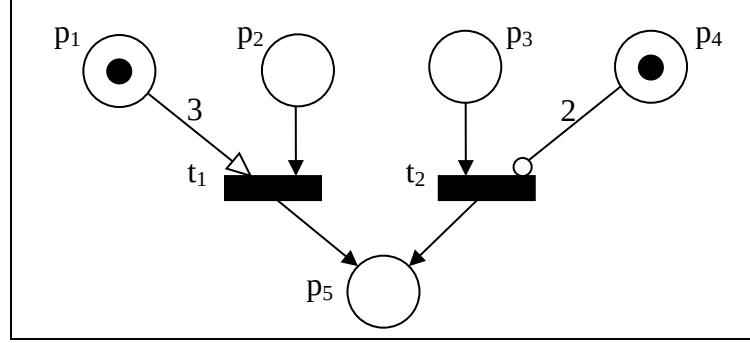
Bir OPA, PA'ya benzer şekilde yerlerden ve geçişlerden oluşmaktadır. OPA'da, PA'daki standart yaya ($\longrightarrow$) ek olarak yasaklayıcı yay ve yetkilendirici yay kullanılmaktadır (bkz. "4.1 Klasik Petri Ağlarına Eklentiler"). Yasaklayıcı yaylar, ucu boş daire ile biten yay ile ($\longrightarrow\bigcirc$) gösterilir. Yetkilendirici yaylar ise ucu boş okla biten yönlü yay ile ($\longrightarrow\triangleright$) gösterilir. Standart yaylar, PA'da olduğu gibi yerleri geçişlere ve geçişleri yerlere bağlamaktadır. Ancak yasaklayıcı yaylar ve yetkilendirici yaylar sadece yerleri geçişlere bağlamaktadır. Yerler ile koşullar ve geçişler ile olaylar ifade edilmektedir. Yerlerin işaretlenmesi için PA'da olduğu gibi jetonlar kullanılmakta ve bu jetonların yerler arasındaki transferi OPA'nın durum geçiş mekanizmasını oluşturmaktadır. Jetonların transferi aktif hale gelip tetiklenen geçişler ile sağlanmaktadır. OPA'da bir geçişin aktif hale gelebilmesi için aşağıdaki koşulların sağlanmış olması gerekmektedir [17]:

1. Eğer bir "t" geçişine ait "p" giriş yeri bu geçişe standart yay ile bağlıysa, "p" giriş yeri en az, kendisini "t" geçişine bağlayan bu standart yayın ağırlık değeri kadar jeton içeriyorsa "t" geçişi aktif hale gelmiş demektir.
2. Eğer bir "t" geçişine ait "p" giriş yeri bu geçişe yetkilendirici yay ile bağlıysa, "p" giriş yeri en az, kendisini "t" geçişine bağlayan bu yetkilendirici yayın ağırlık değeri kadar jeton içeriyorsa "t" geçişi aktif hale gelmiş demektir.
3. Eğer bir "t" geçişine ait "p" giriş yeri bu geçişe yasaklayıcı yay ile bağlıysa, "p" giriş yeri, kendisini "t" geçişine bağlayan bu yasaklayıcı yayın ağırlık değerinden daha az jeton içeriyorsa "t" geçişi aktif hale gelmiş demektir.

OPA'da aktif hale gelen bir "t" geçişi koşullara bağlı olarak tetiklenebilir ya da tetiklenmeyebilir. Bu koşullar, örneğin, sistemdeki sensörlerden okunan geri besleme bilgileri olabilir. Geri besleme bilgileri "AND", "OR" ve "NOT" gibi mantıksal operatörlerden oluşabilir. Bir "t" geçişinin tetiklenip tetiklenmeyeceğine bu mantıksal operatörlerden oluşan geri besleme bilgileri yorumlanarak karar verilmelidir [17].

Aktif hale gelmiş bir "t" geçişi tetiklendiğinde, "t" geçişine bağlı olan her "p" giriş yerinden bu giriş yerlerini "t" geçişine bağlayan standart yayların ağırlık değeri kadar jeton alınır. Ancak "t" geçişine yasaklayıcı yaylar veya yetkilendirici yaylar ile bağlı olan "p" giriş yerlerindeki jetonlar harcanmaz [17]. Tetiklenme sonrasında "t"

geçişine bağlı olan her “p” çıkış yerine “t” geçişini bu çıkış yerlerine bağlayan standart yayların ağırlık değeri kadar jeton konulur. Şekil 4.11’de örnek bir OPA verilmiştir:



Şekil 4.11 : Örnek bir OPA.

5 PETRİ AĞLARINDAN PLC KODUNA DÖNÜŞÜM

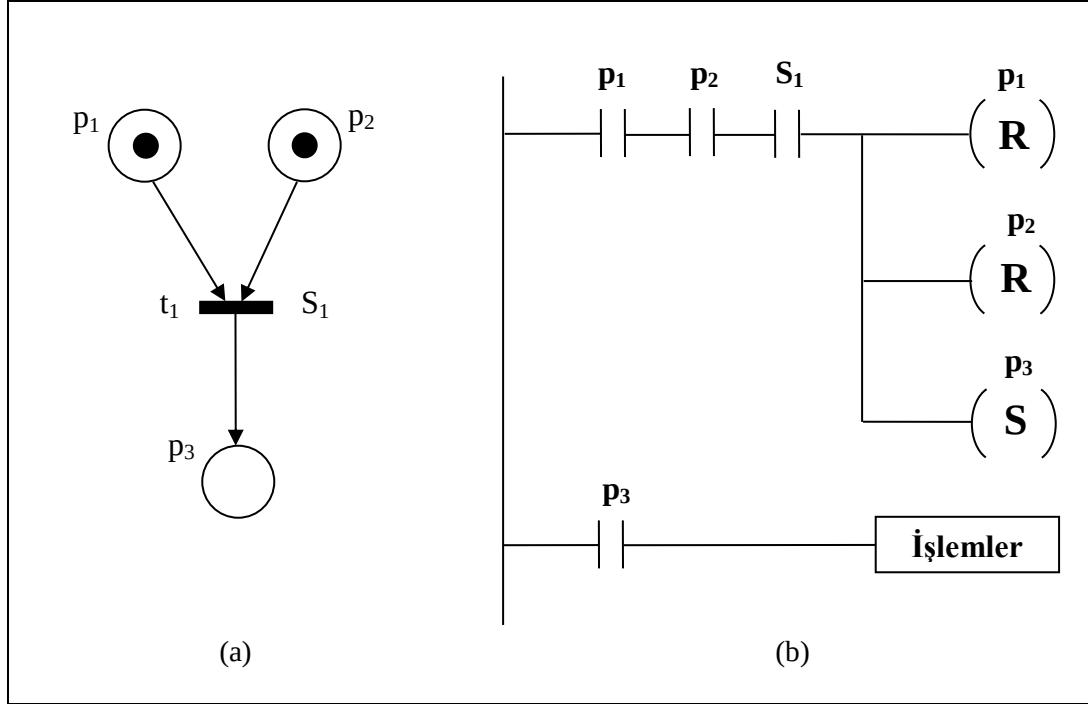
Endüstriyel otomasyon sistemlerinin gerçekleştirilmesinde çoğunlukla PLC kullanılmaktadır. Bir sistemin PLC üzerinde gerçekleştirilmesi için kodlamanın uygun bir şekilde yapılması gerekmektedir. Bu kodlamanın yapılabilmesi için sezgisel ya da formal yöntemler kullanılabilir. Sezgisel yöntemler herhangi bir sistematiğe dayalı olmadığı için çok tercih edilmemektedir. Bu nedenle formal yöntemler daha sık kullanılmaktadır. Formal yöntemlerde öncelikle kodlanacak sistemin temsili bir modeli oluşturulur ve daha sonra belirli kurallar çerçevesinde bu model PLC koduna dönüştürülür. Örneğin formal yöntemler kullanılarak bir sistemin PA modeli oluşturulup PLC koduna dönüştürülebilir. Yapılan literatür taramasında bir sistemin PA modelinden yola çıkılarak PLC kodunun oluşturulmasında uygulanan formal yöntemler incelenmiş ve bu yöntemlerin genellikle “kurma” (*set*) ve “silme” (*reset*) komutları ile gerçekleştirildiği görülmüştür [19-22]. Her ne kadar pratik de olsa “kurma” ve “silme” komutları ile yapılan gerçeklemin birtakım eksiklikleri bulunmaktadır. Bu bölümde bir PA modelinin PLC koduna “kurma” ve “silme” komutları kullanılarak nasıl dönüştürüldüğü gösterilmiş, uygulanan bu yöntemin eksikliklerinden bahsedilmiş ve bu eksikliklerin giderilmesi amacıyla PA’nın matematiksel özellikleri kullanılarak PLC koduna dönüştürülmesi için bir yöntem önerilmiştir.

5.1 PLC Koduna Dönüşüm için Mevcut Yöntemler

5.1.1 Kurma ve silme komutları ile kodlama yöntemi

Literatürde bir sistemin PA modelinin PLC koduna dönüştürülmesi için yöntem olarak genellikle “kurma” ve “silme” komutlarının kullanıldığı görülmüştür. Bu yöntemde, PA’daki jeton içermeyen bir yere tek bir jeton konulması için “kurma”, PA’da tek bir jeton içeren bir yerdeki jetonun harcanması için ise “silme” komutu kullanılır. Görüldüğü gibi bu yöntem sadece 1-sınırlı PA’nın (bkz. “2.4 Sınırlılık Kavramı”) PLC koduna dönüştürülmesinde uygulanabilmektedir. Yöntemin daha iyi

anlaşılabilmesi bakımından Şekil 5.1(a)'da örnek bir PA ele alınmış ve Şekil 5.1(b)'de bu PA'nın PLC kodu karşılığı merdiven diyagramı olarak verilmiştir:



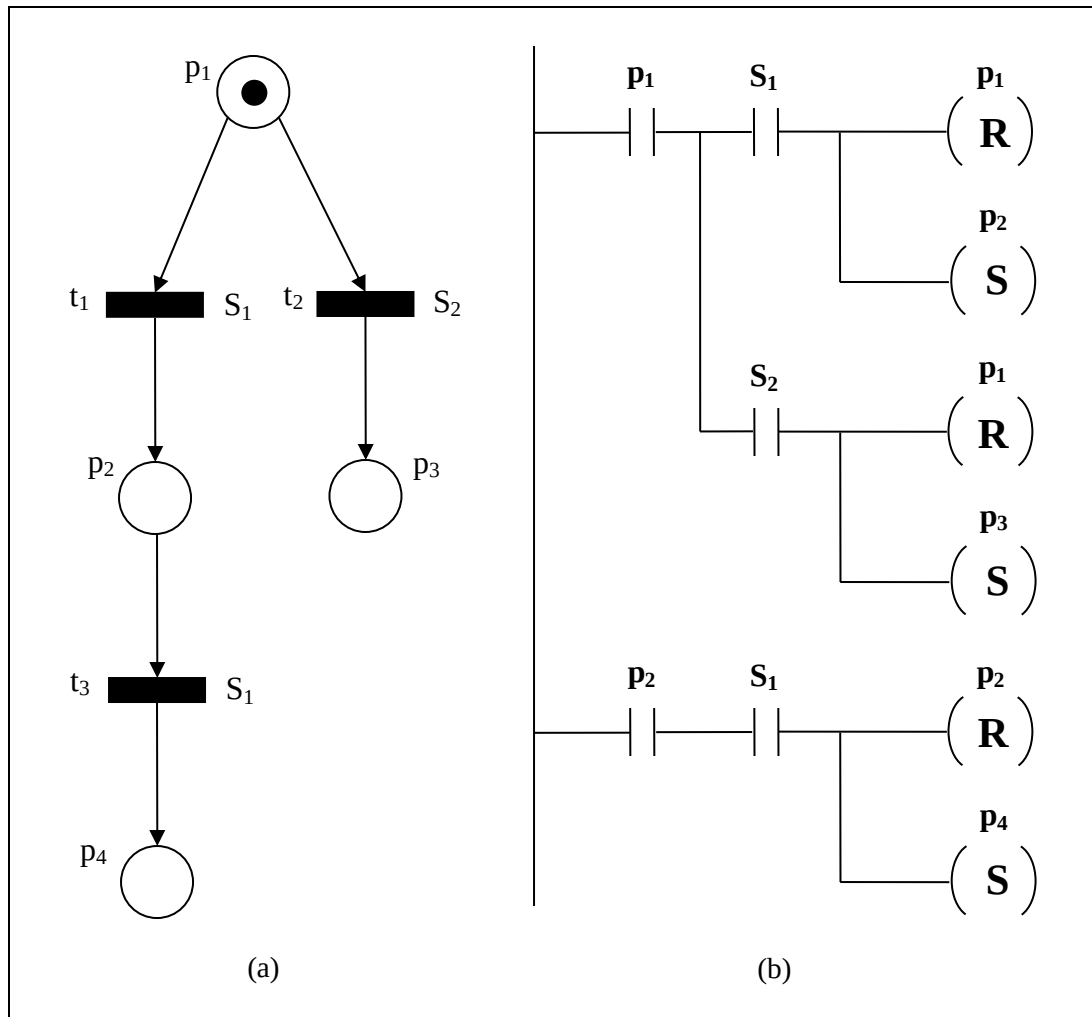
Şekil 5.1 : (a) Örnek bir PA. (b) PA'nın merdiven diyagramı karşılığı.

Şekil 5.1(a)'da verilen PA'da üç tane yer (p_1 , p_2 , p_3) ve bir tane geçiş (t_1) bulunmaktadır. p_1 ve p_2 yerleri t_1 geçişinin giriş yerleri, p_3 yeri ise t_1 geçişinin çıkış yeridir. Dikkat edilecek olursa bu PA'da her bir yerdeki jeton sayısının herhangi bir anda en fazla 1 olduğu görülmektedir. Dolayısıyla bu PA, 1-sınırlı PA olup söz konusu yöntem için uygundur. Yöntemin uygulanabilmesi için PA'nın durum geçiş mekanizması incelenmelidir. t_1 geçişinin tetiklenmesi için p_1 ve p_2 yerlerinde jeton bulunması ve S_1 koşulunun sağlanmış olması gerekmektedir. Bu gereksinimler sağlanıp t_1 geçişi tetiklendiği zaman ise p_1 ve p_2 yerlerindeki jetonlar harcanır ve p_3 yerine jeton konulur. Bu bilgiler ışığında p_1 ve p_2 yerlerindeki jetonların harcanması için “silme”, p_3 yerine jeton konulması için ise “kurma” komutu kullanılarak PA'nın PLC kodu karşılığı elde edilmiş olur. Şekil 5.1(b)'de bu PA'ya karşılık gelen PLC kodu merdiven diyagramı olarak verilmiştir [17].

Yöntemi kısaca özetlemek gerekirse PA'daki her bir geçişe ait giriş yerlerindeki jetonların harcanması için “silme”, çıkış yerlerine jetonların konulması için ise “kurma” komutu kullanılmaktadır. Bu komutlar ortak bir koşula bağlanmakta ve bu koşul söz konusu geçişin tetiklenme koşuluna karşılık gelmektedir. Koşulun PLC koduna dönüştürülmesi için ise mantıksal ifadeler kullanılmaktadır.

5.1.1.1 Çığ etkisinin giderilmesi

Çığ etkisi (*avalanche effect*), bir sistemin giriş değerlerinin sabit tutulmasına rağmen sistemde art arda 1’den fazla durum değişikliğinin meydana gelmesi olarak tanımlanmaktadır. Bu nedenle çığ etkisinin mevcut olduğu bir sistem kendisinden beklenen davranışı gösteremez. Bu anlamda çığ etkisi bir sistemin işleyişini olumsuz bir biçimde etkileyen ve sistemin hatalı bir şekilde çalışmasına neden olan bir etken olarak karşımıza çıkmaktadır. Dolayısıyla bir sistemin gerçekleşmesi için oluşturulan PLC kodu ne kadar hatasız olursa olsun eğer bu PLC kodunda çığ etkisi bulunuyorsa sistem istenildiği gibi çalışmayacaktır [23]. Sistemin düzgün çalışması için çığ etkisinin giderilmesi gerekmektedir. Bu anlatılanlar “kurma” ve “silme” komutları kullanılarak oluşturulmuş bir PLC kodu için de geçerlidir. Şekil 5.2’de çığ etkisine bir örnek verilmiştir:

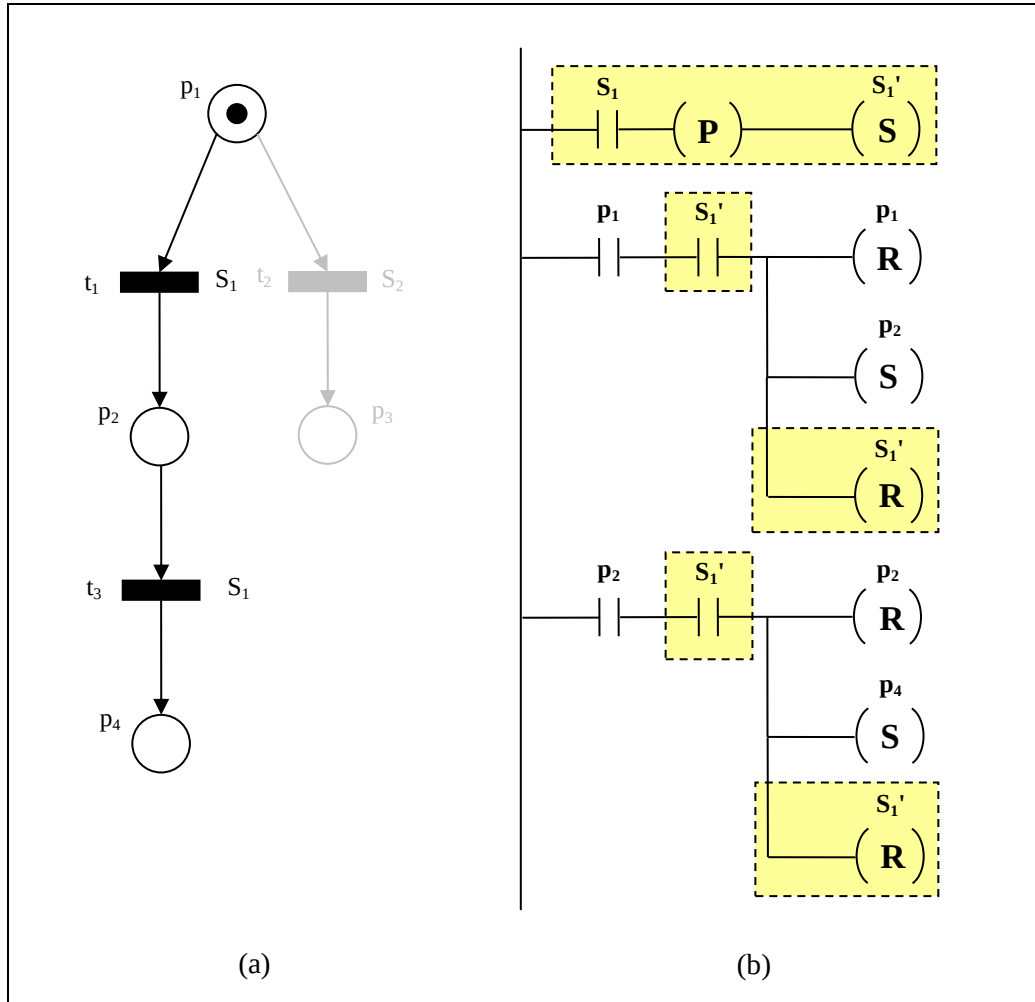


Şekil 5.2 : (a) Örnek bir PA. (b) PA'nın PLC kodu karşılığı (Çığ etkisi var).

Şekil 5.2(a)'da verilen PA'ya dikkatli bakıldığında t_1 ile t_3 geçişlerinin her ikisinin de aynı “S₁” girişine bağlı olduğu görülmektedir. Bu PA'da başlangıçta sadece p_1 yerinde jeton bulunmaktadır. Bu durumda “S₁” girişi bir kez aktif hale getirildiğinde t_1 geçişi aktif hale gelerek tetiklenir ve p_1 yerindeki jeton harcanarak p_2 yerine jeton konulur. “S₁” girişi tekrar aktif hale getirilinceye kadar t_3 geçişi tetiklenmeyeceği için p_2 yerindeki jeton harcanmaz. Böylelikle sistem p_2 yerine ilişkin durumunu korumuş olur. “S₁” girişi tekrar aktif hale getirildiğinde t_3 geçişi tetiklenir ve p_2 yerindeki jeton harcanarak p_4 yerine jeton konulur. PA'daki durum geçişleri bu şekilde gerçekleştiği zaman sistem kendisinden beklenen davranışı göstermiş olur.

PA'nın Şekil 5.2(b)'de verilen PLC kodu karşılığına bakıldığında “S₁” girişi ilk kez aktif hale getirildiğinde sistemden beklendiği gibi p_1 yerindeki jetonun harcanarak p_2 yerine jeton konulduğu görülmektedir. Ancak PLC kodunun son satırında, p_2 yerinde jeton bulunduğu ve “S₁” girişi hâlâ aktif olduğu için p_2 yerindeki jeton da harcanarak p_4 yerine jeton konulmaktadır. Görüldüğü gibi aynı “S₁” girişi altında sistem önce p_2 yerine geçip durum değiştirmekte ve hemen ardından p_4 yerine geçip bu durumdan çıkarak başka bir duruma geçiş yapmaktadır. Buradan söz konusu PLC kodunda çıkış etkisinin bulunduğu sonucuna varılır. Sistemin beklenen davranışı göstermesi için PLC kodundaki mevcut çıkış etkisinin giderilmesi gerekmektedir.

Şekil 5.3(b)'de PLC kodunun çıkış etkisi giderilmiş hâli verilmiştir. Bu PLC kodunda, bir önceki PLC kodunda yapılan değişiklikler kutucuk içinde belirtilmiştir. Dikkat edilecek olursa öncelikle “S₁” giriş değerinden yeni bir “S₁” giriş değeri türetilmektedir. Bunun için “S₁” girişinin çıkan kenarında “S₁” girişine “kurma” komutu ile mantıksal “1” değeri atanmaktadır. Dolayısıyla “S₁” girişi aktif hale getirildiğinde “S₁” girişi de aktif hale gelecektir. Buradan itibaren kodda durum değişikliklerinin gerçekleştirilmesi için artık yeni “S₁” giriş değeri kullanılmaktadır. “S₁” girişi aktif haldeyken p_1 yerinde jeton bulunması durumunda sistem p_2 yerine geçiş yapar. Burada bir önceki PLC kodundan farklı olarak durum geçişi yapıldıktan sonra “S₁” giriş değeri “silme” komutuyla sıfırlanmaktadır. Kodun son satırında ise “S₁” girişi bir önceki satırda sıfırlanmış olduğundan gerekli koşullar sağlanmadığı için sistem p_4 yerine geçiş yapamaz. Böylelikle “S₁” girişi bir kez aktif hale getirildiğinde sistemde 1'den fazla durum değişikliğinin meydana gelmesi engellenmiş olur. Sonuç olarak PLC kodundaki çıkış etkisi giderilmiş ve sistemin istenen davranışı göstermesi sağlanmıştır.



Şekil 5.3 : (a) Örnek bir PA. (b) PA'nın PLC kodu karşılığı (Çığ etkisi giderilmiş).

5.1.2 Mantıksal ifadeler ile kodlama yöntemi

“Kurma” ve “silme” komutlarının kullanıldığı yöntemde PA’dan PLC koduna dönüşüm her ne kadar basit ve etkin bir şekilde yapılsa da aslında bu yöntemin birtakım eksiklikleri bulunmaktadır. Bu eksiklikler aşağıdaki gibi sıralanabilir:

- Her “kurma” ve “silme” komutu normal bit işlemlerine kıyasla PLC’nin belleğinde daha fazla yer kaplamakta ve PLC’nin işlemcisini daha fazla kullanmaktadır. Bu nedenle özellikle büyük ölçekli sistemler için elde edilen kodun PLC üzerinde çalıştırılmasında bellek ve işlemcinin yoğun kullanımına bağlı olarak performans sorunları yaşanabilir.
- Elde edilen PLC kodunun okunabilirliği düşüktür. Bu nedenle kodun anlaşılması ve takip edilmesi güçtür. Bu durum ayrıca kodda hataların ayıklanmasında ve değişikliklerin yapılmasında zorluklar çıkarmaktadır.

- Kodda çok sayıda “kurma” ve “silme” komutu kullanılmaktadır. Kodlama yapılırken sayıca fazla olan bu komutlardan bazılarının gözden kaçırılarak unutulması muhtemeldir. Bu bakımdan yöntem hataya oldukça açıktır.
- Sistemin hatalı bir şekilde çalışmasına neden olan çığ etkisinin giderilmesi için ek bir çaba gerekmektedir. Bu durum PLC koduna dönüşüm sürecinin uzamasına neden olmaktadır (bkz. “5.1.1.1 Çığ etkisinin giderilmesi”).
- “Kurma” ve “silme” komutları ile kodlama yöntemi karmaşık yapıdaki sistemlerin PLC koduna dönüştürülmesi için uygun değildir.

PA’dan PLC koduna dönüşüm işleminin daha etkin ve daha kullanışlı bir şekilde yapılabilmesi için yukarıdaki eksikliklerin giderilmesi gerekmektedir. Bu amaçla daha az bellek ve işlemci gücü gerektirecek, kodun okunabilirliğini arttıracak ve özellikle büyük ölçekli sistemlerin daha kolay bir şekilde kodlanmasını sağlayacak yeni bir yöntem arayışına gidilmiştir. Bunun için ise hâlihazırda otomatlarda uygulanan bir yöntem ele alınmıştır. Otomatların PA’ya olan benzerliği göz önünde bulundurulduğunda bu yöntemin uyarlanarak PA için de kullanılması sağlanabilir.

Yöntemde bir otomatın PLC koduna dönüştürülmesi için mantıksal ifadeler kullanılmaktadır. Her bir durum için o duruma ilişkin mantıksal ifadeler elde edilmekte ve ardından çığ etkisini giderecek bir kodlama yapılarak PLC koduna dönüşüm işlemi tamamlanmaktadır. Her bir duruma ilişkin mantıksal ifadenin genel yapısı (5.1)’de verilmiştir [23]:

$$Q_i = \left(q_i + \sum_{x_i \in X_i} \left(q_{x_i} \sum_{y_i \in Y_i} S_{y_i} \right) \right) \prod_{z_i \in Z_i} \overline{S_{z_i}} \quad (5.1)$$

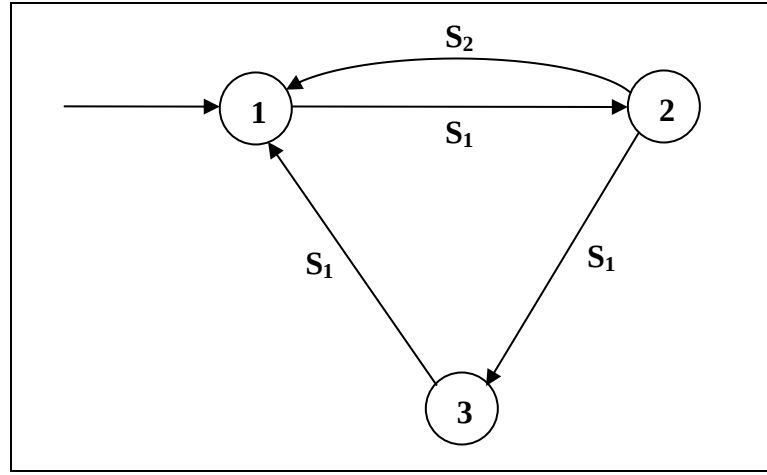
Burada;

- $X_i = \{ x_i \in X_i \mid Q_i \text{ durumuna getiren durumların indisleri} \}$
- $Y_i = \{ y_i \in Y_i \mid \text{Bir } Q_{x_i} \text{ durumundan } Q_i \text{ durumuna getiren girişlerin indisleri} \}$
- $Z_i = \{ z_i \in Z_i \mid Q_i \text{ durumundan çıkaran girişlerin indisleri} \}$

olarak tanımlanmıştır. q_i ise Q_i durumunun bir önceki değerini ifade etmektedir. Dolayısıyla bir Q_i durumuna ilişkin mantıksal ifadenin elde edilebilmesi için şunların bilinmesi gerekmektedir [23]:

- Q_i durumuna getiren durumlar
- Q_i durumuna getiren girişler
- Q_i durumundan çıkaran girişler
- Q_i durumunun bir önceki değeri

Yöntemin daha kolay anlaşılabilmesi bakımından Şekil 5.4'teki örnek otomatı ele alalım:



Şekil 5.4 : Örnek bir otomat diyagramı.

Şekil 5.4'te verilen otomat diyagramında 3 durum ve 2 farklı giriş bulunmaktadır. (5.2)'de sırasıyla bu otomata ait durumlar ve girişler kümesi verilmiştir:

$$Q = \{Q_1, Q_2, Q_3\}$$

$$S = \{S_1, S_2\} \quad (5.2)$$

(5.1) ve (5.2) kullanılarak örnek otomattaki her bir duruma ilişkin mantıksal ifade aşağıdaki gibi elde edilebilir:

- **Q_1 Durumu:** Otomat diyagramında başlangıç durumu olarak gösterilmektedir. Q_2 durumundayken S_2 girişi uygulandığında ve Q_3 durumundayken S_1 girişi uygulandığında bu duruma gelinmektedir. Q_1 durumundayken S_1 girişi uygulandığında ise bu durumdan çıkılmaktadır. Bu bilgilerden yola çıkılarak Q_1 durumuna ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$Q_1 = (q_1 + q_2 S_2 + q_3 S_1) \overline{S_1} \quad (5.3)$$

- **Q₂ Durumu:** Otomat diyagramındaki bir ara durumdur. Bu duruma sadece Q₁ durumundayken S₁ girişi uygulandığında gelinmektedir. Q₂ durumundayken S₁ veya S₂ girişi uygulandığında ise bu durumdan çıkılmaktadır. Bu bilgilerden yola çıkılarak Q₂ durumuna ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$Q_2 = (q_2 + q_1 S_1) \overline{S_1} \overline{S_2} \quad (5.4)$$

- **Q₃ Durumu:** Otomat diyagramındaki bir ara durumdur. Bu duruma sadece Q₂ durumundayken S₁ girişi uygulandığında gelinmektedir. Q₃ durumundayken S₁ girişi uygulandığında ise bu durumdan çıkılmaktadır. Bu bilgilerden yola çıkılarak Q₃ durumuna ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$Q_3 = (q_3 + q_2 S_1) \overline{S_1} \quad (5.5)$$

Bu yöntem kullanılarak otomat diyagramındaki her bir duruma ilişkin mantıksal ifade doğrudan ve hızlı bir şekilde elde edilebilmektedir. Dikkat edilecek olursa her bir durum için iki farklı değişken kullanılmaktadır. Q_i değişkeni söz konusu durumun mevcut değerini ifade ederken, q_i değişkeni ise bu durumun bir önceki değerini ifade etmektedir. Aslında her bir durum için iki farklı değişken kullanılarak çıkış etkisinin giderilmesi sağlanmaktadır. Her bir duruma ilişkin mantıksal ifade elde edildikten sonra PLC koduna dönüşüm işleminin son aşaması olarak (5.6)'da verilen işlemler yapılır [23]:

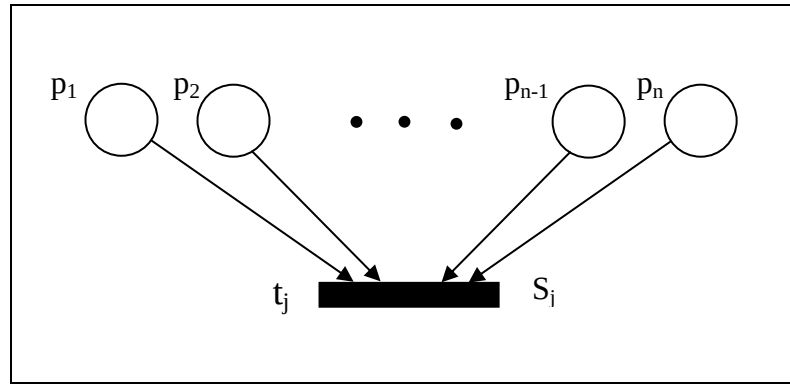
$$\begin{aligned} q_1 &= Q_1 \\ q_2 &= Q_2 \\ q_3 &= Q_3 \end{aligned} \quad (5.6)$$

Bu işlemlerle her bir durumun eski değeri yeni değerine eşitlenmekte ve bir sonraki PLC çevrimi için kullanıma hazır hale getirilmektedir. Böylelikle ek bir işleme gerek duyulmadan çıkış etkisinin giderilmesi sağlanmaktadır. Sonuç olarak bu yöntem sayesinde sadece mantıksal ifadeler kullanılarak bir otomat diyagramından PLC koduna geçiş hızlı ve kolay bir şekilde sağlanabilmektedir [23].

5.2 Mantıksal İfadeler Yönteminin Petri Ağları için Uyarlanması

Otomat diyagramları sadece durumlar ve bu durumlar arasındaki geçişlerden oluşmaktadır. Durumlar arası geçişler ise giriş koşullarına bağlıdır. PA’da ise bu yapı farklıdır. PA’da otomattaki durumlar yerine yerler kullanılmakta ve yerlerdeki koşulların sağlanıp sağlanmadığı jetonlarla belirtilmektedir. Durum değişiklikleri ise jetonların yerler arasındaki transferi ile gerçekleştirilmektedir. Bu jetonların transferi yerleri birbirine bağlayan geçişlerin tetiklenmesi ile mümkün olmaktadır. Geçişlerin tetiklenebilmesi için ilgili giriş yerlerindeki koşulların sağlanmış olması gerekmektedir (bkz. “2 PETRİ AĞLARININ GENEL TANIMI”). Otomat diyagramından PLC koduna dönüşüm için kullanılan mevcut yöntemin PA için de kullanılabilmesi amacıyla, otomat diyagramı ile PA arasındaki farklılıklardan dolayı, bu yöntem üzerinde birtakım değişikliklerin yapılması gerekmektedir.

PA’da yerler ve geçişler olmak üzere iki farklı bileşen bulunduğundan bu bileşenlerin her ikisi için de mantıksal ifadelerin ayrı ayrı elde edilmesi gerekmektedir. Öncelikle geçişlere ilişkin mantıksal ifadelerin elde edilmesi üzerinde duralım. Bu amaçla Şekil 5.5’teki örnek PA’yı inceleyelim:



Şekil 5.5 : Geçişlere ilişkin mantıksal ifadelerin elde edilmesi.

Şekil 5.5’te verilen PA’da t_j geçişinin p₁, p₂, ..., p_{n-1}, p_n olmak üzere “n” adet giriş yeri bulunmaktadır. Bu PA’nın 1-sınırlı PA olduğu varsayımı altında t_j geçişinin tetiklenebilmesi için tüm giriş yerlerinde 1’er tane jeton bulunması ve aynı zamanda S_j koşulunun da sağlanması gerekmektedir. Bu bilgiler ışığında t_j geçişinin tetiklenmesine ilişkin genel mantıksal ifade aşağıdaki gibi elde edilir:

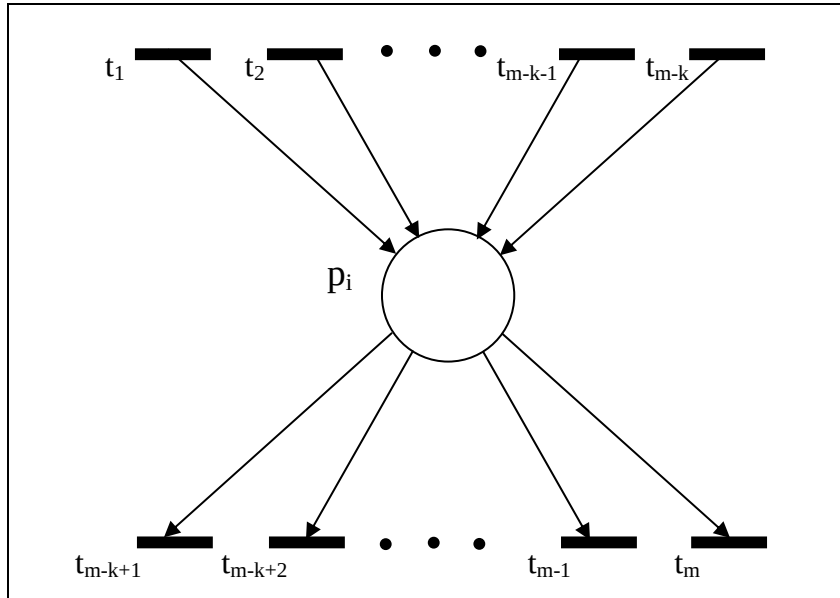
$$t_j = p_1 p_2 \dots p_{n-1} p_n S_j \quad (5.7)$$

(5.7)'de $p_1, p_2, \dots, p_{n-1}, p_n$ ile karşılık gelen yerlerdeki jeton sayısı ifade edilmektedir. 1-sınırlı PA için bu değer mantıksal “0” veya “1” olabilir. Bir geçişe ilişkin mantıksal ifadenin “0” değerini alması o geçişin tetiklenmeyeceğini, “1” değerini alması ise geçişin tetikleneceği anlamına gelmektedir.

Eğer bir geçişin hiçbir giriş yeri yoksa ve bu geçişin tetiklenmesi sadece koşula bağlıysa bu durumda (5.7)'deki tüm giriş yerlerine ilişkin ifadeler düşer ve mantıksal ifade aşağıdaki hale gelir:

$$t_j = S_j \quad (5.8)$$

(5.8)'de verilen denklemde geçişe ilişkin mantıksal ifade sadece koşuldan oluşmaktadır. Böylelikle PA'daki geçişlere ilişkin mantıksal ifadeleri elde etmiş olduk. Şimdi de yerlere ilişkin mantıksal ifadelerin elde edilmesi üzerinde duralım. Bu amaçla Şekil 5.6'da verilen örnek PA'yı ele alalım:



Şekil 5.6 : Yerlere ilişkin mantıksal ifadelerin elde edilmesi.

Şekil 5.6'daki PA'da p_i yeri ile bu yere ilişkin giriş ve çıkış geçişleri bulunmaktadır. Giriş geçişleri olan $t_1, t_2, \dots, t_{m-k-1}, t_{m-k}$ geçişlerinden herhangi biri tetiklendiği zaman p_i yerine 1 adet jeton konulmakta, çıkış geçişleri olan $t_{m-k+1}, t_{m-k+2}, \dots, t_{m-1}, t_m$ geçişlerinden herhangi biri tetiklendiği zaman ise p_i yerinden 1 adet jeton harcanmaktadır. Bu durumda PA'nın 1-sınırlı PA olduğu varsayımı altında herhangi bir anda p_i yerinde en fazla tek bir jeton bulunabilir. Buna göre p_i yerine ilişkin mantıksal ifade (5.9)'daki gibi tanımlanabilir:

$$p_i^+ = \sum_{x_i \in X_i} t_{x_i} + p_i \prod_{z_i \in Z_i} \overline{t_{z_i}} \quad (5.9)$$

Burada;

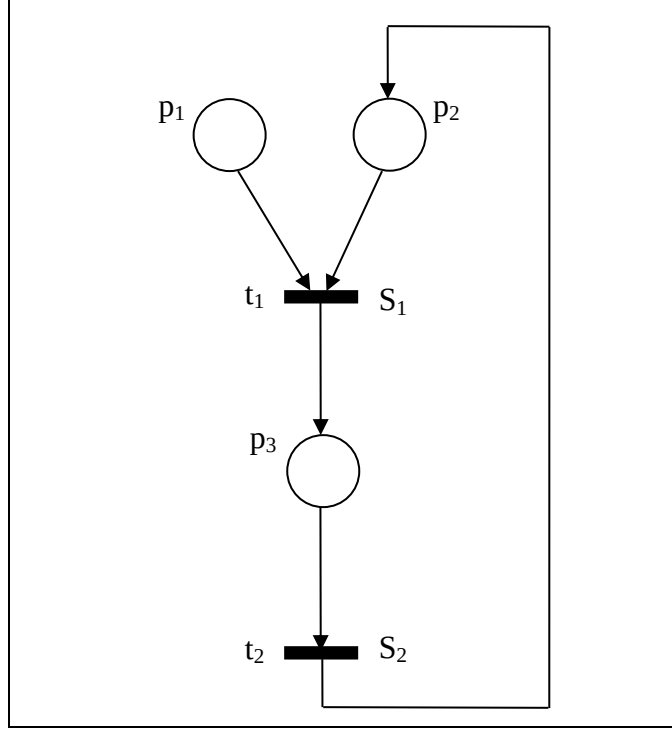
- $X_i = \{ x_i \in X_i \mid p_i \text{ yerinin giriş geçişlerinin indisleri} \}$
- $Z_i = \{ z_i \in Z_i \mid p_i \text{ yerinin çıkış geçişlerinin indisleri} \}$
- $p_i = \{ p_i \text{ yerindeki mevcut jeton sayısı} \}$
- $p_i^+ = \{ p_i \text{ yerindeki tetikleme sonrası jeton sayısı} \}$

olarak tanımlanmıştır. Dikkatli bakılacak olursa (5.9)'da verilen denklem, otomat diyagramlarında durumların ifade edilmesi için kullanılan (5.1) denkleminde oldukça benzerlik göstermektedir. Ancak (5.1) denkleminde farklı olarak (5.9)'da girişler yerine geçişler kullanılmaktadır. Ayrıca (5.9)'da verilen denklem (5.1)'e göre daha az karmaşıktır. Böylelikle PA'daki yerler için elde edilen mantıksal ifadeler otomat diyagramlarındaki durumlar için elde edilenlerden daha basit olmaktadır. (5.9)'da verilen mantıksal ifade aşağıdaki gibi de tanımlanabilir:

$$p_i^+ = \sum_{t_j \in I(p_i)} t_j + p_i \prod_{t_k \in O(p_i)} \overline{t_k} \quad (5.10)$$

PA için uyarlanan bu yöntemde çıkış etkisinin giderilmesi için geçişlere ilişkin mantıksal ifadeler yerlere ilişkin mantıksal ifadelerden önce kodlanmalı ve geçişlere ilişkin mantıksal ifadelerde koşullar kodlanırken inen ya da çıkan kenar kullanılmalıdır. Böyle bir kodlama yapılarak çıkış etkisinin giderilmesi için otomatlardaki gibi ayrı bir kodlamaya ihtiyaç duyulmamaktadır. Böylelikle PA'nın PLC koduna dönüşümü daha hızlı gerçekleştirilebilmekte ve elde edilen kodun boyunun daha kısa olması sağlanmaktadır.

Yapılan bu çalışma ile otomat diyagramlarının PLC koduna dönüşümü için kullanılan yöntem PA için uyarlanmış ve PA'nın PLC koduna dönüştürülebilmesi için geçişler ve yerlere ilişkin mantıksal ifadeler elde edilmiştir. (5.7) veya (5.8)'de geçişler için elde edilen mantıksal ifadeler ile (5.9)'da yerler için elde edilen mantıksal ifadeler kullanılarak PA'nın PLC koduna dönüşümü hızlı ve etkin bir şekilde gerçekleştirilebilir. Yöntemin daha iyi anlaşılması bakımından Şekil 5.7'deki örnek PA'yı PLC koduna dönüştürelim:



Şekil 5.7 : PA'nın PLC koduna dönüştürülmesi.

Şekil 5.7'de verilen örnek PA'yı PLC koduna dönüştürebilmek için p_1 , p_2 ve p_3 yerleri ile t_1 ve t_2 geçişlerine ilişkin mantıksal ifadeleri ayrı ayrı elde etmemiz gerekmektedir. Dönüşüm işlemine öncelikle geçişlerden başlayalım:

- t_1 geçişinin tetiklenip aktif hale gelebilmesi için p_1 ve p_2 yerlerinde 1'er adet jeton bulunmalı ve aynı zamanda S_1 koşulu sağlanmış olmalıdır. Bu bilgiler doğrultusunda (5.7) kullanılarak t_1 geçişine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$t_1 = p_1 p_2 S_1 \quad (5.11)$$

- t_2 geçişinin tetiklenip aktif hale gelebilmesi için sadece p_3 yerinde 1 adet jeton bulunmalı ve aynı zamanda S_2 koşulu sağlanmış olmalıdır. Bu bilgiler doğrultusunda (5.7) kullanılarak t_2 geçişine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$t_2 = p_3 S_2 \quad (5.12)$$

Böylelikle geçişlere ait mantıksal ifadeleri elde etmiş olduk. Şimdi de yerlere ait mantıksal ifadeleri elde edelim:

- p_1 yerine giren herhangi bir geçiş bulunmamaktadır. Bu nedenle p_1 yerine jeton konulması söz konusu olmayacaktır. Bununla birlikte p_1 yerinden çıkan bir t_1 geçişi söz konusudur. t_1 geçişi her tetiklendiğinde ise p_1 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_1 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_1^+ = p_1 \bar{t}_1 \quad (5.13)$$

- p_2 yerine giren bir t_2 geçişi mevcuttur. t_2 geçişi her tetiklendiğinde p_2 yerine jeton konulacaktır. Bununla birlikte p_2 yerinden çıkan bir t_1 geçişi söz konusudur. t_1 geçişi her tetiklendiğinde ise p_2 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_2 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_2^+ = t_2 + p_2 \bar{t}_1 \quad (5.14)$$

- p_3 yerine giren bir t_1 geçişi mevcuttur. t_1 geçişi her tetiklendiğinde p_3 yerine jeton konulacaktır. Bununla birlikte p_3 yerinden çıkan bir t_2 geçişi söz konusudur. t_2 geçişi her tetiklendiğinde ise p_3 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_3 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_3^+ = t_1 + p_3 \bar{t}_2 \quad (5.15)$$

Yerler için de mantıksal ifadelerin elde edilmesiyle PA'nın PLC koduna dönüşümü tamamlanmış olur.

Görüldüğü gibi PA için uyarladığımız PLC koduna dönüşüm yöntemi kullanılarak PA'daki geçişler ve yerlere ilişkin mantıksal ifadeler doğrudan elde edilebilmektedir. Böylelikle PA'nın PLC koduna dönüşümü hızlı ve kolay bir şekilde gerçekleştirilebilmektedir. Burada dikkat edilmesi gereken şey çığ etkisinin oluşmaması için yukarıdaki örnekte de olduğu gibi geçişlere ilişkin mantıksal ifadelerin yerlere ilişkin mantıksal ifadelerden önce kodlanmasıdır. Ayrıca geçişler için mantıksal ifadeler kodlanırken koşullara ait işaretlerin inen ya da çıkan kenarı kullanılmalıdır. Aksi takdirde çığ etkisi ile karşı karşıya kalınabilir ve bu nedenle sistem istendiği gibi çalışmayabilir.

5.3 Petri Ağlarının Sistem Matrisinden PLC Koduna Geçiş

PA'nın PLC koduna dönüştürülmesi için uyguladığımız yöntemde yerlere ilişkin mantıksal ifadeler grafiksel gösterimden yola çıkılarak elde edilmektedir. Bu yöntemden farklı olarak PA'daki yerlere ilişkin mantıksal ifadeler söz konusu PA'nın sistem matrisinden yola çıkılarak da elde edilebilir. Bu amaçla yeni bir yöntem önerilecektir. Bu yöntemde (5.9)'da verilen mantıksal ifade aşağıdaki gibi yeniden yazılabilir:

$$p_i^+ = \sum_{x_i \in X_i} t_{x_i} + p_i \prod_{z_i \in Z_i} \overline{t_{z_i}} \quad (5.16)$$

Burada;

- $X_i = \{ x_i \in X_i \mid x_i = j, \quad i=1, \dots, n, \quad j=1, \dots, m, \quad d_{ji} = 1 \text{ ise} \}$
- $Z_i = \{ z_i \in Z_i \mid z_i = j, \quad i=1, \dots, n, \quad j=1, \dots, m, \quad d_{ji} = -1 \text{ ise} \}$
- $p_i = \{ p_i \text{ yerindeki mevcut jeton sayısı} \}$
- $p_i^+ = \{ p_i \text{ yerindeki tetikleme sonrası jeton sayısı} \}$

olarak yeniden tanımlanmıştır. “n” ve “m” sırasıyla toplam yer ve geçiş sayısıdır. (5.16)'da X_i kümesi p_i yerine ait giriş geçişlerinin indislerini, Z_i kümesi ise p_i yerine ait çıkış geçişlerinin indislerini içermektedir. Önerdiğimiz yöntemi kullanarak Şekil 5.7'de verilen örnek PA'daki yerler için mantıksal ifadeleri bir kez daha elde edelim. Öncelikle PA'nın sistem matrisini elde edelim:

$$D = \begin{bmatrix} -1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \quad (5.17)$$

(5.17)'da verilen sistem matrisinden yola çıkılarak ve (5.16) kullanılarak yerlere ilişkin mantıksal ifadeler aşağıdaki gibi elde edilir:

- p_1 yeri için sistem matrisinde 1. sütuna bakılır. Bu sütunda değeri “1” olan herhangi bir eleman bulunmamaktadır. Dolayısıyla p_1 yerine ait bir giriş geçişi mevcut değildir. Bu durumda $X_1 = \{ \}$ olur. Yine aynı sütunda sadece 1. satırdaki elemanın değeri “-1” olduğundan p_1 yerine ait bir t_1 çıkış geçişi bulunmaktadır. Bu durumda ise $Z_1 = \{1\}$ olur. Bu bilgiler ışığında p_1 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_1^+ = p_1 \overline{t_1} \quad (5.18)$$

- p_2 yeri için sistem matrisinde 2. sütuna bakılır. Bu sütunda sadece 2. satırdaki elemanın değeri “1” olduğundan p_2 yerine ait bir t_2 giriş geçişi bulunmaktadır. Bu durumda $X_2 = \{2\}$ olur. Yine aynı sütunda sadece 1. satırdaki elemanın değeri “-1” olduğundan p_2 yerine ait bir t_1 çıkış geçişi bulunmaktadır. Bu durumda ise $Z_2 = \{1\}$ olur. Bu bilgiler ışığında p_2 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_2^+ = t_2 + p_2 \bar{t}_1 \quad (5.19)$$

- p_3 yeri için sistem matrisinde 3. sütuna bakılır. Bu sütunda sadece 1. satırdaki elemanın değeri “1” olduğundan p_3 yerine ait bir t_1 giriş geçişi bulunmaktadır. Bu durumda $X_3 = \{1\}$ olur. Yine aynı sütunda sadece 2. satırdaki elemanın değeri “-1” olduğundan p_3 yerine ait bir t_2 çıkış geçişi bulunmaktadır. Bu durumda ise $Z_3 = \{2\}$ olur. Bu bilgiler ışığında p_3 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_3^+ = t_1 + p_3 \bar{t}_2 \quad (5.20)$$

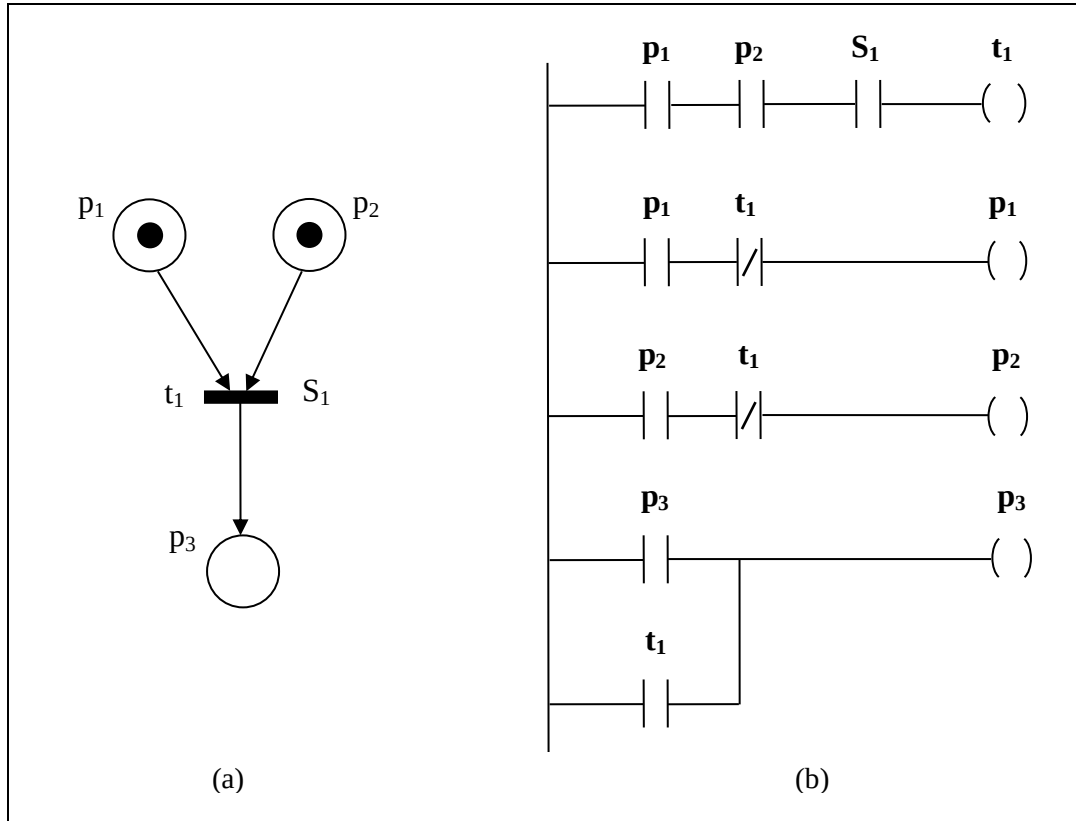
Böylelikle PA’nın sistem matrisinden yola çıkarak yerler için mantıksal ifadeleri elde etmiş olduk. Dikkat edilecek olursa önerdiğimiz bu yöntem kullanılarak elde edilen mantıksal ifadeler ile grafiksel gösterimden yola çıkılarak elde edilen mantıksal ifadelerin aynı olduğu görülmektedir.

5.4 Petri Ağları Yapılarının PLC Koduna Dönüşümü

Bu bölümde birtakım PA yapılarının “5.2 Mantıksal İfadeler Yönteminin Petri Ağları için Uyarlanması” bölümünde tanıtılan yöntem temel alınıp genişletilerek PLC koduna dönüşümü gerçekleştirilmiştir. Söz konusu PA yapılarının 1-sınırlı PA olduğu varsayılmıştır.

5.4.1 Senkronizasyon yapısının PLC kodu karşılığı

Şekil 5.8(a)’da senkronizasyon yapısı verilmiştir. Bu senkronizasyon yapısında t_1 geçişinin tetiklenmesi için p_1 ve p_2 yerlerinde jeton bulunmalı ve S_1 koşulu sağlanmış olmalıdır. Bu koşullar sağlandığında t_1 geçişi tetiklenir ve p_1 ile p_2 yerlerindeki jetonlar harcanarak p_3 yerine jeton konulur. Şekil 5.8(b)’de senkronizasyon yapısının PLC kodu karşılığı verilmiştir:



Şekil 5.8 : (a) Senkronizasyon yapısı. (b) Yapının PLC kodu karşılığı.

Şekil 5.8(a)'da verilen senkronizasyon yapısındaki p_1 , p_2 , p_3 yerleri ve t_1 geçişi için mantıksal ifadeler aşağıdaki gibi elde edilebilir:

- t_1 geçişinin tetiklenmesi için hem p_1 hem de p_2 yerinde jeton bulunmalı ve aynı zamanda S_1 koşulu sağlanmış olmalıdır. Bu durumda (5.7) kullanılarak t_1 geçişine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$t_1 = p_1 p_2 S_1 \quad (5.21)$$

Tetikleme için gerekli koşullar sağlanmazsa (5.21)'deki mantıksal ifade "0" değerini alır. Bu durumda t_1 geçişi tetiklenmez. Tetikleme koşulları sağlanırsa (5.21)'deki mantıksal ifade "1" değerini alır. Bu durumda ise t_1 geçişi tetiklenir.

- p_1 yerine giren bir geçiş bulunmamaktadır. Bununla birlikte p_1 yerinden çıkan bir t_1 geçişi söz konusudur. t_1 geçişi her tetiklendiğinde p_1 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_1 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_1^+ = p_1 \bar{t}_1 \quad (5.22)$$

- p_2 yerine giren bir geçiş bulunmamaktadır. Bununla birlikte p_2 yerinden çıkan bir t_1 geçişi söz konusudur. t_1 geçişi her tetiklendiğinde p_2 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_2 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_2^+ = p_2 \bar{t}_1 \quad (5.23)$$

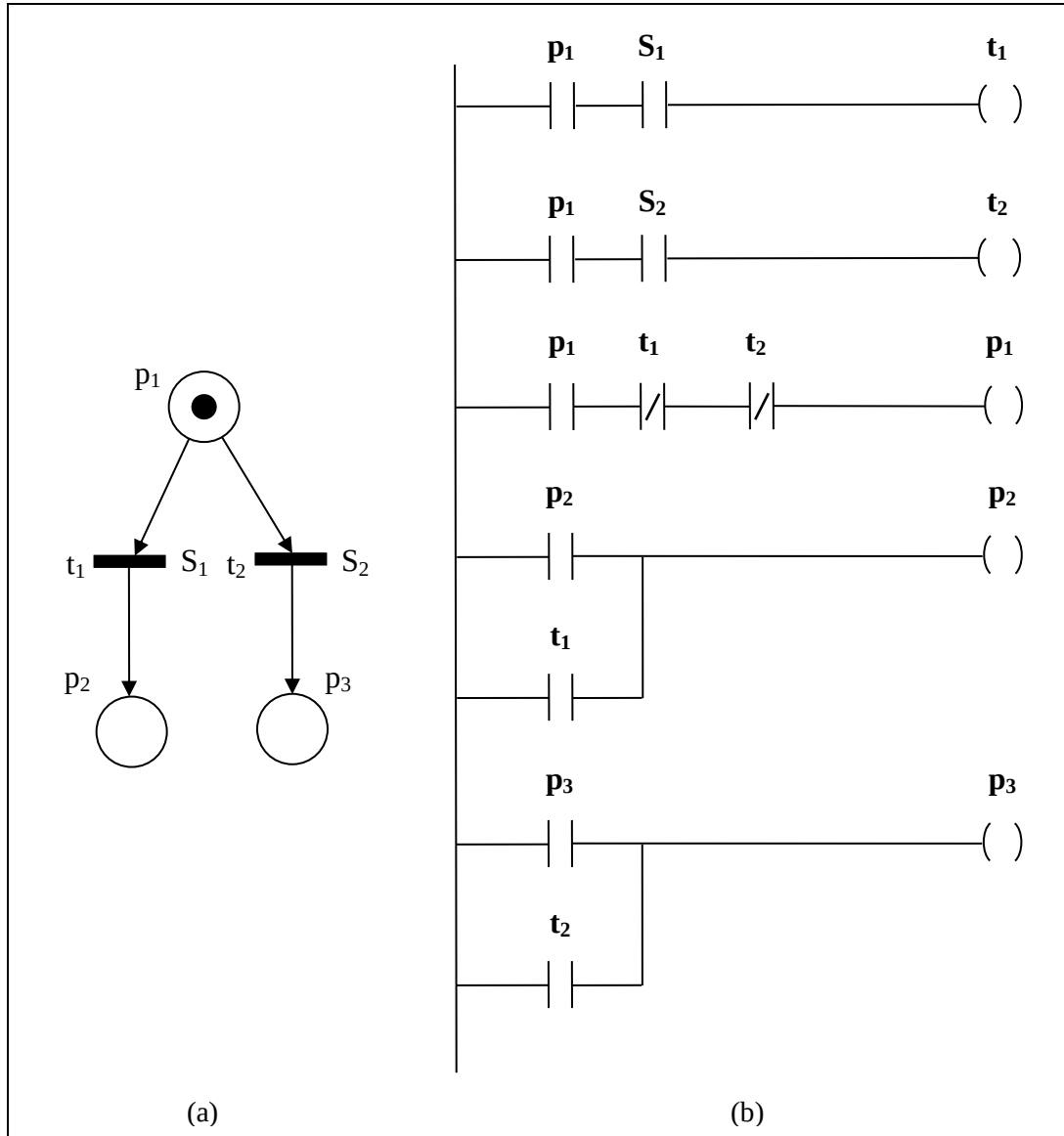
- p_3 yerine giren bir t_1 geçişi bulunmaktadır. t_1 geçişi her tetiklendiğinde p_3 yerine jeton konulacaktır. Bununla birlikte p_3 yerinden çıkan bir geçiş bulunmamaktadır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_3 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_3^+ = t_1 + p_3 \quad (5.24)$$

(5.21)'de geçiş için elde edilen mantıksal ifade ile (5.22), (5.23) ve (5.24)'te yerler için elde edilen mantıksal ifadeler kullanılarak senkronizasyon yapısının PLC kodu karşılığı Şekil 5.8(b)'de verilmiştir. Dikkat edilecek olursa çıkış etkisinin giderilmesi için geçişe ilişkin mantıksal ifade yerlere ilişkin mantıksal ifadelerden önce konumlandırılmıştır.

5.4.2 Çakışma yapısının PLC kodu karşılığı

Şekil 5.9(a)'da çakışma yapısı verilmiştir. Bu çakışma yapısında t_1 ve t_2 geçişlerinin aktif hale gelebilmesi için ortak giriş yeri olan p_1 yerinde jeton bulunmalıdır. Aktif hale gelmiş bir geçişin tetiklenmesi için ise o geçişe ilişkin koşulun sağlanmış olması gerekmektedir. Bu durum göz önüne alınarak S_1 koşulu sağlandığında t_1 geçişinin, S_2 koşulu sağlandığında ise t_2 geçişinin tetikleneceği söylenebilir. Başka bir deyişle aktifleşmiş durumda olan iki geçişten sadece biri tetiklenecek ve hangi geçişin tetikleneceğini sağlanmış olan koşul belirleyecektir. Çakışma yapısının davranışı bu şekildedir. Hangi geçiş tetiklenirse tetiklensin p_1 ortak giriş yeri olduğu için p_1 yerindeki jeton harcanacaktır. Eğer t_1 geçişi tetiklenirse p_2 yerine, t_2 geçişi tetiklenirse p_3 yerine jeton konulacaktır. Şekil 5.9(b)'de çakışma yapısının PLC kodu karşılığı verilmiştir:



Şekil 5.9 : (a) Çakışma yapısı. (b) Yapının PLC kodu karşılığı.

Şekil 5.9(a)'da verilen çakışma yapısındaki p_1 , p_2 , p_3 yerleri ve t_1 , t_2 geçişleri için mantıksal ifadeler aşağıdaki gibi elde edilebilir:

- t_1 geçişinin tetiklenmesi için ortak giriş yeri olan p_1 yerinde jeton bulunmalı ve aynı zamanda S_1 koşulu sağlanmış olmalıdır. Bu durumda (5.7) kullanılarak t_1 geçişine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$t_1 = p_1 S_1 \quad (5.25)$$

Tetikleme için gerekli koşullar sağlanmazsa (5.25)'teki mantıksal ifade "0" değerini alır. Bu durumda t_1 geçişi tetiklenmez. Tetikleme koşulları sağlanırsa (5.25)'teki mantıksal ifade "1" değerini alır. Bu durumda ise t_1 geçişi tetiklenir.

- t_2 geçişinin tetiklenmesi için ortak giriş yeri olan p_1 yerinde jeton bulunmalı ve aynı zamanda S_2 koşulu sağlanmış olmalıdır. Bu durumda (5.7) kullanılarak t_2 geçişine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$t_2 = p_1 S_2 \quad (5.26)$$

Tetikleme için gerekli koşullar sağlanmazsa (5.26)'daki mantıksal ifade "0" değerini alır. Bu durumda t_2 geçişi tetiklenmez. Tetikleme koşulları sağlanırsa (5.26)'daki mantıksal ifade "1" değerini alır. Bu durumda ise t_2 geçişi tetiklenir.

- p_1 yerine giren bir geçiş bulunmamaktadır. Bununla birlikte p_1 yerinden çıkan t_1 ve t_2 geçişleri söz konusudur. t_1 veya t_2 geçişi her tetiklendiğinde p_1 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_1 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_1^+ = p_1 \overline{t_1} \overline{t_2} \quad (5.27)$$

- p_2 yerine giren bir t_1 geçişi bulunmaktadır. t_1 geçişi her tetiklendiğinde p_2 yerine jeton konulacaktır. Bununla birlikte p_2 yerinden çıkan bir geçiş bulunmamaktadır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_2 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_2^+ = t_1 + p_2 \quad (5.28)$$

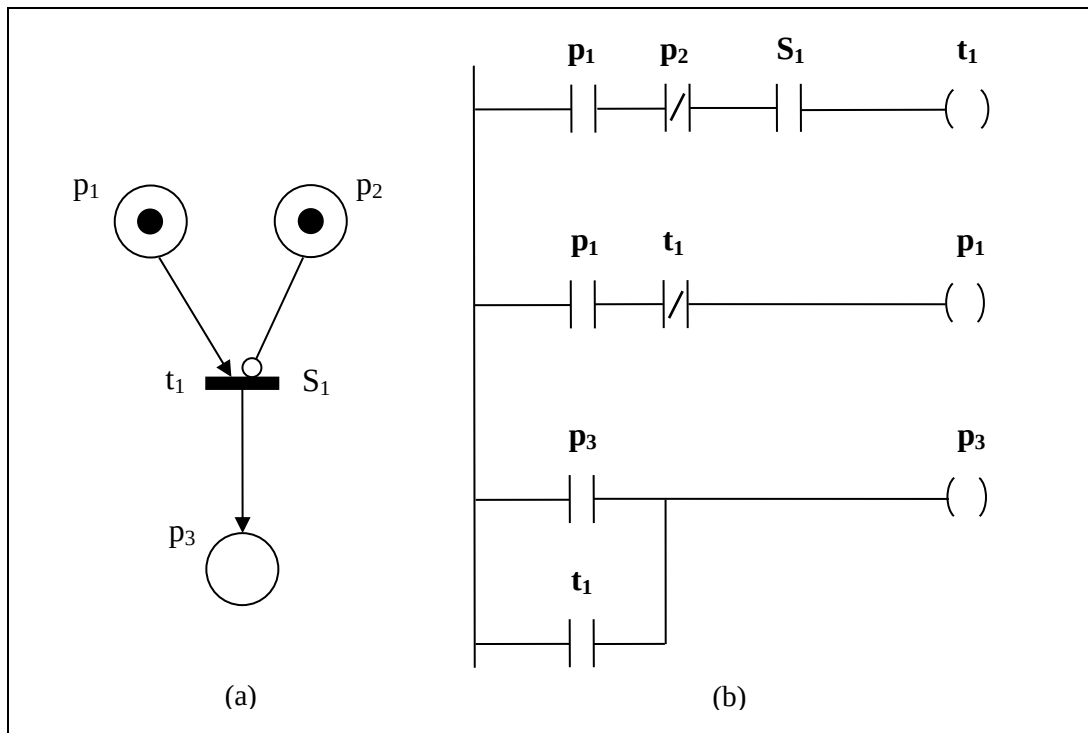
- p_3 yerine giren bir t_2 geçişi bulunmaktadır. t_2 geçişi her tetiklendiğinde p_3 yerine jeton konulacaktır. Bununla birlikte p_3 yerinden çıkan bir geçiş bulunmamaktadır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_3 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_3^+ = t_2 + p_3 \quad (5.29)$$

(5.25) ve (5.26)'da geçişler için elde edilen mantıksal ifadeler ile (5.27), (5.28) ve (5.29)'da yerler için elde edilen mantıksal ifadeler kullanılarak çakışma yapısının PLC kodu karşılığı Şekil 5.9(b)'de verilmiştir. Dikkat edilecek olursa çık etkisinin giderilmesi için geçişlere ilişkin mantıksal ifadeler yerlere ilişkin mantıksal ifadelerden önce konumlandırılmıştır.

5.4.3 Yasaklayıcı yay içeren Petri Ağları'nın PLC kodu karşılığı

Şekil 5.10(a)'da yasaklayıcı yay içeren bir PA verilmiştir. Bu PA'da p_2 yeri ile t_1 geçişi arasında bir yasaklayıcı yay bulunmaktadır. p_1 yerinde jeton bulursa ve S_1 koşulu sağlanmış olsa bile p_2 yerinde jeton bulunması durumunda söz konusu yasaklayıcı yay t_1 geçişinin aktif hale gelip tetiklenmesini engelleyecektir. Bu nedenle t_1 geçişinin tetiklenmesi için p_1 yerinde jeton bulunması ve S_1 koşulunun sağlanmış olmasının yanı sıra p_2 yerinde jeton bulunmaması gerekmektedir. t_1 geçişi tetiklendiğinde p_1 yerindeki jeton harcanır ve p_3 yerine jeton konulur. Yasaklayıcı yayın bağlı olduğu p_2 yeri ile ilgili herhangi bir jeton transferi gerçekleştirilmez. Şekil 5.10(b)'de yasaklayıcı yay içeren PA'nın PLC kodu karşılığı verilmiştir:



Şekil 5.10 : (a) Yasaklayıcı yay içeren PA. (b) PA'nın PLC kodu karşılığı.

Şekil 5.10(a)'da verilen PA'da p_1 , p_2 , p_3 yerleri ve t_1 geçişi için mantıksal ifadeler aşağıdaki gibi elde edilebilir:

- t_1 geçişinin tetiklenmesi için p_1 yerinde jeton bulunmalı ancak p_2 yerinde jeton olmamalıdır. Aynı zamanda S_1 koşulu da sağlanmış olmalıdır. Bu durumda (5.7) kullanılarak t_1 geçişine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$t_1 = p_1 \overline{p_2} S_1 \quad (5.30)$$

(5.30)'a bakıldığında p_2 yerinde jeton bulunmaması koşulunun diğer koşullardan farklı olarak mantıksal DEĞİL operatörü kullanılarak belirtildiği görülmektedir. Bunun dışında diğer koşullar olduğu gibi kullanılmıştır. Buradan yola çıkılarak (5.7) ve (5.8)'de geçişler için verilen mantıksal ifadelerde yasaklayıcı yaylar için şöyle bir istisna uygulanmalıdır: Bir geçişe ilişkin mantıksal ifade oluşturulurken o geçişe giren yasaklayıcı yaylar varsa bu yasaklayıcı yayların bağlı olduğu yerlere ait koşullar için mantıksal DEĞİL operatörü kullanılmalıdır.

Tetikleme için gerekli koşullar sağlanmazsa (5.30)'daki mantıksal ifade "0" değerini alır. Bu durumda t_1 geçişi tetiklenmez. Tetikleme koşulları sağlanırsa (5.30)'daki mantıksal ifade "1" değerini alır. Bu durumda ise t_1 geçişi tetiklenir.

- p_1 yerine giren bir geçiş bulunmamaktadır. Bununla birlikte p_1 yerinden çıkan bir t_1 geçişi söz konusudur. t_1 geçişi her tetiklendiğinde p_1 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_1 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_1^+ = p_1 \bar{t}_1 \quad (5.31)$$

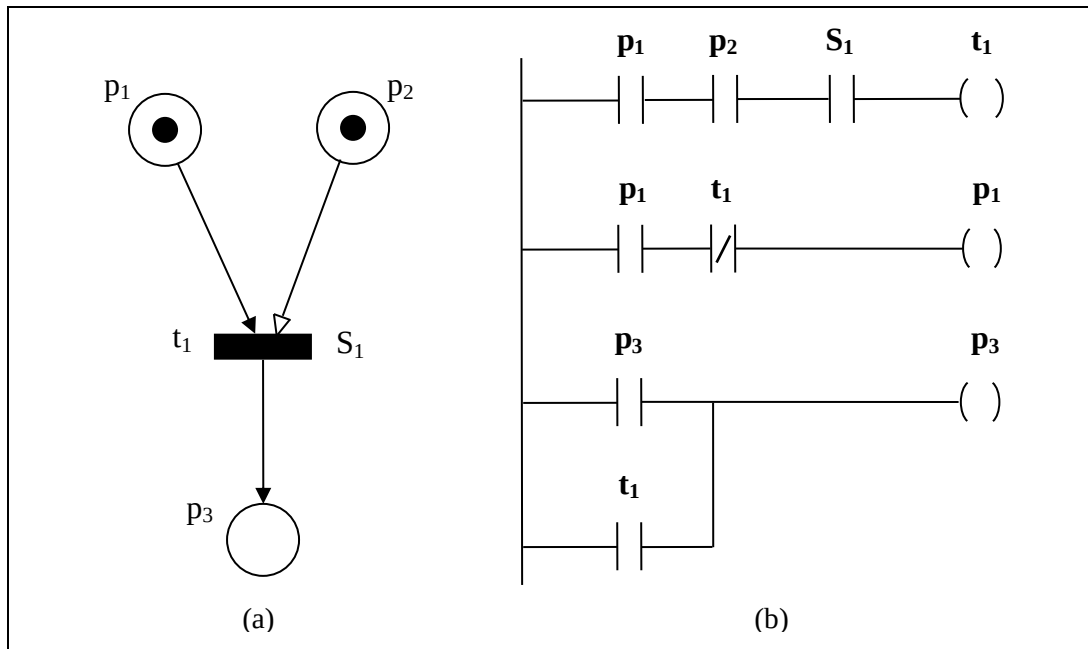
- t_1 geçişi tetiklendiğinde yasaklayıcı yayın bağlı olduğu p_2 yeri ile ilgili herhangi bir jeton transferi gerçekleştirilmemektedir. Dolayısıyla bu PA'nın PLC kodunda p_2 yerine ilişkin bir mantıksal ifade bulunmayacaktır.
- p_3 yerine giren bir t_1 geçişi bulunmaktadır. t_1 geçişi her tetiklendiğinde p_3 yerine jeton konulacaktır. Bununla birlikte p_3 yerinden çıkan bir geçiş bulunmamaktadır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_3 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_3^+ = t_1 + p_3 \quad (5.32)$$

(5.30)'da geçiş için ve (5.31) ile (5.32)'de yerler için elde edilen mantıksal ifadeler kullanılarak yasaklayıcı yay içeren PA'nın PLC kodu karşılığı Şekil 5.10(b)'de verilmiştir. Bu PLC kodu karşılığına bakıldığında yasaklayıcı yayın bağlı olduğu p_2 yerine ilişkin bir mantıksal ifadenin olmadığı görülmektedir. Bunun dışında olması gerektiği gibi çıkış etkisinin giderilmesi için geçişe ilişkin mantıksal ifade yerlere ilişkin mantıksal ifadelerden önce konumlandırılmıştır.

5.4.4 Yetkilendirici yay içeren Petri Ağları'nın PLC kodu karşılığı

Şekil 5.11(a)'da yetkilendirici yay içeren bir PA verilmiştir. Bu PA'da p_2 yeri ile t_1 geçişi arasında bir yetkilendirici yay bulunmaktadır. t_1 geçişinin tetiklenmesi için söz konusu yetkilendirici yayın bağlı olduğu p_2 yeri ile ilgili sağlanması gereken koşulda herhangi bir değişiklik olmayacaktır. Dolayısıyla t_1 geçişinin tetiklenmesi için gerekli olan koşullar oluşturulurken p_2 ile t_1 geçişi arasında yetkilendirici yayın yerinde standart yay varmış gibi düşünülebilir. O halde t_1 geçişinin tetiklenmesi için hem p_1 hem de p_2 yerinde jeton bulunmalı ve S_1 koşulu sağlanmış olmalıdır. Bu koşullar sağlanıp t_1 geçişi tetiklendiğinde p_1 yerindeki jeton harcanır, ancak yetkilendirici yayın bağlı olduğu p_2 yerindeki jeton harcanmaz ve olduğu gibi bırakılır. Bunun dışında olması gerektiği gibi p_3 yerine jeton konulur. Şekil 5.11(b)'de yetkilendirici yay içeren PA'nın PLC kodu karşılığı verilmiştir:



Şekil 5.11 : (a) Yetkilendirici yay içeren PA. (b) PA'nın PLC kodu karşılığı.

Şekil 5.11(a)'da verilen PA'da p_1 , p_2 , p_3 yerleri ve t_1 geçişi için mantıksal ifadeler aşağıdaki gibi elde edilebilir:

- t_1 geçişinin tetiklenmesi için hem p_1 hem de p_2 yerinde jeton bulunmalı ve aynı zamanda S_1 koşulu sağlanmış olmalıdır. Bu durumda (5.7) kullanılarak t_1 geçişine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$t_1 = p_1 p_2 S_1 \quad (5.33)$$

(5.33)'e bakıldığında yetkilendirici yayın bağlı olduğu p_2 yeri ile ilgili sağlanması gereken koşulun p_1 yeri ile ilgili sağlanması gereken koşul ile aynı şekilde belirtildiği görülmektedir. Başka bir deyişle p_2 yerine ilişkin koşul oluşturulurken yetkilendirici yayın varlığı dikkate alınmamıştır. Buradan yola çıkılarak (5.7) ve (5.8)'de geçişler için verilen mantıksal ifadelerin yetkilendirici yaylar için aynen kullanılabileceği sonucuna varılmaktadır.

Tetikleme için gerekli koşullar sağlanmazsa (5.33)'teki mantıksal ifade “0” değerini alır. Bu durumda t_1 geçişi tetiklenmez. Tetikleme koşulları sağlanırsa (5.33)'teki mantıksal ifade “1” değerini alır. Bu durumda ise t_1 geçişi tetiklenir.

- p_1 yerine giren bir geçiş bulunmamaktadır. Bununla birlikte p_1 yerinden çıkan bir t_1 geçişi söz konusudur. t_1 geçişi her tetiklendiğinde p_1 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_1 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_1^+ = p_1 \bar{t}_1 \quad (5.34)$$

- t_1 geçişi tetiklendiğinde yetkilendirici yayın bağlı olduğu p_2 yeri ile ilgili herhangi bir jeton transferi gerçekleştirilmemektedir. Dolayısıyla bu PA'nın PLC kodunda p_2 yerine ilişkin bir mantıksal ifade bulunmayacaktır.
- p_3 yerine giren bir t_1 geçişi bulunmaktadır. t_1 geçişi her tetiklendiğinde p_3 yerine jeton konulacaktır. Bununla birlikte p_3 yerinden çıkan bir geçiş bulunmamaktadır. Bu bilgiler doğrultusunda (5.9) kullanılarak p_3 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_3^+ = t_1 + p_3 \quad (5.35)$$

(5.33)'te geçiş için elde edilen mantıksal ifade ile (5.34) ve (5.35)'te yerler için elde edilen mantıksal ifadeler kullanılarak yetkilendirici yay içeren PA'nın PLC kodu karşılığı Şekil 5.11(b)'de verilmiştir. Bu PLC kodu karşılığına bakıldığında yetkilendirici yayın bağlı olduğu p_2 yerine ilişkin bir mantıksal ifadenin olmadığı görülmektedir. Bunun dışında olması gerektiği gibi çıkış etkisinin giderilmesi için geçişe ilişkin mantıksal ifade yerlere ilişkin mantıksal ifadelerden önce konumlandırılmıştır.

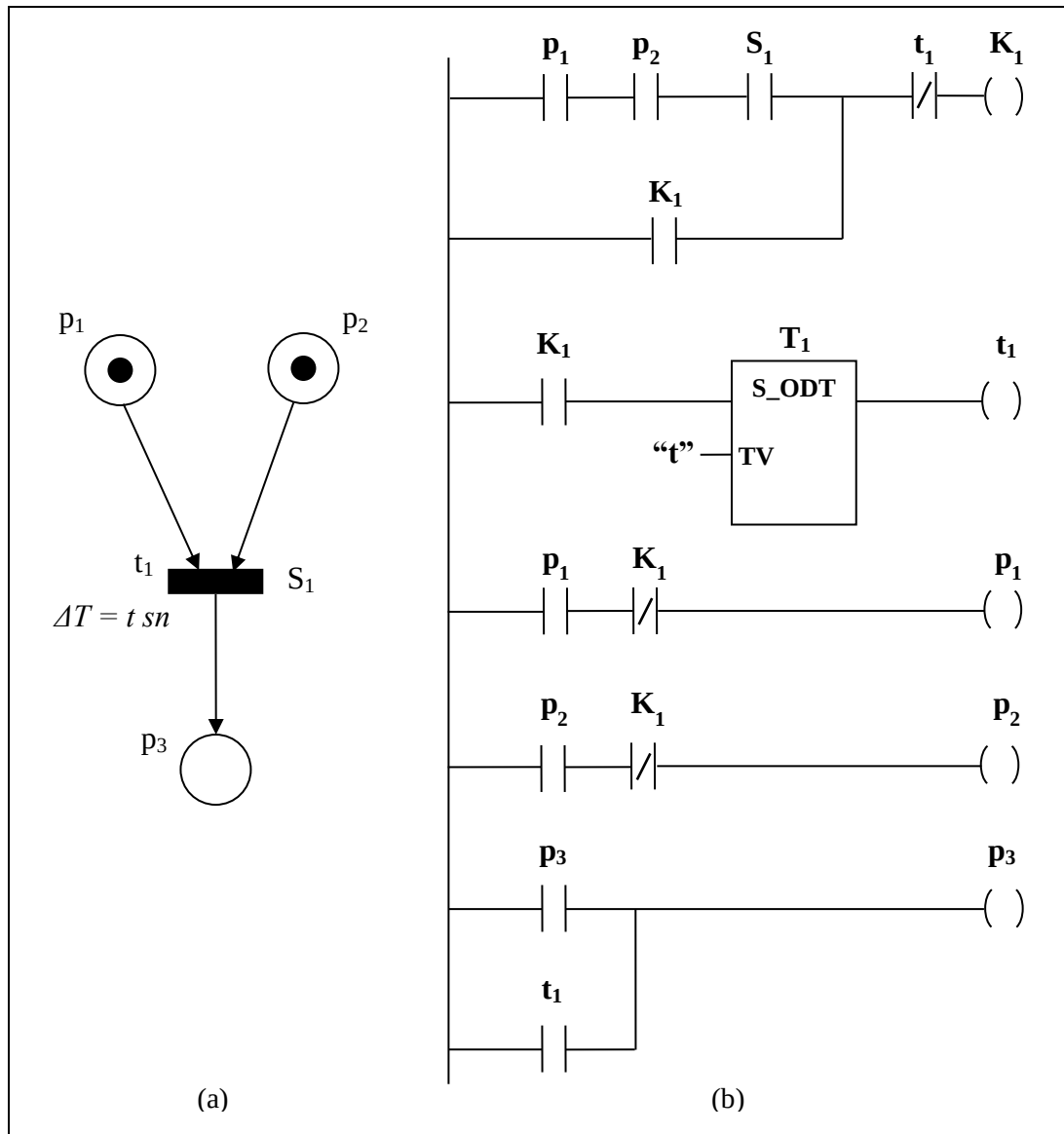
5.4.5 Zaman Etiketli Petri Ağları'nın PLC kodu karşılığı

Ayrık olaylı sistemlerde zamanlama büyük bir önem taşımaktadır. Bu nedenle sistemlerin PA modelinde zamanlamanın da ifade edilmesi gereklilik haline gelmiştir. Zamana bağımlı olayların modellenenebilmesi için “Zaman Etiketli PA” kullanılmaktadır (bkz. “3.2.3 Zaman Etiketli Petri Ağları”). Zaman Etiketli PA’da zamanlamanın ifade edilmesi için ilgili geçişlere zaman etiketleri atanmaktadır. Bu zaman etiketleri kullanılarak zamanlama öncelikli ve olay öncelikli olmak üzere iki farklı davranış modellenenebilmektedir:

1. **Zamanlama Öncelikli:** Bu çalışma şeklinde bir geçişin tetiklenmesine ilişkin koşullar sağlandığında o geçiş hemen tetiklenmez. Koşulların sağlanması ile bu geçişin giriş yerlerindeki jetonlar harcanır ve zamanlayıcı çalıştırılır. Zamanlayıcının süresi dolduğunda geçiş tetiklenir ve çıkış yerlerine jetonlar konulur. Zamanlama öncelikli çalışma şeklinde esas olan zaman etiketi atanmış bir geçişe ilişkin koşulların bir kere sağlanmasının tetikleme işlemi için yeterli olmasıdır. Bu çalışma şeklinde eğer giriş yerleri birden fazla geçişe ait tetikleme koşullarını sağlıyorsa bu geçişler arasında zaman etiketi atanmış olan geçişin tetiklenmesine öncelik verilmektedir.
2. **Olay Öncelikli:** Bu çalışma şeklinde ise bir geçişin tetiklenmesine ilişkin koşullar sağlandığında o geçişin giriş yerlerindeki jetonlar hemen harcanmaz. Koşulların sağlanması ile sadece zamanlayıcı çalıştırılır. Tetikleme koşulları sağlandığı sürece zamanlayıcı çalışmasını sürdürür. Ancak zamanlayıcı çalışırken eğer giriş yerlerindeki jetonlar başka bir geçişin tetiklenmesi için kullanılıp harcanırsa zamanlayıcı devre dışı bırakılır ve zaman etiketi atanmış olan geçişin tetiklenmesine izin verilmez. Söz konusu geçişin tetiklenmesi için zamanlayıcının süresi doluncaya kadar giriş yerlerindeki jetonların başka bir geçiş tarafından harcanmamış olması gerekmektedir. Bu durumda giriş yerlerindeki jetonlar harcanır ve çıkış yerlerine jetonların konulması sağlanır. Bu çalışma şeklinde eğer giriş yerleri birden fazla geçişe ait tetikleme koşullarını sağlıyorsa bu geçişler arasında zaman etiketi atanmamış olan geçişin tetiklenmesine öncelik verilmektedir.

5.4.5.1 Zamanlama öncelikli Petri Ağları

Şekil 5.12(a)'da zamanlama öncelikli PA yapısı verilmiştir. Bu PA yapısında t_1 geçişine atanmış olan bir zaman etiketi bulunmaktadır ($\Delta T = t \text{ sn}$). Bu zaman etiketi ile “zamanlama öncelikli” davranış belirtilmektedir. Bu çalışma şeklinde p_1 ve p_2 yerlerinde jeton bulunması ve S_1 koşulunun sağlanması durumunda t_1 geçişi hemen tetiklenmez. Ancak p_1 ile p_2 yerlerindeki jetonlar harcanır ve T_1 zamanlayıcısı “t” sn. boyunca çalıştırılır. “t” sn’lik süre dolduğunda ise t_1 geçişinin tetiklenmesi sağlanır ve p_3 yerine jeton konulur. Şekil 5.12(b)'de zamanlama öncelikli PA yapısının PLC kodu karşılığı verilmiştir:



Şekil 5.12 : (a) Zamanlama öncelikli PA yapısı. (b) PA'nın PLC kodu karşılığı.

Şekil 5.12(a)'da verilen zamanlama öncelikli PA yapısındaki p_1 , p_2 , p_3 yerleri ve t_1 geçişine ilişkin mantıksal ifadeler aşağıdaki gibi elde edilebilir:

- t_1 geçişinin tetiklenmesi için T_1 zamanlayıcısının “t” sn. kadar çalışması gerekmektedir. T_1 zamanlayıcısının çalışmaya başlaması için de p_1 ile p_2 yerlerinde jeton bulunmalı ve S_1 koşulu sağlanmış olmalıdır. T_1 zamanlayıcısı “t” sn. kadar çalıştıktan sonra t_1 geçişi tetiklenir ve zamanlayıcı durdurulur. T_1 zamanlayıcısının çalıştırılması ve durdurulmasına ilişkin koşul K_1 mantıksal ifadesi kullanılarak aşağıdaki gibi belirtilmiştir:

$$K_1 = (p_1 p_2 S_1 + K_1) \bar{t}_1 \quad (5.36)$$

(5.36)'da verilen K_1 mantıksal ifadesi “1” değerini aldığı anda T_1 zamanlayıcısı çalıştırılır. Bu mantıksal ifade “0” değerini aldığı anda ise T_1 zamanlayıcısı durdurulur.

- p_1 yerindeki jetonun harcanması için T_1 zamanlayıcısının çalışmaya başlaması gerekmektedir. Dolayısıyla K_1 mantıksal ifadesi “1” değerini aldığı anda p_1 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda p_1 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_1^+ = p_1 \bar{K}_1 \quad (5.37)$$

- p_2 yerindeki jetonun harcanması için T_1 zamanlayıcısının çalışmaya başlaması gerekmektedir. Dolayısıyla K_1 mantıksal ifadesi “1” değerini aldığı anda p_2 yerindeki jeton harcanacaktır. Bu bilgiler doğrultusunda p_2 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

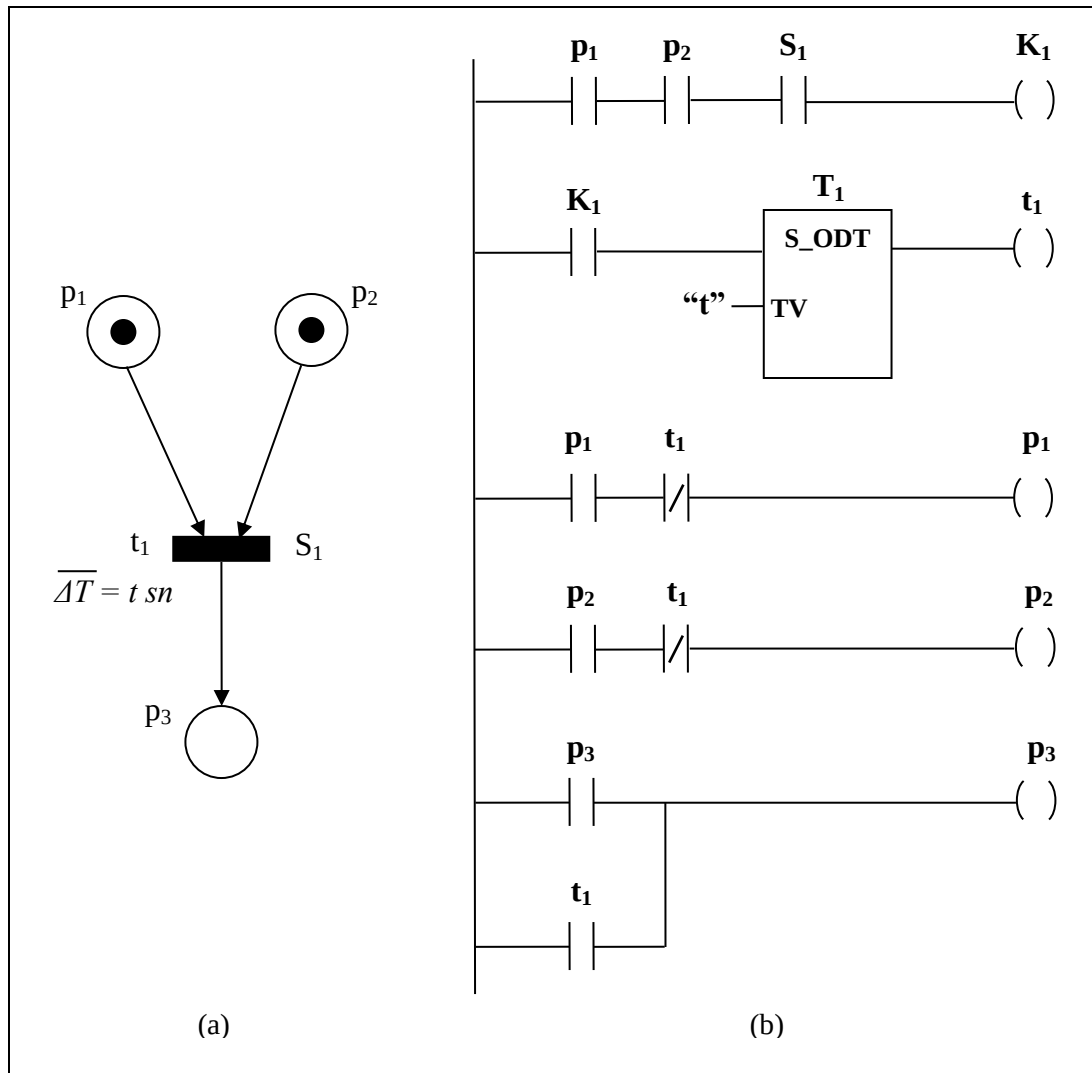
$$p_2^+ = p_2 \bar{K}_1 \quad (5.38)$$

- p_3 yerine jeton konulması için t_1 geçişinin tetiklenmesi gerekmektedir. Bunun için de T_1 zamanlayıcısı “t” sn. kadar çalışmalıdır. “t” sn'lik sürenin sonunda t_1 geçişi tetiklenir ve p_3 yerine jeton konulur. Bu bilgiler doğrultusunda p_3 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_3^+ = t_1 + p_3 \quad (5.39)$$

5.4.5.2 Olay öncelikli Petri Ağları

Şekil 5.13(a)'da olay öncelikli PA yapısı verilmiştir. Bu PA yapısında t_1 geçişine atanmış olan bir zaman etiketi bulunmaktadır ($\overline{\Delta T} = t \text{ sn}$). Bu zaman etiketi ile “olay öncelikli” davranış belirtilmektedir. Bu çalışma şeklinde p_1 ve p_2 yerlerinde jeton bulunması ve S_1 koşulunun sağlanması durumunda t_1 geçişi hemen tetiklenmez ve söz konusu giriş yerlerindeki jetonlar harcanmaz. Bu durumda sadece T_1 zamanlayıcısı “t” sn. boyunca çalıştırılır. “t” sn’lik süre dolmadan giriş yerlerindeki jetonlar başka bir geçişin tetiklenmesi için harcanırsa T_1 zamanlayıcısı durdurulur ve t_1 geçişinin tetiklenmesine izin verilmez. T_1 zamanlayıcısı “t” sn’lik çalışmasını tamamladığında ise t_1 geçişi tetiklenir. Böylece p_1 ve p_2 yerlerindeki jetonlar harcanır ve p_3 yerine jeton konulur. Şekil 5.13(b)'de olay öncelikli PA yapısının PLC kodu karşılığı verilmiştir:



Şekil 5.13 : (a) Olay öncelikli PA yapısı. (b) PA'nın PLC kodu karşılığı.

Şekil 5.13(a)'da verilen olay öncelikli PA yapısındaki p_1 , p_2 , p_3 yerleri ve t_1 geçişine ilişkin mantıksal ifadeler aşağıdaki gibi elde edilebilir:

- t_1 geçişinin tetiklenmesi için T_1 zamanlayıcısının “t” sn. kadar çalışması gerekmektedir. T_1 zamanlayıcısının çalışmaya başlaması için de p_1 ile p_2 yerlerinde jeton bulunmalı ve S_1 koşulu sağlanmış olmalıdır. T_1 zamanlayıcısının çalışmasını “t” sn. boyunca sürdürebilmesi için bu koşullar “t” sn’lik süre boyunca sağlanmalıdır. Aksi takdirde T_1 zamanlayıcısı durdurulur. T_1 zamanlayıcısı “t” sn. kadar çalıştıktan sonra t_1 geçişi tetiklenir. T_1 zamanlayıcısının çalıştırılması ve durdurulmasına ilişkin koşul K_1 mantıksal ifadesi kullanılarak aşağıdaki gibi belirtilmiştir:

$$K_1 = p_1 p_2 S_1 \quad (5.40)$$

(5.40)’ta verilen K_1 mantıksal ifadesi “1” değerini aldığı anda T_1 zamanlayıcısı çalıştırılır. Bu mantıksal ifade “0” değerini aldığı anda ise T_1 zamanlayıcısı durdurulur.

- p_1 yerindeki jetonun harcanması için t_1 geçişinin tetiklenmesi gerekmektedir. Bunun için de T_1 zamanlayıcısı “t” sn. kadar çalışmalıdır. “t” sn’lik sürenin sonunda t_1 geçişi tetiklenir ve p_1 yerindeki jeton harcanır. Bu bilgiler doğrultusunda p_1 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_1^+ = p_1 \bar{t}_1 \quad (5.41)$$

- p_2 yerindeki jetonun harcanması için t_1 geçişinin tetiklenmesi gerekmektedir. Bunun için de T_1 zamanlayıcısı “t” sn. kadar çalışmalıdır. “t” sn’lik sürenin sonunda t_1 geçişi tetiklenir ve p_2 yerindeki jeton harcanır. Bu bilgiler doğrultusunda p_2 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_2^+ = p_2 \bar{t}_1 \quad (5.42)$$

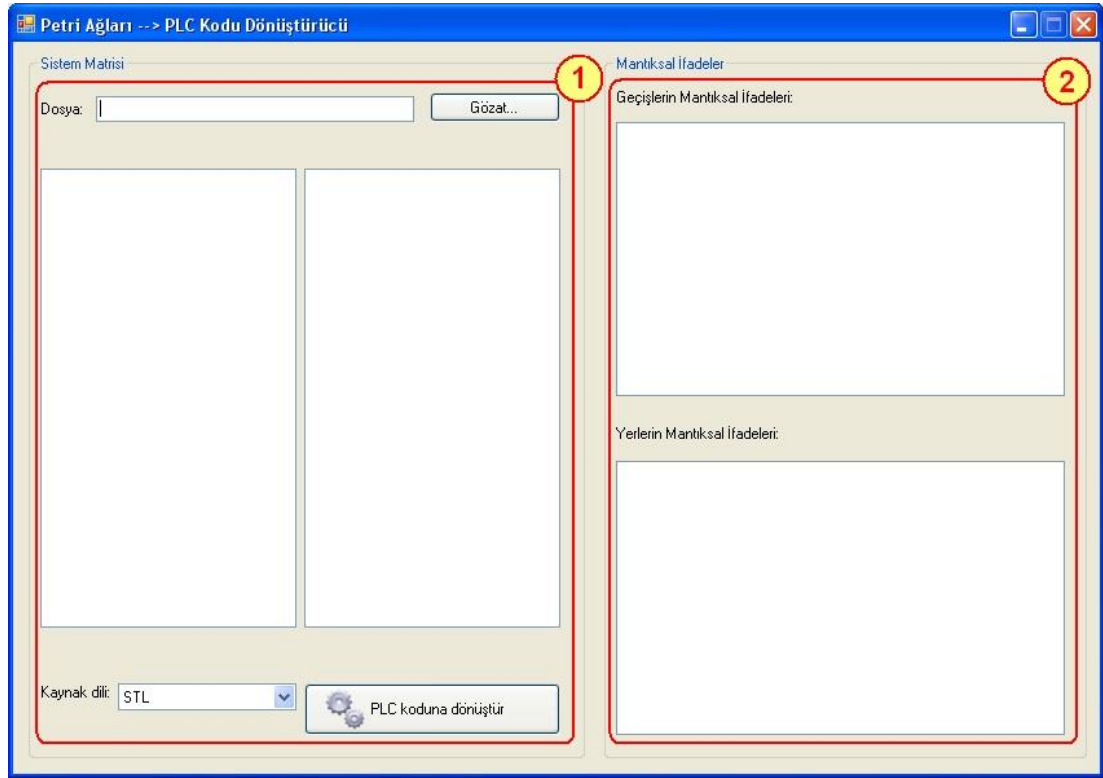
- p_3 yerine jeton konulması için t_1 geçişinin tetiklenmesi gerekmektedir. Bunun için de T_1 zamanlayıcısı “t” sn. kadar çalışmalıdır. “t” sn’lik sürenin sonunda t_1 geçişi tetiklenir ve p_3 yerine jeton konulur. Bu bilgiler doğrultusunda p_3 yerine ilişkin mantıksal ifade aşağıdaki gibi elde edilir:

$$p_3^+ = t_1 + p_3 \quad (5.43)$$

5.5 Petri Ağlarından PLC Koduna Otomatik Dönüşüm Yapan Yazılım

5.5.1 Yazılıma genel bakış

“5.3 Petri Ağlarının Sistem Matrisinden PLC Koduna Geçiş” bölümünde önerdiğimiz yöntemin gerçekleşmesi amacıyla bu tez kapsamında konsept bir yazılım geliştirilmiştir. Bu yazılım kullanılarak sistem matrisi verilen bir PA’nın PLC kodu “STL” veya “SCL” kaynak dilinde otomatik olarak elde edilebilmektedir. Yazılımın ana penceresi Şekil 5.14’te verilmiştir:



Şekil 5.14 : Yazılımın ana penceresi.

Görüldüğü gibi yazılımın ana penceresi iki ana bölümden oluşmaktadır. Bu bölümler açıklamalarıyla birlikte aşağıda verilmiştir:

1. **Sistem Matrisi:** Bu bölümde kullanıcı PLC koduna dönüştürmek istediği PA’ya ilişkin sistem matrisinin bulunduğu dosyayı seçer. Bu dosya sistem matrisinin satırlarından ve geçişlerin tetiklenme koşullarından oluşmaktadır. Dosyada tetiklenme koşullarının işaretin “inen” ya da “çıkan” kenarında etkin olacağı ayrıca belirtilir. Örnek vermek gerekirse bu dosyanın her bir satırında sistem matrisinin ilgili satırı, bu satıra ilişkin geçişin tetiklenme koşulu ve bu koşulun işaretin hangi kenarında etkin olacağı bilgisi bulunmaktadır.

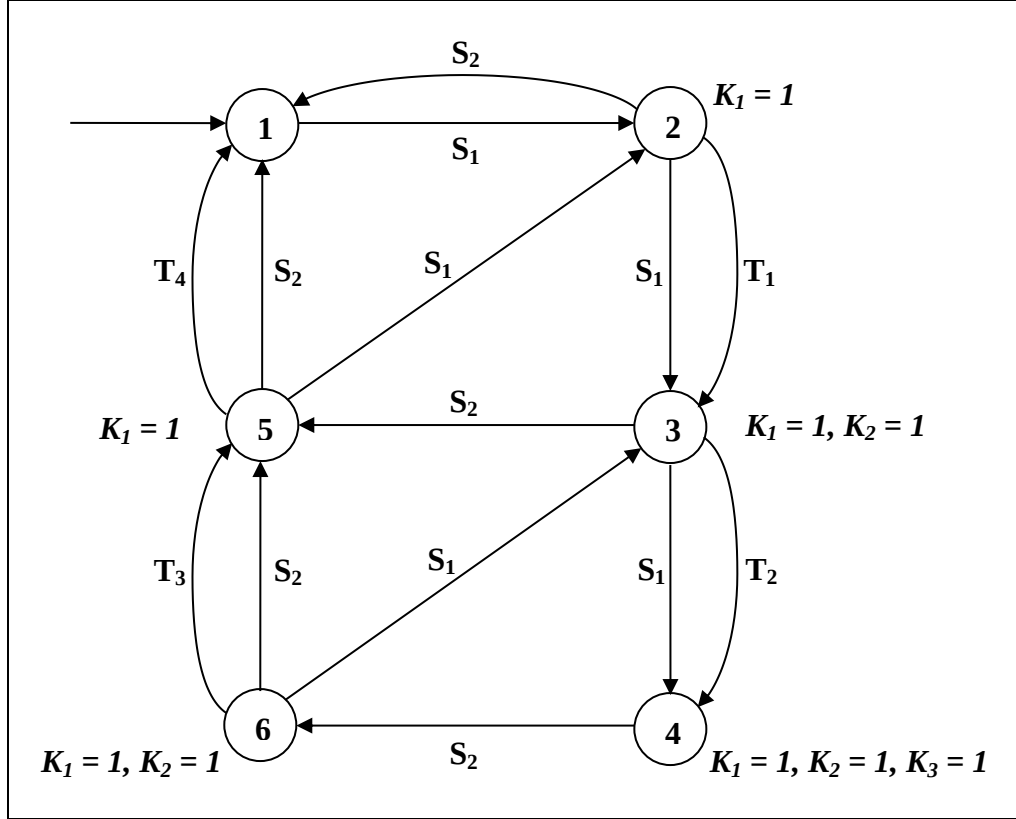
2. **Mantıksal İfadeler:** Sistem matrisinin ve tetiklenme koşullarının bulunduğu dosya seçildikten sonra yazılım PLC koduna dönüşüm işlemi için hazır hale gelir. Bunun için kalan son adım istenen PLC kodunun hangi kaynak dilinde üretileceğinin belirtilmesidir. Bu amaçla bir önceki “Sistem matrisi” bölümünde “Kaynak dili” alanında istenen dil “STL” veya “SCL” olarak seçilmelidir. Ardından PLC koduna dönüşüm işleminin başlatılması için “PLC koduna dönüştür” butonuna tıklanır. Bu butona tıklandıktan sonra yazılım tarafından sistem matrisi verilen PA’nın PLC koduna dönüşümü gerçekleştirilir ve aynı zamanda üretilen PLC kodu için sembol tablosu oluşturulur. Kullanıcının incelemesi açısından üretilen PLC kodu ayrı bir “notepad” penceresinde otomatik olarak açılır. Bununla birlikte PLC kodundaki geçişler ve yerlere ait mantıksal ifadeler sırasıyla “Geçişlerin Mantıksal İfadeleri” ve “Yerlerin Mantıksal İfadeleri” alanlarında da görüntülenir. Böylelikle üretilen PLC kodunun incelenmesinde kullanıcıya kolaylık sağlanır.

Yazılım tarafından üretilen PLC kodu PA’yı tam ve doğru bir şekilde gerçekleştirecek işlevselliğe sahiptir. Bu işlevselliğin test edilmesi amacıyla uygun bir benzetim yazılımı (örneğin “SIMATIC Manager”) kullanılarak yeni bir proje oluşturulur. Daha sonra üretilen kodun kaynak diline göre (STL veya SCL) bu proje altında içi boş bir kaynak kodu oluşturulur. Bu kaynak kodu açılır ve kodun içine yazılım tarafından üretilen PLC kodu kopyalanıp yapıştırılır. Ardından yazılım tarafından oluşturulan sembol tablosu da projeye dâhil edilir ve kaynak kodu derlenir. Derleme işleminin hatasız olarak tamamlandığından emin olunduktan sonra yeni bir fonksiyon bloğu elde edilmiş olur. Bundan sonra yeni bir ana program bloğu oluşturulur ve elde edilen fonksiyon bloğu bu ana program bloğunun içine gömülür. Bu fonksiyon bloğunun giriş ve çıkışları doğru bir şekilde ayarlandıktan sonra ana program bloğu tamamlanmış olur. Nihayetinde ana program bloğunun benzetimi yapılarak beklenen davranışı gösterip göstermediği test edilir.

Yazılım tarafından üretilen PLC kodu ile PA’nın sadece durum geçiş mekanizması gerçekleşmektedir. Başlangıç durumunda yapılması gereken ilklendirmeler, durum geçişlerinde gerçekleştirilmesi gereken işlemler ve diğer hususlar kullanıcı tarafından dikkate alınıp üretilen koda uygun bir şekilde dâhil edilmelidir. Yazılım bu yönüyle kullanıcının PA’yı PLC koduna dönüştürmesi için bir ana çatı sunmaktadır.

5.5.2 Örnek sistemin tanımı

Ele alınan örnek endüstriyel otomasyon sistemi en genel anlamda 3 tane motordan (K_1 , K_2 , K_3) oluşmaktadır. Sistemin S_1 ve S_2 olmak üzere iki tane girişi bulunmaktadır. S_1 tuşu ile sistemdeki motorlar devreye alınıp çalıştırılmakta, S_2 tuşu ile de motorlar devre dışı bırakılıp durdurulmaktadır. Sistemin otomat diyagramı Şekil 5.15'te verilmiştir:



Şekil 5.15 : Örnek sistemin otomat diyagramı.

Şekil 5.15'te otomat diyagramı verilen sistemin çalışma şekli aşağıda maddeler halinde verilmiştir. Her bir maddenin sıra numarası sistemin bulunduğu durumun numarasını ifade etmektedir:

1. Sistem başlangıç durumundadır ve herhangi bir motor çalışmamaktadır. Bu durumda S_1 tuşuna basıldığında sistem 2 no'lu duruma geçer ve K_1 motoru devreye alınır. Bununla birlikte T_1 zamanlayıcısı çalıştırılır.
2. Bu durumda sadece K_1 motoru çalışmaktadır. Ayrıca T_1 zamanlayıcısı da devrededir. Burada iki türlü durum oluşabilir:

- a. Kullanıcı S_2 tuşuna basarsa K_1 motoru devre dışı bırakılır, T_1 zamanlayıcısı iptal edilir ve sistem 1 no'lu başlangıç durumuna tekrar geri döner.
 - b. S_1 tuşuna basıldığında veya T_1 zamanlayıcısının süresi dolduğunda sistem 3 no'lu duruma geçer ve K_2 motoru devreye alınır. Ayrıca T_2 zamanlayıcısı çalıştırılır.
3. Bu durumda hem K_1 hem de K_2 motoru çalışmaktadır. Ayrıca T_2 zamanlayıcısı da devrededir. Burada iki türlü durum oluşabilir:
 - a. Kullanıcı S_2 tuşuna basarsa K_2 motoru devre dışı bırakılır ve T_2 zamanlayıcısı iptal edilir. Sistem 5 no'lu duruma geçer ve T_4 zamanlayıcısı çalıştırılır.
 - b. S_1 tuşuna basıldığında veya T_2 zamanlayıcısının süresi dolduğunda sistem 4 no'lu duruma geçer ve K_3 motoru devreye alınır.
4. Bu durumda tüm motorlar (K_1 , K_2 ve K_3) çalışmaktadır. Kullanıcı S_2 tuşuna basarsa K_3 motoru devre dışı bırakılır. Sistem 6 no'lu duruma geçer ve T_3 zamanlayıcısı çalıştırılır.
5. Bu durumda sadece K_1 motoru çalışmaktadır. Ayrıca T_4 zamanlayıcısı da devrededir. Burada iki türlü durum oluşabilir:
 - a. Kullanıcı S_2 tuşuna basarsa veya T_4 zamanlayıcısının süresi dolarsa K_1 motoru devre dışı bırakılır ve sistem 1 no'lu başlangıç durumuna geçer.
 - b. S_1 tuşuna basıldığında T_4 zamanlayıcısı iptal edilir ve sistem 2 no'lu duruma geçer.
6. Bu durumda hem K_1 hem de K_2 motoru çalışmaktadır. Ayrıca T_3 zamanlayıcısı da devrededir. Burada iki türlü durum oluşabilir:
 - a. Kullanıcı S_2 tuşuna basarsa veya T_3 zamanlayıcısının süresi dolarsa K_2 motoru devre dışı bırakılır ve sistem 5 no'lu duruma geçer.
 - b. S_1 tuşuna basıldığında T_3 zamanlayıcısı iptal edilir ve sistem 3 no'lu duruma geçer.

5.5.3 Sistemin Petri Ağları modelinin oluşturulması

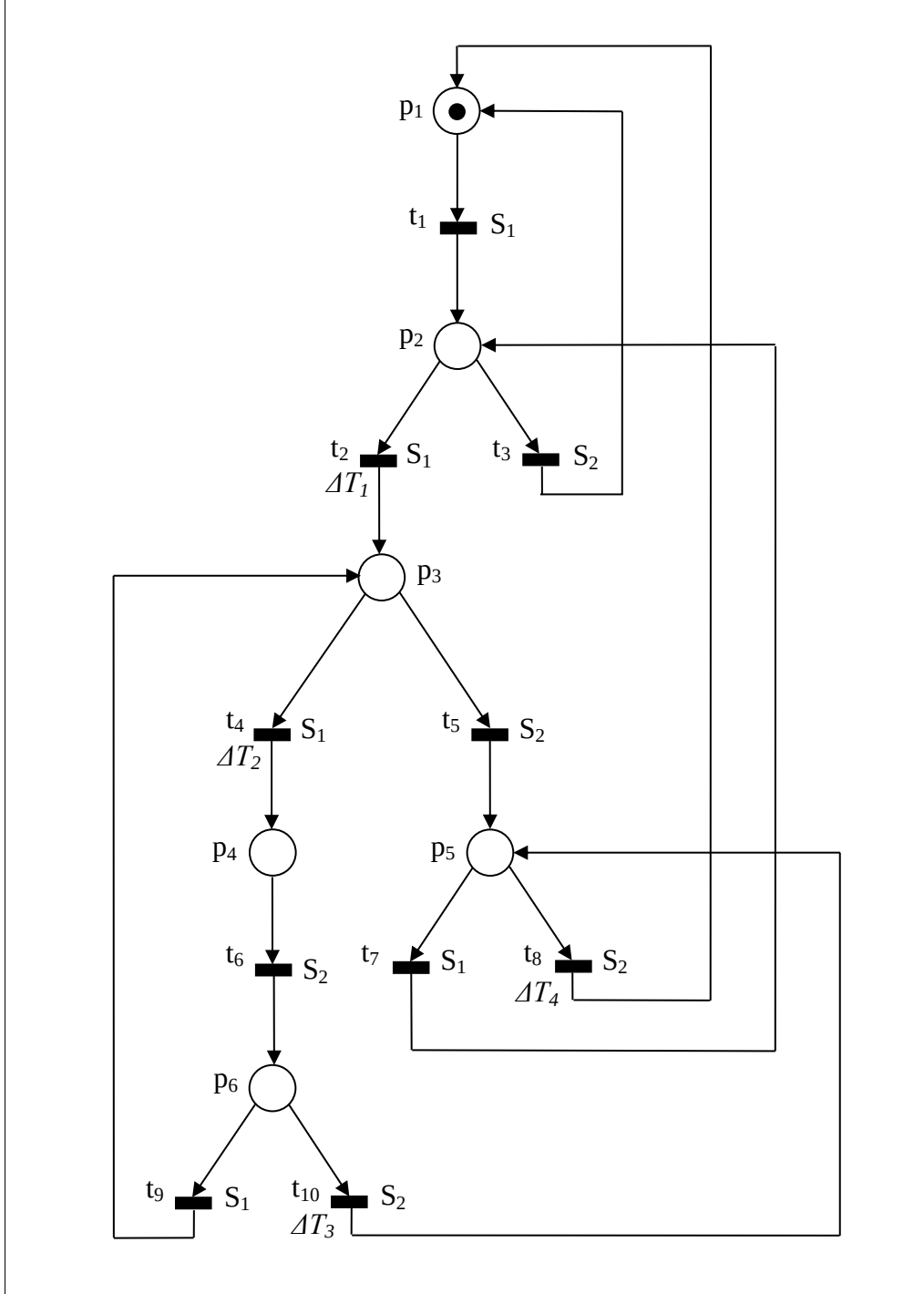
Örnek endüstriyel otomasyon sisteminin PA modelinin elde edilmesi için yerler, geçişler ve bu geçişlerin tetiklenmesi için gerekli koşullar Çizelge 5.1'deki gibi belirlenmiştir:

Çizelge 5.1 : Örnek sisteme ait PA'daki yerler ve geçişler.

Yer	Açıklama	Geçiş	Koşul	Çıkış Yeri
p ₁	Başlangıç	t ₁	S ₁	p ₂
p ₂	K ₁ motoru devreye alınır ve T ₁	t ₂	S ₁ + T ₁	p ₃
	zamanlayıcısı çalıştırılır.	t ₃	S ₂	p ₁
p ₃	K ₂ motoru devreye alınır ve T ₂	t ₄	S ₁ + T ₂	p ₄
	zamanlayıcısı çalıştırılır.	t ₅	S ₂	p ₅
p ₄	K ₃ motoru devreye alınır.	t ₆	S ₂	p ₆
p ₅	K ₂ motoru devre dışı bırakılır ve	t ₇	S ₁	p ₂
	T ₄ zamanlayıcısı çalıştırılır.	t ₈	S ₂ + T ₄	p ₁
p ₆	K ₃ motoru devre dışı bırakılır ve	t ₉	S ₁	p ₃
	T ₃ zamanlayıcısı çalıştırılır.	t ₁₀	S ₂ + T ₃	p ₅

Çizelge 5.1'de örnek sistemin PA modelindeki yerler, yerlerin açıklamaları, yerlerin bağlı olduğu geçişlerin tetiklenme koşulları ve bu koşullar sağlandığı zaman geçiş yapılacak çıkış yerleri gösterilmektedir. Sistem herhangi bir durumda iken gerekli koşullar sağlandığında ilgili giriş yerindeki jeton harcanır ve çıkış yerine jeton konulur.

Çizelge 5.1'den faydalanılarak sistemin PA modeli elde edilebilir. Şekil 5.16'da sistemin PA modeli verilmiştir. Dikkat edilecek olursa sistem ilk çalıştırıldığında başlangıç durumunda olacağından p₁ yerinde jeton bulunmaktadır. Bu durumda S₁ tuşuna basıldığında t₁ geçişi tetiklenir ve p₁ yerindeki jeton harcanıp p₂ yerine jeton konulur. p₂ yerinde jeton varken T₁ zamanlayıcısının süresi dolduğunda veya S₁ tuşuna basıldığında t₂ geçişi tetiklenerek p₂ yerindeki jeton harcanır ve p₃ yerine jeton konulur. Eğer p₂ yerinde jeton varken T₁ zamanlayıcısının süresi dolmadan S₂ tuşuna basılırsa bu sefer t₃ geçişi tetiklenir ve p₂ yerindeki jeton harcanıp p₁ yerine jeton konulur. Bu durum geçiş mekanizması PA'daki diğer yerler için aynı şekilde Çizelge 5.1'deki gibi gerçekleştirilir.



Şekil 5.16 : Örnek sistemin PA modeli.

5.5.4 Petri Ağlarının PLC koduna otomatik dönüşümü

Şekil 5.16’da verilen örnek sisteme ait PA’nın PLC koduna dönüştürülmesi için geliştirmiş olduğumuz yazılım kullanılacaktır (bkz. “5.5.1 Yazılıma genel bakış”). Bu amaçla örnek sistemi tanımlayan dosyanın yazılımın ihtiyaç duyduğu şekilde oluşturulması gerekmektedir. Bilindiği gibi bu dosya örnek sisteme ait PA’nın sistem

matrisinden ve geçişlerin tetiklenme koşullarından oluşmaktadır. PA'nın sistem matrisi (5.44)'teki gibi elde edilir:

$$D = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad (5.44)$$

PA'nın sistem matrisi elde edildikten sonra geçişlerin tetiklenme koşulları da belirlenmelidir. Şekil 5.16'da verilen PA'ya bakıldığında geçişlerin tetiklenme koşulları açıkça görülebilmektedir. Dikkat edilecek olursa bazı koşullar zamanlamaya bağlıdır. Basitliğin sağlanması açısından zamanlamaya bağlı "T₁", "T₂", "T₃" ve "T₄" koşulları göz ardı edilecek ve sadece motorların devreye alınmasını sağlayan "S₁" koşulu ile motorların durdurulmasını sağlayan "S₂" koşulu dikkate alınacaktır. Bununla birlikte bu koşulların etkin olması için işaretin "çıkan" kenarı kullanılacaktır. Bu bilgiler ışığında PLC kodunun üretilmesi için yazılım tarafından kullanılacak dosya Şekil 5.17'deki gibi elde edilir:

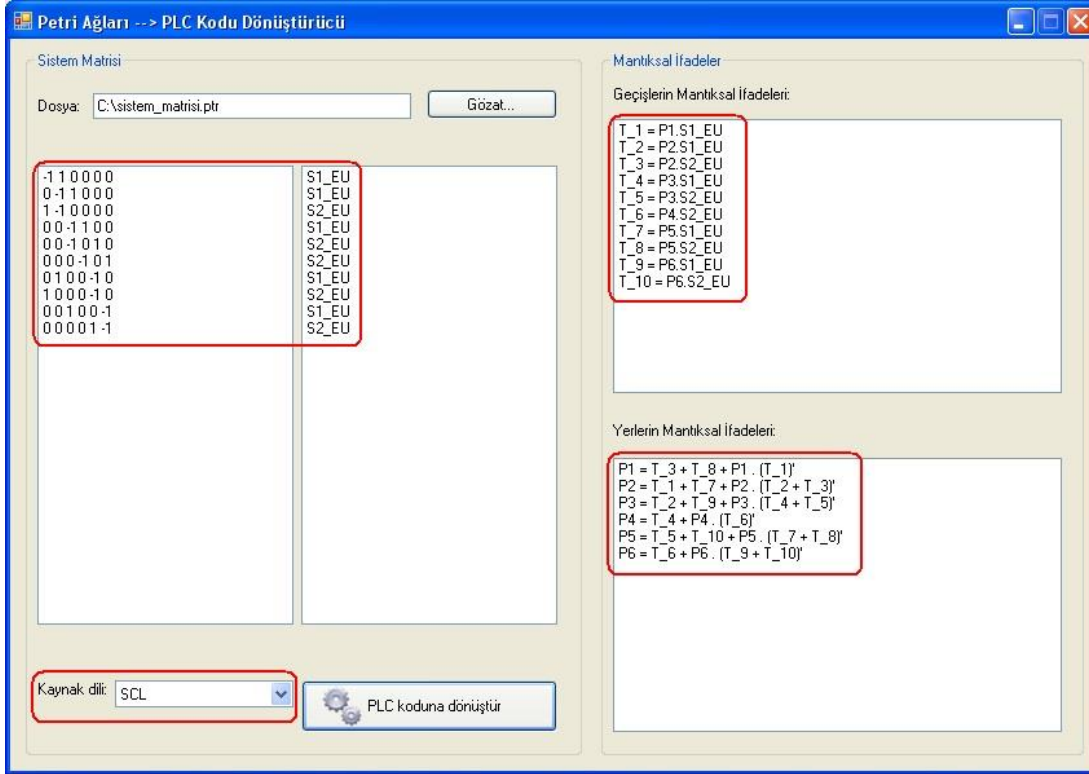
```

sistem_matrisi.ptr - Notepad
File Edit Format View Help
-1 1 0 0 0 0, S1, EU
0 -1 1 0 0 0, S1, EU
1 -1 0 0 0 0, S2, EU
0 0 -1 1 0 0, S1, EU
0 0 -1 0 1 0, S2, EU
0 0 0 -1 0 1, S2, EU
0 1 0 0 -1 0, S1, EU
1 0 0 0 -1 0, S2, EU
0 0 1 0 0 -1, S1, EU
0 0 0 0 1 -1, S2, EU

```

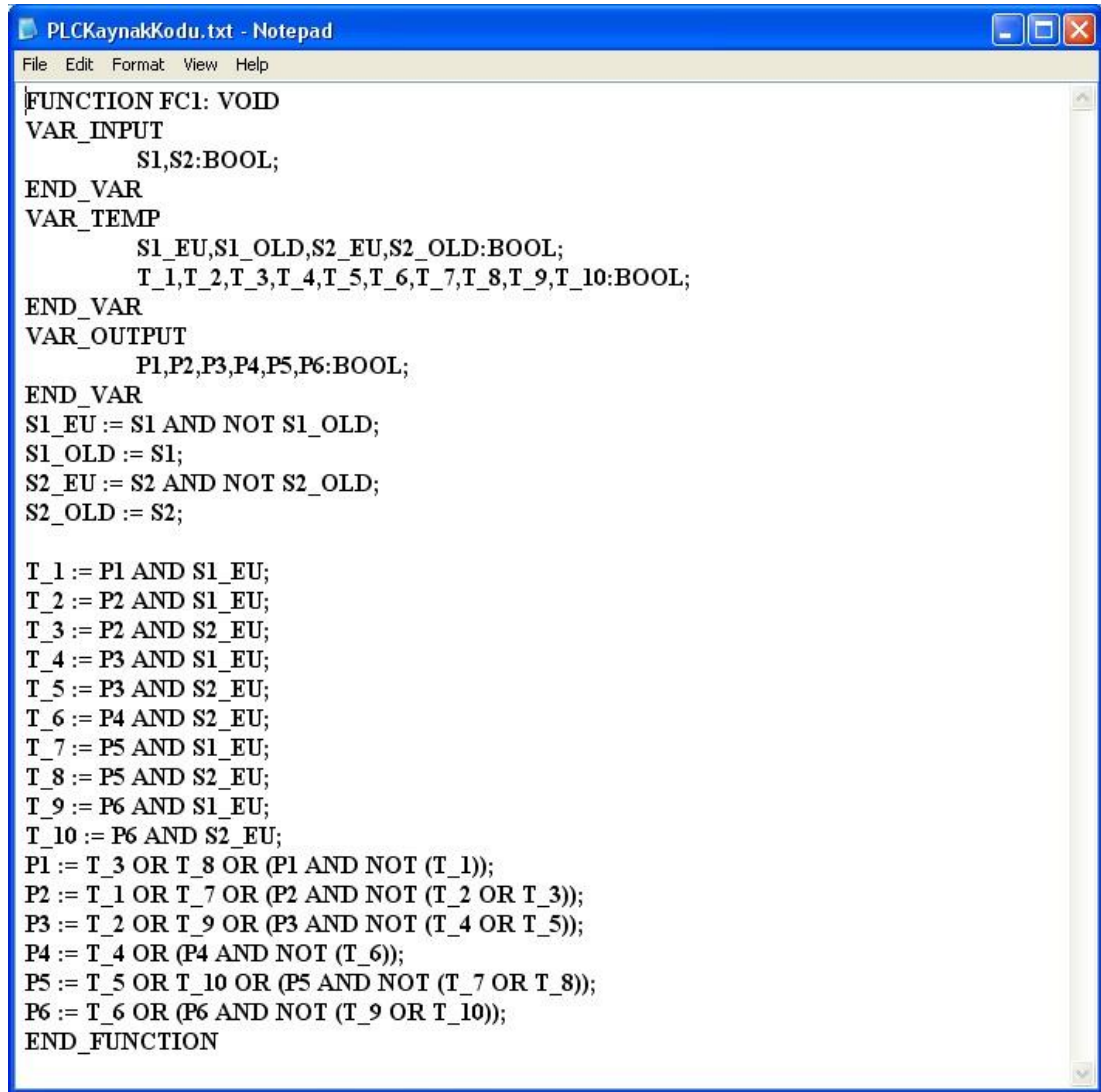
Şekil 5.17 : PLC kodunun üretilmesi için kullanılacak dosya.

Dosya elde edildikten sonra PA'nın PLC koduna dönüştürülmesi için yazılım hazır hale gelir. Bunun için öncelikle yazılımın ana penceresinde “Gözet...” butonuna tıklanarak dosya seçilir. Ardından üretilecek PLC kodunun kaynak dili “STL” veya “SCL” olarak seçilir ve “PLC koduna dönüştür” butonuna tıklanır. Bu butona tıklandıktan sonra örnek sisteme ait PA'nın PLC koduna dönüşümü yazılım tarafından gerçekleştirilir (Şekil 5.18):



Şekil 5.18 : Örnek sisteme ait PA'nın PLC koduna dönüştürülmesi.

Şekil 5.18’de görüldüğü gibi PA’daki geçişler ve yerlere ilişkin mantıksal ifadeler de yazılımın ana penceresinde kullanıcıya gösterilmektedir. Böylelikle yazılım tarafından üretilen PLC kodunun incelenmesinde kullanıcıya kolaylık sağlanmaktadır. Şekil 5.19’da örnek sistemin yazılım tarafından “SCL” kaynak dilinde üretilen PLC kodu verilmiştir:



```
FUNCTION FC1: VOID
VAR_INPUT
    S1,S2:BOOL;
END_VAR
VAR_TEMP
    S1_EU,S1_OLD,S2_EU,S2_OLD:BOOL;
    T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8,T_9,T_10:BOOL;
END_VAR
VAR_OUTPUT
    P1,P2,P3,P4,P5,P6:BOOL;
END_VAR
S1_EU := S1 AND NOT S1_OLD;
S1_OLD := S1;
S2_EU := S2 AND NOT S2_OLD;
S2_OLD := S2;

T_1 := P1 AND S1_EU;
T_2 := P2 AND S1_EU;
T_3 := P2 AND S2_EU;
T_4 := P3 AND S1_EU;
T_5 := P3 AND S2_EU;
T_6 := P4 AND S2_EU;
T_7 := P5 AND S1_EU;
T_8 := P5 AND S2_EU;
T_9 := P6 AND S1_EU;
T_10 := P6 AND S2_EU;
P1 := T_3 OR T_8 OR (P1 AND NOT (T_1));
P2 := T_1 OR T_7 OR (P2 AND NOT (T_2 OR T_3));
P3 := T_2 OR T_9 OR (P3 AND NOT (T_4 OR T_5));
P4 := T_4 OR (P4 AND NOT (T_6));
P5 := T_5 OR T_10 OR (P5 AND NOT (T_7 OR T_8));
P6 := T_6 OR (P6 AND NOT (T_9 OR T_10));
END_FUNCTION
```

Şekil 5.19 : Örnek sistemin yazılım tarafından üretilen PLC kodu.

6 DEMİRYOLU SİNYALİZASYON SİSTEMİ

Günümüzde ulaşım, insanların en önemli sorunlarından biri hâline gelmiştir. Özellikle büyük kentlerde yaşayan insanlar bu sorunla daha sık karşılaşmaktadırlar. Hızlı kentleşme, yoğun nüfus artışı, hava kirliliği ve enerji sıkıntısı gibi büyük sorunlar, ulaşımında raylı sisteme geçişi zorunlu hale getirmiştir [24].

Raylı sistem taşımacılığının yatırım maliyetleri yüksek olmakla birlikte işletme maliyetleri karayolu taşımacılığına göre daha düşüktür. Ayrıca kaza riskleri, enerji tüketimi, trafik sıkışıklığı ve personel istihdamı kara yolu taşımacılığına göre daha azdır. Bununla birlikte raylı sistemlerin taşıma kapasitesi, karayolu taşımacılığına göre çok daha yüksektir [24].

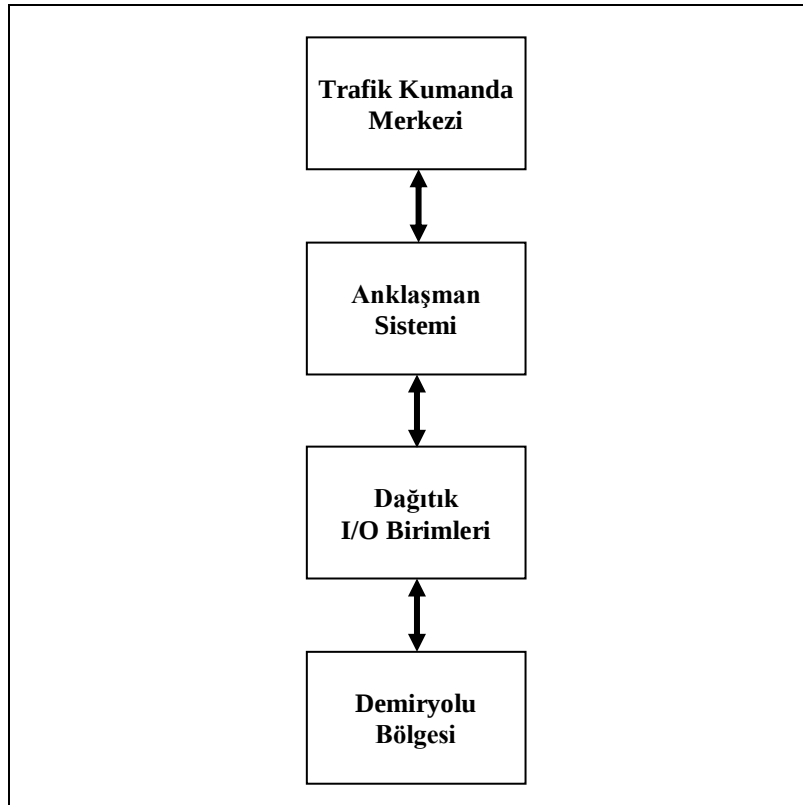
İlk kullanılan demiryolu sistemlerinde tren hareketlerinin kontrolü belirli bölgelerde ve ellerinde bulunan kırmızı renkli bayrak ile makinistleri uyaran demiryolu bekçileri tarafından sağlanmaktaydı. Demiryollarındaki trafiğin artması ile birlikte demiryolu bekçilerinden veya makinistlerden kaynaklanan ya da mekanik sistemlerden kaynaklanan birçok kaza meydana gelmiştir. Tüm bunların sonucu olarak ilk anlaşılan sistemi 1843 yılında Bricklayers Arms, İngiltere’de kurulmuştur [25]. Günümüzde trenlerin ulaştığı hızlar ve tren yoğunluğu göz önüne alındığında ise bazen çok küçük bir hatanın ölümcül kazalara [26] sebep olabildiği görülmektedir. Bu nedenle demiryollarındaki yük ve yolcu taşımacılığının güvenliğini sağlamak amacıyla güvenilir sinyalizasyon ve anlaşılan sistemlerine ihtiyaç duyulmaktadır [27].

Demiryolu sinyalizasyon sistemleri, raylı ulaşımın emniyetli ve planlanan zamanda gerçekleştirilmesi ve sistem emniyetinin sağlanması açısından büyük önem taşımaktadır. Sinyalizasyon sistemlerinin birincil amacı trenler arasındaki kazaların önlenmesidir. Gerek dünyada gerekse ülkemizde meydana gelen kazalar sinyalizasyon sistemlerinin önemini açıkça göstermektedir.

Demiryolu sinyalizasyon sistemi trafik kumanda merkezi, saha elemanları ve anlaşılan sistemi olmak üzere üç ana kısımdan oluşmaktadır.

6.1 Trafik Kumanda Merkezi

Trafiğin tek bir merkezden kontrol edildiği sistemlerde TKM kendi yetkisi altında bulunan tüm demiryolu trafiğinden sorumlu olan birimdir. Güzergâh tanzimleri, tren konumları ve bilgileri, makas konumları, sinyal durumları bu merkeze gelmekte ve burada gerekli işlemler gerçekleştirilmektedir. Güzergâh tanzim talepleri TKM’de bulunan dispeçer tarafından gerçekleştirilmektedir. TKM ile anklâşman sistemi arasında çift yönlü bir veri alışverişi mevcuttur. TKM ile yapılan komutlar anklâşman sistemiyle sahaya aktarılırken aynı şekilde sahadan gelen uyarılar da anklâşman sistemiyle TKM’ye aktarılmaktadır. Anklâşman yazılımına iletilen tanzim talepleri sahanın o andaki durumu ile karşılaştırılarak kabul edilmekte veya ret edilmektedir. Anklâşman yazılımı sadece demiryolu trafiğini tehlikeye düşürmeyecek taleplere izin vermektedir. Bu nedenle sinyalizasyon sisteminin en önemli kısmını oluşturmaktadır. TKM’nin çalışma yapısı Şekil 6.1’de gösterilmiştir [28]:



Şekil 6.1 : Sinyalizasyon sistemi blok diyagramı.

Dispeçer tarafından yapılan güzergâh tanzim talebi anklâşman yazılımı tarafından kabul edildikten sonra bu kabul bilgisi TKM’ye bildirilmektedir. TKM’den gelen onaydan sonra güzergâh tanzimi ile ilgili komutlar sahaya gönderilmekte ve saha

elemanları elektronik olarak kilitlenmektedir. Tren kendisi için tanzim edilen güzergâhta ilerledikçe geçmiş olduğu ray devreleri serbest hale gelmekte ve bir sonraki tanzim için uygun hale gelmektedir [28].

6.2 Saha Elemanları

Saha elemanları, demiryolu sahasının altyapısını oluşturan elemanlardır. Anklaşman sistemi saha elemanlarından bilgi alır ve bunları kumanda eder. Saha elemanları makaslar, sinyaller ve ray devreleri olmak üzere üç ana başlık altında incelenebilir.

6.2.1 Makaslar

Trenlerin demiryolu üzerinde yön değiştirmesi makaslar tarafından gerçekleştirilmektedir. Makaslar mekanik elemanlar olup makas motorları üzerinden kumanda edilir. Makas motorları kalkış anında yüksek akım çektikleri için aynı güç kaynağına bağlı bulunan makasların aynı anda çalıştırılmaya başlanması tercih edilmeyen bir durumdur. Bu nedenle aynı güç kaynağına bağlı makaslar belli bir sırada çalıştırılır ve böylece motorun aşırı akım çekerek yanması engellenir [29].

Her makasın “normal” ve “ters” olmak üzere iki farklı konumu bulunmaktadır. Trenin makas üzerinden yön değiştirmeden gittiği yön “normal” konum, yön değiştirerek gittiği yön ise “ters” konum olarak adlandırılır [30]. Makasların anlık konum bilgisi, sahadaki sensörler aracılığıyla anklaşman sistemine bildirilmektedir. Böylelikle herhangi bir anda makasların konumları takip edilebilmektedir [31]. Makasların konum bilgisi için iki adet mantıksal çıkış kullanılmaktadır. Bu mantıksal çıkışların anlamları Çizelge 6.1’de verilmiştir:

Çizelge 6.1 : Makasların konum bilgisini belirleyen mantıksal çıkışlar.

M_1	M_2	Anlamı
0	0	İndikasyonsuzluk hatası
1	0	Makas “normal” konumda
0	1	Makas “ters” konumda
1	1	Veri bağdaşım hatası

Hatasız çalışma durumunda, makas “normal” veya “ters” konumda olacaktır. Çizelge 6.1’den yola çıkılarak makasın konumlandırılmasında karşılaşılabilecek olası hata durumları şu şekilde özetlenebilir:

- **İndikasyonsuzluk Hatası:** Makas konum değiştirirken doğal olarak bir süre “normal” ve “ters” konum arasında kalmaktadır. Ancak makas konum değiştirirken eğer 7 sn. içinde istenen konuma gelmezse “indikasyonsuzluk hatası” oluşur. Bu hata durumunda, M_1 ve M_2 çıkışları 7 sn’den daha uzun sürede mantıksal “0” değerinde kalmaktadır.
- **Veri Bağdaşım Hatası:** Hem M_1 hem de M_2 çıkışları mantıksal “1” değerini aldığı anda gerçekleşen hata durumudur. Makasın, herhangi bir anda, hem “normal” hem de “ters” konumda bulunması mümkün olmadığından, çıkışların her ikisinin de aynı anda mantıksal “1” değerini alması anormal bir durumdur. Normal durumda, her iki çıkıştan aynı anda sadece birinin mantıksal “1” değerini alması beklenmektedir.
- **Makasın Hareketsiz Kalması:** Makas, konum değiştirme isteğine cevap vermeyip 7 sn’den daha uzun süre eski konumunda kalıyorsa bu hata durumu oluşur. Bu hatanın, “indikasyonsuzluk hatası”ndan farkı, makasın hiçbir şekilde hareket etmemesi ve arada kalma durumunun yaşanmamasıdır.

6.2.2 Sinyaller

Sinyaller anlaşılan sistemi vasıtasıyla makiniste bilgi veren ve makinistin uymakla yükümlü olduğu ışıkların genel adıdır. Sinyaller tren gidiş güzergâhının sağında bulunur ve makiniste aşağıdaki konularda bilgi verir [31]:

- Öndeki bloğun boş veya dolu olduğu
- Makinistin ilerlemeye izni olup olmadığı
- Sapma olup olmadığı
- Trenin hangi hızda seyretmesi gerektiği
- Bir sonraki sinyalin durumu
- İstasyona olan uzaklık
- İstasyondaki seyir hızı

Eski zamanlarda mekanik olarak çalışan sistemler günümüzde yerini elektronik sinyallere bırakmıştır. Sinyaller görüşlerine göre dörtlü yüksek, üçlü yüksek ve üçlü cüce olmak üzere 3’e ayrılabilir. Makaslarda olduğu gibi sinyaller de hem

anklaşman sistemi tarafından kumanda edilen hem de ankaşman sistemine veri gönderen saha elemanlarıdır. Sinyaller ankaşman sistemiyle makinist arasındaki ara yüzü oluşturması sebebiyle sinyalizasyon sisteminin güvenliği açısından en önemli elemanlardır.

6.2.2.1 Dörtlü yüksek sinyaller

Bu tür sinyaller genellikle istasyon giriş çıkışlarında ve makas bölgelerine yaklaşırken kullanılır. Bu sinyallerde renk sıralaması üstten başlayarak “Sarı-Yeşil-Kırmızı-Sarı” şeklindedir. En alttaki sarı lamba sapma olup olmadığının bilgisini taşır. En alttaki sarı lamba yanıyorsa tren bir sonraki makas bölgesinde sapma yapacaktır.

Bir dörtlü yüksek sinyalde; sarı, yeşil, sarı üzeri sarı, sarı üzeri yeşil ve sarı üzeri kırmızı açma yetkisi olmak üzere toplam 5 adet çıkış bulunur. Aynı şekilde sarı bildirim, yeşil bildirim, kırmızı bildirim, sarı üzeri sarı bildirim, sarı üzeri yeşil bildirim ve sarı üzeri kırmızı bildirim indikasyonları olmak üzere toplam 6 adet giriş bulunur (Çizelge 6.2):

Çizelge 6.2 : Dörtlü yüksek sinyal renk bildirim açıklamaları.

Renk Bildirimi	Açıklama
	Bir öndeki blok dolu veya hat tanzimi yapılmamış (Dur)
	Bir öndeki blok boş, ondan sonraki dolu (Kontrollü ilerle)
	Öndeki en az iki blok boş (İlerle)
	Bir sonraki makas bölgesinden sapma yapılacak, ancak bir öndeki blok dolu (Her an durabilecek şekilde ilerle)
	Bir sonraki makas bölgesinden sapma yapılacak ve bir öndeki blok boş (Kontrollü ilerle)
	Bir sonraki makas bölgesinden sapma yapılacak ve öndeki en az iki blok boş (İlerle)

6.2.2.2 Üçlü yüksek sinyaller

Bu tür sinyaller genellikle istasyondaki ana hat üzerinde kullanılır. Renk sıralaması yukarıdan aşağıya “Sarı-Yeşil-Kırmızı” şeklindedir. Bu sinyallerde dörtlü sinyallerden farklı olarak herhangi bir sapma bilgisi bulunmamaktadır.

Bir üçlü yüksek sinyalde; sarı ve yeşil açma yetkisi olmak üzere toplam 2 adet çıkış bulunur. Aynı şekilde sarı bildirim, yeşil bildirim, kırmızı bildirim indikasyonları olmak üzere toplam 3 adet giriş bulunur (Çizelge 6.3):

Çizelge 6.3 : Üçlü yüksek sinyal renk bildirim açıklamaları.

Renk Bildirimi	Açıklama
	Bir öndeki blok dolu veya hat tanzimi yapılmamış (Dur)
	Bir öndeki blok boş, ondan sonraki dolu (Kontrollü ilerle)
	Öndeki en az iki blok boş (İlerle)

6.2.2.3 Üçlü cüce sinyaller

Bu tür sinyaller barınma yollarının çıkışlarına ve istasyonların yan yollarına yerleştirilir. Renk sıralaması yukarıdan aşağıya “Kırmızı-Yeşil-Sarı” şeklindedir. Sinyalin yanıp sönmesi (flaş yapması) ilgili sinyalin, sinyalsiz yoldaki tren için açıldığını gösterir.

Bir üçlü cüce sinyalde; sarı, yeşil, flaş sarı, flaş yeşil, sarı üzeri kırmızı ve flaş sarı üzeri kırmızı açma yetkisi olmak üzere toplam 6 adet çıkış bulunur. Aynı şekilde sarı bildirim, yeşil bildirim, kırmızı bildirim, flaş sarı bildirim, flaş yeşil bildirim, flaş sarı üzeri kırmızı bildirim ve sarı üzeri kırmızı bildirim indikasyonları olmak üzere toplam 7 adet giriş bulunur (Çizelge 6.4):

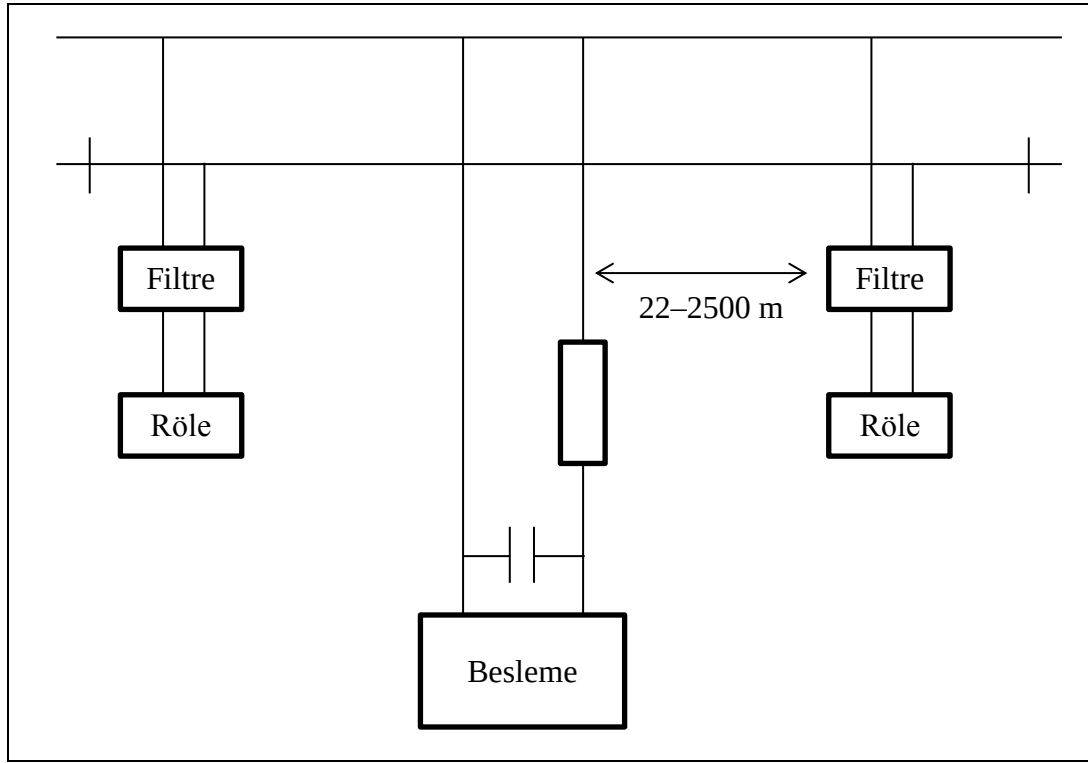
Çizelge 6.4 : Üçlü cüce sinyal renk bildirim açıklamaları.

Renk Bildirimi	Açıklama
	Bir sonraki makas bölgesinden sapma yapılacak, ancak bir öndeki blok dolu veya hat tanzimi yapılmamış (Dur)
	Bir sonraki makas bölgesinden sapma yapılacak ve bir öndeki blok boş (Kontrollü ilerle)
	Bir sonraki makas bölgesinden sapma yapılacak ve öndeki en az iki blok boş (İlerle)
	Bir sonraki makas bölgesinden sapma yapılacak, ancak bir öndeki blok dolu (Her an durabilecek şekilde ilerle)
	Bir sonraki makas bölgesinden sapma yapılacak ve bir öndeki blok boş (Kontrollü ilerle)
	Bir sonraki makas bölgesinden sapma yapılacak ve öndeki en az iki blok boş (İlerle)
	Bir sonraki makas bölgesinden sapma yapılacak, ancak bir öndeki blok dolu (Her an durabilecek şekilde ilerle)

6.2.3 Ray devreleri

Ray devreleri, bir ray bloğunun üzerinde tren olup olmadığının bilgisini vermek üzere tasarlanmış, bu bilgiyi anlaşılan sistemine ileten elektronik donanımlardır. Ray devreleri, AC ve DC ray devreleri olmak üzere ikiye ayrılır. DC ray devreleri, TCDD'nin istasyonlarda kullandığı ana ray devresi türüdür [32]. Çalışma prensibi ray bloğuna 1,5 – 12V doğru gerilim verilmesine dayanır. Ray bloğu üzerinden tren geçerken bir kısa devre oluşur. Bu durum rayla bağlı olan rölenin düşmesine neden olarak ray bloğunun üzerinde tren olduğunun anlaşılmasını sağlar (Şekil 6.2). Ray devrelerinin başlıca fonksiyonları aşağıdaki gibidir [33]:

- Trenin gideceği güzergâhı belirler.
- Makaslar bölgesinde hareket hâlinde veya durmakta olan trenleri belirler.
- Trenlerin hatlarda paralel gidişlerini sağlar.
- Trenlerin geçişlerini yapabilmesi için güzergâhın emniyetli ve serbest kalmasını sağlar.
- Önde bulunan trenin durumunu belirler ve izlenmesini sağlar.



Şekil 6.2 : DC ray devresi.

Ray devreleri, ilgili ray bloğunun üzerinde tren bulunmadığı durumda anlaşılan sistemine mantıksal “1” sinyalini, tren bulunması durumunda ise rayların elektriksel olarak tren üzerinden kısa devre olmasından dolayı anlaşılan sistemine mantıksal “0” sinyalini gönderir. Bu çalışma mekanizması ile kablo kopmalarına karşı da bir güvenlik sağlanmış olur.

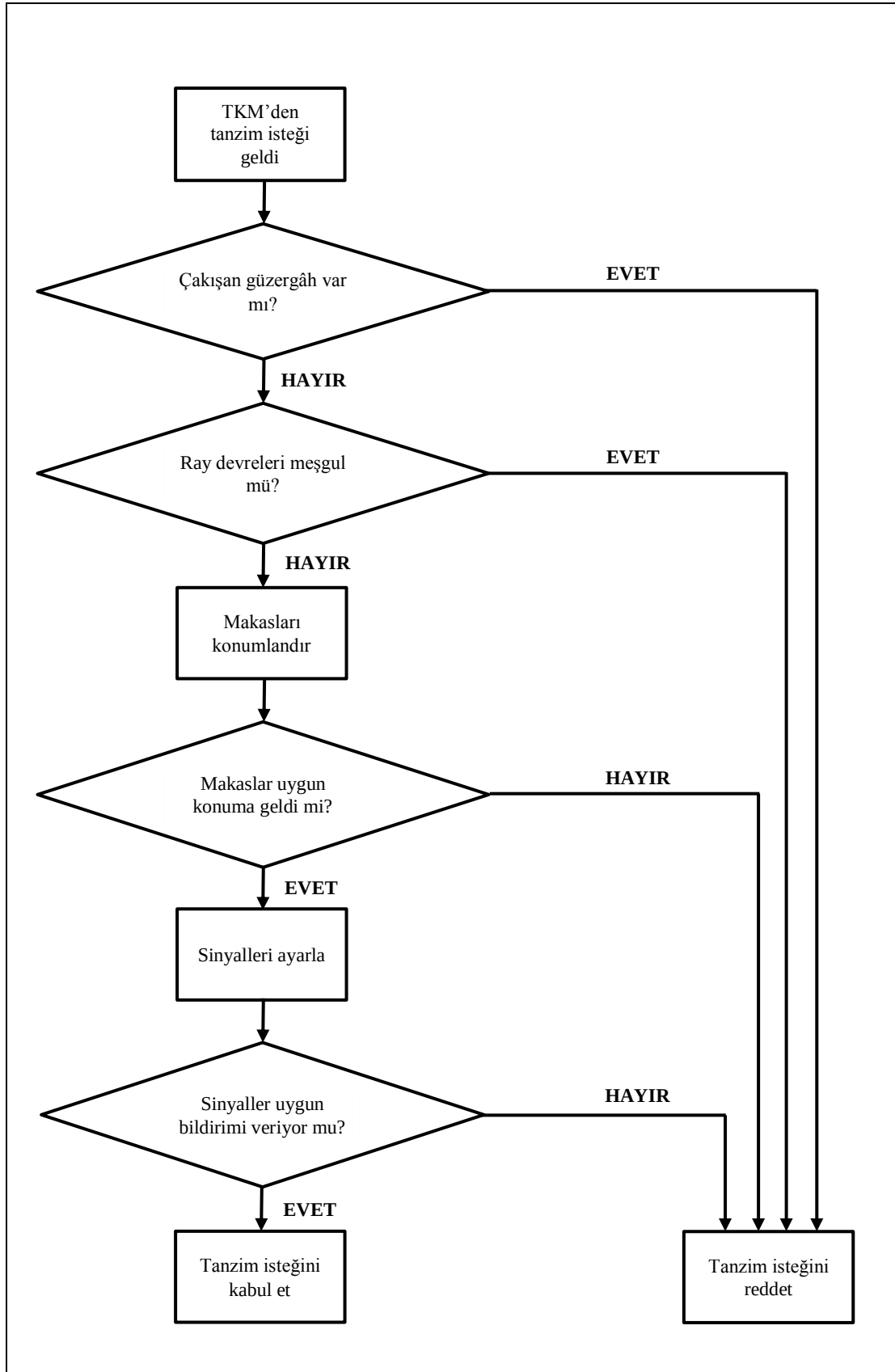
6.3 Anlaşılan Sistemi

Demiryolu sinyalizasyon sistemlerinde temel karar verme mekanizması anlaşılan sistemidir. Anlaşılan sistemlerinin esas görevi, TKM’den gelen hat tanzim isteklerinin değerlendirilip uygulanmasıdır. Değerlendirme esnasında mevcut hat tanzim istekleri ile sinyal ve makas gibi saha elemanlarının durumları göz önünde bulundurulur. Bu değerlendirme sonucunda uygun taleplerin karşılanması için ilgili saha elemanları belirli durumlara getirilir ve kilitlenir; uygun bulunmayan istekler ise reddedilir. Açıkça görülebileceği gibi bir sinyalizasyon sisteminin kazaya sebebiyet vermemesi, karar alıcı anlaşılan sistemi ile doğrudan alakalıdır. Bu nedenle tren yolu sinyalizasyonu için CENELEC tarafından EN50126, EN50128 ve EN50129 güvenlik standartları belirlenmiştir. Karar verme mekanizması olarak anlaşılan sistemi de hem donanımsal hem de yazılımsal olarak bu standartları sağlamalıdır.

Yazılım standartlarının sağlanması için ise tasarım aşamasında kullanılacak yöntemler büyük önem taşımaktadır. Gereksinimlerin sağlanabilmesi için tasarımın formal yöntemler kullanılarak gerçekleştirilmesi esastır [34].

Günümüzde Türkiye’de sinyalizasyonu yapılmış olan hatlarda röleli anlaşıman sistemleri kullanılmaktadır. Karar verme mekanizması olarak bazı bölgelerde sadece rölelerle oluşturulan lojik devreler kullanıldığı gibi bazı bölgelerde rölelerle birlikte özel elektronik devreler de kullanılmaktadır [35]. Bununla birlikte bu yazılımlar PLC ile de gerçekleştirilebilmektedir. İster röleler isterse yazılımsal olarak gerçekleştirilmiş olsun anlaşıman sistemleri anlaşıman tabloları üzerinden değerlendirilir. Anlaşıman tabloları, ilgili bölge için gelebilecek güzergâh tanzim isteklerinin hangi durumlarda kabul edilmesi gerektiğini belirleyen tablolardır. Herhangi bir güzergâh tanziminin güvenli bir şekilde yapılabilmesi için anlaşıman sistemi tarafından aşağıdaki prosedürler uygulanır (Şekil 6.3):

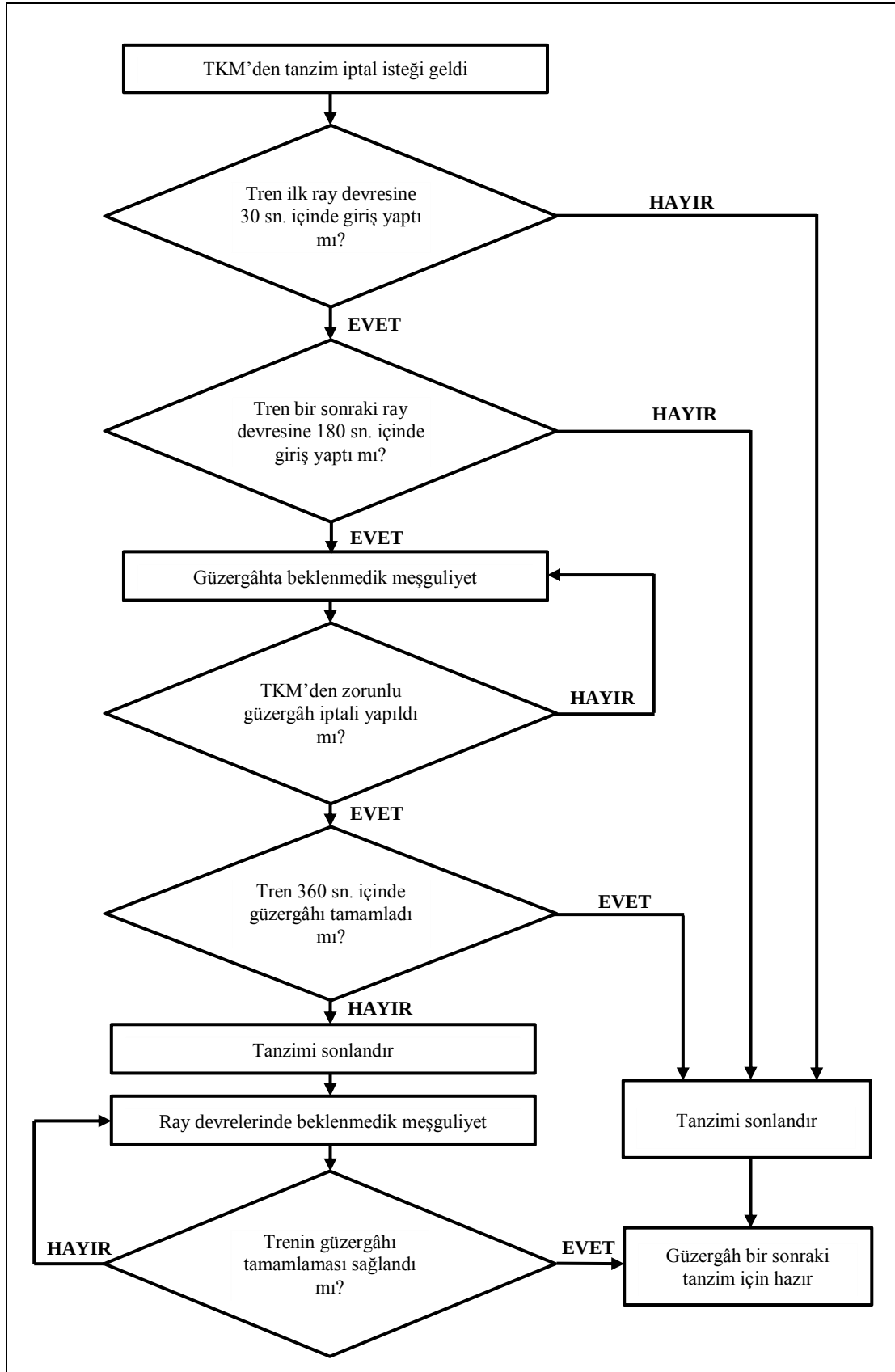
- Tanzim isteği yapılan güzergâh ile çakışan ve aktif olarak kullanılmakta olan güzergâh(lar) kontrol edilir. Eğer çakışan bir güzergâh var ise tanzim isteği reddedilir.
- Çakışan başka bir güzergâh yok ise, tanzim isteği yapılan güzergâha ait ray devrelerinin üzerinde tren olup olmadığı kontrol edilir.
- Ray devreleri de boş ise elektronik olarak kilitlenir ve makasları istenen konumlarına kilitlemek için komut gönderilir. Eğer makaslar 7 sn. içinde istenen konumlarına gelmezse veya hata durumuna düşerse ray devrelerinin kilitleri çözülür ve tanzim isteği reddedilir.
- Makaslar istenen konumlarına gelirse elektronik olarak kilitlenir ve tanzim isteği yapılan güzergâh için tüm sinyaller doğru bildirim verecek şekilde ayarlanır. Sinyaller doğru olarak ayarlanamazsa ray devreleri ile makasların kilitleri çözülür ve tanzim isteği reddedilir.
- Sinyaller doğru olarak kilitlenirse tanzim isteği kabul edilir. Güzergâh üzerinde tren bir ray devresinden diğer ray devresine doğru ilerlerken sinyaller de buna uygun olarak ayarlanır. Tren tanzim edilen güzergâhtaki son ray devresine geldiğinde tanzim sonlandırılır ve ray devreleri ile makasların kilitleri çözülür. Böylelikle ilgili güzergâh bir sonraki tanzim için hazır hale getirilmiş olur.



Şekil 6.3 : Tanzim isteğinin değerlendirilmesine ilişkin akış diyagramı.

TKM tarafından tanzim isteđi yapılabildiđi gibi tanzim iptali de gerekleřtirilebilir. Kurulu olan bir guzergâh iin TKM’den iptal isteđi gonderildiđinde anklâşman sistemi tarafından ařađıdaki prosedrler uygulanır (řekil 6.4):

- İlgili guzergâha ait ilk ray devresinin hemen nndeki sinyal kırmızı bildirim verecek řekilde ayarlanır.
- Eđer bu guzergâhtaki ilk ray devresine 30 sn. boyunca tren giriř yapmazsa ray devreleri ile makasların kilitleri zlr ve tanzim sonlandırılır.
- Guzergâhtaki ilk ray devresine 30 sn. iinde tren girerse guzergâh kurulu bırakılır ve 180 sn. beklemede kalınır.
- Eđer 180 sn. boyunca tren ilk ray devresinde kalıp bir sonraki ray devresine giriř yapmazsa ray devreleri ile makasların kilitleri zlr ve tanzim sonlandırılır.
- Tren 180 sn. iinde ilk ray devresinden ıkıp bir sonraki ray devresine giriř yaparsa “guzergâhta beklenmedik meřguliyet” hatası oluřur. Bu hatanın giderilebilmesi iin TKM tarafından “zorunlu guzergâh iptal” isteđi yapılmalıdır. Bu istek yapılana kadar anklâşman sistemi tarafından guzergâh kurulu bırakılır.
- TKM tarafından “zorunlu guzergâh iptali” yapıldıđında 360 sn. beklenir ve bu sre boyunca guzergâh kurulu olarak bırakılmaya devam edilir.
- Eđer tren 360 sn. dolmadan guzergâhı tamamlarsa ray devreleri ile makasların kilitleri zlr ve tanzim sonlandırılır.
- Tren 360 sn. iinde guzergâhı tamamlamazsa ray devreleri ile makasların kilitleri zlr ve tanzim sonlandırılır. Bu durumda trenin artık kurulu olmayan bir guzergâha ait ray devreleri zerinde bulunması nedeniyle “ray devrelerinde beklenmedik meřguliyet” hatası oluřur. Bu hatanın giderilebilmesi iin trenin tamamen son ray devresine getirilerek guzergâhı tamamlaması sađlanmalıdır.
- “Ray devrelerinde beklenmedik meřguliyet” hatası giderildiđinde guzergâh bir sonraki tanzim iin hazır hale getirilmiř olur.

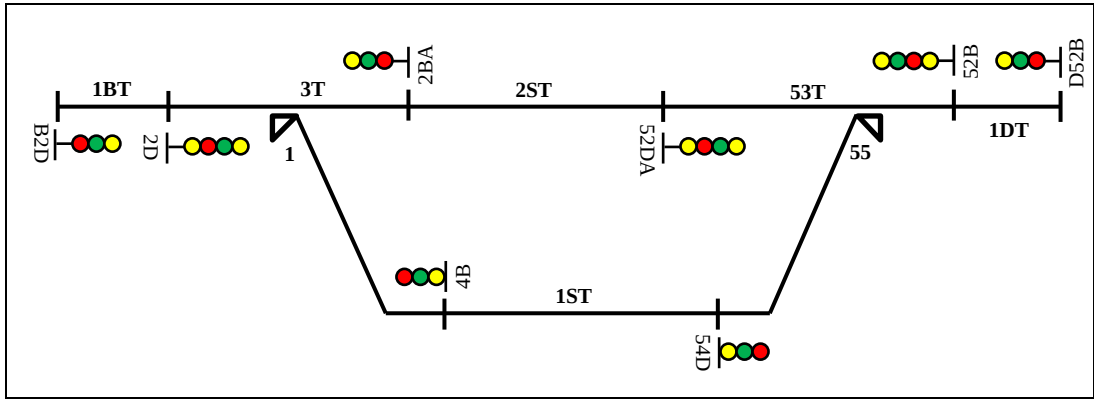


Şekil 6.4 : Tanzim iptal isteğinin değerlendirilmesine ilişkin akış diyagramı.

6.3.1 Örnek anlaşılan tablosu

Demiryolu sinyalizasyon sistemini oluşturan saha elemanlarının ve anlaşılan sisteminin daha iyi anlaşılması bakımından Şekil 6.5'te örnek bir demiryolu bölgesi verilmiştir. Bu demiryolu bölgesi aşağıdaki saha elemanlarından oluşmaktadır:

- “1BT”, “3T”, “1ST”, “2ST”, “53T” ve “1DT” no’lu 6 adet ray devresi,
- “1” ve “55” no’lu 2 adet makas,
- “2D”, “52DA” ve “52B” no’lu 3 adet dörtlü yüksek sinyal,
- “B2D”, “D52B” ve “2BA” no’lu 3 adet üçlü yüksek sinyal ve
- “54D” ve “4B” no’lu 2 adet üçlü cüce sinyal.



Şekil 6.5 : Örnek demiryolu bölgesi.

Şekil 6.5'te verilen örnek demiryolu bölgesindeki olası güzergâhlar için oluşturulan anlaşılan tablosu Çizelge 6.5'te verilmiştir:

Çizelge 6.5 : Örnek anlaşılan tablosu.

Rota	Sinyal No	Sinyal Durumu	Makas Konumu	Ray Devresi
001BT-1ST	2D	SY	54D S	1(T)
		SS	54D K	3T, 1ST
		Y	52DA S	1(N)
001BT-2ST	4B	S	52DA K	3T, 2ST
		S	—	1(T)
		S	—	1(N)
1ST-001BT	2BA	SY	4B S	55(T)
		SS	4B K	53T, 1ST
		Y	2BA S	55(N)
001DT-1ST	52B	S	2BA K	53T, 2ST
		S	—	55(T)
		S	—	55(N)
1ST-001DT	54D	S	—	53T, 001DT
2ST-001DT	52DA	S	—	53T, 001DT

Çizelge 6.5'teki anlaşıman tablosunun her bir satırında o satırdaki güzergâhın tanzimi için sinyallerin vermesi gereken bildirim, makasların kilitlenmesi gereken konum ve kontrol edilmesi gereken ray devreleri ile ilgili bilgiler bulunmaktadır. Aslında anlaşıman tablosu bir güzergâha ait tanzim isteğinin kabul edilmesi için saha elemanları tarafından sağlanması gereken koşulları içeren bir tablodur. Örneğin “001BT-1ST” güzergâhı için TKM'den tanzim isteği geldiğini varsayalım. Bu tanzim isteğinin kabul edilmesi için aşağıdaki koşulların sağlanması gerekir:

1. “001BT-1ST” güzergâhı ile çakışan önceden tanzim edilmiş başka bir güzergâh bulunmamalıdır. Anlaşıman tablosuna bakıldığında bu güzergâh ile çakışan güzergâhların “001BT-2ST”, “1ST-001BT”, “2ST-001BT” olduğu görülmektedir.
2. “001BT-1ST” güzergâhına ait ray devrelerinin üzerinde tren bulunmamalıdır. Anlaşıman tablosunda bu güzergâhın bulunduğu satırdaki “Ray Devresi” sütununa bakıldığında söz konusu ray devrelerinin “3T” ve “1ST” olduğu görülmektedir. Buradan “3T” ve “1ST” ray devrelerinin boş olması gerektiği sonucuna varılmaktadır.
3. Trenin “001BT-1ST” güzergâhı üzerinde ilerleyebilmesi için ilgili makas(lar) uygun konuma kilitlenmelidir. Anlaşıman tablosunda ilgili satırdaki “Makas Konumu” sütununa bakıldığında “1(T)” ifadesinin olduğu görülmektedir. Burada “1” kilitlenecek makasın numarasını belirtmektedir. Parantez içindeki harf ise makasın hangi konuma kilitlenmesi gerektiğini ifade eder. Başka bir deyişle, “T” makasın “ters” konuma, “N” ise makasın “normal” konuma kilitlenmesi gerektiğini belirtir. Bu bilgilerden yola çıkılarak “1” no'lu makasın “ters” konuma kilitlenmesi gerektiği sonucuna varılmaktadır.
4. “001BT-1ST” güzergâhı için “2D” sinyali uygun bildirim verecek şekilde ayarlanmalıdır. Bu amaçla güzergâh üzerinde bu sinyalden bir sonraki sinyal olan “54D” sinyalinin verdiği bildirim dikkate alınır. “54D” sinyali “sarı” bildirim veriyorsa “2D” sinyali “sarı üzeri yeşil”, “54D” sinyali “kırmızı” bildirim veriyorsa “2D” sinyali “sarı üzeri sarı” bildirim verecek şekilde ayarlanmalıdır. Bu bilgi anlaşıman tablosunda ilgili satırdaki “Sinyal No” ve “Sinyal Durumu” sütunlarından elde edilebilir.

7 UYGULAMA

Bu bölümde gerçek bir tren istasyonuna ait anlaşıman sisteminin PA ile modellenerek PLC ile gereklenmesi saėlanmıřtır. Bu amala Adapazarı Mithatpařa tren istasyonu ele alınmıřtır. Öncelikle bu tren istasyonu iin anlaşıman tablosu elde edilmiřtir. Anlaşıman sisteminin modellenmesi iin modüler bir yaklařım uygulanmıř ve anlaşıman sistemi bir takım alt sistemlere bölünmüřtür. Bunlar tanzim, makas, tren takip, cüce sinyal ve yüksek sinyal alt sistemleri olarak tanımlanmıřtır. Tanzim alt sistemi ile tanzim istekleri deėerlendirilir ve tanzim isteėi kabul edildiėinde ray devrelerinin kilitlenmesi saėlanır. Makas alt sistemi ile makasların uygun konuma ayarlanması, kilitlenmesi, kilidinin özölmesi ve hatalı duruma getirilmesi saėlanır. Tren takip alt sistemi ile trenlerin ray devreleri üzerinde ilerlerken takibinin yapılması, tanzimin iptal edilmesi, tanzimin sonlandırılması ve ray devrelerinin kilidinin özölmesi işlemleri gerekleřtirilir. Cüce sinyal alt sistemi ile cüce sinyallerin, yüksek sinyal alt sistemi ile de yüksek sinyallerin uygun bildirim vermesi saėlanır. Bu alt sistemlerin her biri iin ayrı ayrı PA modeli oluřturulmuř ve “5.3 Petri Aėlarının Sistem Matrisinden PLC Koduna Geiř” bölümünde önerilen yöntem kullanılarak her bir PA modelinin PLC koduna dönüşümü gerekleřtirilmiřtir. Böylelikle Mithatpařa tren istasyonuna ait anlaşıman sisteminin PLC ile gereklenmesi saėlanmıřtır. Son olarak gereklenen anlaşıman sisteminin test edilmesi amacıyla bir SCADA uygulaması tasarlanmıř ve bu uygulama ile anlaşıman sisteminin alıřması gözlemlenerek doėrulanmıřtır.

7.1 Mithatpařa Tren İstasyonu

řekil 7.1’de Mithatpařa tren istasyonu verilmiřtir. Bu tren istasyonu ařaėıdaki saha elemanlarından oluřmaktadır:

- “001BT”, “3T”, “1T”, “1ST”, “2ST”, “3ST”, “51T”, “53T”, “001DT”, “(002B)” ve “(002D)” no’lu 11 adet ray devresi,
- “1”, “51” ve “53” no’lu 3 adet S makas ile “3” ve “53” no’lu 2 adet basit makas,

- “2D”, “52DA” ve “52B” no’lu 3 adet 4’lü yüksek sinyal, “2BA” no’lu 1 adet 3’lü yüksek sinyal ve “2BB”, “4B”, “4D”, “52DB”, “54D” ve “54B” no’lu 6 adet 3’lü cüce sinyal.

Mithatpaşa tren istasyonu için oluşturulan anklaşman tablosunun 1 – 14 no’lu güzergâhlara ait olan kısmı Çizelge 7.1’de, 15 – 20 no’lu güzergâhlara ait olan kısmı ise Çizelge 7.2’de verilmiştir. Anklaşman tablosunun her bir satırında o satırdaki güzergâhın tanzimi için sinyallerin vermesi gereken bildirim, makasların kilitlenmesi gereken konum ve kontrol edilmesi gereken ray devreleri ile ilgili bilgiler bulunmaktadır. Örneğin “001BT-1ST” güzergâhı için TKM’den tanzim isteği geldiğini varsayalım. Anklaşman tablosuna bakılarak bu tanzim isteğinin kabul edilmesi için sağlanması gereken koşullar aşağıdaki gibi elde edilir:

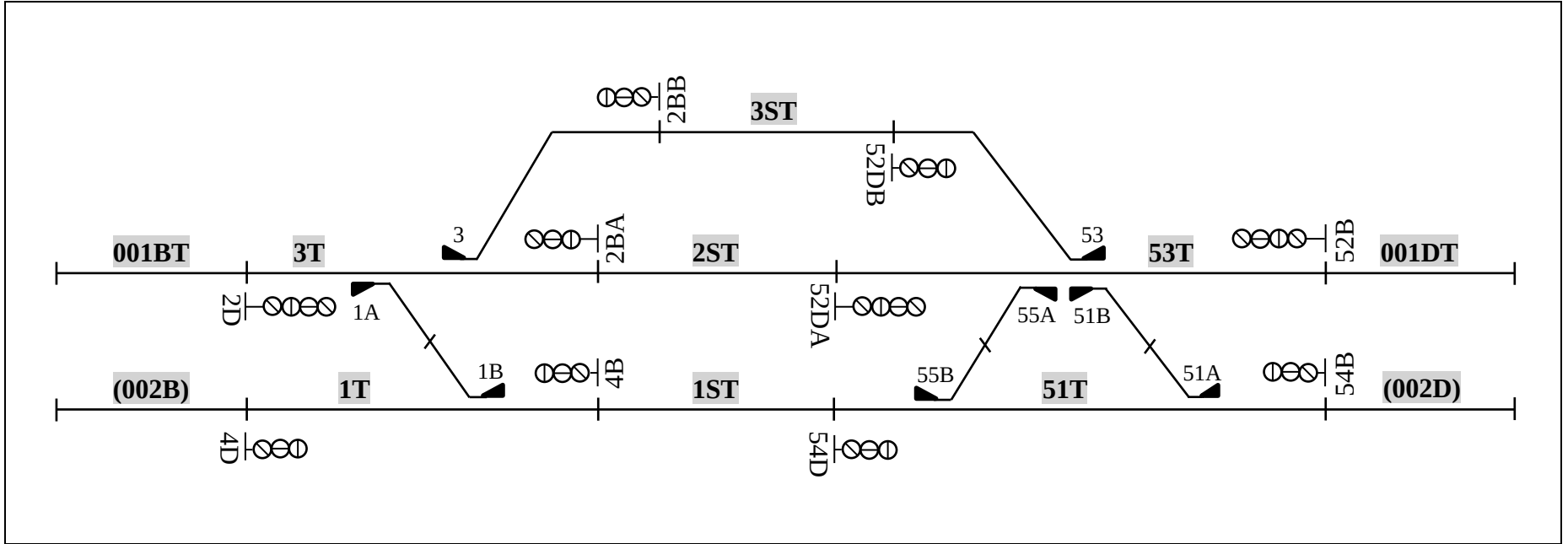
1. “001BT-1ST” güzergâhı ile çakışan önceden tanzim edilmiş başka bir güzergâh bulunmamalıdır. Anklaşman tablosuna bakıldığında bu güzergâh ile çakışan güzergâhların “001BT-2ST”, “001BT-3ST”, “(002B)-1ST”, “1ST-001BT”, “1ST-(002B)”, “2ST-001BT”, “3ST-001BT”, “001DT-1ST” ve “(002D)-1ST” olduğu görülmektedir.
2. “001BT-1ST” güzergâhına ait ray devrelerinin üzerinde tren bulunmamalıdır. Anklaşman tablosunda bu güzergâhın bulunduğu satırdaki “Sinyal Kontrol veya Dedektör Kilit” sütununa bakıldığında söz konusu ray devrelerinin “3T”, “1T” ve “1ST” olduğu görülmektedir. Buradan “3T”, “1T” ve “1ST” ray devrelerinin boş olması gerektiği sonucuna varılmaktadır.
3. Trenin “001BT-1ST” güzergâhı üzerinde ilerleyebilmesi için ilgili makas(lar) uygun konuma kilitlenmelidir. Anklaşman tablosunda ilgili satırdaki “Kilit” sütununa bakıldığında yuvarlak içine alınmış “1” ifadesinin olduğu görülmektedir. Burada “1” ile kilitlenecek makasın numarası belirtilmektedir. “1” ifadesinin yuvarlak içinde olması ise “1” no’lu makasın “ters” konuma kilitlenmesi gerektiğini ifade etmektedir. Eğer “1” ifadesi yuvarlak içinde olmasaydı bu sefer “1” no’lu makasın “normal” konuma kilitlenmesi gerektiği ifade edilecekti. Bununla birlikte “Mevcut Konumda” sütununa bakıldığında “3” ifadesinin olduğu görülmektedir. Burada ise “3” ile mevcut konuma kilitlenmesi gereken makasın numarası belirtilmektedir. Bu bilgilerden yola çıkılarak “1” no’lu makasın “ters”, “3” no’lu makasın ise mevcut konuma kilitlenmesi gerektiği sonucuna varılmaktadır.

4. “001BT-1ST” güzergâhı için “2D” sinyali uygun bildirimi verecek şekilde ayarlanmalıdır. Bu amaçla güzergâh üzerinde bu sinyalden bir sonraki sinyal olan “54D” sinyalinin verdiği bildirim dikkate alınır. “54D” sinyali “sarı” bildirim veriyorsa “2D” sinyali “sarı üzeri yeşil”, “54D” sinyali “kırmızı” veya “sarı üzeri kırmızı” bildirim veriyorsa “2D” sinyali “sarı üzeri sarı” bildirim verecek şekilde ayarlanmalıdır. Bu bilgi anlaşılan tablosunda ilgili satırdaki “Sinyal No” ve “Sinyal Durumu” sütunlarından elde edilebilir.

7.2 Anlaşman Sisteminin Petri Ağları ile Modellenmesi

Mithatpaşa tren istasyonuna ait anlaşman sisteminin modellenmesi için modüler bir yaklaşım uygulanmıştır. Bu amaçla anlaşman sistemi bir takım fonksiyon bloklarına bölünmüş ve her bir fonksiyon bloğu için PA modeli oluşturulmuştur. Bu fonksiyon blokları aşağıda verilmiş olup fonksiyon bloklarına ait PA modelleri arasındaki ilişki Şekil 7.7’de verilmiştir:

- **Tanzim Bloğu:** TKM’den gelen güzergâh tanzim isteklerini değerlendirir. Güzergâh tanzimi için saha elemanları tarafından gerekli koşullar sağlanıyorsa tanzim isteğini kabul eder ve ilgili ray devrelerini kilitler. Aksi takdirde tanzim isteğini reddeder.
- **Makas Bloğu:** Tanzim isteği kabul edilen güzergâhlar için makasları uygun olarak konumlandırır ve kilitler. Tanzim sonlandığında makasların kilitlerini çözer. Ayrıca makasların hatalı duruma getirilmesini sağlar.
- **Tren Takip Bloğu:** Trenin güzergâh üzerinde ilerlerken takibinin yapılmasını sağlar. Tanzim iptal edildiğinde veya tren güzergâhtaki son ray devresine geldiğinde tanzimin sonlandırılmasını ve ray devrelerinin kilitlerinin çözülmesini sağlar.
- **Cüce Sinyal Bloğu:** Tanzim isteğinin kabul edilmesi/reddedilmesi veya tanzimin iptal edilmesi/sonlanması durumunda ilgili cüce sinyalin uygun bildirimi vermesini sağlar.
- **Yüksek Sinyal Bloğu:** Tanzim isteğinin kabul edilmesi/reddedilmesi veya tanzimin iptal edilmesi/sonlanması durumunda ilgili yüksek sinyalin uygun bildirimi vermesini sağlar.



Şekil 7.1 : Mithatpaşa tren istasyonu.

Çizelge 7.1 : Mithatpaşa tren istasyonuna ait anlaşıman tablosu (1 – 14 no’lu rotalar).

Tanım			Sinyal No	Sinyal Durumu	Kilit			Sinyal Kontrol veya Dedektör Kilit	Rota Kilit	
İsim	Rota No	Rota			Mevcut Konumda					
GİRİŞ SİNYALİ	rt1	001BT-1ST	1S	SY SS SK	54D S 54D K, SK	①	3	4B 52B1 54B1	3T 1T 1ST (3T) (1T)	
	rt2	001BT-2ST	2D	2S Y S SK	52DA S 52DA K, SK	1	3	2BA 52B2 54B2	3T 1T 1ST 3T 2ST (3T)	
	rt3	001BT-3ST	3S	SY SS SK	52DB S 52DB K, SK	1	③	2BB 52B3 54B3	3T 2ST 3T 3ST (3T)	
	rt4	(002B)-1ST	4D	Y S SK	54D S 54D K, SK	1		4B 52B1 54B1	3T 3ST 3T 3ST (3T)	
	rt5	1ST-001BT	4B	001B	S	①	3	2D B2D	1T 1ST 1T 1ST (1T)	
ÇIKIŞ SİNYALİ	rt6	1ST-(002B)	002B	SK		1		4D	1T 3T 001BT 1T (1T) (3T)	
	rt7	1ST-001DT	54D	001D	S	⑤⑤	53	51	52B D52B	51T 53T 001DT (51T) (53T)
	rt8	1ST-(002D)	002D	SK		55	51	54B	51T (51T)	
	rt9	2ST-001BT	2BA	S		3	1	2D B2D	3T 001BT (3T)	
	rt10	2ST-001DT	52DA	001D	S	55	53	51	52B D52B	53T 001DT (53T)
	rt11	2ST-(002D)	002D	SK		55	53	⑤①	54B	53T 51T (53T) (51T)
	rt12	3ST-001BT	2BB	S		③	1	2D B2D	3T 001BT (3T)	
	rt13	3ST-001DT	52DB	001D	S	55	⑤③	51	52B D52B	53T 001DT (53T)
	rt14	3ST-(002D)	002D	SK		⑤③	⑤①	55	54B	53T 51T (53T) (51T)

Çizelge 7.2 : Mithatpaşa tren istasyonuna ait anlaşıman tablosu (15 – 20 no’lu rotalar).

Tanım			Sinyal No	Sinyal Durumu	Kilit			Sinyal Kontrol veya Dedektör Kilit	Rota Kilit	
İsim	Rota No	Rota			Mevcut Konumda					
GİRİŞ SİNYALİ	rt15	001DT-1ST	1S	SY SS SK	4B S 4B K, SK	51	53 (55)	54D 2D1 4D1	53T 51T 1ST 53T 51T(1ST)	(53T) (51T)
	rt16	001DT-2ST	52B 2S	Y S SK	2BA S 2BA K	51	53 55	52DA 2D2	53T 2ST 53T(2ST)	(53T)
	rt17	001DT-3ST	3S	SY SS SK	2BB S 2BB K	51	(53) 55	52DB 2D3	53T 3ST 53T(3ST)	(53T)
	rt18	(002D)-1ST	1S	Y S SK	4B S 4B K, SK	51	55	54D 2D1 4D1	51T 1ST 51T(1ST)	(51T)
	rt19	(002D)-2ST	54B 2S	Y S SK	2BA S 2BA K	(51)	53 55	52DA 2D2	51T 53T 2ST 51T 53T(2ST)	(51T) (53T)
	rt20	(002D)-3ST	3S	Y S SK	2BB S 2BB K	(51)	(53) 55	52DB 2D3	51T 53T 3ST 51T 53T(3ST)	(51T) (53T)

7.2.1 Tanzim bloğunun Petri Ağları modeli

Tanzim bloğuna ait PA modeli Şekil 7.2’de, bu PA modelindeki yerlerin tanımı ise Çizelge 7.3’te verilmiştir:

Çizelge 7.3 : Tanzim bloğuna ait PA’daki yerlerin tanımı.

Yer	Tanım
P_RD_X	“X” ray devresi başka bir güzergâh için kullanılmamaktadır.
P_RD_Y	“Y” ray devresi başka bir güzergâh için kullanılmamaktadır.
P_RD_Z	“Z” ray devresi başka bir güzergâh için kullanılmamaktadır.
P_T_KBL_1	Güzergâha ait ray devrelerinin hiçbiri kilitli değildir.
P_T_KBL_2	Güzergâha ait ray devrelerinin hepsi boş durumdadır.
P_T_RET_1_ONY	Tanzim isteği reddedilecek (çakışan güzergâh).
P_T_RET_2_ONY	Tanzim isteği reddedilecek (dolu ray devresi).
P_T_RET_1	Güzergâh ile çakışan başka bir güzergâh mevcuttur.
P_T_RET_2	Güzergâha ait ray devrelerinden en az biri doludur.
P_T_RET_3	Güzergâh üzerindeki makaslardan en az biri hatalıdır.
P_T_RET	Tanzim isteği reddedildi.
P_MX_NH	“X” no’lu makas “normal” konumdan hatalı duruma geçti.
P_MX_TH	“X” no’lu makas “ters” konumdan hatalı duruma geçti.
P_MY_NH	“Y” no’lu makas “normal” konumdan hatalı duruma geçti.
P_MY_TH	“Y” no’lu makas “ters” konumdan hatalı duruma geçti.
P_MZ_NH	“Z” no’lu makas “normal” konumdan hatalı duruma geçti.
P_MZ_TH	“Z” no’lu makas “ters” konumdan hatalı duruma geçti.
P_MX_K_AC	“X” no’lu makasın kilidi çözülecek.
P_MY_K_AC	“Y” no’lu makasın kilidi çözülecek.
P_TK_MX	“X” no’lu makas uygun konuma kilitlenecek.
P_M_X	“X” no’lu makas uygun konuma kilitlendi.
P_T_HZR_X	“Y” no’lu makasın kilitlenmesi için sistem hazır durumdadır.
P_TK_MY	“Y” no’lu makas uygun konuma kilitlenecek.
P_M_Y	“Y” no’lu makas uygun konuma kilitlendi.
P_T_HZR_Y	“Z” no’lu makasın kilitlenmesi için sistem hazır durumdadır.
P_TK_MZ	“Z” no’lu makas uygun konuma kilitlenecek.
P_M_Z	“Z” no’lu makas uygun konuma kilitlendi.
P_T_HZR	Tüm makaslar uygun konuma kilitlendi (tanzim kabul).

P_RD_X, P_RD_Y ve P_RD_Z yerlerinde birer adet jeton bulunması sırasıyla “X”, “Y” ve “Z” ray devrelerinin başka bir güzergâh tarafından kullanılmadığı ve dolayısıyla bu ray devrelerinin kilitli durumda olmadığı anlamına gelmektedir. Bu durumda TKM’den tanzim isteği geldiğinde P_T_KBL_1 yerine jeton konulur. Aksi takdirde sistem P_T_RET_1 yerine geçer ve tanzim isteği reddedilir. P_T_KBL_1 yerinde jeton varken güzergâha ait ray devreleri kontrol edilir. Tüm ray devrelerinin boş olması durumunda P_T_KBL_2 yerine jeton konularak ray devreleri kilitlenir. Aksi takdirde sistem P_T_RET_2 yerine geçer ve tanzim isteği reddedilir.

P_T_KBL_2 yerine jeton konulduğunda “X” no’lu makasın, P_T_HZR_X yerine jeton konulduğunda “Y” no’lu makasın, P_T_HZR_Y yerine jeton konulduğunda ise “Z” no’lu makasın kilitlenmesi sağlanır. Makasların kilitlenebilmesi için hatalı durumda olmaması gerekmektedir. Eğer güzergâh üzerindeki makaslardan en az biri hatalı durumdaysa P_MX_NH, P_MX_TH yerlerinin birinde ve/veya P_MY_NH, P_MY_TH yerlerinin birinde ve/veya P_MZ_NH, P_MZ_TH yerlerinin birinde jeton bulunacaktır. Bu durumda P_T_RET_3 yerine jeton konulur, ray devrelerinin kilitleri çözülür ve tanzim isteği reddedilir. Ayrıca “X” ve/veya “Y” no’lu makaslar hatalı durumda olmayıp kilitlenmişse bu makasların kilitlerinin çözülmesi için makas bloğuna ait olan P_MX_K_AC ve/veya P_MY_K_AC yerlerine jeton konulur.

Güzergâh üzerindeki makasların hiçbiri hatalı durumda değilse bu makasların uygun konuma getirilip kilitlenmesi sağlanır. Bu amaçla sırasıyla P_TK_MX, P_TK_MY ve P_TK_MZ yerlerine jeton konularak makas bloğuna komutlar gönderilmektedir. Bu komutlar makas bloğu tarafından işlendikten sonra P_M_X, P_M_Y ve P_M_Z yerlerine jeton konularak sırasıyla “X”, “Y” ve “Z” no’lu makasın uygun konuma kilitlenmesi sağlanmakta ve P_T_HZR yerine jeton konulup tanzim isteği kabul edilmektedir.

7.2.2 Makas bloğunun Petri Ağları modeli

Makas bloğuna ait PA modeli Şekil 7.3’te, bu PA modelindeki yerlerin tanımı ise Çizelge 7.4’te verilmiştir:

Çizelge 7.4 : Makas bloğuna ait PA’daki yerlerin tanımı.

Yer	Tanım
P_TK_MT	Makas “ters” konuma kilitlenecek.
P_TK_MN	Makas “normal” konuma kilitlenecek.
P_TK_MK	Makas mevcut bulunduğu konuma kilitlenecek.
P_M_N	Makas “normal” konumda bulunmaktadır.
P_MN_H	Makas “normal” konumdayken hatalı duruma geçti.
P_M_N_T	Makas “normal” konumdan “ters” konuma geçmektedir.
P_M_T	Makas “ters” konumda bulunmaktadır.
P_MT_H	Makas “ters” konumdayken hatalı duruma geçti.
P_M_T_N	Makas “ters” konumdan “normal” konuma geçmektedir.
P_MT_K	Makas “ters” konuma kilitlendi.
P_MN_K	Makas “normal” konuma kilitlendi.
P_M_K	Makas uygun konuma kilitlendi.
P_M_K_YOK	Makas kilitli değildir.
P_M_K_AC	Makasın kilidi çözülecek.

P_M_N yerinde jeton bulunması makasın “normal”, P_M_T yerinde jeton bulunması ise makasın “ters” konumda olduğunu göstermektedir. P_M_K_YOK yerinde jeton bulunması makasın kilitli olmayıp güzergâh için kullanılabilir durumda olduğu anlamına gelmektedir. P_TK_MT, P_TK_MN ve P_TK_MK yerleri sırasıyla makasın “ters”, “normal” veya mevcut bulunduğu konuma kilitlenmesini sağlamaktadır. Aynı makas için bu yerlerden sadece birinde jeton bulunur. Jeton konulacak yerin seçilmesi ve seçilen bu yere jetonun konulması işlemleri tanzim bloğu tarafından yapılmaktadır.

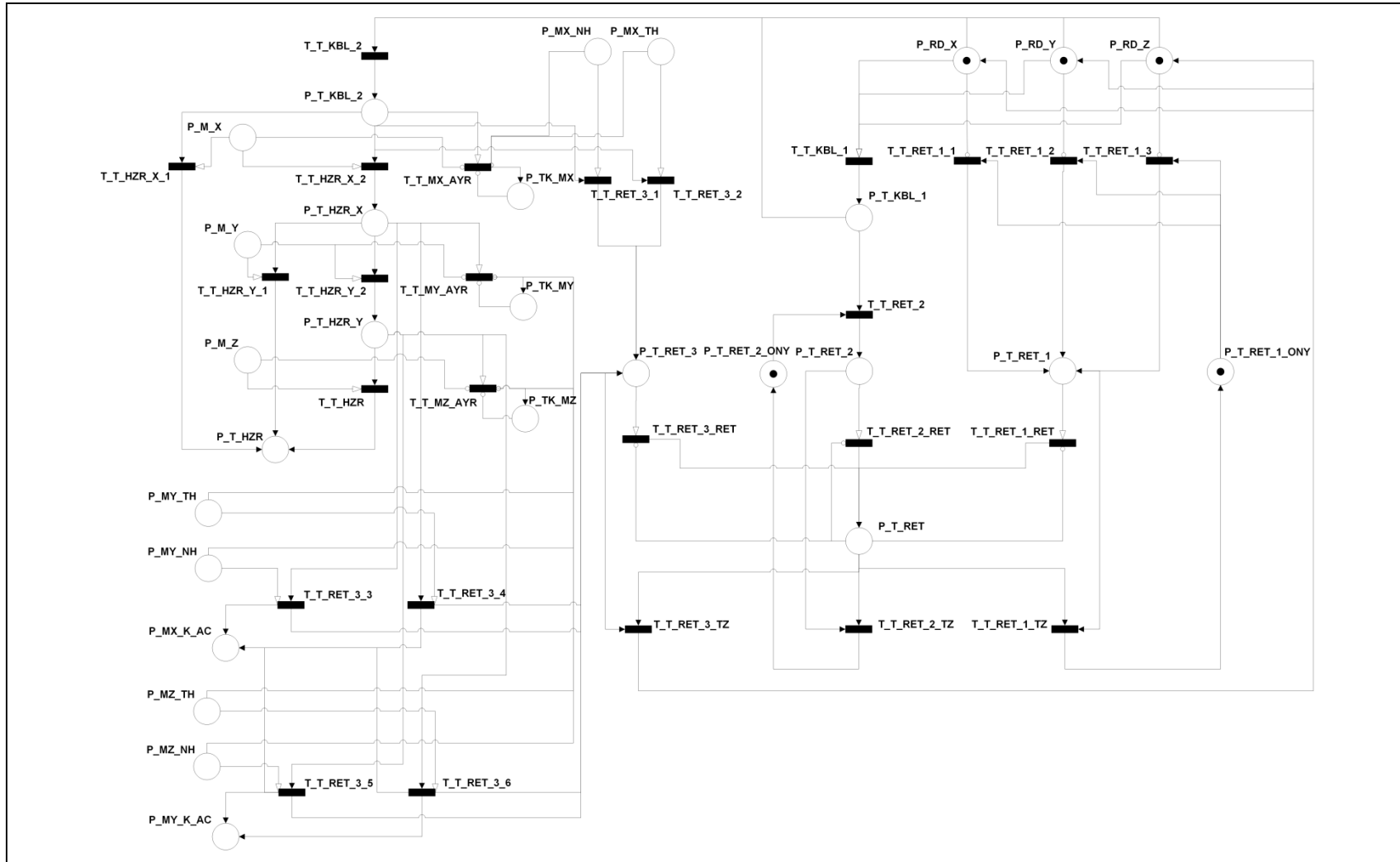
Sistem P_M_N yerindeyken P_TK_MT yerine jeton konulduğunda sistem önce P_M_N_T sonra da P_M_T yerine geçiş yapar. Böylelikle makas “ters” konuma getirilmiş olur. Bu durumda P_MT_K ve P_M_K yerlerine jeton konularak makasın “ters” konuma kilitlenmesi sağlanır ve bu bilgi tanzim bloğuna iletilir.

Sistem P_M_T yerindeyken P_TK_MN yerine jeton konulduğunda sistem önce P_M_T_N sonra da P_M_N yerine geçiş yapar. Böylelikle makas “normal” konuma getirilmiş olur. Bu durumda P_MN_K ve P_M_K yerlerine jeton konularak makasın “normal” konuma kilitlenmesi sağlanır ve bu bilgi tanzim bloğuna iletilir.

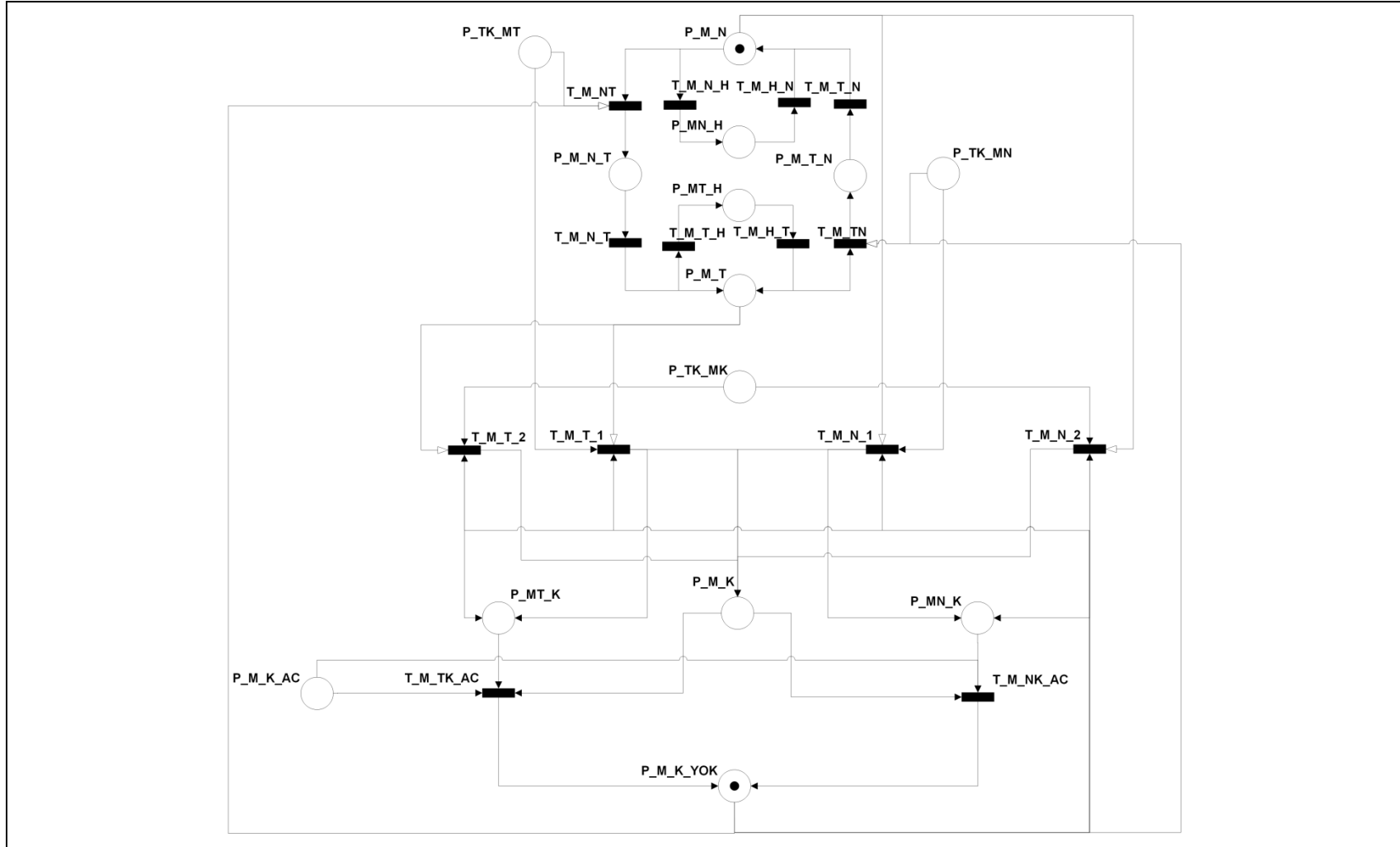
P_M_N yerinde jeton varken P_TK_MK yerine jeton konulduğunda P_MN_K ve P_M_K yerlerine jeton konularak makasın “normal” konuma kilitlenmesi sağlanır. P_M_T yerinde jeton varken P_TK_MK yerine jeton konulduğunda ise P_MT_K ve P_M_K yerlerine jeton konularak makasın “ters” konuma kilitlenmesi sağlanır. Her iki durumda da makasın mevcut bulunduğu konuma kilitlenmesi sağlanır ve bu bilgi tanzim bloğuna iletilir.

P_M_N yerinde jeton varken makas hata durumuna geçerse P_MN_H yerine jeton konulur. P_M_T yerinde jeton varken makas hata durumuna geçerse P_MT_H yerine jeton konulur. Aynı anda P_MN_H veya P_MT_H yerlerinin sadece birinde jeton bulunabilir. Bu yerlerde jeton bulunması makasın hata durumunda olduğu göstermektedir.

Tanzim isteğinin reddedilmesi veya tanzimin sonlanması/iptal edilmesi durumunda bu tanzime ilişkin güzergâh üzerinde bulunan makasın kilidinin çözülmesi gerekmektedir. Bu amaçla tren takip bloğu tarafından P_M_K_AC yerine jeton konulur ve makasın kilidinin çözülmesi sağlanır.



Şekil 7.2 : Tanzim bloğunun PA modeli.



Şekil 7.3 : Makas bloğunun PA modeli.

7.2.3 Tren takip bloğunun Petri Ağları modeli

Tren takip bloğuna ait PA modeli Şekil 7.4’te, bu PA modelindeki yerlerin tanımı ise Çizelge 7.5’te verilmiştir:

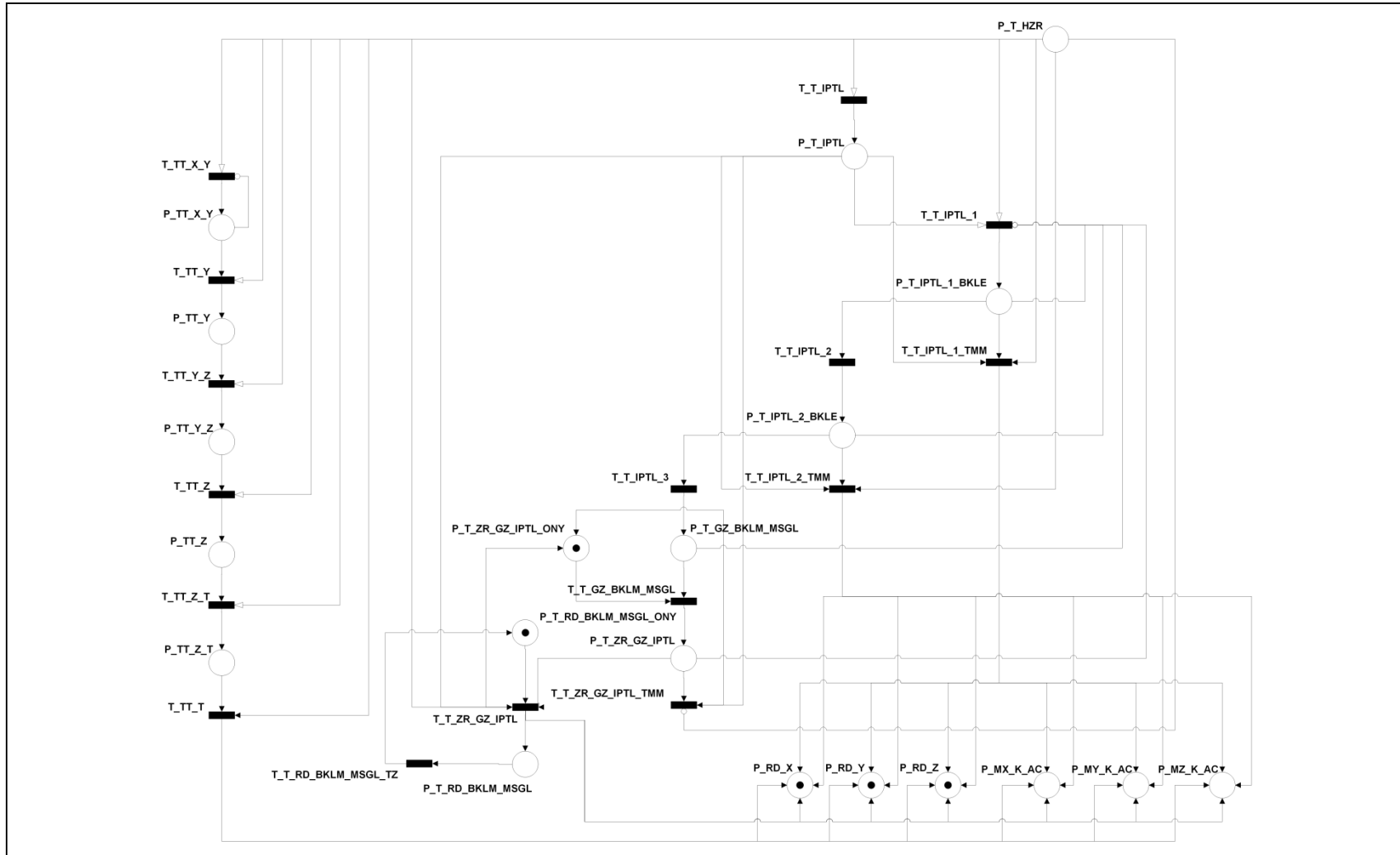
Çizelge 7.5 : Tren takip bloğuna ait PA’daki yerlerin tanımı.

Yer	Tanım
P_T_HZR	Güzergâh trenin ilerlemesi için hazır.
P_T_IPTL	Güzergâha ait tanzim iptal edilecek.
P_TT_X_Y	Tren “X” ile “Y” ray devrelerinin arasındadır.
P_TT_Y	Tren sadece “Y” ray devresi üzerindedir.
P_TT_Y_Z	Tren “Y” ile “Z” ray devrelerinin arasındadır.
P_TT_Z	Tren sadece “Z” ray devresi üzerindedir.
P_TT_Z_T	Tren “Z” ile “T” ray devrelerinin arasındadır.
P_T_IPTL_1_BKLE	Trenin ilk ray devresine gelmesi beklenecek.
P_T_IPTL_2_BKLE	Trenin ikinci ray devresine gelmesi beklenecek.
P_T_ZR_GZ_IPTL_ONY	Zorunlu güzergâh iptali yapılmadı.
P_T_RD_BKLM_MSGL_ONY	Ray devrelerinde beklenmedik meşguliyet yok.
P_T_GZ_BKLM_MSGL	“Güzergâhta beklenmedik meşguliyet”
P_T_ZR_GZ_IPTL	Zorunlu güzergâh iptali yapıldı.
P_T_RD_BKLM_MSGL	“Ray devrelerinde beklenmedik meşguliyet”
P_RD_X	“X” ray devresinin kilidi çözülecek.
P_RD_Y	“Y” ray devresinin kilidi çözülecek.
P_RD_Z	“Z” ray devresinin kilidi çözülecek.
P_MX_K_AC	“X” no’lu makasın kilidi çözülecek.
P_MY_K_AC	“Y” no’lu makasın kilidi çözülecek.
P_MZ_K_AC	“Z” no’lu makasın kilidi çözülecek.

P_T_HZR yerinde jeton bulunması tanzim isteğinde bulunulan güzergâha ait ray devrelerinin kilitlenmiş ve makasların uygun konuma ayarlanmış olduğu anlamına gelmektedir. P_T_HZR yeri tanzim bloğuna ait olup tren takip bloğu için giriş yeri olarak kullanılmaktadır. Sistem P_T_HZR yerindeyken trenin güzergâh üzerinde ilerlerken takibinin yapılması sağlanır. Tanzim edilen güzergâhın tamamlanması için tren “X” ray devresi üzerindeyken sırasıyla “Y” ve “Z” ray devrelerinden geçtikten sonra tamamen “T” ray devresinin üzerine gelmelidir. Tren bu şekilde hareket ettiğinde sırasıyla P_TT_X_Y, P_TT_Y, P_TT_Y_Z, P_TT_Z ve P_TT_Z_T yerlerine jeton konulur. P_TT_Z_T yerinde jeton varken tren güzergâhın son ray devresi olan “T” ray devresine geldiğinde tanzim sonlanır. Tanzim ister bu şekilde sonlansın isterse iptal edilerek sonlandırılınsın tanzim bloğuna ait olan P_RD_X, P_RD_Y ve P_RD_Z yerlerine jeton konularak “X”, “Y” ve “Z” ray devrelerinin kilitlerinin çözülmesi sağlanır. Ayrıca sırasıyla “X”, “Y” ve “Z” makas bloklarının

giriş yerleri olan P_MX_K_AC, P_MY_K_AC ve P_MZ_K_AC yerlerine jeton konularak “X”, “Y” ve “Z” no’lu makasların kilitleri de çözülür. Böylelikle güzergâh bir sonraki tanzim için hazır hale getirilmiş olur.

P_T_HZR yerindeyken TKM’den tanzim iptal isteği geldiği zaman P_T_IPTL yerine jeton konulur. P_T_IPTL yerine jeton konularak hâlihazırdaki tanzimin iptal edileceği belirtilmektedir. Ardından P_T_IPTL_1_BKLE yerine jeton konulur ve 5 sn’lik bir zamanlayıcı çalıştırılır. 5 sn’lik süre içinde tren güzergâhın ilk ray devresi olan “X” ray devresine giriş yapmazsa tanzim sonlandırılır. Eğer 5 sn. dolmadan tren “X” ray devresine giriş yaparsa tanzim kurulu bırakılır ve P_T_IPTL_2_BKLE yerine jeton konulur. Bu durumda 10 sn’lik ikinci zamanlayıcı çalıştırılır. 10 sn’lik süre içinde tren güzergâhın ikinci ray devresi olan “Y” ray devresine giriş yapmazsa tanzim sonlandırılır. Eğer 10 sn. dolmadan tren “Y” ray devresine giriş yaparsa “güzergâhta beklenmedik meşguliyet” hatası oluşur ve P_T_GZ_BKLM_MSGL yerine jeton konulur. Bu hatanın giderilmesi için kullanıcı tarafından “zorunlu güzergâh iptal” isteği yapılmalıdır. Kullanıcı tarafından bu istek yapıldığında P_T_ZR_GZ_IPTL yerine jeton konulur ve 15 sn’lik üçüncü zamanlayıcı çalıştırılır. 15 sn’lik süre içinde tren güzergâha ait son ray devresi olan “T” ray devresine gelirse tanzim sonlandırılır ve “güzergâhta beklenmedik meşguliyet” hatası giderilmiş olur. Eğer 15 sn’lik süre dolduğunda tren henüz “T” ray devresine gelmediyse “ray devrelerinde beklenmedik meşguliyet” hatası oluşur ve P_T_RD_BKLM_MSGL yerine jeton konulur. Bu durumda tanzim sonlandırılır ancak hatanın giderilmesi için trenin tamamen “T” ray devresine gelmesi beklenir. Tren “T” ray devresine geldiğinde ise güzergâh tamamlanır ve kullanıcının “ray devrelerinde beklenmedik meşguliyet” hatasını temizlemesine izin verilir. Kullanıcı bu hatayı temizlediğinde tanzim güvenli bir şekilde sonlandırılır ve güzergâh sonraki tanzim istekleri için hazır hale getirilmiş olur.



Şekil 7.4 : Tren takip bloğunun PA modeli.

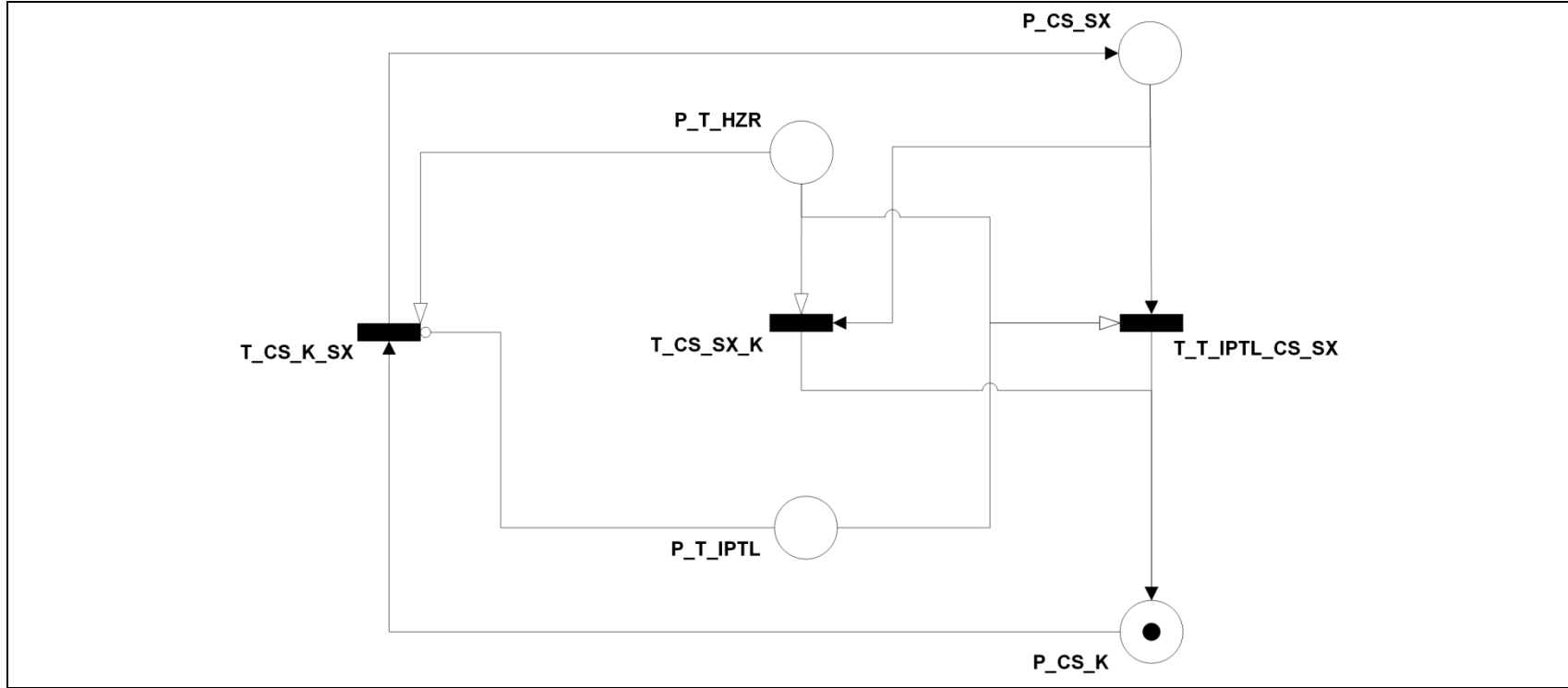
7.2.4 Cüce sinyal bloğunun Petri Ağları modeli

Cüce sinyal bloğuna ait PA modeli Şekil 7.5’te, bu PA modelindeki yerlerin tanımı ise Çizelge 7.6’da verilmiştir:

Çizelge 7.6 : Cüce sinyal bloğuna ait PA’daki yerlerin tanımı.

Yer	Tanım
P_T_HZR	Güzergâh trenin ilerlemesi için hazır.
P_T_IPTL	Güzergâha ait tanzim iptal edilecek.
P_CS_K	Cüce sinyal “kırmızı” bildirim vermektedir.
P_CS_SX	Cüce sinyal “sarı” veya “sarı üzeri kırmızı” bildirim vermektedir.

P_T_HZR yerinde jeton bulunması tanzim isteğinde bulunulan güzergâha ait ray devrelerinin kilitlenmiş ve makasların uygun konuma ayarlanmış olduğu anlamına gelmektedir. P_T_IPTL yerinde jeton bulunması söz konusu güzergâha ait tanzimin iptal edileceğini belirtmektedir. P_T_HZR yeri tanzim bloğuna, P_T_IPTL yeri ise tren takip bloğuna ait olup bu yerler cüce sinyal bloğu için giriş yerleri olarak kullanılmaktadır. Herhangi bir güzergâh için tanzim yapılmamışsa bu güzergâha ait cüce sinyalin “kırmızı” bildirim vermesi gerekmektedir. Bunun için P_CS_K yerinde jeton bulunmalıdır. P_CS_K yerinde jeton varken P_T_HZR yerine jeton konulursa P_CS_K yerindeki jeton harcanır ve P_CS_SX yerine jeton konulur. Böylelikle cüce sinyalin ilgili güzergâh için uygun bildirimi vermesi sağlanır. Tren güzergâh üzerinde ilerleyip cüce sinyalin bir önündeki ray devresine geldiğinde ise P_CS_K yerine tekrar jeton konulur ve cüce sinyalin “kırmızı” bildirim vermesi sağlanır. Böylelikle arkadan gelebilecek trenlere güzergâhın dolu olduğu belirtilmiş olur. P_CS_SX yerinde jeton varken tanzim iptal edildiğinde de P_CS_K yerine jeton konulur ve ilgili cüce sinyalin “kırmızı” bildirim vermesi sağlanır.



Şekil 7.5 : Cüce sinyal bloğunun PA modeli.

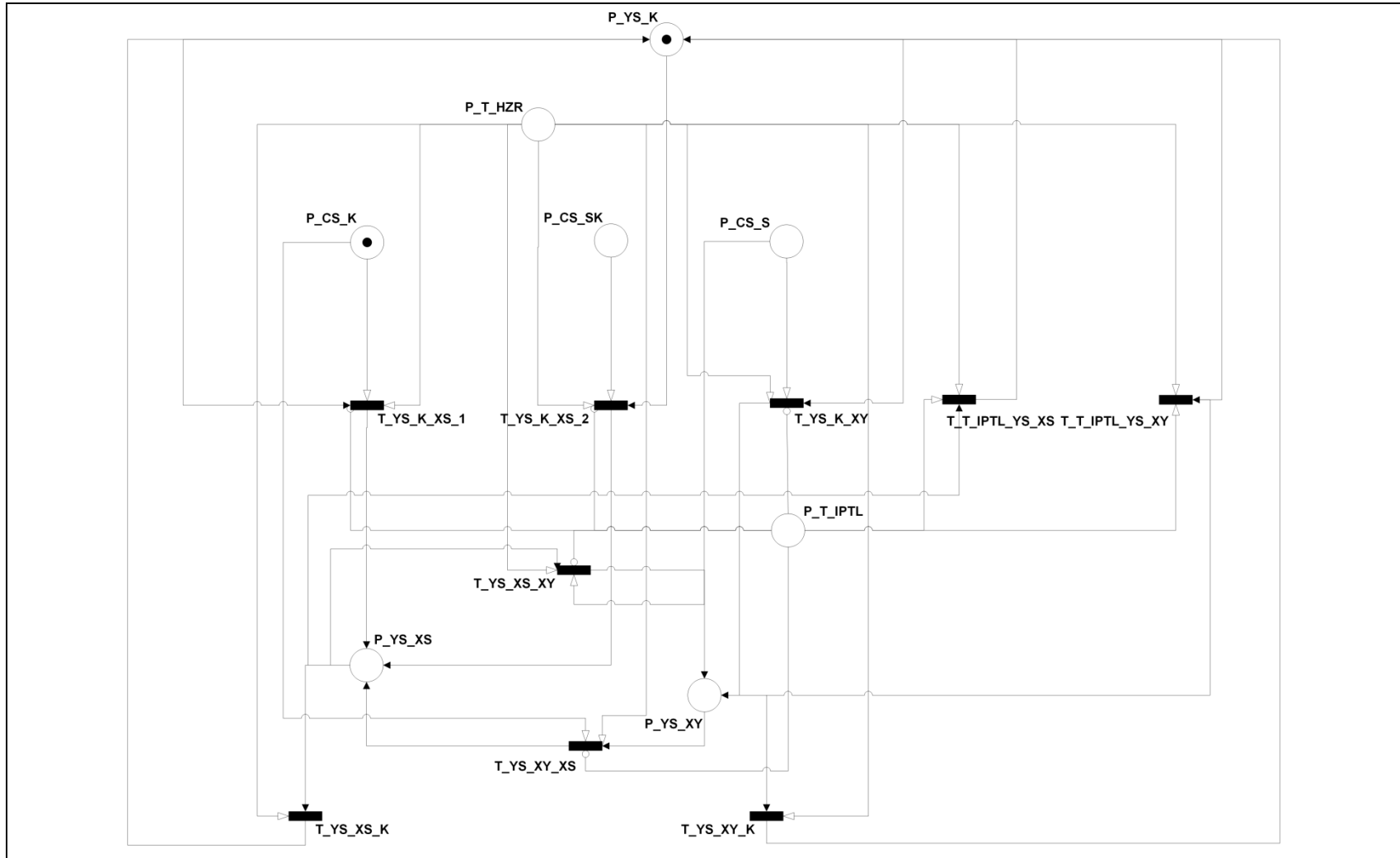
7.2.5 Yüksek sinyal bloğunun Petri Ağları modeli

Yüksek sinyal bloğuna ait PA modeli Şekil 7.6’da, bu PA modelindeki yerlerin tanımı ise Çizelge 7.7’de verilmiştir:

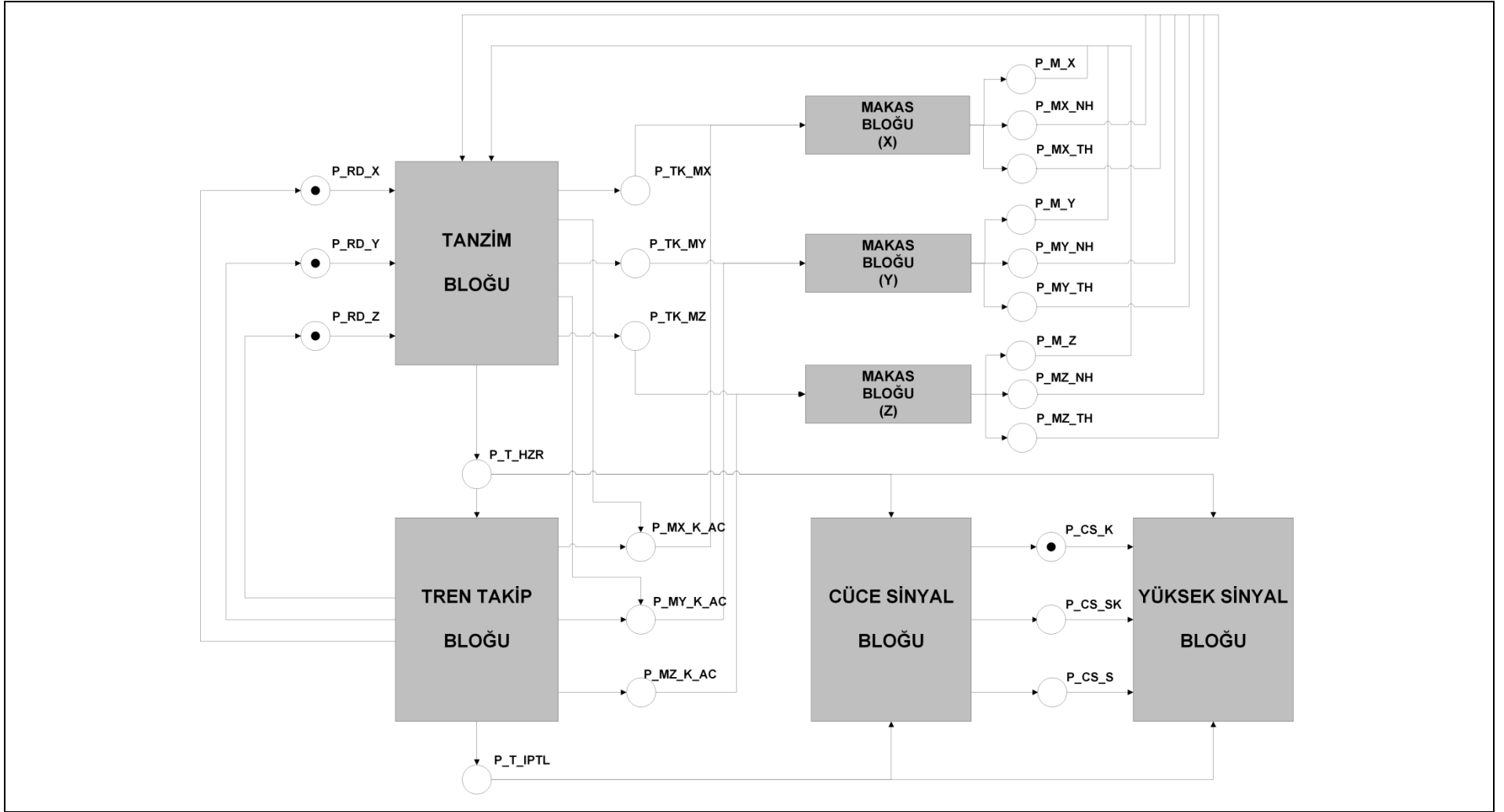
Çizelge 7.7 : Yüksek sinyal bloğuna ait PA’daki yerlerin tanımı.

Yer	Tanım
P_T_HZR	Güzergâh trenin ilerlemesi için hazır.
P_T_IPTL	Güzergâha ait tanzim iptal edilecek.
P_CS_K	Cüce sinyal “kırmızı” bildirim vermektedir.
P_CS_SK	Cüce sinyal “sarı üzeri kırmızı” bildirim vermektedir.
P_CS_S	Cüce sinyal “sarı” bildirim vermektedir.
P_YS_K	Yüksek sinyal “kırmızı” bildirim vermektedir.
P_YS_XS	Yüksek sinyal “sarı” veya “sarı üzeri sarı” bildirim vermektedir.
P_YS_XY	Yüksek sinyal “yeşil” veya “sarı üzeri yeşil” bildirim vermektedir.

P_T_HZR yerinde jeton bulunması tanzim isteğinde bulunulan güzergâha ait ray devrelerinin kilitlenmiş ve makasların uygun konuma ayarlanmış olduğu anlamına gelmektedir. P_T_IPTL yerinde jeton bulunması söz konusu güzergâha ait tanzimin iptal edileceğini belirtmektedir. P_CS_K, P_CS_SK ve P_CS_S yerleri ise ilgili cüce sinyalin verdiği bildirimi göstermektedir. P_T_HZR yeri tanzim bloğuna, P_T_IPTL yeri tren takip bloğuna ve P_CS_K, P_CS_SK, P_CS_S yerleri ise cüce sinyal bloğuna ait olup bu yerler yüksek sinyal bloğu için giriş yerleri olarak kullanılmaktadır. Herhangi bir güzergâh için tanzim yapılmamışsa bu güzergâha ait yüksek sinyalin “kırmızı” bildirim vermesi gerekmektedir. Bunun için P_YS_K yerinde jeton bulunmalıdır. P_YS_K ile P_CS_K veya P_CS_SK yerlerinde jeton varken P_T_HZR yerine jeton konulduğunda P_YS_K yerindeki jeton harcanır ve P_YS_XS yerine jeton konulur. Böylelikle yüksek sinyalin ilgili güzergâh için “sarı” veya “sarı üzeri sarı” bildirim vermesi sağlanır. P_YS_K ile P_CS_S yerlerinde jeton varken P_T_HZR yerine jeton konulduğunda P_YS_K yerindeki jeton harcanır ve bu sefer P_YS_XY yerine jeton konulur. Böylelikle yüksek sinyalin ilgili güzergâh için “yeşil” veya “sarı üzeri yeşil” bildirim vermesi sağlanır. P_YS_XS veya P_YS_XY yerlerinde jeton varken tanzim iptal edildiğinde yüksek sinyalin “kırmızı” bildirim vermesini sağlamak amacıyla bu yerlerdeki jetonlar harcanır ve P_YS_K yerine jeton konulur. P_YS_XS veya P_YS_XY yerlerinde jeton varken tren yüksek sinyalin bir önündeki ray devresine geldiğinde de P_YS_K yerine jeton konulur ve yüksek sinyalin “kırmızı” bildirim vermesi sağlanır.



Şekil 7.6 : Yüksek sinyal bloğunun PA modeli.



Şekil 7.7 : Fonksiyon bloklarının PA modelleri arasındaki bağlantılar.

7.3 SCADA Sistemi Tasarımı

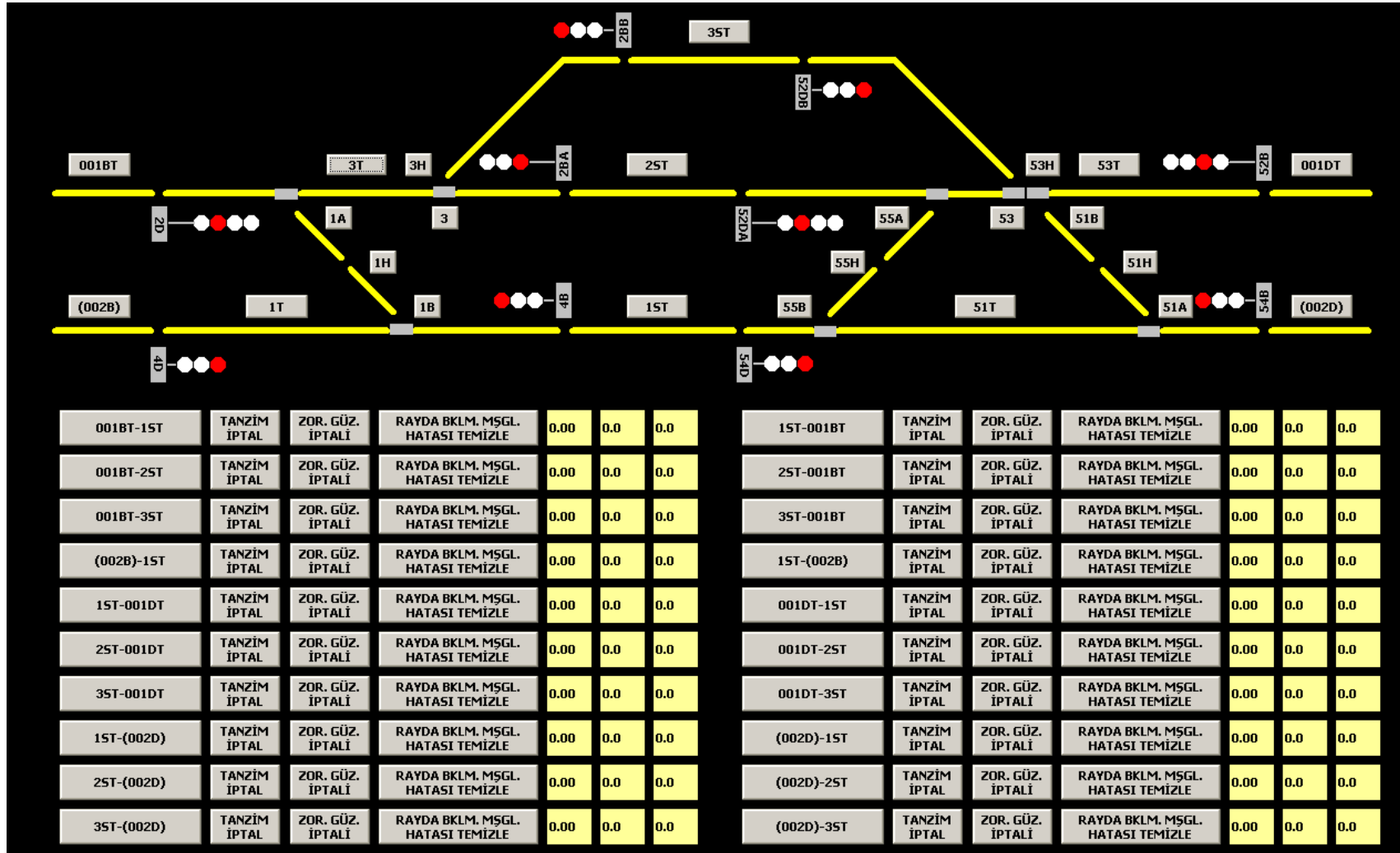
Tanzim, makas, tren takip, cüce sinyal ve yüksek sinyal bloklarının PA modelleri “5.3 Petri Ağlarının Sistem Matrisinden PLC Koduna Geçiş” bölümde önerilen yöntem kullanılarak PLC koduna dönüştürülmüştür. Böylelikle anlaşılan sisteminin PLC ile gerçekleştirilmesi sağlanmıştır. Bununla birlikte anlaşılan sisteminin çalışmasının gözlemlenmesi amacıyla bir SCADA uygulaması tasarlanmıştır. Bu SCADA uygulaması ile Mithatpaşa tren istasyonundaki tüm güzergâhlar için aşağıdaki işlemler gerçekleştirilebilmektedir:

- Güzergâh tanzim isteği yapılabilir.
- Güzergâh tanzim iptal isteği yapılabilir.
- Zorunlu güzergâh iptali gerçekleştirilebilir.
- Rayda beklenmedik meşguliyet hatası giderilebilir.

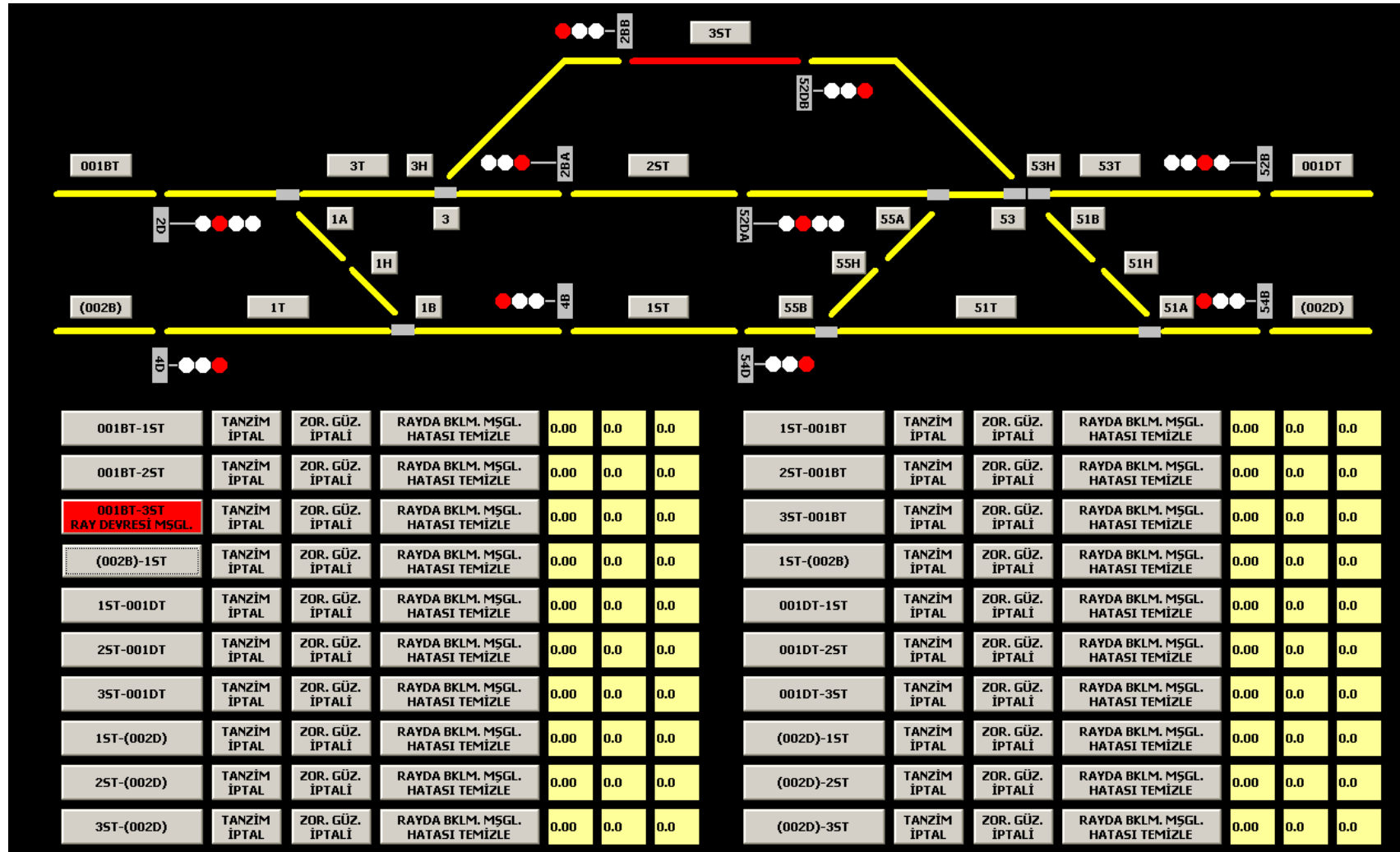
Bunlarla birlikte SCADA uygulaması kullanıcıya aşağıdaki imkânları sunmaktadır:

- Makaslar, sinyaller ve ray devrelerinin durumları gözlemlenebilir.
- Trenlerin güzergâh üzerindeki hareketleri takip edilebilir.
- Makaslar “normal” ya da “ters” konuma alınabilir. Ayrıca makaslar hatalı duruma getirilebilir.
- Tanzim isteklerinin kabul veya reddedildiği kullanıcıya bildirilir.
- Tanzim isteği reddedildiyse bunun nedeni kullanıcıya gösterilebilir.

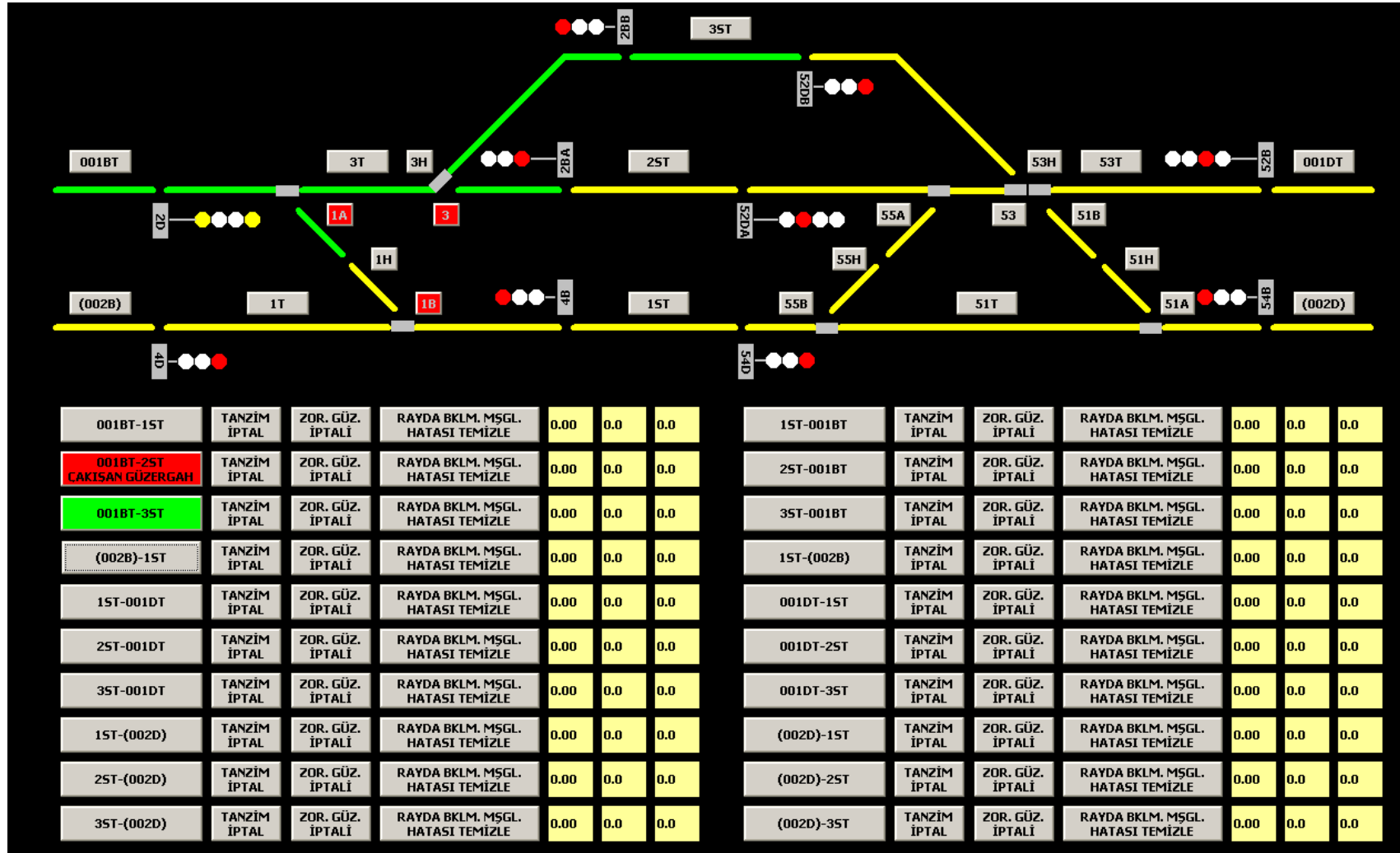
Şekil 7.8’de tren istasyonunun herhangi bir tanzim isteği gelmeden önceki ilk durumu verilmiştir. Ray devreleri ve makaslar kilitlenmemiş durumdadır. Ayrıca hiçbir ray devresinde tren bulunmamaktadır. Dikkat edilecek olursa herhangi bir güzergâh tanzimi yapılmamış olduğu için tüm sinyal lambaları “kırmızı” bildirim vermektedir. Bir güzergâh için tanzim yapılabilmesi için ilgili güzergâhın isminin bulunduğu butona basılmalıdır. Söz konusu butona basıldığında eğer güzergâhın tanzimi ile ilgili koşullardan herhangi biri sağlanmamışsa tanzim isteği reddedilir ve buton kırmızı yakılarak tanzim isteğinin reddedildiği nedeniyle birlikte kullanıcıya bildirilir (Şekil 7.9, Şekil 7.10, Şekil 7.11). Eğer güzergâhın tanzimi ile ilgili koşullar sağlanmışsa tanzim isteği kabul edilir ve buton bu sefer yeşil yakılarak tanzim isteğinin kabul edildiği kullanıcıya bildirilir (Şekil 7.12). Bu şekilde kullanıcıya tanzim isteğinin kabul edilip edilmediği ile ilgili geri bildirim yapılır.



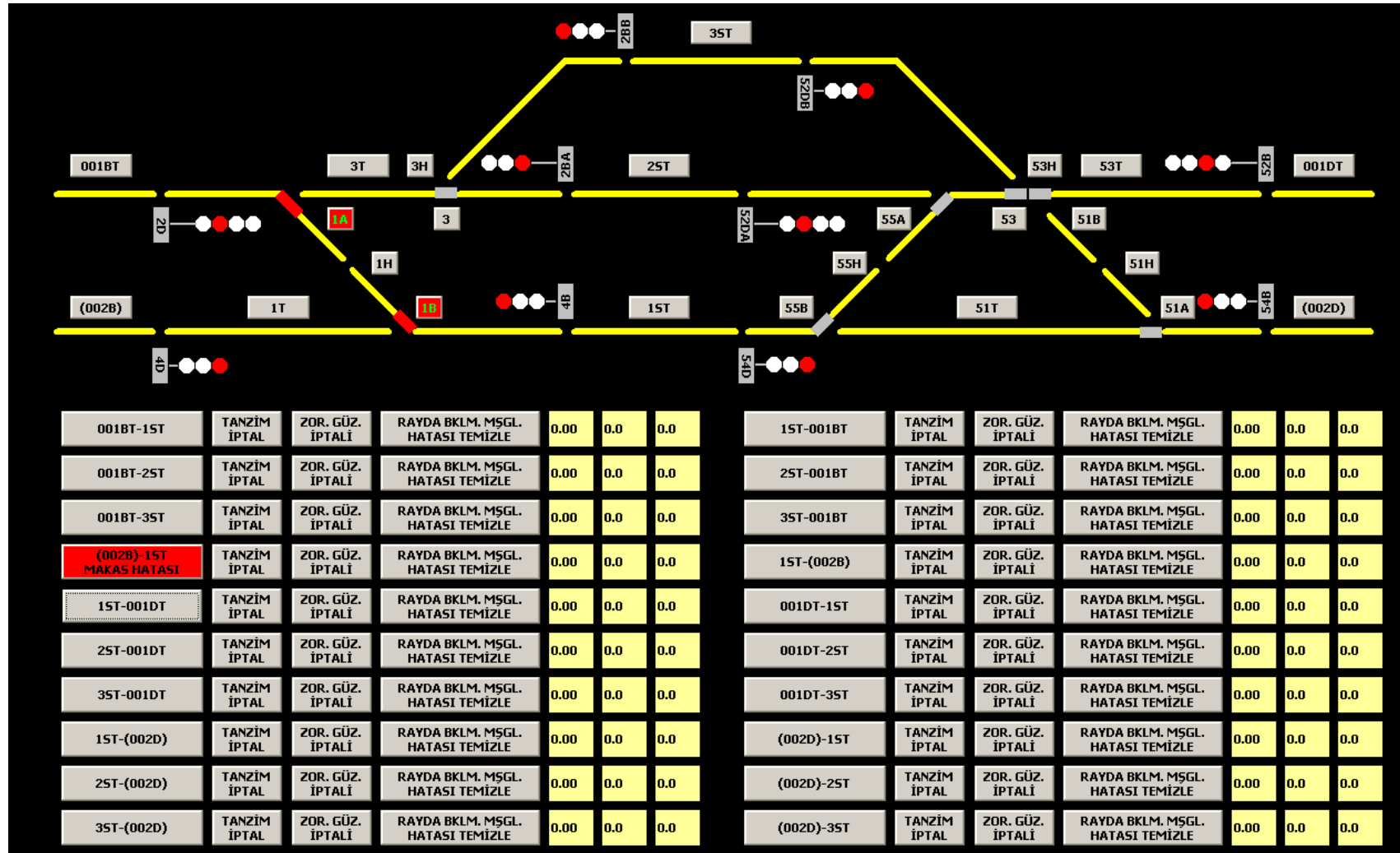
Şekil 7.8 : Tren istasyonunun ilk durumu.



Şekil 7.9 : “001BT-3ST” güzergâhı için tanzim isteğinin reddedilmesi (ray devresi meşgul).



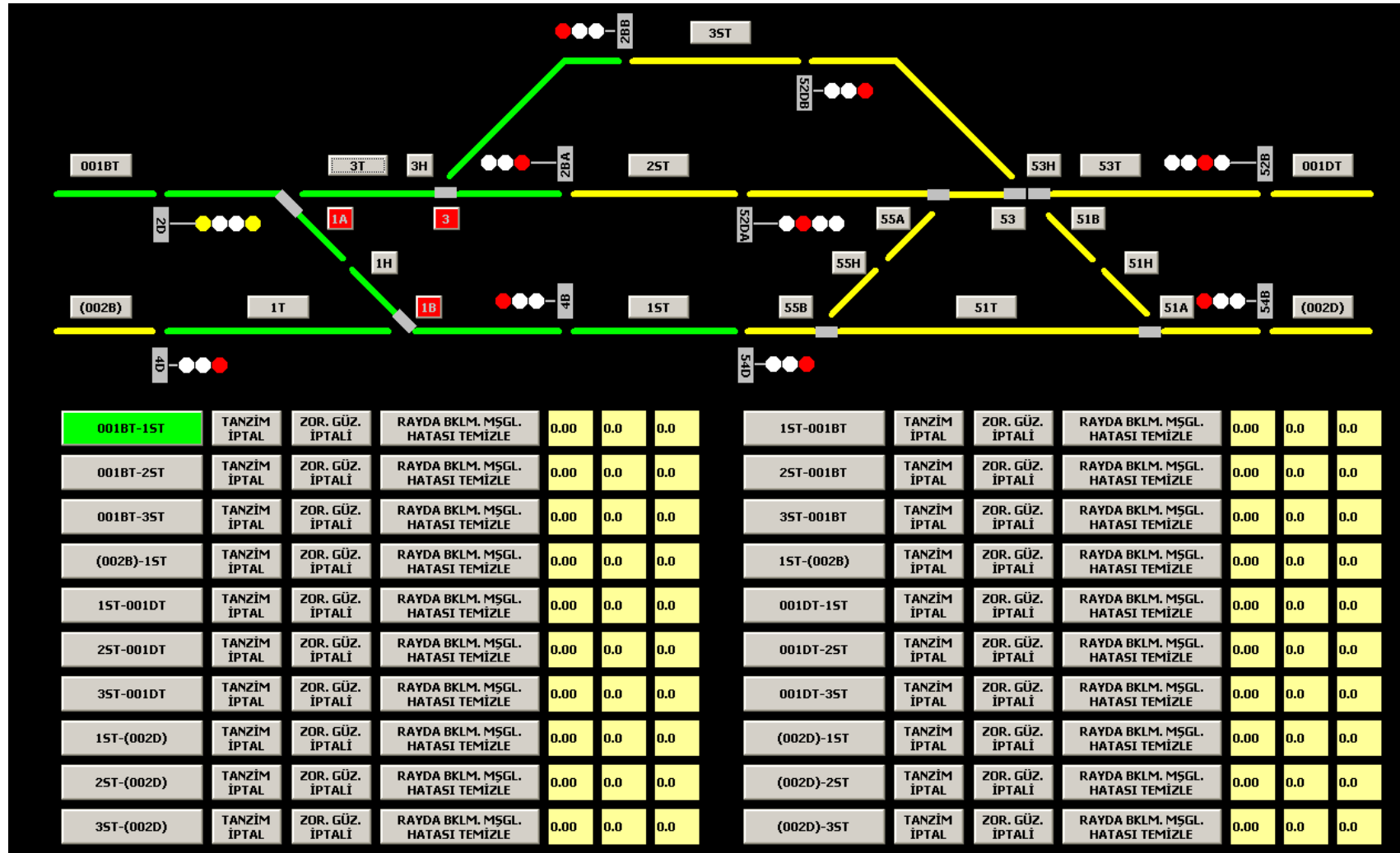
Şekil 7.10 : “001BT-2ST” güzergâhı için tanzim isteğinin reddedilmesi (çakışan güzergâh).



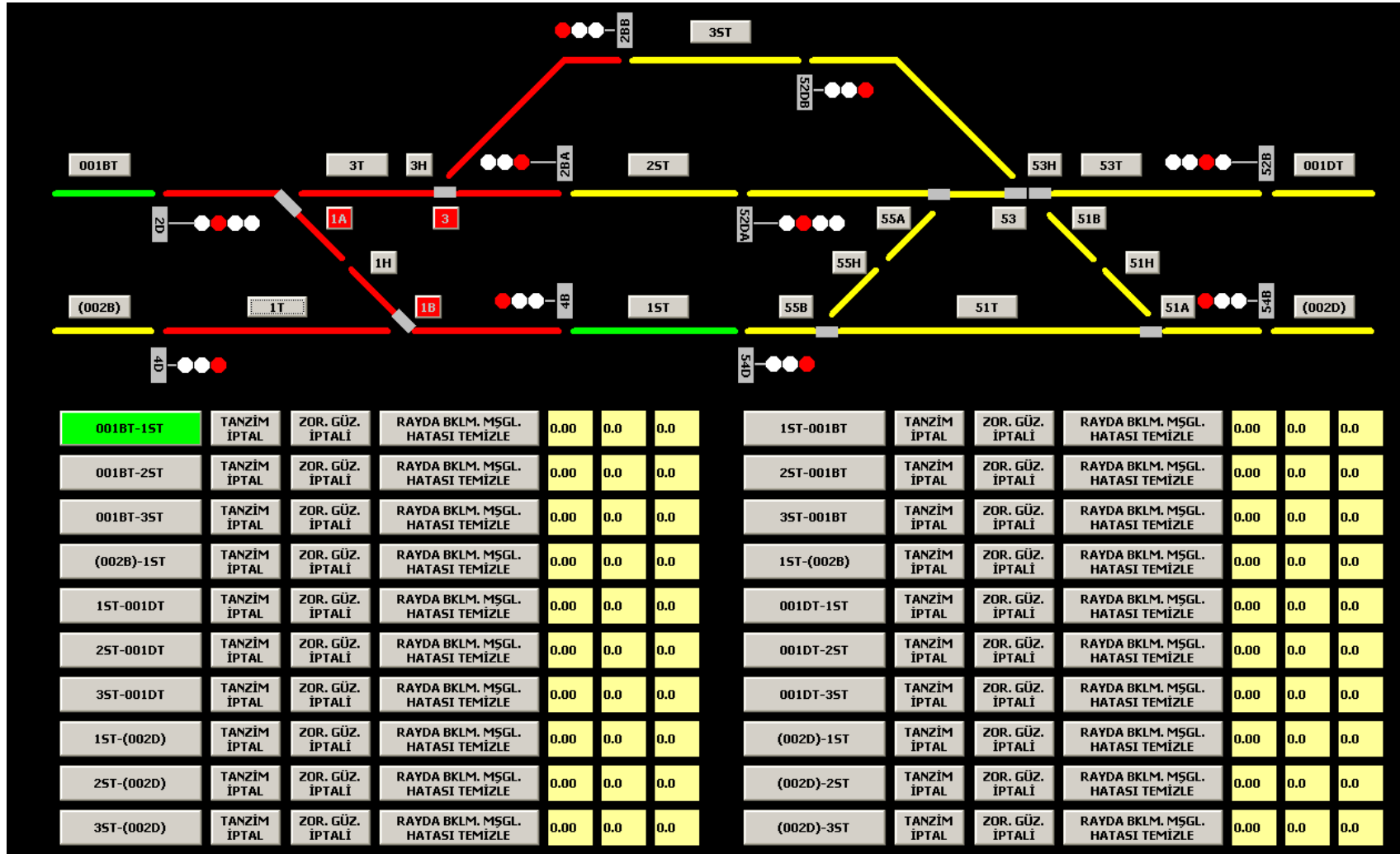
Şekil 7.11 : “(002B)-1ST” güzergâhı için tanzim isteğinin reddedilmesi (makas hatası).

Şekil 7.12’de “001BT-1ST” rotası için tanzim isteğinin yapıldığı durum ele alınmıştır. Başkaca çakışan herhangi bir rota olmadığı, “3T”, “1T”, “1ST” ray devrelerinde tren bulunmadığı ve güzergâhtaki makasların hiçbirisi hatalı durumda olmadığı için bu tanzim isteği kabul edilmiştir. Tanzim isteği kabul edildiği için güzergâhın isminin yazılı olduğu buton yeşil yakılır. Ayrıca “3T”, “1T” ve “1ST” ray devreleri kilitlenir ve bu ray devreleri de yeşil yakılır. Güzergâh üzerindeki makasların trenin ilerlemesine izin verecek konuma ayarlanıp kilitlenmesi gerekmektedir. Bu nedenle “1” no’lu makas “ters” konuma, “3” no’lu makas ise olduğu konumda değişiklik yapılmadan kilitlenmiştir. Henüz “1ST-(002D)” ve “1ST-001DT” güzergâhları için tanzim yapılmamış olduğundan “54D” sinyal lambası “kırmızı” bildirim vermektedir. Çizelge 7.1’de verilen anlaşılan tablosuna bakıldığında “54D” sinyal lambası “kırmızı” bildirim verdiği için “2D” sinyal lambasının “sarı üzeri sarı” bildirim vermesi gerektiği anlaşılmaktadır. Bunun dışında güzergâh üzerinde olmayan diğer tüm sinyal lambaları “kırmızı” bildirim vermelidir. Bu bilgiler doğrultusunda Şekil 7.12’deki sinyal lambalarının doğru bildirimleri verdiği görülmektedir. Artık trenin “001BT-1ST” rotası üzerinde ilerlemesi için gerekli bütün işlemler yerine getirilmiştir. Şekil 7.13’te trenin “001BT-1ST” rotası üzerindeki ilerleyişi gösterilmektedir. Tren “001BT” ray devresini terk edip tamamen “3T” ray devresine geldiği anda “2D” sinyal lambası “kırmızı” bildirim vermektedir. Böylece arkadan gelebilecek bir trenin aynı güzergâha geçişi engellenmektedir. Bundan sonra tren sırasıyla “3T” ray devresinden “1T” ray devresine ve “1T” ray devresinden “1ST” ray devresine doğru ilerler. Tren tamamen “1ST” ray devresine geldiğinde ise tanzim sonlandırılır. Tanzimin sonlanması ile birlikte “3T”, “1T” ve “1ST” ray devrelerinin ve “1” ile “3” no’lu makasların kilitleri çözülür (Şekil 7.14).

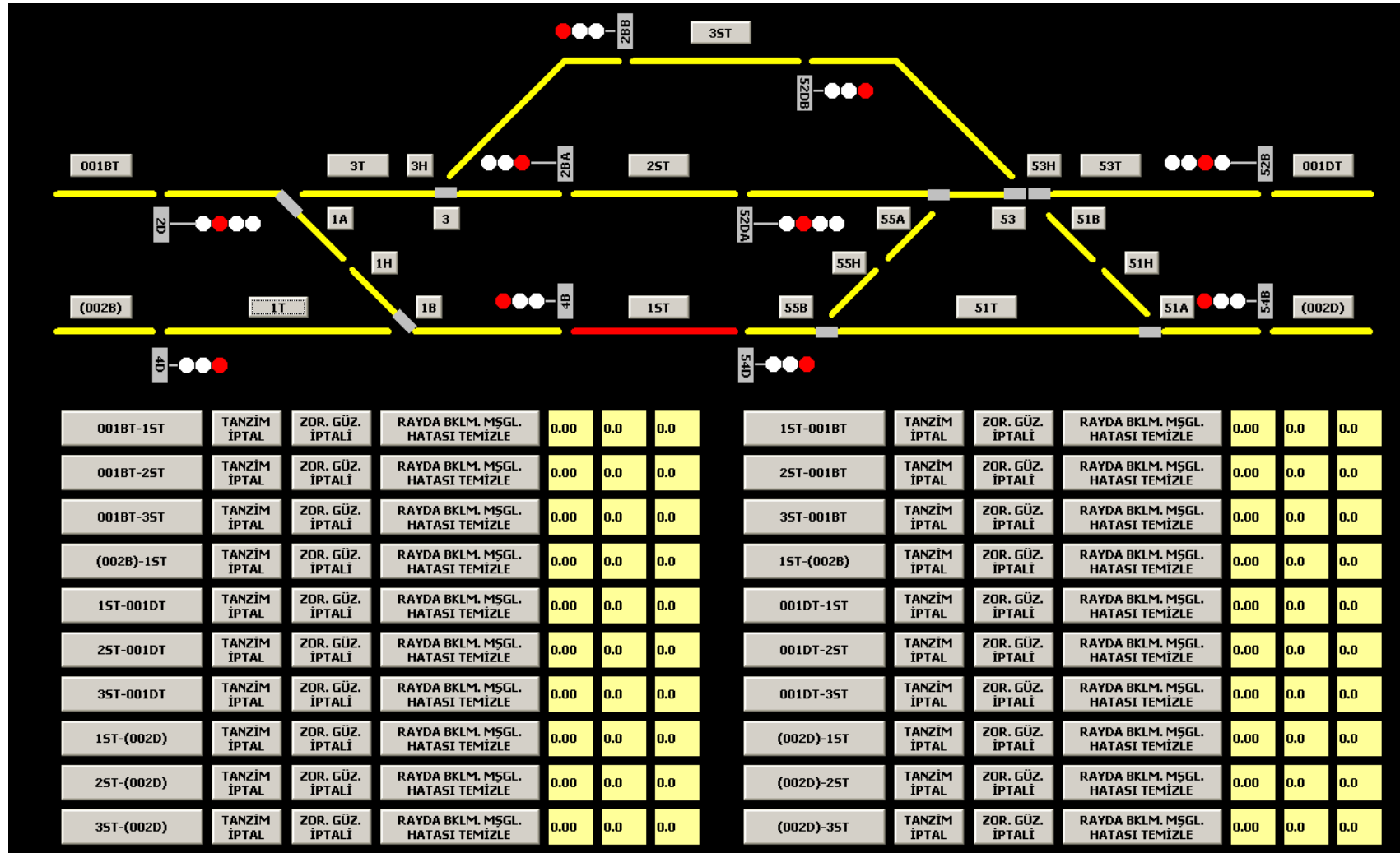
SCADA uygulamasında tanzim isteğinin ardışık güzergâhlar için de yapılabilmesine olanak sağlanmıştır. Şekil 7.15’te birbirini izleyen “001BT-1ST” ve “1ST-001DT” güzergâhları için aynı anda yapılan tanzim isteklerinin kabul edildiği durum gösterilmiştir.



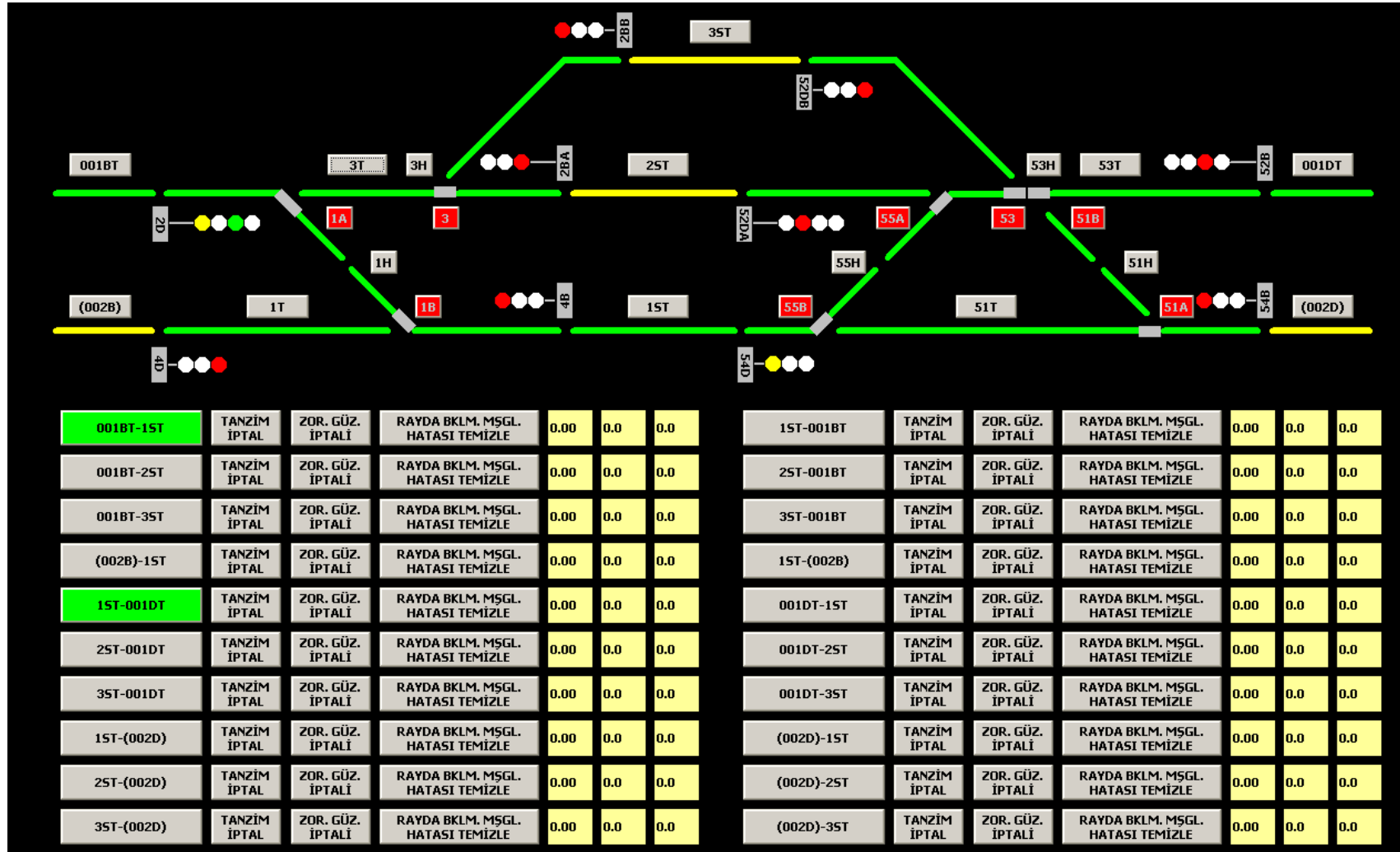
Şekil 7.12 : “001BT-1ST” güzergâhı için tanzim isteğinin kabul edilmesi.



Şekil 7.13 : “001BT-1ST” güzergâhında trenin ilerlemesi.

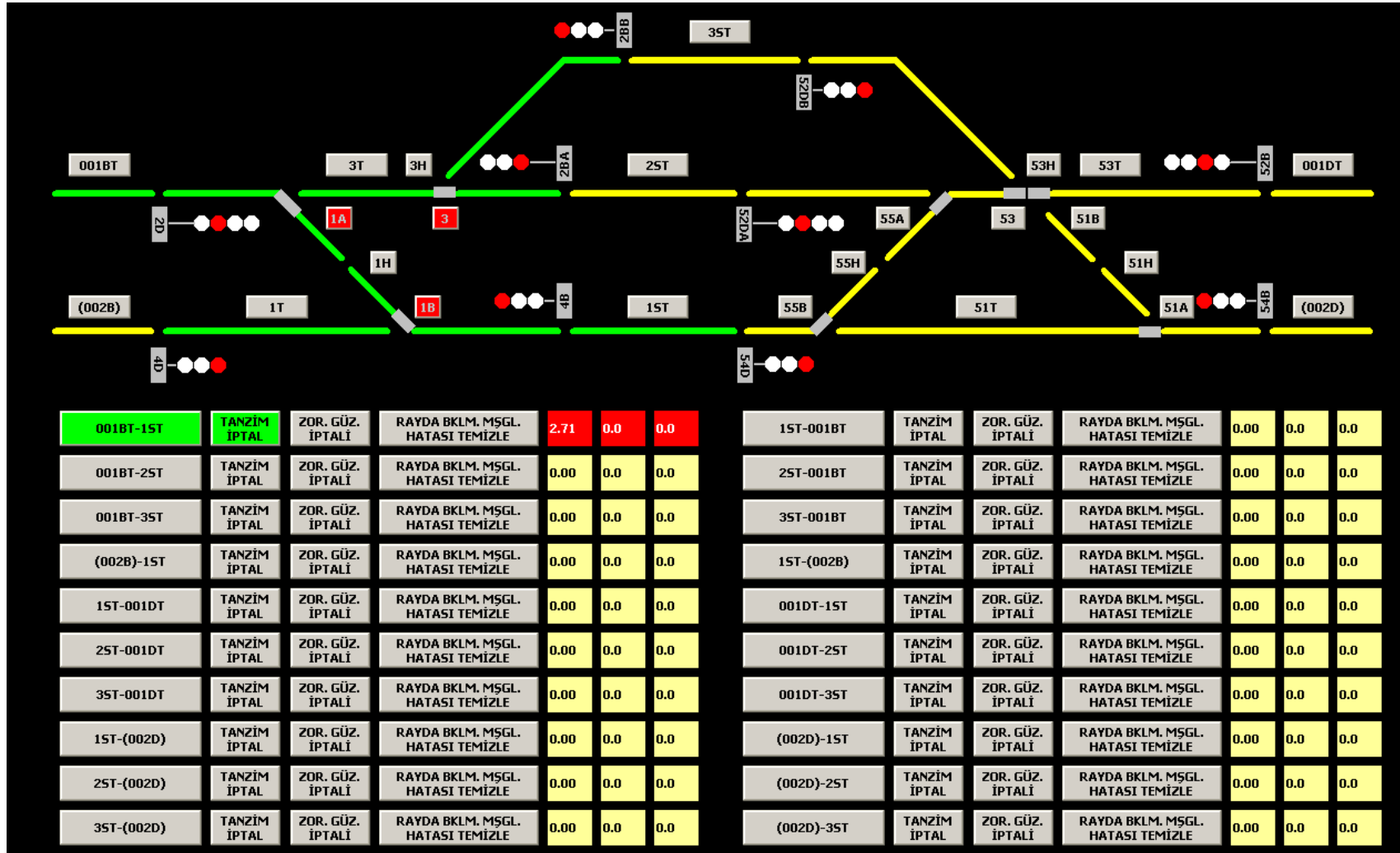


Şekil 7.14 : “001BT-1ST” güzergâhı için yapılan tanzimin sonlanması.

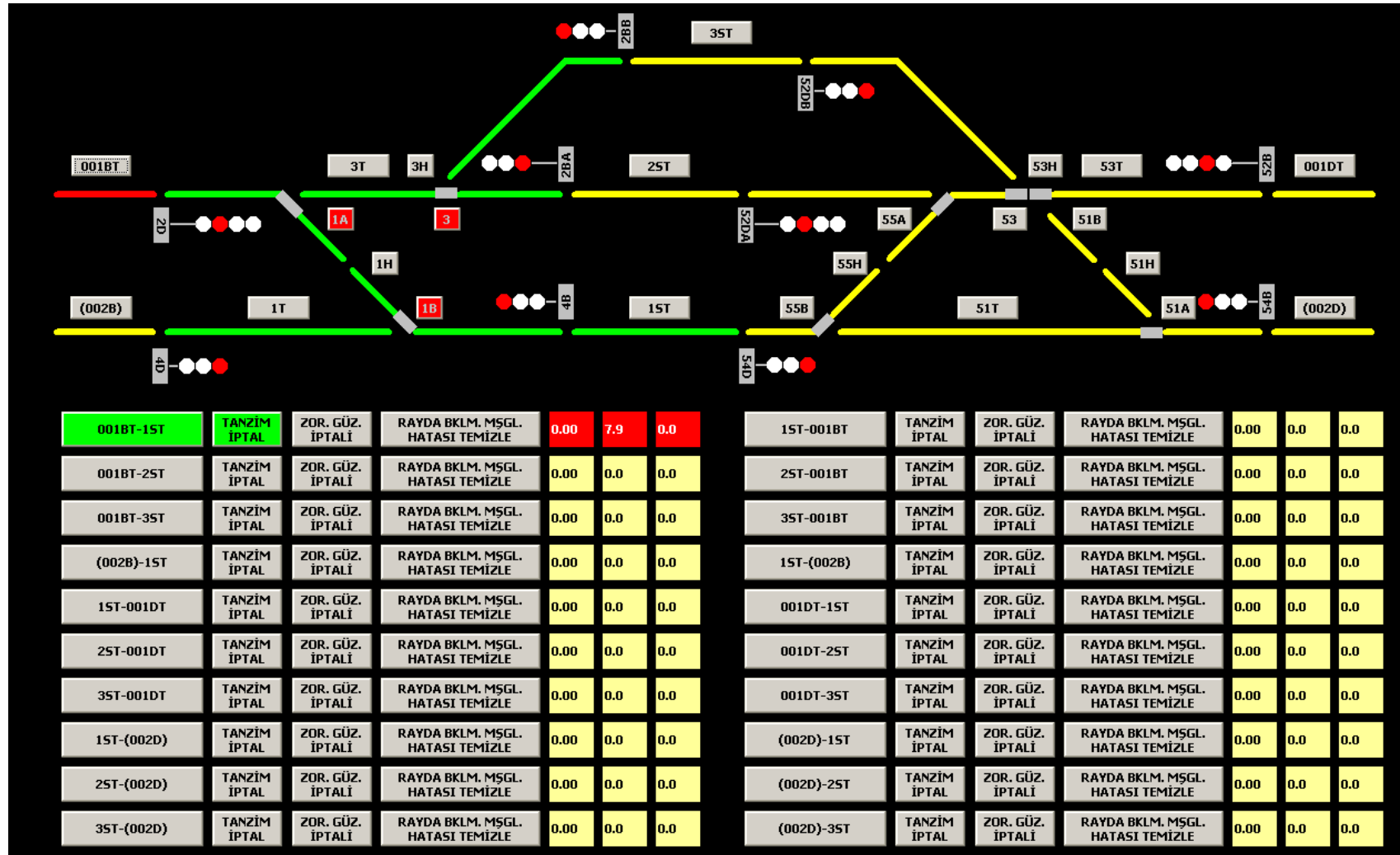


Şekil 7.15 : “001BT-1ST” ve “1ST-001DT” ardışık güzergâhları için yapılan tanzim isteği.

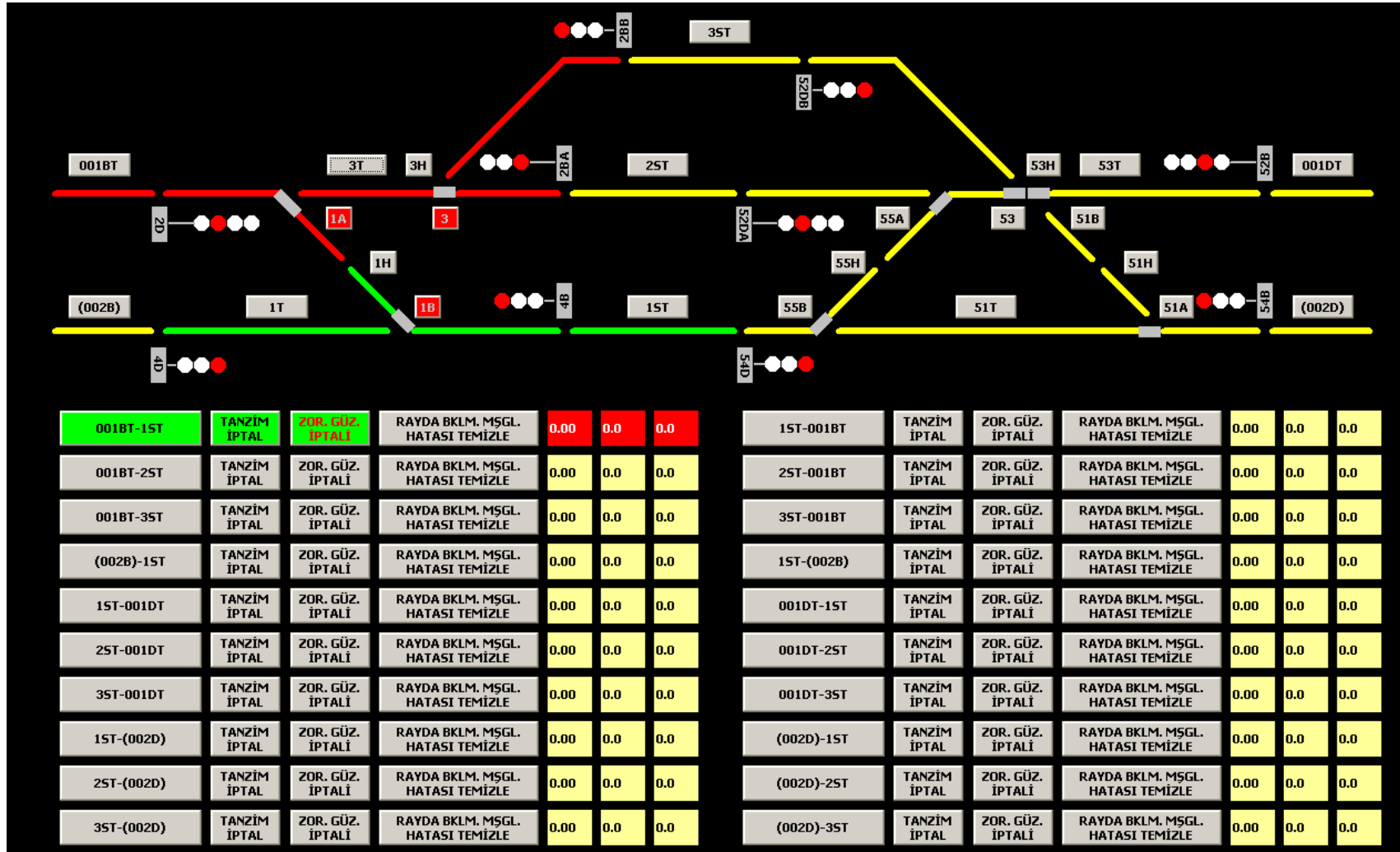
Şekil 7.16’da “001BT-1ST” rotası için kabul edilmiş olan tanzimin iptal edildiği durum ele alınmıştır. Tanzim iptal butonuna basıldığında öncelikle “2D” sinyal lambasının “kırmızı” bildirim vermesi sağlanır. Böylelikle güzergâha girecek olan trenlere tanzimin iptal edildiği belirtilmiş olur. Bu durumdayken ilk zamanlayıcı çalıştırılır ve 5sn’den geriye doğru sayılmaya başlanır. Bu süre normalde 30 sn’dir, ancak SCADA uygulamasında uzun süre beklenmemesi için 5 sn. olarak seçilmiştir. Tren “001BT” ray devresine 5 sn’lik süre içinde giriş yapmazsa tanzim sonlandırılır. Eğer 5 sn. dolmadan tren “001BT” ray devresine giriş yaparsa güzergâh kurulu bırakılır. Bu durumda ikinci zamanlayıcı çalıştırılır ve 10 sn’den geriye doğru sayılmaya başlanır (Şekil 7.17). Bu süre normalde 180 sn’dir, ancak SCADA uygulamasında uzun süre beklenmemesi için 10 sn. olarak seçilmiştir. Tren “3T” ray devresine 10 sn’lik süre içinde giriş yapmazsa tanzim sonlandırılır. Eğer 10 sn. dolmadan tren “3T” ray devresine giriş yaparsa “güzergâhta beklenmedik meşguliyet” hatası oluşur (Şekil 7.18). Bu durumda kullanıcı tarafından “zorunlu güzergâh iptal” isteği yapılmalıdır. Bu istek yapılana kadar güzergâh kurulu bırakılır. “Zorunlu güzergâh iptal” isteği için ilgili butona basıldığında üçüncü zamanlayıcı çalıştırılır ve 15 sn’den geriye doğru sayılmaya başlanır (Şekil 7.19). Bu süre normalde 360 sn’dir, ancak SCADA uygulamasında uzun süre beklenmemesi için 15 sn. olarak seçilmiştir. 15 sn’lik süre içinde tren “1ST” ray devresine gelip güzergâhı tamamlarsa tanzim sonlandırılır. Aksi takdirde “ray devrelerinde beklenmedik meşguliyet” hatası oluşur (Şekil 7.20). Bu durumda tanzim sonlandırılır ancak hatanın giderilebilmesi için trenin tamamen “1ST” ray devresine gelmesi beklenir. Tren “1ST” ray devresine gelip güzergâhı tamamladığında ise “ray devrelerinde beklenmedik meşguliyet” hatası ilgili butona basılarak giderilir. Böylelikle tanzim güvenli bir şekilde sonlandırılır ve güzergâh sonraki tanzim istekleri için hazır hale getirilmiş olur.



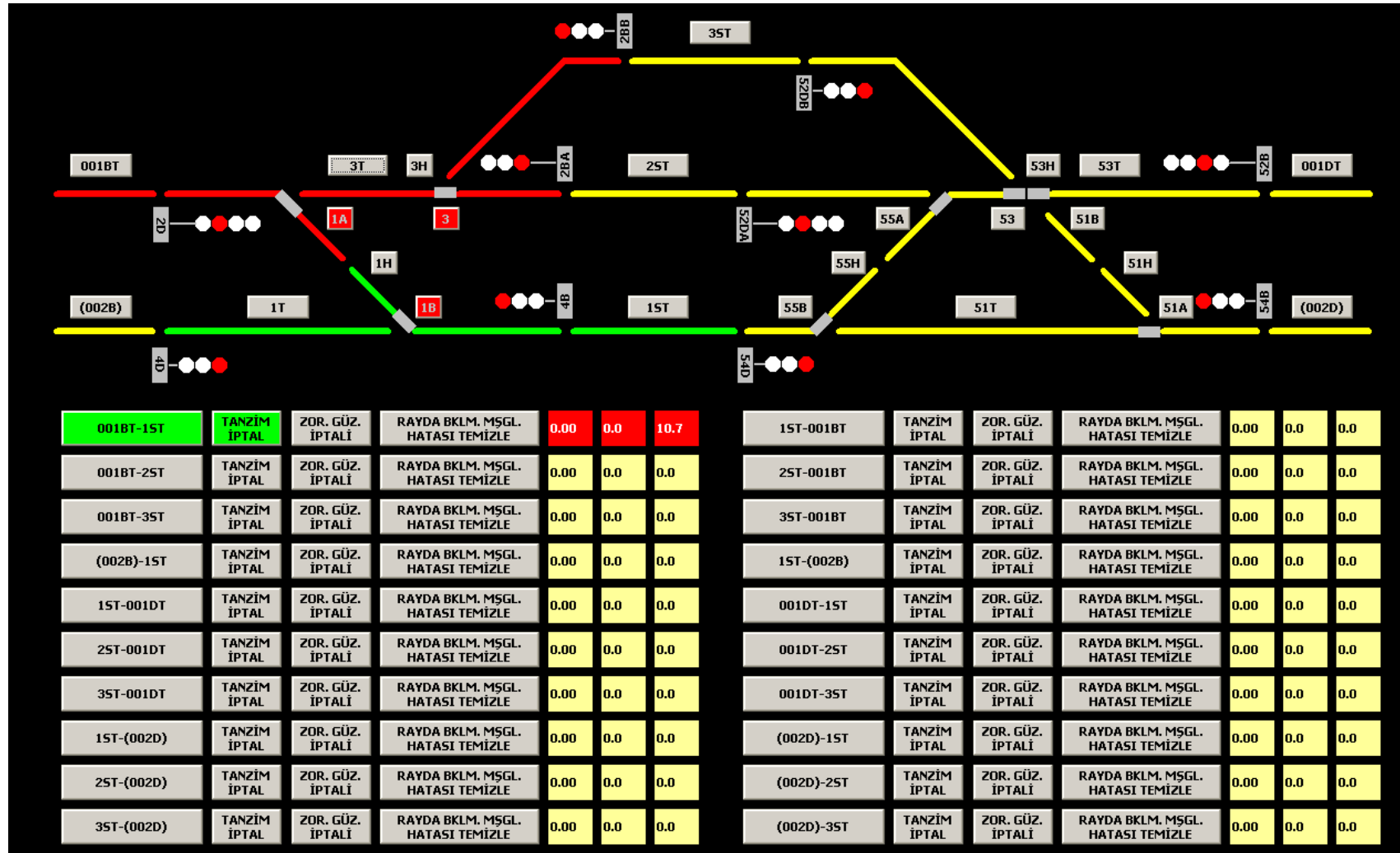
Şekil 7.16 : “001BT-1ST” güzergâhı için yapılmış tanzimin iptal edilmesi (1).



Şekil 7.17 : “001BT-1ST” güzergâhı için yapılmış tanzimin iptal edilmesi (2).



Şekil 7.18 : “001BT-1ST” güzergâhı için zorunlu güzergâh iptalinin yapılması (1).



Şekil 7.19 : “001BT-1ST” güzergâhı için zorunlu güzergâh iptalinin yapılması (2).

8 SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında raylı ulaşım sistemlerine ait anlaşılan sistemlerinin tasarımı ve gerçekleşmesi için bir yöntem önerilmiş ve bir istasyonun verileri kullanılarak benzetimi yapılan bir SCADA uygulaması ile bu yöntemin uygulanabilirliği, geçerliliği ve doğruluğu gösterilmiştir. Petri Ağları'ndan yararlanılarak geliştirilen bu yöntemin tasarım sürecinde daha etkin kullanımı için PA'nın matematiksel modeli üzerinde bazı düzenlemeler yapılmıştır. PA'nın modelleme yeteneğinin artırılması amacıyla tanımlanmış yaylar ele alınmış ve bu yayların tümünü kapsayan bir PA'nın matematiksel modeli için bilgisayar ortamında kolayca işlem yapmaya olanak sağlayan yeni bir "genelleştirilmiş durum denklemi" yapısı önerilmiştir. Bu durum denkleminin hızlı, etkin ve daha az hatayla elde edilmesi için bir yazılım geliştirilmiş, böylece bütün yayları kapsayan bir PA'nın durum denklemlerinin hızlı bir şekilde elde edilmesi sağlanmıştır.

Genelleştirilmiş durum denklemleri elde edilen PA'ya ilişkin PLC programının elde edilmesi için de bir yöntem önerilmiş ve bu işlemin bilgisayar ortamında daha az hatayla, hızlı ve etkin bir biçimde yapılabilmesi amacıyla bir yazılım geliştirilmiştir. Bu yazılım kullanılarak, genelleştirilmiş durum denklemleri verilen bir PA'ya ilişkin PLC programının IEC 61131-3 endüstriyel programlama standardı ile uyumlu biçimde elde edilmesi sağlanmıştır. IEC 61131-3 standardında tanımlı STL veya SCL dillerinde oluşturulan bu programlar PLC programlama editörü üzerinden PLC'ye yüklenebilmektedir.

Önerilen yöntem ve geliştirilen yazılımlar kullanılarak, bir demiryolu istasyonuna ilişkin tüm bileşenlerin tasarımı gerçekleşmiştir. Tasarım sürecinin ilk adımı olan PA modelinin oluşturulmasında Adapazarı Mithatpaşa istasyonuna ilişkin, trafik kumanda merkezi, saha elemanları, anlaşılan tablosu, güzergâh tanzimi, güzergâh iptali, zorunlu güzergâh iptali ile ilgili tüm isteklerden yararlanılmıştır.

Demiryolu sinyalizasyon sisteminin en temel ve önemli bileşeni olan anlaşıman sisteminin PA ile modellenmesi için modüler bir yapı kullanılmış ve anlaşıman sistemi bir takım alt sistemlere bölünmüştür. Tanzim, makas, tren takip, cüce sinyal, yüksek sinyal gibi alt sistemlerin her biri için ayrı ayrı PA modelleri oluşturulmuş ve önerilen yöntemler ve geliştirilen yazılımlar kullanılarak her bir alt sisteme ait endüstriyel programlama standartlarına uyumlu PLC program blokları elde edilmiştir. Gerçeklenen anlaşıman sisteminin geçerliliğinin ve doğruluğunun sınanması amacıyla Mithatpaşa istasyonunun ana bölümüne ilişkin bir benzetim programı uygulaması bir SCADA yazılım ortamında tasarlanmış ve gerçekleştirilmiştir. Bu SCADA uygulaması üzerinde fonksiyonel özellikler ve sınaama süreçleri gözlemlenmiştir. Elde edilen sonuçlar PLC ile gerçeklemek üzere PA ile yapılan tasarımın raylı ulaşım anlaşıman sistemleri için geçerli ve uygulanabilir olduğunu göstermiştir.

Bu tezde önerilen yöntem ve geliştirilen yazılımlar için ileriye dönük olarak Petri Ağları'nın grafiksel olarak çizilmesi ve bu çizimden IEC 61131-3 endüstriyel programlama standardında program üreten bir tümleşik yazılım geliştirilmesi konusu ele alınabilir. Bu konuya uygun bir çözüm bulunursa, PA modeli oluşturulması dışındaki süreçlerde insandan kaynaklanan hatalar en aza indirilebilir. Bir raylı sisteme ilişkin anlaşıman sisteminin tasarımında en önemli konu matematiksel modelin geçerliliği ve doğruludur. Genellikle sezgisel bir yaklaşımla oluşturulan ve başarımı büyük ölçüde tasarımcının deneyimine bağlı olan otomat veya Petri Ağları gibi modellerin doğruluk ve geçerlilik durumlarının belirlenmesi çok önemlidir. Bir anlaşıman sistemine ilişkin PA modelinin geçerlilik ve doğruluk sınaamaları için gerekli senaryoları üreten ve bu senaryolara göre sonucu bildiren yazılımlar geliştirilerek tasarım sürecine önemli katkılar yapılabilir.

KAYNAKLAR

- [1] **Fay, A.** (2000). A Fuzzy Knowledge-based System for Railway Traffic Control, *Engineering Applications of Artificial Intelligence*, 13, 719–729.
- [2] **Dotoli, M., & Fanti, M.P.** (2006). An Urban Traffic Network Model via Timed Coloured Petri Nets, *Control Engineering Practice*, 14, 1213–1229.
- [3] **Di Febbraro, A. & Sacco, N.** (2004). On Modelling Urban Transportation Networks via Hybrid Petri Nets, *Control Engineering Practice*, 12, 1225–1239.
- [4] **Kaakai, F., Hayat, S., & El Moudni, A.** (2007). A Hybrid Petri Nets-based Simulation Model for Evaluating the Design of Railway Transit Stations, *Simulation Modelling Practice and Theory*, 15, 935–969.
- [5] **Fanti, M.P., Giua, A., & Seatzu, C.** (2006). Monitor Design for Colored Petri Nets: An Application to Deadlock Prevention in Railway Networks, *Control Engineering Practice*, 14, 1231–1247.
- [6] **Bartkevicius, S., Bagdonas, V., Juraska, M., & Sarkauskas, K.** (2005). Train Traffic Simulation with Coloured Petri Nets and Schedule Optimisation, *Elektronika ir Elektrotechnika*, 59 (3), 18–23.
- [7] **Ren, X., & Zhou, M.** (1995). Tactical Scheduling of Rail Operations: A Petri Net Approach, *IEEE International Conference on Intelligent Systems for the 21st Century*, 4, s. 3087–3092. Vancouver, BC.
- [8] **Url-1** <<http://vast.uccs.edu/~tboult/CS330/NOTES/CS330-Specification.ppt>>, erişim tarihi 15.01.2014.
- [9] **Cassandras, C.G., & Lafortune, S.** (1999). *Introduction to Discrete Event Systems*. Dordrecht, Netherlands: Kluwer Academic Publishers.
- [10] **Moody, J.O., & Antsaklis, P.J.** (1998). *Supervisory Control of Discrete Event Systems Using Petri Nets*. Norwell, MA, USA: Kluwer Academic Publishers.
- [11] **Url-2** <http://www.wis.win.tue.nl/~wvdaalst/workflowcourse/slides/pn2refresher_long.ppt>, erişim tarihi 15.01.2014.
- [12] **Cassandras, C.G., & Lafortune, S.** (2008). *Introduction to Discrete Event Systems (2nd Ed.)*. New York, NY, USA: Springer.
- [13] **Reisig, W.** (2013). *Understanding Petri Nets: Modeling Techniques, Analysis Methods*. Berlin, Germany: Springer-Verlag.
- [14] **Url-3** <<http://people.unipmn.it/bobbio/BIBLIO/PAPERS/ANNO90/kluwerpetrinet.pdf>>, erişim tarihi 15.01.2014.
- [15] **Dahl, O.M.** (2005). *Using Colored Petri Nets in Penetration Testing*. (Yüksek Lisans Tezi). Gjøvik University College, Department of Computer Science and Media Technology, Norway.

- [16] **Hruz, B., & Zhou, M.C.** (2007). *Modeling and Control of Discrete-event Dynamic Systems with Petri Nets and Other Tool*. London, UK: Springer-Verlag.
- [17] **Uzam, M.** (1998). *Petri-net-based Supervisory Control of Discrete Event Systems and Their Ladder Logic Diagram Implementations*. (Doktora Tezi). University of Salford, Telford Research Institute, Salford.
- [18] **Durmuş, M. S., & Söylemez, M. T.** (2008). Petri Ağları ile Demiryolu Ankleşman ve Sinyalizasyon Tasarımı, *ELECO'08*, (s. 447–451). Bursa.
- [19] **Jimenez, I., Lopez, E., & Ramirez, A.** (2001). Synthesis of Ladder Diagrams from Petri Nets Controller Models, *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, (s. 225–230). Mexico City.
- [20] **Moreira, M. V., Botelho, D. S., & Babilio, J. C.** (2009). Ladder Diagram Implementation of Control Interpreted Petri Nets: A State Equation Approach, *IFAC Discrete-Event System Design*, (s. 85–90). Spain.
- [21] **Uzam, M., & Jones, A. H.** (1998). Discrete Event Control System Design Using Automation Petri Nets and Their Ladder Diagram Implementation, *The International Journal of Advanced Manufacturing Technology*, 14 (10), 716–728.
- [22] **Uzam, M., Jones, A. H., & Yücel, İ.** (2000). Using a Petri Net Based Approach for the Real Time Supervisory Control of an Experimental Manufacturing System, *The International Journal of Advanced Manufacturing Technology*, 16, 498–515.
- [23] **Hasdemir, İ.T., Kurtulan, S., & Gören, L.** (2008). An Implementation Methodology for Supervisory Control Theory, *The International Journal of Advanced Manufacturing Technology*, 36, 373–385.
- [24] **Url-4** <http://emezun.meb.gov.tr/doc/tanitimmodulu/35-Rayli_Sistemler.pdf>, erişim tarihi 15.01.2014.
- [25] **Hall, S.** (2001). *Modern Signalling Handbook*. England: Ian Allan Publishing.
- [26] **Kuepper, G.J.** (1999). 150 Years of Train-Disasters - Practical Approaches for Emergency Responders, *9-1-1 Magazine*, September/October issue, 30–33.
- [27] **Durmuş, M. S., Yıldırım, U., & Söylemez, M. T.** (2011). Demiryolu Sinyalizasyon Tasarımında Fonksiyonel Güvenlik ve Ayrık Olay Sistem Yaklaşımı, *Elektrikli Ulaşım Sistemleri Sempozyumu ve Sergisi (EUSİS 2011)*. Bursa.
- [28] **Yıldırım, U., Durmuş, M. S., Durmuş, M. S., & Söylemez, M. T.** (2010). Demiryolu Hemzemin Geçitleri için Hatada-Güvenli Sinyalizasyon ve Ankleşman Tasarımı, *Otomatik Kontrol Ulusal Toplantısı (TOK'10)*, (s. 235–240). Gebze.
- [29] **Sonat, A.** (2010). *Raylı Ulaşım Sistemlerinde Ankleşman Algoritması Tasarımı ve Otomat Yaklaşımı ile Otomatik Kod Üretme*. (Yüksek Lisans Tezi). İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

- [30] **Yıldırım, U., Durmuş, M. S., Durmuş, M. S., & Söylemez, M. T.** (2011). Demiryolu Sinyalizasyon Sistemleri için Otomatik Anlaşman Tablosu Oluşturulması, *Elektrikli Ulaşım Sistemleri Sempozyumu ve Sergisi (EUSİS 2011)*. Bursa.
- [31] **Eriş, O.** (2010). *Demiryolu Anlaşman Sistemlerinin PLC ile Gerçeklenmesi*. (Yüksek Lisans Tezi). İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- [32] **Babacan, V. K.** (2010). *Raylı Sistemlerde Sinyalizasyon Tekniğine Giriş*. Haydarpaşa, İstanbul.
- [33] **Url-5** <http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Ray%20Devreleri.pdf>, erişim tarihi 15.01.2014.
- [34] **Eriş, O., & Kurtulan, S.** (2011). Application of Functional Safety on Railways Part III: Development of a Main Controller, *Proceedings of 2011 8th Asian Control Conference (ASCC)*, (s. 1102–1104). Taiwan.
- [35] **Söyler, H., & Açıkbaş, S.** (2007). Raylı Toplu Taşımada Sinyalizasyon Sistemleri, *Elektrik-Elektronik-Bilgisayar Mühendisliği 11. Ulusal Kongresi*. İstanbul.

ÖZGEÇMİŞ



Ad Soyad: Cem BAŞKOCAGİL

Doğum Yeri ve Tarihi: Ereğli/KONYA, 25.01.1979

E-Posta: baskocagil@yahoo.com

Lisans: İstanbul Teknik Üniversitesi, Kontrol ve Bilgisayar Mühendisliği
(1997–2001)

Yüksek Lisans: İstanbul Teknik Üniversitesi, Kontrol ve Otomasyon Mühendisliği
(2001–2004)

Mesleki Deneyim ve Ödüller:

- Gömülü Yazılım Geliştirme Uzmanı, TÜBİTAK–BİLGEM (2007–)
- Ağ Güvenliği Uzmanı, TÜBİTAK–BİLGEM (2001–2007)

Yayın ve Patent Listesi:

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

- Başkocagil, C., & Kurtulan, S. (2011). Generalized State Equation for Petri Nets, *WSEAS Transactions on Systems*, 10, 295–305.

