



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟΣΧΟΛΗ
ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ
ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΣΥΣΤΗΜΑΤΑ ΑΥΤΟΜΑΤΙΣΜΟΥ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**«Ανάπτυξη προγράμματος ελέγχου plc για
ευέλικτο κύτταρο κατεργασιών με βάση δίκτυα
Coloured Petri»**

Φοιτήτρια : Παπάζογλου Κατερίνα
Επιβλέπων : Γεώργιος Βοσνιάκος
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2016

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Γεώργιο Βοσνιάκο, καθηγητή του ΕΜΠ, για την ανάθεση της διπλωματικής εργασίας καθώς και την επιστημονική του συμβολή κατά τη διεκπεραίωσή της.

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η μοντελοποίηση της παραγωγικής διαδικασίας ενός ευέλικτου συστήματος κατεργασιών με τη βοήθεια ενός μαθηματικά ορισμένου εργαλείου που ονομάζεται δίκτυα Petri.

Στο πρώτο κεφάλαιο εισάγεται η έννοια της ευελιξίας στα αυτόματα κύτταρα κατεργασιών και παρουσιάζονται τα βασικά χαρακτηριστικά τους και τα κίνητρα για την υιοθέτησή τους στη βιομηχανία. Στη συνέχεια παρουσιάζεται η διάταξη του ευέλικτου συστήματος κατεργασιών που πρόκειται να μελετηθεί. Παράλληλα, περιγράφεται η διαδικασία παραγωγής των τεμαχίων, η κατανομή των εργασιών στα επιμέρους στοιχεία του κυττάρου και η επιβολή προτεραιοτήτων που θα τηρηθεί μεταξύ των εκτελούμενων εργασιών.

Το δεύτερο κεφάλαιο αποτελεί μια γενική εισαγωγή στη θεωρία των δικτύων Petri στους κανόνες εκτέλεσης, τις δυνατότητες μοντελοποίησης και τις επεκτάσεις τους. Ακόμα, οι μέθοδοι ανάλυσης των ιδιοτήτων τους, αλλά και οι περιπτώσιολογικές μελέτες που παρουσιάζονται στη σχετική βιβλιογραφική έρευνα, αποδεικνύουν τη χρησιμότητά τους στον προγραμματισμό συστημάτων διακριτών γεγονότων όπως τα ευέλικτα κύτταρα κατεργασιών. Παράλληλα, αναφέρονται και ορισμένες κατηγορίες των δικτύων Petri ως επέκταση των δυνατοτήτων των κλασικών μοντέλων. Στην τελευταία ενότητα του κεφαλαίου διερευνάται ο τρόπος μοντελοποίησης του συστήματος της παρούσας εργασίας με δίκτυα Petri. Πιο συγκεκριμένα γίνεται μια προμελέτη σχετικά με τον αριθμό των δομικών στοιχείων αλλά και την κατηγορία των δικτύων που επιλέγεται στην τρέχουσα εφαρμογή.

Στο τρίτο κεφάλαιο παρουσιάζεται το πρόγραμμα CPN tools που χρησιμοποιήθηκε για την μοντελοποίηση του συστήματος. Δίνονται βασικές πληροφορίες σχετικά με το γραφικό περιβάλλον, τη δήλωση των στοιχείων και μεταβλητών του μοντέλου και τις δυνατότητες προσομοίωσης του συγκεκριμένου προγράμματος. Η τελευταία ενότητα του κεφαλαίου αναφέρεται στα εργαλεία υπολογισμού του χώρου καταστάσεων του μοντέλου CPN και παράλληλα εξηγείται πως τα αποτελέσματα αξιοποιούνται για την εξακρίβωση των ιδιοτήτων του πραγματικού συστήματος.

Στο τέταρτο κεφάλαιο δίνεται αναλυτικά βήμα-βήμα η μοντελοποίηση του μελετούμενου συστήματος με Coloured δίκτυα Petri χρησιμοποιώντας το πρόγραμμα CPN tools. Παρουσιάζεται το πλήρες μοντελοποιημένο σύστημα πριν και μετά την προσομοίωση επαληθεύοντας την αναμενόμενη τελική κατάσταση του συστήματος. Ακολουθεί η ανάλυση του χώρου καταστάσεων του μοντέλου που επιβεβαιώνει τη σωστή λειτουργία των πόρων και αποθηκευτικών χώρων καθώς και την απουσία ανεπιθύμητων κολλημάτων και αδιεξόδων του συστήματος.

Στο πέμπτο κεφάλαιο πραγματοποιείται η σύνδεση του μοντέλου των Coloured Petri δικτύων με τη γλώσσα λογικών διαγραμμάτων τύπου σκάλας (LLD) που χρησιμοποιείται στους εμπορικά διαθέσιμους προγραμματιζόμενους λογικούς ελεγκτές (PLC). Δίνεται συνοπτική βιβλιογραφική μελέτη που αναφέρεται σε μετατροπή απλών μοντέλων Petri σε προγράμματα LLD με στόχο τη διατήρηση των ιδιοτήτων που έχουν διερευνηθεί μέσω της προηγούμενης μορφής. Με βάση αυτή τη βιβλιογραφία, έγινε μια προσπάθεια μετάφρασης των μεταβάσεων του CPN μοντέλου του συστήματος κατεργασιών σε rungs της γλώσσας LLD. Κατά τη μετατροπή αυτή αναφέρονται και οι απαραίτητες τροποποιήσεις και κυρίως ο τρόπος ενσωμάτωσης των σημάτων εισόδου/εξόδου του ελεγκτή που δεν εμπεριέχονται στα μοντέλα Petri. Τέλος, εξετάστηκε η ορθότητα του συνολικού προγράμματος LLD που πρέκυψε, για διάφορα σενάρια λειτουργίας του συστήματος, χρησιμοποιώντας εργαλείο προσομοίωσης του λογισμικού *Cx-programmer* της εταιρίας *Omron*.

Summary

The objective of this thesis is modeling the production process of a flexible manufacturing system (FMS) by applying a mathematically based tool called Petri Nets.

The first chapter introduces the idea of flexibility in automated manufacturing cells as well as their basic features and benefits from integrating this type of systems in industry. The final unit of the chapter presents the layout of the manufacturing cell under study and sets the sequence and priority of several milling, turning and transferring jobs comprising the production process.

The second chapter is an overview of Petri Net theory, their modeling power, classes and extensions. Moreover, there is a concise reference to related studies, concerning applications of Petri Nets, which prove their importance in programming discrete event systems. The last section of the chapter deals with the choice of Petri Net class, type and number of structural elements (places and transitions) used to model the studied cell.

The third chapter is a detailed presentation of the computer tool 'CPN tools' which was applied for editing, simulating and analyzing the FMS studied in the current thesis. Firstly, there is analytical information about declaring of variables, composing inscriptions and creating model's graphical structure. The next chapters deal with simulation of the model and calculation of state space. It is also explained how the state space analysis results can be used to verify system's properties.

The fourth chapter is a step by step presentation of modeling of the FMS under study using CPN tools program. The created CPN model is shown before and after simulation, verifying its expected final state. Then, the state space analysis proves accurate functionality of system's resources as well as the absence of unwanted states such as deadlocks.

The fifth chapter deals with the connection of Petri Net theory with Ladder Logic diagrams used in commercial programmable logic controllers (plc). The chapter provides a brief literature research concerning the conversion of Petri Net models into plc programming language. This field of research aims at generating plc programs which maintain all the desirable properties previously checked by Petri Net tool. Based on the existing literature, there was an attempt to translate the transitions of the created CPN model into rungs of ladder logic diagram. During this conversion, all the necessary amendments are stated and most significantly the integration of input and output signals which is not included in Petri Net theory. Finally, the performance and accuracy of the created ladder logic diagram was

checked for several operation scenarios using Cx-programmer simulation tool of Omron company.

Περιεχόμενα

Κεφάλαιο 1: Ευέλικτα Συστήματα Κατεργασιών

1.1 Γενικά

1.1.1 Ορισμός Ευέλικτου Συστήματος Κατεργασιών

1.1.2 Βασικά στοιχεία Ευέλικτου Συστήματος Κατεργασιών

1.1.3 Οφέλη από τη χρήση Ευέλικτων Συστημάτων Κατεργασιών

1.2 Παρουσίαση συστήματος παρούσας εργασίας

1.2.1 Εξοπλισμός διάταξης συστήματος κατεργασιών

1.2.2 Παραγωγική διαδικασία συστήματος

Κεφάλαιο 2: Δίκτυα Petri

2.1 Γενικά

2.1.1 Εισαγωγή στα δίκτυα Petri

2.1.2 Πλεονεκτήματα των δικτύων Petri

2.1.3 Τα δίκτυα Petri στον τομέα της αυτοματοποίησης της παραγωγής

2.2 Θεωρία δικτύων Petri

2.2.1 Βασικά στοιχεία δικτύων Petri

2.2.2 Κανόνες ενεργοποίησης των μεταβάσεων

2.2.3 Μηδενικός έλεγχος

2.2.4 Ιδιότητες δικτύων Petri

2.2.5 Μέθοδοι ανάλυσης δικτύων Petri

2.2.6 Δυνατότες μοντελοποίησης δικτύων Petri

2.3 Επεκτάσεις δικτύων Petri

2.4 Βιβλιογραφική μελέτη εφαρμογών με δίκτυα Petri

2.5 Μοντελοποίηση συστήματος της παρούσας εργασίας με δίκτυα Petri

Κεφάλαιο 3 : Το πρόγραμμα CPN tools

3.1 Εισαγωγή στο πρόγραμμα CPN tools

3.2 Γραφικό περιβάλλον χρήστη προγράμματος CPN tools

3.3 Ορισμός θέσεων μεταβάσεων και τόξων στο CPN tools

3.4 Προσομοίωση στο CPN tools

3.5 Ανάλυση χώρου καταστάσεων (state space) στο CPN tools

Κεφάλαιο 4 : Μοντελοποίηση Ευέλικτου Συστήματος Κατεργασιών με το πρόγραμμα CPN tools

4.1 Δομή του μοντέλου – θέσεις/μεταβάσεις

4.2 Δημιουργία colourset και μεταβλητών

4.3 Εισαγωγή συνθηκών στα τόξα και τις μεταβάσεις του μοντέλου

4.4 Προσομοίωση μοντέλου

4.5 Ανάλυση χώρου καταστάσεων (state space) μοντέλου

Κεφάλαιο 5 : Μετατροπή δικτύων Petri σε διαγράμματα ladder

5.1 Γενικά

5.1.1 Εισαγωγή στους προγραμματιζόμενους λογικούς ελεγκτές

5.1.2 Αρχή λειτουργίας PLC

5.1.3 Προγραμματισμός PLC

5.1.4 Σύνδεση θεωρίας δικτύων Petri με τη λογική των PLC

5.2 Βιβλιογραφική μελέτη

5.3 Μετατροπή μοντέλου CPN παρούσας εργασίας σε διάγραμμα Ladder

5.4 Προσομοίωση προγράμματος ladder

Κεφάλαιο 6 : Συμπεράσματα

6.1 Χρησιμότητα δικτύων Petri

6.2 Αξιολόγηση προγράμματος CPN tools

6.3 Γενικότητα μοντέλου

6.4 Επεκτασιμότητα μοντέλου

6.5 Ευελιξία συστήματος

6.6 Μετατροπή μοντέλου CPN σε γλώσσα plc

6.7 Αντικείμενα για περαιτέρω μελέτη

Κεφάλαιο 1: Ευέλικτα Συστήματα Κατεργασιών

1.1 Εισαγωγή

1.1.1 Ορισμός Ευέλικτου Συστήματος Κατεργασιών:

Σύστημα που ασχολείται με την επεξεργασία και αυτόματη ροή υλικού χρησιμοποιώντας μηχανές αριθμητικού ελέγχου (CNC) , κύτταρα συναρμολόγησης, βιομηχανικά ρομπότ, μηχανές ταυτοποίησης/επιθεώρησης και ενσωματωμένα συστήματα διαχείρισης υλικού και αποθήκευσης ελεγχόμενα από υπολογιστή (Ranky 1983).

Ο όρος ευελιξία στα συστήματα κατεργασιών μπορεί να αναφέρεται σε διαφορετικά θέματα που αφορούν το σύστημα ή την παραγωγή. Έτσι ο όρος ευελιξία μπορεί να έχει διαφορετικές έννοιες:

- Ευελιξία μηχανής. Εκφράζει την ευκολία με την οποία γίνονται οι απαραίτητες αλλαγές σε μια μηχανή ώστε να παραχθεί ένα συγκεκριμένο σύνολο τύπων τεμαχίων.
- Ευελιξία κατεργασίας. Εκφράζει τη δυνατότητα παραγωγής ενός συνόλου τύπων τεμαχίων με διαφορετικούς τρόπους, δηλαδή με διαφορετικές κατεργασίες.
- Ευελιξία προϊόντος. Εκφράζει την ευκολία με την οποία είναι δυνατή η αλλαγή από ένα μείγμα τύπων τεμαχίων σε ένα διαφορετικό μείγμα.
- Ευελιξία διαδρομής . Αναφέρεται στη δυνατότητα εναλλακτικών λύσεων όταν ένα ή περισσότερα σημεία της διαδρομής ενός ή περισσότερων τεμαχίων γίνονται προσπελάσιμα λόγω π.χ. βλάβης μηχανών
- Ευελιξία όγκου παραγωγής. Εκφράζει τη δυνατότητα να λειτουργεί το Σύστημα Κατεργασιών με κέρδος σε όσο το δυνατόν μεγαλύτερο εύρος μεγέθους μερίδων.
- Ευελιξία επέκτασης. Εκφράζει τη δυνατότητα αλλαγών μεγέθους του συστήματος.
- Ευελιξία διεργασιών. Αναφέρεται στη δυνατότητα αλλαγής της διαδοχής των φάσεων κατεργασίας που αποτελούν το φασεολόγιο παραγωγής ενός τεμαχίου
- Ευελιξία παραγωγής. Ουσιαστικά περιλαμβάνει όλα τα είδη ευελιξίας που αναφέρθηκαν και αναφέρεται στο εύρος τύπων τεμαχίων που μπορεί να παραχθεί οικονομικά από ένα Σύστημα Κατεργασιών [2].

1.1.2 Βασικά στοιχεία Ευέλικτου Συστήματος Κατεργασιών:

Τα βασικά και χαρακτηριστικά ενός Συστήματος Κατεργασιών όπως περιγράφονται σε διάφορες πηγές είναι :

Πρωτεύοντα στοιχεία :

- **Ανεξάρτητες μηχανές αριθμητικού ελέγχου** : είναι συνήθως προγραμματιζόμενοι σταθμοί κατεργασιών στους οποίους περιλαμβάνονται κέντρα τόννευσης, κέντρα κατεργασιών, κέντρα διαμόρφωσης/κοπής ελάσματος. Σημαντική είναι και η παρουσία βοηθητικών σταθμών επεξεργασίας όπως κέντρα καθαρισμού και μέτρησης.
- **Αυτόματο σύστημα διαχείρισης υλικού** το οποίο πραγματοποιεί τη μεταφορά του υλικού μεταξύ των σταθμών αλλά και το χειρισμό του σε κάθε σταθμό (φορτο-εκφόρτωση). Η μεταφορά πραγματοποιείται με σταθερές διατάξεις τύπου ταινιόδρομου ή τύπου γερανογέφυρας καθώς και με οχήματα αυτόματης πλοήγησης (Rail Guided Vehicles –RGVs). Χειρισμός σε κάθε σταθμό γίνεται με ρομπότ ή και αυτόματες συσκευές ενσωματωμένες στο σταθμό κατεργασίας. Στα συστήματα χειρισμού του υλικού ανήκουν και οι σταθμοί παλετοποίησης τεμαχίων.
- **Διατάξεις αποθήκευσης** υπάρχουν στην είσοδο , έξοδο αλλά και σε ενδιάμεσα στάδια της παραγωγικής διαδικασίας. Μπορεί να είναι αυτόματα εξυπηρετούμενα σύνολα με ράφια (Automatic Storage and Retrieval Systems), σταθερά ή περιστρεφόμενα τραπέζα με λίγες θέσεις αποθήκευσης ή και απλά ράφια εξυπηρετούμενα από προσωπικό.
- **Ολοκληρωμένη μέθοδος ελέγχου** που συντονίζει τις λειτουργίες των εργαλειομηχανών και των συστημάτων διαχείρισης υλικού ώστε να επιτύχει ευελιξία. Παράλληλα, αντικείμενο του σχεδιασμού ελέγχου του συστήματος είναι η κατανομή των εργασιών στο κύτταρο στοχεύοντας στην ελαχιστοποίηση του συνολικού μη παραγωγικού χρόνου, τη μεγιστοποίηση του βαθμού χρήσης των μηχανών και άλλα κριτήρια κατά περίπτωση [1],[2].

Ο έλεγχος συστημάτων κατεργασιών πραγματοποιείται με την υλοποίηση κάποιας λογικής η οποία, για αυτοματοποιημένα συστήματα, ενσωματώνεται σε ένα πρόγραμμα λογισμικού και εκτελείται σε Προγραμματιζόμενους Λογικούς Ελεγκτές (PLC) ή σε βιομηχανικού τύπου PC. Για την επικοινωνία του ελεγκτή με τα υπόλοιπα στοιχεία του κυττάρου χρησιμοποιούνται τεχνολογίες δικτύων ή διαύλων (RS-232, RS 422, Field bus, Profibus) [2].

Δευτερεύοντα στοιχεία :

- **τεχνολογία αριθμητικού ελέγχου NC** αναφέρεται στις λειτουργίες κατανομής και ελέγχου της ροής προγραμμάτων CNC σε αντίστοιχες εργαλειομηχανές ενός ΕΣΚ (συμπεριλαμβανόμενης και της διόρθωσης προγραμμάτων) και παρακολούθησης της κατάστασης των εργαλειομηχανών (συμπεριλαμβανόμενης και της διάγνωσης και δυνατής επιδιόρθωσης των προβλημάτων)
- **Διατάξεις αυτόματης αλλαγής και διαχείρισης εργαλείων (spindle tooling).** Η σημασία της επιλογής του σωστού εργαλείου, της παρακολούθησης της φθοράς, του πλήθους, της εναλλαξιμότητας και του κόστους των εργαλείων ενός ΕΣΚ δημιουργεί την ανάγκη για ένα ξεχωριστό σύστημα διαχείρισης εργαλείων. Σε ένα τέτοιο σύστημα είναι δυνατόν η αυτόματη αλλαγή των εργαλείων με κάποιο κριτήριο π.χ. την ελαχιστοποίηση αλλαγών εργαλείων.
Η αυτόματη καταγραφή εργαλείων μπορεί να γίνεται με αισθητήρες ανάγνωσης barcode ή EPROMS με ραδιο-ζεύξη. Η καταγραφή της κατάστασης των εργαλείων μπορεί να γίνεται εκτός γραμμής από προσωπικό ή αυτόματα βάσει μέτρησης των διαστάσεων από ειδικούς αισθητήρες επαφής ή με έμμεση εκτίμηση των διαστάσεων μέσω μέτρησης δυνάμεων κατεργασίας ή απορροφούμενης ισχύος.
- **Προγραμματιζόμενες ιδιοσυσκευές συγκράτησης των τεμαχίων (work holding features)[1],[2].**

1.1.3 Οφέλη από τη χρήση Ευέλικτων Συστημάτων Κατεργασιών

Οι κύριες καταστάσεις παραγωγής που εμφανίζουν την ανάγκη προσαρμογής Ευέλικτων Συστημάτων Κατεργασιών είναι :

- Επεξεργασία ποικιλίας τεμαχίων υψηλής ακρίβειας
- Επεξεργασία υλικού από μεγάλο αριθμός DNC μηχανών
- Αυτόματη διαχείριση υλικού για τη μεταφορά των τεμαχίων προς το κύτταρο κατεργασιών ,μέσα και έξω από αυτό.
- On line έλεγχος από υπολογιστή ολόκληρου του κυττάρου κατεργασιών υπό μεταβαλλόμενες συνθήκες (μεταβαλλόμενους συνδυασμούς παραγόμενων προϊόντων και μεταβαλλόμενες προτεραιότητες) [1]

Το βασικό κίνητρο για την υιοθέτηση των ΕΣΚ είναι η μείωση του κόστους στην παραγωγή και η προσαρμοστικότητα σε ένα συνεχώς μεταβαλλόμενο περιβάλλον. Επιπλέον λόγοι αποτελούν η απαιτούμενη μείωση του χρόνου παραγωγής των προϊόντων και η αυξανόμενη πολυπλοκότητά τους. Εταιρίες που έχουν εγκαταστήσει ΕΣΚ έχουν αναφέρει τα ακόλουθα αποτελέσματα :

- Ωφέλη σχετικά με προγράμματα μείωσης κόστους
- Ωφέλη σχετικά με τη βελτίωση της ανταπόκρισης στη ζήτηση της αγοράς
- Ωφέλη σχετικά με την ευελιξία στην παραγωγή [1]

Στον ακόλουθο πίνακα φαίνεται μία σύγκριση των ΕΣΚ με τη συμβατική βιομηχανική τεχνολογία αναφορικά με χρόνους που σχετίζονται με την παραγωγική διαδικασία :

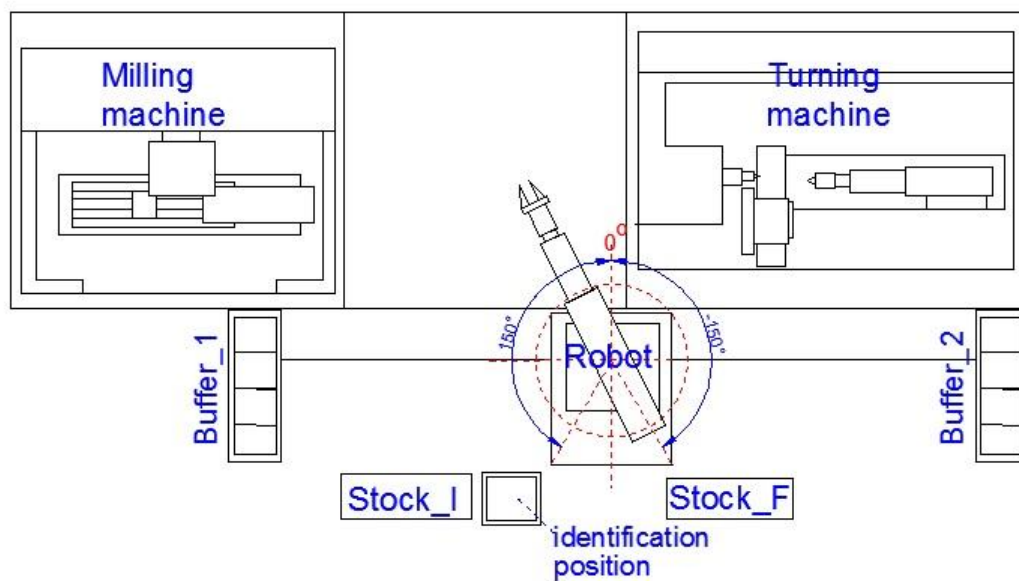
Παράμετρος	Συμβατικά συστήματα κατεργασιών	Ευέλικτα συστήματα κατεργασιών
Ποσοστό του χρόνου που η μηχανή είναι διαθέσιμη	50	Έως και 5
Ποσοστό του χρόνου που βρίσκεται τεμάχιο στη μηχανή	50	Έως και 95
Ποσοστό του χρόνου που η μηχανή δε βρίσκεται σε φάση κατεργασίας ενώ έχει τεμάχιο	70	Έως και 7
Ποσοστό του χρόνου που η μηχανή βρίσκεται σε φάση κατεργασίας ενώ έχει τεμάχιο	30	Έως και 93
Ποσοστό του χρόνου αναμονής και μεταφοράς του τεμαχίου σε σχέση με το χρόνο παραγωγής του	95	Έως και 85
Ποσοστό του χρόνου που το τεμάχιο βρίσκεται στη μηχανή σε σχέση με το χρόνο παραγωγής του	5	Έως και 15
<i>Σύγκριση ΕΣΚ με συμβατική βιομηχανική τεχνολογία ,Πηγή: [1]</i>		

Ο σχεδιασμός και η απόδοση των ευέλικτων συστημάτων κατεργασιών απαιτεί ανάλυση και σχεδιασμό για την επιλογή του βέλτιστου σχεδιασμού και τακτική λειτουργίας. Τα δίκτυα Petri τα οποία θα αναλυθούν στο επόμενο κεφάλαιο ως ένα μαθηματικό γραφικό εργαλείο παρέχουν ένα εννιαίο περιβάλλον για τη μοντελοποίηση, την ανάλυση και το σχεδιασμό των συστημάτων διακριτών

γεγονότων. Ένα από τα βασικά πλεονεκτήματα είναι ότι το ίδιο μοντέλο μπορεί να χρησιμοποιηθεί για την ανάλυση των ιδιοτήτων και την συστηματική δημιουργία του προσομοιωτή και του ελεγκτή.

1.2 Παρουσίαση συστήματος παρούσας εργασίας

Η γενική ιδέα της διάταξης του υπό μελέτη συστήματος βασίστηκε στον εξοπλισμό του εργαστηρίου και έχει την παρακάτω μορφή.



Διάταξη του υπό μελέτη ευέλικτου συστήματος κατεργασιών

Βέβαια , η παρούσα διπλωματική εργασία αφορά κυριώς τον προγραμματισμό της παραγωγικής διαδικασίας και τον συντονισμό των στοιχείων που λαμβάνουν μέρος σε αυτήν. Οι ακριβείς θέσεις των μηχανών κατεργασίας και των αποθηκών δεν έχουν καμία σημασία στη συγκεκριμένη εργασία.

1.2.1 Εξοπλισμός διάταξης συστήματος κατεργασιών

Ο εξοπλισμός του μελετούμενου συστήματος κατεργασιών αποτελείται από:

- το κέντρο τórνευσης
- τη φρέζα
- το ρομπότ
- μια αρχική αποθήκη τεμαχίων (Stock_I)

(Το ρομπότ παραλαμβάνει τα τεμάχια ξεκινώντας από την πρώτη γραμμή του Stock_1 και διατρέχοντας όλες τις στήλες)

- μια θέση δίπλα στην αρχική αποθήκη των τεμαχίων όπου γίνεται ταυτοποίηση του είδους του τεμαχίου από ειδική κάμερα.
- μια ενδιάμεση αποθήκη τεμαχίων στα οποία τοποθετούνται κατεργασμένα τεμάχια που απαιτούν περαιτέρω επεξεργασία (Buffer). Στη διάταξη υπάρχουν δύο διαφορετικές ενδιάμεσοι χώροι αποθήκευσης (Buffer_1 για τεμάχια με επόμενη κατεργασία στη φρέζα και Buffer_2 για τεμάχια με επόμενη κατεργασία στο κέντρο τόννευσης)
- μια τελική αποθήκη (Stock_F) όπου τοποθετούνται τα πλήρως κατεργασμένα τεμάχια

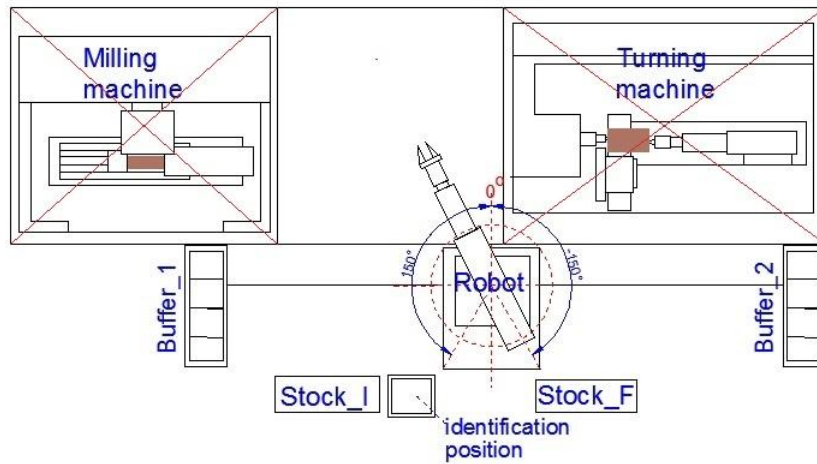
1.2.2 Παραγωγική διαδικασία συστήματος

Στη θέση ταυτοποίησης “διαβάζεται” το είδος του τεμαχίου και στέλνεται στον ελεγκτή το αντίστοιχο σήμα για το process plan (αριθμός και σειρά κατεργασιών) του συγκεκριμένου είδους. Στα τεμάχια που απαιτούν παραπάνω από μία φάσεις επεξεργασίας, οι κατεργασίες γίνονται πάντοτε εναλλακτικά στις δύο μηχανές. Η φόρτωση και εκφόρτωση των τεμαχίων στον τόρνο, τη φρέζα και τους αποθηκευτικούς χώρους γίνεται αυτόματα από το ρομπότ.

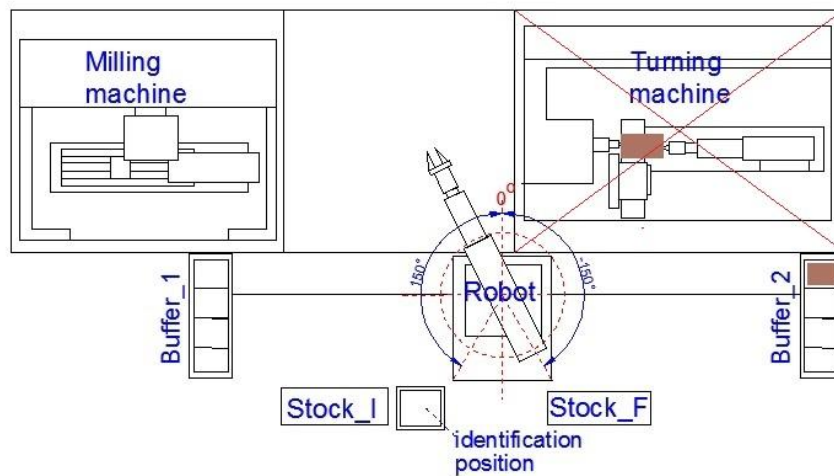
- Αν η μηχανή που εκτελεί την 1^η κατεργασία στο process plan του ρομπότ είναι ελεύθερη τότε το ρομπότ πηγαίνει στη θέση ταυτοποίησης, παραλαμβάνει το τεμάχιο και το φορτώνει στην αντίστοιχη μηχανή.
- Αν η μηχανή που εκτελεί την 1^η κατεργασία στο process plan του ρομπότ είναι απασχολημένη, τότε ο ελεγκτής “περιμένει” να τελειώσει κάποια κατεργασία.

Τα δύο buffers που υπάρχουν στη διάταξη χρησιμοποιούνται στις περιπτώσεις όπου ακολουθεί επόμενη επεξεργασία του τεμαχίου και η μηχανή που απαιτείται είναι απασχολημένη. Ο προγραμματισμός αυτός στοχεύει στην όσο το δυνατόν συνεχόμενη απασχόληση και των δύο μηχανών. Στο ακόλουθο παράδειγμα φαίνεται μια φάση της παραγωγικής διαδικασίας που αποδεικνύει τη χρησιμότητα των buffers :

Αρχικά βρίσκονται σε εξέλιξη δύο κατεργασίες και επομένως και οι δύο μηχανές δεν είναι διαθέσιμες. Υποθέτουμε ότι πρώτα ολοκληρώνεται η κατεργασία του τεμαχίου στη φρέζα το οποίο όμως απαιτεί επόμενη επεξεργασία στον τόρνο. Δεδομένου ότι ο τόρνος είναι προσωρινά απασχολημένος, το τεμάχιο θα έπρεπε να περιμένει στη φρέζα μέχρι να γίνει διαθέσιμος. Η ύπαρξη των ενδιάμεσων buffers δίνει τη δυνατότητα στο σύστημα να τοποθετήσει τεμάχια σε ‘θέσεις αναμονής’ για επόμενη επεξεργασία ώστε να αποδεσμεύει τις μηχανές κατεργασίες όταν αυτές δε βρίσκονται σε λειτουργία.

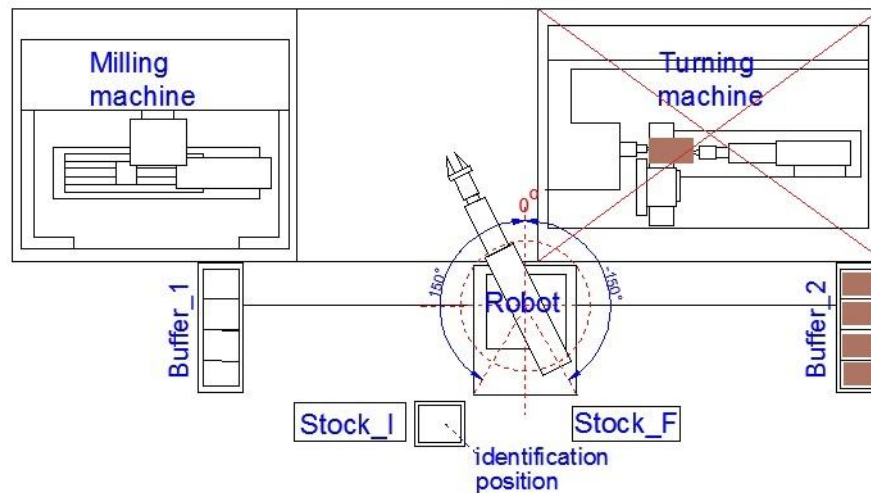


Σύστημα σε φάση όπου και οι δύο μηχανές κατεργασίας είναι μη διαθέσιμες



Ολοκλήρωση κατεργασίας φρεζαρίσματος τεμαχίου - Αποδέσμευση της φρέζας από το τεμάχιο που απαιτεί επόμενη κατεργασία στον τόρνο με τοποθέτηση του τεμαχίου στο Buffer_2

Όπως φαίνεται στο παρακατω σχήμα η φρέζα μπορεί να συνεχίσει εξυπηρετεί τεμάχια (μέχρι και τέσσερα τεμάχια που απαιτούν επόμενη επεξεργασία στον τόρνο)



Πλήρωση του Buffer_2 όσο ο τóρνος συνεχίζει να είναι μη-διαθέσιμος

Ακόμα, το σύστημα προγραμματίζεται έτσι ώστε να δίνει προτεραιότητα στην επεξεργασία των τεμαχίων που βρίσκονται στα buffers και περιμένουν να κατεργαστούν στην αντίστοιχη μηχανή έναντι των τεμαχίων που βρίσκονται στην αρχική αποθήκη. Με αυτόν τον τρόπο επιτυγχάνεται η ελαχιστοποίηση του συνολικού χρόνου παραγωγής των τεμαχίων.

Εργασίες εκτελούμενες από τις μηχανές κατεργασίας:

Στο σύστημα που σχεδιάστηκε είναι δυνατόν να υπάρχουν όλοι οι παρακάτω συνδυασμοί κατεργασιών :

- A: τóρνευση
- B: τóρνευση-φρεζάρισμα
- C: τóρνευση-φρεζάρισμα- τóρνευση
- D: φρεζάρισμα
- E: φρεζάρισμα- τóρνευση
- F: φρεζάρισμα- τóρνευση- φρεζάρισμα

Εργασίες εκτελούμενες από το ρομπότ :

Σύμφωνα με την παραγωγική διαδικασία που περιγράφηκε παραπάνω, το ρομπότ της διάταξης εκτελεί τις παρακάτω εργασίες:

- μεταφορά τεμαχίου από την αρχική αποθήκη στον τóρνο
- μεταφορά τεμαχίου από την αρχική αποθήκη στη φρέζα
- μεταφορά τεμαχίου από τον τóρνο στην τελική αποθήκη
- μεταφορά τεμαχίου από τη φρέζα στην τελική αποθήκη
- μεταφορά τεμαχίου από τον τóρνο στη φρέζα
- μεταφορά τεμαχίου από τη φρέζα στον τóρνο
- μεταφορά τεμαχίου από τον τóρνο στην ενδιάμεση αποθήκη

- μεταφορά τεμαχίου από τη φρέζα στην ενδιάμεση αποθήκη
- μεταφορά τεμαχίου από την ενδιάμεση αποθήκη στη φρέζα
- μεταφορά τεμαχίου από την ενδιάμεση αποθήκη στον τόρνο

Σε αυτό το σημείο αλλά και κατά τη μοντελοποίηση του συστήματος δε λαμβάνονται υπ' όψη:

- i) ο τρόπος λήψης και τοποθέτησης από το ρομπότ που είναι πιθανόν να διαφοροποιείται σε κάθε τεμάχιο
- ii) οι διαφορετικές θέσεις τοποθέτησης των τεμαχίων στις ενδιάμεσες αποθήκες και η ενημέρωση του ελεγκτή σε σχέση με τη δέσμευση/ελευθέρωση αυτών

Οι απλοποιήσεις αυτές δεν επηρεάζουν την ακρίβεια της μοντελοποίησης καθώς οι διαφοροποιημένες ενδεχομένως κινήσεις μιας εργασίας του ρομπότ αντιπροσωπεύουν την ίδια κατάσταση για το σύστημα. Θα ληφθούν όμως υπ' όψη κατά τον τελικό προγραμματισμό του συστήματος που θα γίνει σε γλώσσα προγραμματιζόμενου λογικού ελεγκτή.

Αναφορικά με τους ορισμούς που δόθηκαν παραπάνω για την ευελιξία, θα μπορούσαμε να πούμε ότι το συγκεκριμένο σύστημα χαρακτηρίζεται από ευελιξία διαδρομής καθώς υπάρχει η δυνατότητα να ακολουθήσει εναλλακτική σειρά των εκτελούμενων εργασιών (τηρώντας όμως πάντοτε τη σειρά του process plan κάθε τεμαχίου) προκειμένου να μειωθούν οι χρόνοι αναμονής του ρομπότ και των μηχανών κατεργασίας.

Κεφάλαιο 2 - Δίκτυα Petri

2.1 Γενικά

2.1.1 Εισαγωγή στα δίκτυα Petri

Τα δίκτυα Petri πήραν το όνομά τους από τον Carl A. Petri ο οποίος δημιούργησε ένα μαθηματικό εργαλείο για τη μελέτη της επικοινωνίας μεταξύ αυτόματων μηχανών το 1962. Ο Petri περιέγραψε γραφικά, με τη μορφή δικτύου, τη μαθηματική σχέση μεταξύ των γεγονότων και καταστάσεων σε ένα σύστημα υπολογιστών. Η περαιτέρω ανάπτυξη της συγκεκριμένης θεωρίας προέκυψε από την ανάγκη προσδιορισμού και μοντελοποίησης συστημάτων διακριτών γεγονότων [5].

Τα συστήματα διακριτών γεγονότων (Discrete event systems – DES) περιγράφουν διαδικασίες που εξελίσσονται με βάση καθορισμένα γεγονότα που συμβαίνουν σε διακριτά χρονικά διαστήματα. Κάθε γεγονός έχει σαν αποτέλεσμα τη μετάβαση της διαδικασίας από μια κατάσταση σε κάποια άλλη. Κύρια χαρακτηριστικά των συστημάτων αυτών είναι ότι περιέχουν εργασίες που μπορεί να είναι σειριακές ή παράλληλες, σύγχρονες ή και ασύγχρονες και πιθανόν μη ντετερμινιστικές [?].

Έως τώρα, μοντελα δικτύων Petri έχουν χρησιμοποιηθεί σε πολλά διαφορετικά πεδία εφαρμογών όπως συστήματα λογισμικού, βιομηχανικά συστήματα ελέγχου παραγωγής, συστήματα ελέγχου κυκλοφορίας, χημικές διαδικασίες, πρωτόκολλα επικοινωνίας [5],[9].

2.1.2 Πλεονεκτήματα των δικτύων Petri

Τα δίκτυα Petri αποτελούν ένα γραφικό εργαλείο για το σχεδιασμό συστημάτων διακριτών γεγονότων και χαρακτηρίζονται από τα παρακάτω πλεονεκτήματα :

- Ευκολία στη μοντελοποίηση χαρακτηριστικών πολύπλοκων βιομηχανικών συστημάτων όπως : παράλληλα, σύγχρονες και ασύγχρονες λειτουργίες, συγκρούσεις, αμοιβαίος αποκλεισμός, σχέσεις προτεραιότητας, μη ντετερμινισμός και κολλήματα συστημάτων (deadlocks)
- Δυνατότητα δημιουργίας κώδικα κατευθείαν από τη γραφική παράσταση του δικτύου Petri. Ένας εκτελέσιμος κώδικας δικτύου Petri μπορεί επίσης να δημιουργηθεί σε εφαρμογές πραγματικού χρόνου χρησιμοποιώντας προγραμματιζόμενους λογικούς ελεγκτές ή υπολογιστές.
- Δυνατότητα διερεύνησης βασικών ιδιοτήτων του συστήματος (όπως η ύπαρξη κολλημάτων μέσω ανάλυσης σε υπολογιστικά προγράμματα βασισμένα στα δίκτυα Petri αποφεύγοντας πολυάριθμες προσομοιώσεις για πολλά σενάρια λειτουργίας του συστήματος)
- Δυνατότητα ανάλυσης λειτουργίας και απόδοσης του συστήματος αξιολογώντας χρόνους παραγωγής, χρόνους αναμονής, βαθμό χρήσης πόρων.

- Η προσομοίωση είναι δυνατόν να οδηγείται από το ίδιο το μοντέλο
- Παροχή χρήσιμων πληροφοριών που επιτρέπουν την παρακολούθηση και τον έλεγχο του μοντέλου αλλά και διόρθωση σφαλμάτων πριν την εφρμογή του στο πραγματικό σύστημα[1],[5].

2.1.3 Τα δίκτυα Petri στον τομέα της αυτοματοποίησης της παραγωγής

Όπως έχει αποδειχθεί, τα δίκτυα Petri μπορούν να χρησιμοποιηθούν για τη μελέτη πληθώρας αυτόματων βιομηχανικών συστημάτων που χαρακτηρίζονται από ασύγχρονα γεγονότα, παράλληλες λειτουργίες, συγκρούσεις, αμοιβαίο αποκλεισμό και κατανομή πόρων.

Το αρχικό ενδιαφέρον για τα δίκτυα Petri στο συγκεκριμένο τομέα ξεκίνησε από την ανάγκη του προσδιορισμού και μοντελοποίησης διακριτών συστημάτων παραγωγής. Η βασική ιδιαιτερότητα των συστημάτων αυτών είναι η παραλληλία και ο μη ντετερμινισμός το οποίο συνεπάγεται ότι η εκτέλεση αυτών μπορεί να γίνει με πολλούς διαφορετικούς τρόπους. Είναι εύκολο, κατά τη σχεδίαση τέτοιων συστημάτων να αγνοηθούν σημαντικά σημεία αλληλεπίδρασης οδηγώντας σε εσφαλμένη λειτουργία κατά την εκτέλεσή τους. Έτσι, για να αντιμετωπιστεί η πολυπλοκότητα των σύγχρονων παράλληλων συστημάτων είναι βασικό να παρέχονται μέθοδοι που επιτρέπουν τον έλεγχο και την επιδιόρθωση πριν από τη εφαρμογή τους. Ένας τρόπος προσέγγισης του συγκεκριμένου προβλήματος είναι να χτιστεί ένα εκτελέσιμο μοντέλο του συστήματος. Η κατάστρωση και προσομοίωση ενός μοντέλου μπορεί να δώσει μια ολοκληρωμένη εικόνα στο σχεδιασμό και τη λειτουργία του συστήματος [9].

Σε πρώτα στάδια, τα δίκτυα Petri χρησιμοποιήθηκαν για την αναπαράσταση απλών γραμμών παραγωγής με ουρές, μηχανουργείων, αυτοματοποιημένων συστημάτων παραγωγής, και στη συνέχεια χρησιμοποιήθηκαν για την μοντελοποίηση ευέλικτων συστημάτων παραγωγής (ΕΣΚ), αυτοματοποιημένων γραμμών συναρμολόγησης, συστημάτων με κοινή χρήση-διαχείριση πόρων, και προσφάτως σε συστήματα παραγωγής με μεθόδους just-in-time και Kanban.

Η εργασία προχώρησε στην προσπάθεια προσδιορισμού και μοντελοποίησης συστημάτων παραγωγής μεγάλης κλίμακας κάτω από διαφορετικές επιχειρησιακές συνθήκες. Τελευταίως, η έρευνα προσανατολίστηκε στην αναζήτηση των ποιοτικών χαρακτηριστικών των μοντέλων PN για συστήματα παραγωγής. Για παράδειγμα, η ανάλυση της προσβασιμότητας (reachability analysis) εξετάζει αν το σύστημα μπορεί να φτάσει σε μια συγκεκριμένη κατάσταση. Επιθυμητές ακολουθίες γεγονότων ελέγχονται και επικυρώνονται σύμφωνα με τις ανάγκες του συστήματος. Άλλες ιδιότητες των PN χρησιμοποιούνται για να προκύψει η σταθερότητα των DEDS, η κυκλική συμπεριφορά, και η απουσία αδιεξόδων. Ειδικά οι τεχνικές αποφυγής των αδιεξόδων και η εκτίμηση των αποτελεί ένα ιδιαίτερο πλαίσιο έρευνας ακόμα και σήμερα.

Σήμερα υπάρχουν πολλά πακέτα λογισμικού προσομοίωσης τα οποία στηρίζονται σε δίκτυα Petri. Η χρήση τεχνικών προσομοίωσης για την εκτίμηση ευέλικτων συστημάτων παραγωγής και ρομποτικών συστημάτων συναρμολόγησης επιρρεπών σε λάθη είναι μόνο μερικοί τομείς [5].

2.2 Θεωρία Δικτύων Petri

2.2.1 Βασικά στοιχεία δικτύων Petri

Η ενότητα αυτή αποτελεί μια συνοπτική παρουσίαση της θεωρίας των δικτύων Petri με σκοπό την εισαγωγή στα επόμενα κεφάλαια. Εκτενέστερη ανάλυση γύρω από τη συγκεκριμένη θεωρία μπορεί να βρεθεί στις πηγές [2],[5],[6].

Τα βασικά δομικά στοιχεία ενός δικτύου Petri είναι οι θέσεις (places), οι μεταβάσεις (transitions), τα τόξα ή κλάδοι (arcs) και τα κουπόνια (tokens). Οι θέσεις εκφράζουν συνθήκες, κατάσταση ενός συστατικού, ή μια εργασία του συστήματος και παριστάνονται με κύκλους. Οι μεταβάσεις εκφράζουν συνήθως γεγονότα και παριστάνονται με τετράγωνα. Δύο χαρακτηριστικά γεγονότα που μπορεί να εμπεριέχει μία μετάβαση είναι η «έναρξη» και η «λήξη». Οι θέσεις και οι μεταβάσεις συνδέονται μεταξύ τους με τόξα. Ένα τόξο μπορεί να έχει κατεύθυνση από μια θέση προς μια μετάβαση (τόξο εισόδου στη μετάβαση) ή από μια μετάβαση σε μια θέση (τόξο εξόδου από τη μετάβαση). Επίσης ένα τόξο μπορεί να έχει ταυτόχρονα και τις δύο κατευθύνσεις (διπλό τόξο).

Οι θέσεις, οι μεταβάσεις, και τα τόξα διαμορφώνουν ένα προσανατολισμένο γράφημα το οποίο δίνει τη βασική δομή του δικτύου Petri. Η δυναμική του δικτύου καθορίζεται από την δυνατότητα των θέσεων να έχουν κανένα ή θετικό αριθμό token, τα οποία αναπαριστάνται με μικρούς γεμάτους κύκλους. Τα tokens μέσα σε κάποια θέση μπορούν να σημαίνουν τον αριθμό των πόρων, να υποδηλώνουν αν μια συνθήκη είναι αληθής, ή να δηλώνουν αν μια εργασία βρίσκεται σε εξέλιξη. Η ροή των tokens μέσα στο δίκτυο ρυθμίζεται από τις μεταβάσεις σύμφωνα με τις συνθήκες που περιγράφονται παρακάτω [5].

Ορισμοί στοιχείων δικτύου Petri

Ένα δίκτυο Petri $Z = (P, T, I, O, M)$ ορίζεται από πέντε στοιχεία, όπου:

- $P = \{p_1, p_2, \dots, p_n\}$, $n > 0$, είναι ένας πεπερασμένος πίνακας θέσεων
- $T = \{t_1, t_2, \dots, t_n\}$, $n > 0$, είναι ένας πεπερασμένος πίνακας μεταβάσεων όπου $P \cap T = \emptyset$ και $P \cap T = \emptyset$
- $I: P \times T \rightarrow \mathbb{N}$, είναι η συνάρτηση εισόδου που ορίζει το σύνολο των προσανατολισμένων τόξων από τις P στις T , όπου $\mathbb{N} = \{0, 1, 2, \dots\}$. Ένα τόξο εισόδου μπορεί να εκφραστεί ως $I(p, t)$ όπου p η θέση από την οποία ξεκινά και t η μετάβαση στην οποία καταλήγει.
- $O: P \times T \rightarrow \mathbb{N}$, είναι η συνάρτηση εξόδου που ορίζει το σύνολο των προσανατολισμένων τόξων από τις T στις P . Ένα τόξο εξόδου μπορεί να εκφραστεί

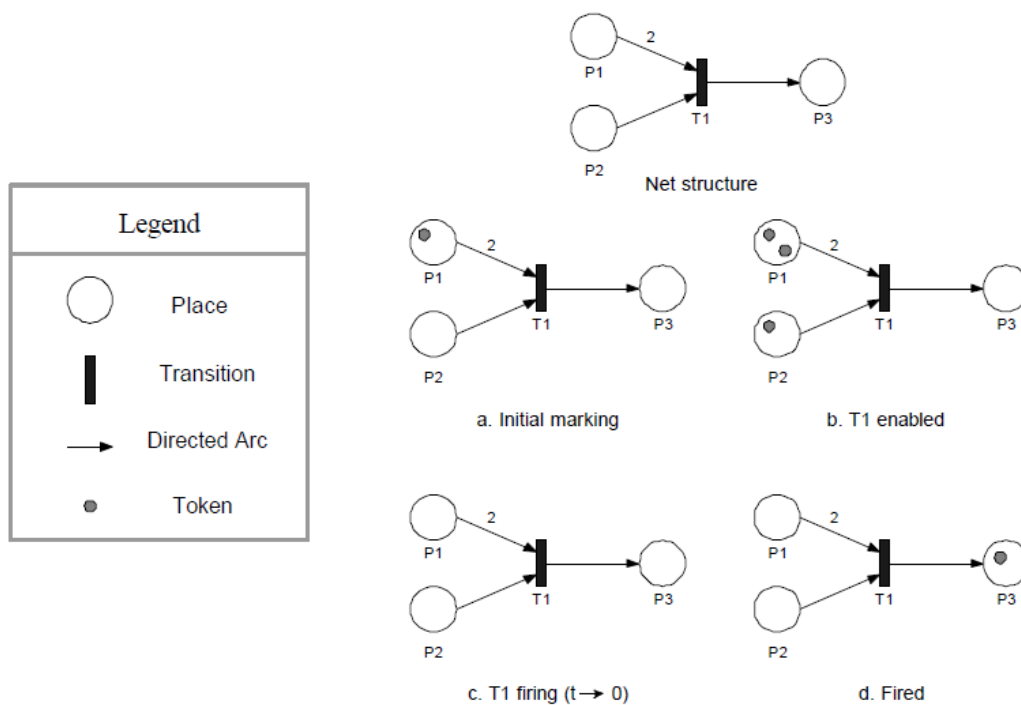
ως $I(t, p)$ όπου t η μετάβαση από την οποία ξεκινά και p η θέση στην οποία καταλήγει.

- $m : P \rightarrow \mathbb{N}$, είναι η σήμανση της οποίας ο i -οστός όρος παριστάνει τον αριθμό των tokens στην i -οστή θέση. Η αρχική σήμανση δηλώνεται ως m_0 .

Οι αριθμοί που αντιστοιχούν στα στοιχεία $I(p, t)$ και $O(p, t)$ των πινάκων ονομάζονται βάρη των κλάδων (τόξων). Τα 'βάρη' δείχνουν τον αριθμό των tokens που διατρέχουν ένα τόξο όταν ενεργοποιείται μια μετάβαση. Το βάρος ενός κλάδου σημειώνεται δίπλα στο τόξο και αν είναι ίσο με τη μονάδα συνήθως παραλείπεται. Ένα δίκτυο Petri μπορεί εναλλακτικά να οριστεί και ως (P, T, F, W, m) , όπου F είναι ένα υποσύνολο $\{P \times T\} \cup \{T \times P\}$, που αναπαριστά το σύνολο όλων των τόξων, ενώ το $W: F \rightarrow \mathbb{N}$ αναπαριστά το βάρος (ή πολλαπλότητα) των τόξων[5].

2.2.2 Κανόνες ενεργοποίησης των μεταβάσεων :

Σε ένα σημασμένο PN μια μετάβαση t_j ενεργοποιείται όταν ισχύει $m(p) \geq I(p, t)$, $\forall p \in P$. Η συνθήκη αυτή εκφράζει ότι για να ενεργοποιηθεί μια μετάβαση πρέπει όλες οι θέσεις εισόδου σε αυτήν να έχουν αριθμό token ίσο ή μεγαλύτερο από το βάρος του αντίστοιχου τόξου.



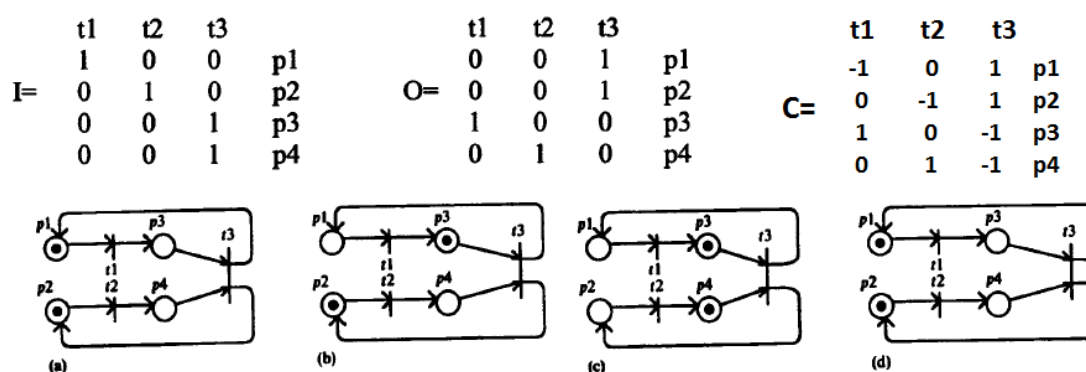
Παράδειγμα ροής token στα δίκτυα Petri

Η ενεργοποίηση της μετάβασης t_j συνεπάγεται μια νέα σήμανση m_i' που προκύπτει από την αφαίρεση $I(p_i, t_j)$ κουπονιών από τις θέσεις εισόδου και πρόσθεση $O(p_i, t_j)$ κουπονιών στις θέσεις εξόδου. Η σχέση που συνδέει τη νέα σήμανση σε μια θέση με την προηγούμενη σήμανση στην ίδια θέση είναι :

$$m'(p) = m(p) - I(p, t) + O(p, t), \forall p \in P$$

Παρακάτω φαίνεται ένα δίκτυο Petri τεσσάρων θέσεων και οι διαφορετικές καταστάσεις στις οποίες μπορεί να βρεθεί. Οι αντίστοιχες σημάνσεις είναι :

$M_a = \{1, 1, 0, 0\}$, $M_b = \{0, 1, 1, 0\}$, $M_c = \{0, 0, 1, 1\}$, $M_d = \{1, 1, 0, 0\}$



Καταστάσεις δικτύου Petri, πίνακες τόξων εισόδου εξόδου και πίνακας συμβάντων, Πηγή : [2]

Η παραπάνω σχέση μπορεί να εκφραστεί και σε μορφή μητρώου συμπεριλαμβάνοντας όλες τις θέσεις του PN:

$$M_k = M_{k-1} + O \cdot u_k - I \cdot u_k = M_{k-1} + (O - I) \cdot u_k = M_{k-1} + C \cdot u_k$$

όπου:

- M_k το διάνυσμα σήμανσης του PN στην κατάσταση k όπου M_k είναι μια σήμανση άμεσα προσβάσιμη από τη M_{k-1}
- C ο πίνακας συμβάντων ο οποίος ορίζει την τοπολογία του PN. Ο C προκύπτει από την αφαίρεση του I από το O ($C = O - I$). Η τιμή 1 στον πίνακα C αντιστοιχεί σε ύπαρξη εισόδου από τη μετάβαση και η τιμή -1 σε ύπαρξη εξόδου από τη μετάβαση
- $u_k = \{u(t1), u(t2), u(t3), \dots\}$ το διάνυσμα ενεργοποίησης των μεταβάσεων στην κατάσταση k . Το u_k είναι ένα διάνυσμα-στήλη με ένα μόνο στοιχείο ίσο με 1 και

όλα τα υπόλοιπα ίσα με 0. Αν η μετάβαση t_i εκκινήσει στην k -οστή εκκίνηση, τότε το i -οστό στοιχείο του u_k είναι 1 και όλα τα άλλα στοιχεία είναι 0.

Με διαδοχικές ενεργοποιήσεις u_1, u_2 το σύστημα μπορεί να φθάσει από μια αρχική κατάσταση (σήμανση) M_0 σε μια τελική M_d :

$$M_d = M_0 + C \sum_{k=1}^d u_k = M_0 + C \cdot y$$

όπου y είναι το διάνυσμα μέτρησης ενεργοποιήσεων το οποίο δείχνει πόσες φορές ενεργοποιείται μία μετάβαση[2],[5].

2.2.3 Μηδενικός έλεγχος

Μια σημαντική επέκταση των δυνατοτήτων μοντελοποίησης των δικτύων Petri είναι ο μηδενικός έλεγχος (zero testing ability), δηλαδή η δυνατότητα ελέγχου αν μια θέση έχει μηδενικά tokens. Για να υλοποιηθεί ο μηδενικός έλεγχος χρησιμοποιείται ένα δεσμευτικό τόξο. Το δεσμευτικό τόξο συνδέει πάντοτε μια θέση εισόδου με μια μετάβαση, και στο τέλος του αντί για βελάκι σχεδιάζεται ένας μικρός κύκλος και αυτό διότι στα ψηφιακά κυκλώματα ο μικρός κύκλος υποδηλώνει το NOT.

Η παρουσία του δεσμευτικού τόξου στην είσοδο μιας μετάβασης αλλάζει τις συνθήκες ενεργοποίησης της. Έτσι, μια μετάβαση ενεργοποιείται αν κάθε θέση εισόδου που είναι συνδεδεμένη με την μετάβαση με ένα κανονικό τόξο περιέχει αριθμό tokens τουλάχιστον ίσο με το βάρος του τόξου και κανένα token δεν είναι παρόν στις θέσεις εισόδου που συνδέονται με το δεσμευτικό τόξο. Η ενεργοποίηση της μετάβασης δεν μεταβάλλει τον αριθμό token στις θέσεις που είναι συνδεδεμένες με το δεσμευτικό τόξο.

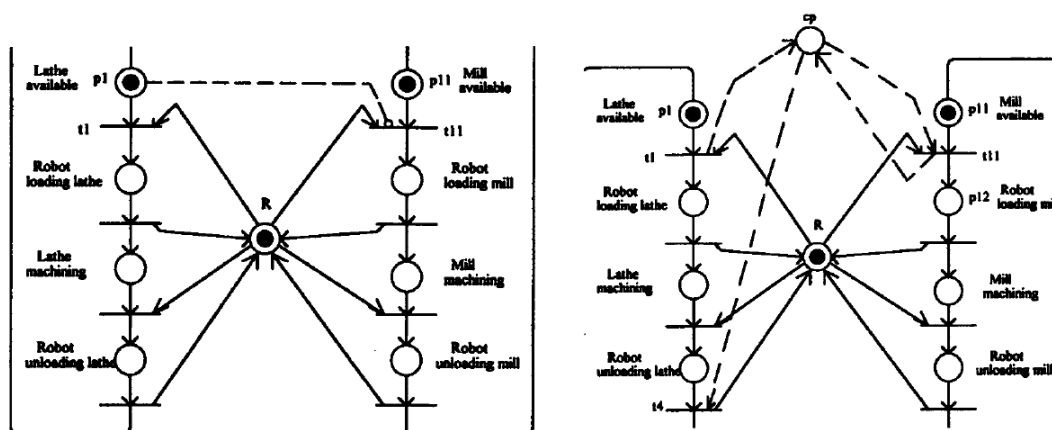
Το δεσμευτικό τόξο μπορεί να γενικευθεί με μια πολλαπλότητα. Έτσι, όταν ο αριθμός των κουπονιών στις θέσεις είναι μικρότερος από τον αριθμό πολλαπλότητας του δεσμευτικού τόξου, η μετάβαση επιτρέπεται να ενεργοποιηθεί από τις κανονικές θέσεις εισόδου.

Το δεσμευτικό τόξο μπορεί να συμπεριληφθεί στον ορισμό του δικτύου Petri, μέσω μιας συνάρτησης δέσμευσης $H: P \times T \rightarrow N$. Μία μετάβαση $t \in T$ θεωρείται ενεργοποιημένη όταν και μόνο όταν:

$$m(p) \geq l(p, t) \text{ και } m(p) < H(p, t), \forall p \in P$$

Το δεσμευτικό τόξο ή αποτρεπτικός κλάδος μπορεί να χρησιμοποιηθεί και για την επιβολή προτεραιοτήτων σε ένα PN. Αυτό είναι χρήσιμο όταν έχουμε ανταγωνισμό μεταξύ δραστηριοτήτων για τον ίδιο πόρο ή/και από τον ανταγωνισμό μιας δραστηριότητας να επιλέξει ανάμεσα σε πολλούς πόρους [5].

Στο παράδειγμα που δίνεται παρακάτω, ο αποτρεπτικός κλάδος από τη θέση p_1 (τόρνος διαθέσιμος) στη μετάβαση t_{11} (έναρξη φόρτωσης φρέζας) δίνει προτεραιότητα στην εξυπηρέτηση του τόρνου από το ρομπότ.



Τρόποι επιβολής προτεραιοτήτων σε δίκτυα Petri, Πηγή : [2]

Σε ένα μοντέλο PN, ο αποτρεπτικός κλάδος μπορεί να υλοποιηθεί και εναλλακτικά με μια πρόσθετη θέση και κατάλληλα τοποθετημένα τόξα. Όπως φαίνεται στο παραπάνω παράδειγμα, ενεργοποίηση της t_{11} εξαρτάται (εκτός από την p_{11}) και από τη θέση cp . Όμως, η cp λαμβάνει token από την ενεργοποίηση της t_1 . Άρα πάντοτε θα πρέπει να προηγείται η ενεργοποίηση της t_1 [2].

2.2.4 Ιδιότητες Δικτύων Petri

Η θεωρία των δικτύων Petri ως ένα μαθηματικά – ορισμένο εργαλείο μπορεί να αποδείξει την παρουσία ή απουσία ενός αριθμού ιδιοτήτων ενός δικτύου άρα και ενός συστήματος. Οι ιδιότητες αυτές διακρίνονται σε :

- ιδιότητες που αφορούν στην δομή του συστήματος οι οποίες δεν εξαρτώνται από την αρχική σήμανση
- ιδιότητες που αφορούν στην συμπεριφορά του συστήματος οι οποίες εξαρτώνται από την αρχική σήμανση του δικτύου.

Οι πιο σημαντικές, από πλευράς πρακτικότητας, ιδιότητες είναι η προσβασιμότητα (reachability), η φραγή (boundness), η ασφάλεια (safeness), η ζωντάνια (liveness), η αντιστρεπτικότητα (reversibility), και η αρχική κατάσταση (home state). Επιπρόσθετες δομικές ιδιότητες περιλαμβάνουν την επαναληψιμότητα (repetitiveness) και τη συνέπεια (consistency), η κάλυψη (coverability), η επιμονή (consistency), η απόσταση συγχρονισμού (synchronic distance), η αμεροληψία (fairness)[5]. Παρακάτω παρουσιάζονται οι πιο σημαντικές από αυτές που αφορούν το μελετούμενο μοντέλο του ΕΣΚ.

Προσβασιμότητα

Σε ένα δίκτυο Petri $Z = (P, T, I, O, m_0)$, μία σήμανση m είναι προσβάσιμη από μία σήμανση m_0 εφόσον υπάρχει τέτοια ακολουθία ενεργοποιήσεων μεταβάσεων που μπορεί να οδηγήσει από τη m_0 σε m . Η προσβασιμότητα, ως μια ιδιότητα συμπεριφοράς, εξαρτάται από την αρχική σήμανση. Μια πολύ σημαντική παράμετρος κατά την σχεδίαση ενός συστήματος κατεργασιών είναι το ερώτημα κατά πόσον το σύστημα μπορεί να φτάσει σε μια συγκεκριμένη κατάσταση, ή να επιδείξει συγκεκριμένη λειτουργική συμπεριφορά.

Σημειώνεται ότι τα πραγματικά συστήματα μπορούν να φτάσουν σε επιθυμητή κατάσταση ως αποτέλεσμα διαφορετικών λειτουργικών συμπεριφορών. Σε ένα μοντέλο δικτύου Petri αυτό πρέπει να αντικατοπτρίζεται στην ύπαρξη συγκεκριμένων ακολουθιών εκκινήσεων μεταβάσεων, που δηλώνουν τις λειτουργικές συμπεριφορές, που θα μετατρέψουν την m_0 σε m . Η ύπαρξη επιπλέον ακολουθιών στο δίκτυο Petri που μετατρέπουν την m_0 σε m υποδηλώνει ότι το μοντέλο μπορεί να μην αντικατοπτρίζει ακριβώς την δομή και την δυναμική του μοντελοποιημένου συστήματος. Επίσης, μπορεί να υποδηλώνει την ύπαρξη ανεπιθύμητων όψεων της λειτουργικής συμπεριφοράς του πραγματικού συστήματος, δεδομένου ότι το μοντέλο του δικτύου Petri αντικατοπτρίζει πλήρως τις προδιαγραφόμενες απαιτήσεις του πραγματικού συστήματος [5].

Φραγή και Ασφάλεια

Ένα δίκτυο Petri $Z = (P, T, I, O, m_0)$ ονομάζεται κ-φραγμένο όταν ο αριθμός token σε οποιαδήποτε θέση δεν ξεπερνά το κ για κάθε προσβάσιμη σήμανση από την αρχική σήμανση m_0 . Το Z ονομάζεται ασφαλές αν είναι κ-φραγμένο. Ακόμα, το Z είναι δομικά φραγμένο αν το Z είναι φραγμένο για οποιοδήποτε πεπερασμένη αρχική σήμανση m_0 . Σε περιπτώσεις όπου μία θέση ορίζει χώρο αποθήκευσης ή πόρο, η ιδιότητα της φραγής μας επιτρέπει να προσδιορίσουμε αν ο προτεινόμενος έλεγχος εμποδίζει την υπερχειλίση των αποθηκευτικών χώρων και την υπερχειλίση των χωρητικότητων των πόρων. Σε περιπτώσεις όπου μια θέση ορίζει μια εργασία, η ασφάλεια εγγυάται ότι ο ελεγκτής δεν θα προχωρήσει σε καμία προσπάθεια να εκκινήσει μια υπό-εκτέλεση εργασία[5].

Ζωτικότητα

Σε ένα δίκτυο Petri $Z = (P, T, I, O, m_0)$ μία μετάβαση t είναι ζωντανή αν από την αρχική σήμανση m_0 υπάρχει μια ακολουθία σημάτων η οποία ενεργοποιεί την t . Ένα δίκτυο Petri είναι ζωντανό αν κάθε μετάβαση σε αυτό είναι ζωντανή.

Μία μετάβαση t είναι νεκρή αν από την αρχική σήμανση m_0 δεν υπάρχει καμία ακολουθία σημάτων η οποία να ενεργοποιεί την t .

Μια νεκρή σήμανση είναι μια σήμανση από την οποία δεν ενεργοποιείται καμία μετάβαση. Ένα δίκτυο Petri περιέχει ένα αδιέξοδο αν υπάρχει νεκρή σήμανση δηλαδή αν υπάρχει μια σήμανση $m \in R$, έτσι ώστε καμία μετάβαση να μην ενεργοποιείται. Καταστάσεις με

αδιέξοδα είναι αποτέλεσμα λανθασμένης κατανομής των πόρων ή εξάντλησης συγκεκριμένων τύπων πόρων [1],[2],[5],[6].

Σε ένα αυτοματοποιημένο σύστημα κατεργασιών, πολλοί πόροι χρησιμοποιούνται από κοινού. Τέτοιοι πόροι μπορούν να είναι μηχανές, συστήματα χειρισμού υλικού συμπεριλαμβανομένων ρομπότ και ΟΑΠ, και χώροι αποθήκευσης. Για παράδειγμα, τα ρομπότ χρησιμοποιούνται από κοινού από διάφορες μηχανές, οι μηχανές από διαφορετικές ή ίδιες εργασίες, και οι αποθήκες από πρώτη ύλη, ενδιάμεσο, και τελικό προϊόν. Μια κατάσταση διεξόδου που συνήθως συμβαίνει σε ένα μικρό υποσύστημα του δικτύου, μπορεί να διογκωθεί και να επηρεάσει ένα μεγάλο κομμάτι του συστήματος. Αυτό συνήθως συνεπάγεται ολοκληρωτική στασιμότητα του συστήματος [1],[2],[5],[6].

Αντιστρεπτότητα και Αρχική Κατάσταση

Ένα δίκτυο Petri $Z = (P, T, I, O, m_0)$ είναι αντιστρεπτό αν για κάθε σήμανση $m \in R(m_0)$ η αρχική σήμανση $m_0 \in R(m)$ όπου

- $R(m_0)$ το σύνολο δυνατών σημάνσεων από τη m_0
- $R(m)$ το σύνολο δυνατών σημάνσεων από τη m .

Η σήμανση $m' \in R(m_0)$ ονομάζεται αρχική κατάσταση (home marking) αν $\forall m \in R(m_0)$, και η m' είναι προσβάσιμη από την m . Δηλαδή η αρχική κατάσταση είναι μία σήμανση στην οποία καταλήγει πάντα το σύστημα σε όποια σήμανση του $R(m_0)$ και αν βρίσκεται.

Ένα δίκτυο Petri είναι δομικά αντιστρεπτό αν υπάρχει πεπερασμένη αρχική σήμανση που κάνει το δίκτυο αντιστρεπτό. Η αντιστρεπτότητα ενός δικτύου Petri σημαίνει ότι για μια οποιαδήποτε σήμανση m προσβάσιμη από την m_0 , η m_0 είναι επίσης προσβάσιμη από την m . Η συγκεκριμένη ιδιότητα εγγυάται την κυκλική συμπεριφορά του συστήματος, κάτι το οποίο απαιτείται ιδιαίτερα σε περιπτώσεις ανάκαμψης από λάθος δηλαδή από μια κατάσταση αποτυχίας στις επερχόμενες σωστές καταστάσεις [5].

2.2.5 Μέθοδοι ανάλυσης δικτύων Petri

Η μαθηματική ανάλυση των δικτύων Petri πραγματοποιείται κυρίως με τις εξής μεθόδους.

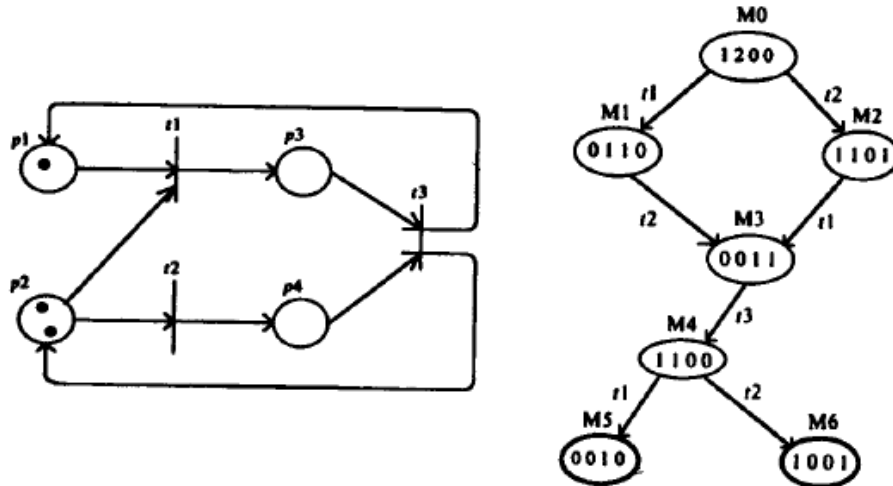
- μέθοδο ανάλυσης μέσω προσβασιμότητας (Reachability Analysis Method)
- μέθοδο ανάλυσης μέσω αναλλοίωτων (Invariant Analysis Method) [2],[5].

Μέθοδος Ανάλυσης μέσω Προσβασιμότητας

Σε αυτή τη μέθοδο, ξεκινώντας από την αρχική κατάσταση του συστήματος, απαριθμούνται διαδοχικά όλες οι πιθανές καταστάσεις στις οποίες μπορεί να βρεθεί το σύστημα. Η βασική μεθοδολογία που ακολουθείται είναι ότι για κάθε σήμανση αναγνωρίζονται όλες οι ενεργοποιούμενες μεταβάσεις και η αντίστοιχη σήμανση που προκύπτει για καθεμία από αυτές. Ένα τόξο που συνδέει έναν κόμβο με έναν άλλο έχει σημειωμένη τη μετάβαση η οποία εκκίνησε και δημιούργησε την αλλαγή στην σήμανση. Η μεθοδολογία αυτή μπορεί να απεικονιστεί με ένα γράφημα που ονομάζεται Δένδρο Προσβασιμότητας (reachability tree

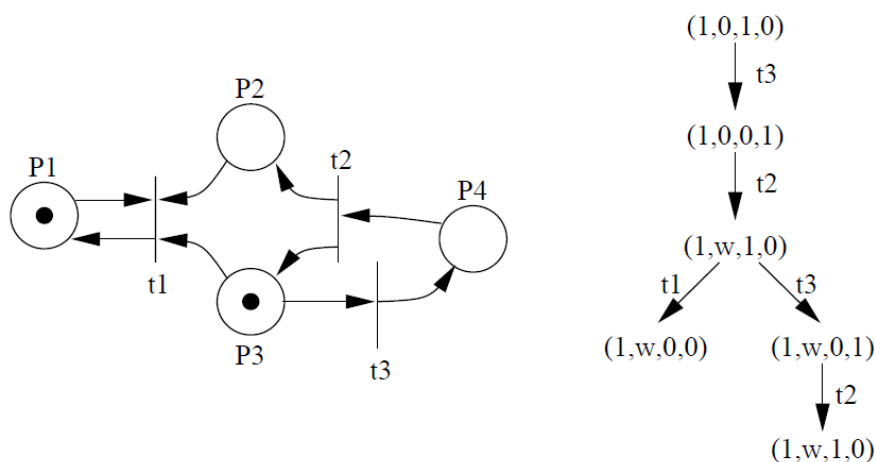
or graph). Στο δένδρο αυτό κάθε σήμανση απεικονίζεται ως κόμβος και συνδέεται μέσω τόξων με τις προκύπτουσες σημάνσεις.

Μέσω του δένδρου προσβασιμότητας είναι δυνατή η άμεση επαλήθευση προσβασιμότητας, ζωτικότητας και αντιστρεπτότητας [1],[2],[5],[6].



Δένδρο προσβασιμότητας μοντέλου δικτύου Petri, Πηγή : [2]

Όταν ένα δίκτυο Petri είναι μη-φραγμένο ή ο αριθμός των καταστάσεων του μοντελοποιημένου συστήματος είναι μη πεπερασμένος, τότε η παραπάνω διαδικασία θα συνεχίζεται για πάντα. Για να διατηρηθεί το δένδρο πεπερασμένο χρησιμοποιούνται ορισμένοι κόμβοι που αντιπροσωπεύουν άπειρο αριθμό καταστάσεων. Στη σήμανση των συγκεκριμένων κόμβων μία ή περισσότερες θέσεις περιέχουν απροσδιόριστο αριθμό token ο οποίος παριστάνεται με w . Ακολουθεί ένα παράδειγμα δημιουργίας πεπερασμένου δένδρου προσβασιμότητας :



Δημιουργία πεπερασμένου δένδρου προσβασιμότητας, Πηγή: [6]

Η αρχική σήμανση του συγκεκριμένου δικτύου έχει tokens στις θέσεις $p1$ και $p3$ και παριστάνεται με τον κόμβο $(1,0,1,0)$ στη ρίζα του δένδρου προσβασιμότητας (η σήμανση του κόμβου αναφέρεται στις θέσεις $p1$, $p2$, $p3$ και $p4$ αντίστοιχα).

Από αυτήν την κατάσταση η μόνη ενεργοποιημένη μετάβαση είναι η $t3$. Όταν αυτή εκκινεί ένα token προστίθεται στη θέση $p4$ οδηγώντας στη σήμανση $(1,0,0,1)$. Εν συνεχεία, ενεργοποιείται η $t2$ οδηγώντας στη σήμανση $(1,1,1,0)$. Εφόσον η $(1,0,1,0)$ είναι υποσύνολο της $(1,1,1,0)$, η αλληλουχία $t3 - t2$ θα εμφανίζεται ξανά και ξανά στο δένδρο προσβασιμότητας. Για το λόγο αυτό, σε αυτό το σημείο, ο αριθμός token στη θέση $p2$ αντικαθίσταται με το σύμβολο w . Από αυτή τη σήμανση ενεργοποιούνται οι $t1$ και $t3$ παράγοντας τις σημάνσεις $(1,w,0,0)$ και $(1,w,0,1)$ αντίστοιχα. Η πρώτη από αυτές δεν ενεργοποιεί καμία μετάβαση (νεκρή σήμανση) ενώ η δεύτερη ενεργοποιεί την $t2$. Η εκκίνηση της $t2$ παράγει την $(1,w,1,0)$ που υπάρχει ήδη. Άρα το δένδρο προσβασιμότητας έχει ολοκληρωθεί[6].

Μέθοδος ανάλυσης μέσω αναλλοίωτων

Με τη μέθοδο αυτή εξετάζονται βασικές ιδιότητες των δικτύων Petri μελετώντας τις γραμμικές εξισώσεις που προκύπτουν από τους κανόνες εκτέλεσης και τον πίνακα συμβάντων C .

P-αναλλοίωτες

Οι θετικές ακέραιες λύσεις του x στην εξίσωση $C^T \cdot x = 0$, αποκαλούνται P-αναλλοίωτες. Αποδεικνύεται ότι οι P-αναλλοίωτες δείχνουν την ελάχιστη δομή των κλειστών διαδρομών μέσα στο δίκτυο Petri. Ακόμα αν κάθε θέση στο δίκτυο καλύπτεται από μια P-αναλλοίωτη τότε το δίκτυο είναι φραγμένο [1],[2],[5].

Για το PN του παραδείγματος της ενότητας 2.2.2 με $x^T = \{w1, w2, w3, w4\}$ η εξίσωση $x^T A = 0$ δίνει σύστημα τριών εξισώσεων με αγνώστους: $-w1 + w3 = 0$, $-w2 + w4 = 0$, $w1 + w2 - w3 - w4 = 0$. Το σύστημα αυτό έχει λύσεις $x_1 = \{1, 0, 1, 0\}$, $x_2 = \{0, 1, 0, 1\}$, $x_3 = \{1, 1, 1, 1\}$. Τα διανύσματα x_1 και x_2 καλύπτουν το x_3 άρα υπάρχουν δύο P-αναλλοίωτες οι οποίες αντιστοιχούν στα κυκλώματα $p1-p3$ και $p2-p4$. Επίσης και οι τέσσερις θέσεις του PN καλύπτονται από P-αναλλοίωτη επομένως το δίκτυο είναι φραγμένο [2].

T-αναλλοίωτες

Οι ακέραιες λύσεις του y στην εξίσωση $C \cdot y = 0$ αποκαλούνται T-αναλλοίωτες. Μια T-αναλλοίωτη υποδεικνύει πόσες φορές ενεργοποιείται μια μετάβαση μέσα σε ένα κύκλο ώστε να επανέλθει η αρχική σήμανση στον εαυτό της.

Για το PN του παραδείγματος της ενότητας 2.2.2 με $y^T = \{w1, w2, w3, w4\}$ η εξίσωση $Ay = 0$ δίνει σύστημα τεσσάρων εξισώσεων με τρεις αγνώστους: $-w1 + w3 = 0$, $-w2 + w3 = 0$, $w1 - w3 = 0$, $w2 - w3 = 0$, από το οποίο προκύπτει η λύση $y^T = \{1, 1, 1\}$. Αυτό σημαίνει ότι και οι τρεις μεταβάσεις ενεργοποιούνται μία φορά μέσα σε ένα κύκλο [2].

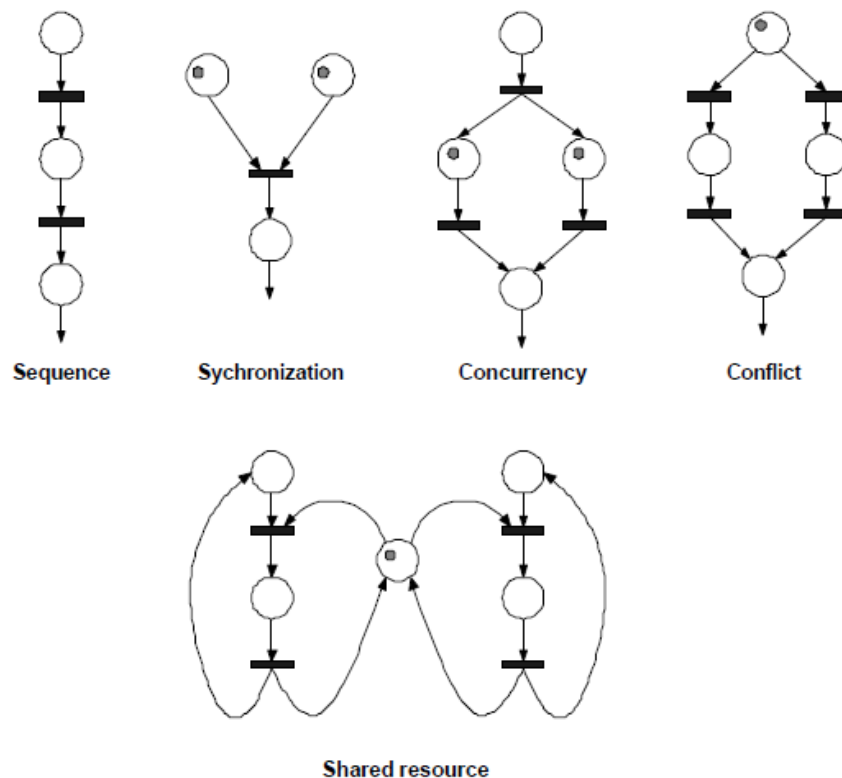
2.2.6 Δυνατότες μοντελοποίησης δικτύων Petri

Ακολουθία: διαφορετικά γεγονότα συμβαίνουν διαδοχικά με καθορισμένη σειρά

Συγχρονισμός: ένα γεγονός εξαρτάται ταυτόχρονα από περισσότερες από μια δραστηριότητες /γεγονότα.

Παράλληλεια: διαφορετικά γεγονότα που ενεργοποιούνται παράλληλα και δεν αλληλεπιδρούν συμβαίνουν ανεξάρτητα

Σύγκρουση: διαφορετικά γεγονότα διεκδικούν κάποιον κοινό πόρο για να ενεργοποιηθούν



Δυνατότες μοντελοποίησης δικτύων Petri

2.3 Επεκτάσεις δικτύων Petri

Παράλληλα με τα κλασικά δίκτυα Petri έχουν παρουσιαστεί διαφορές επεκτάσεις αυτών με μια σειρά χαρακτηριστικών όπως ο χρόνος, οι τύποι μεταβλητών και η ιεραρχία. Στόχος τους είναι η παροχή πιο περιεκτικών και παραστατικών μοντέλων με περισσότερες δυνατότητες μοντελοποίησης.

Χρονισμένα δίκτυα Petri

Το βασικά μοντέλα PN είναι ασύγχρονα και δεν περιλαμβάνουν καμία πληροφορία χρόνου. Η ενσωμάτωση μηχανισμών χρόνου επιτρέπουν στα PN να χρησιμοποιηθούν στην

προσομοίωση συστημάτων όπου είναι συχνά σημαντικό να περιγραφεί χρονικά η συμπεριφορά του συστήματος. Η υλοποίηση τέτοιων μηχανισμών γίνεται συνήθως με την προσθήκη χρονικών καθυστερήσεων στις θέσεις ή τις μεταβάσεις του μοντέλου.

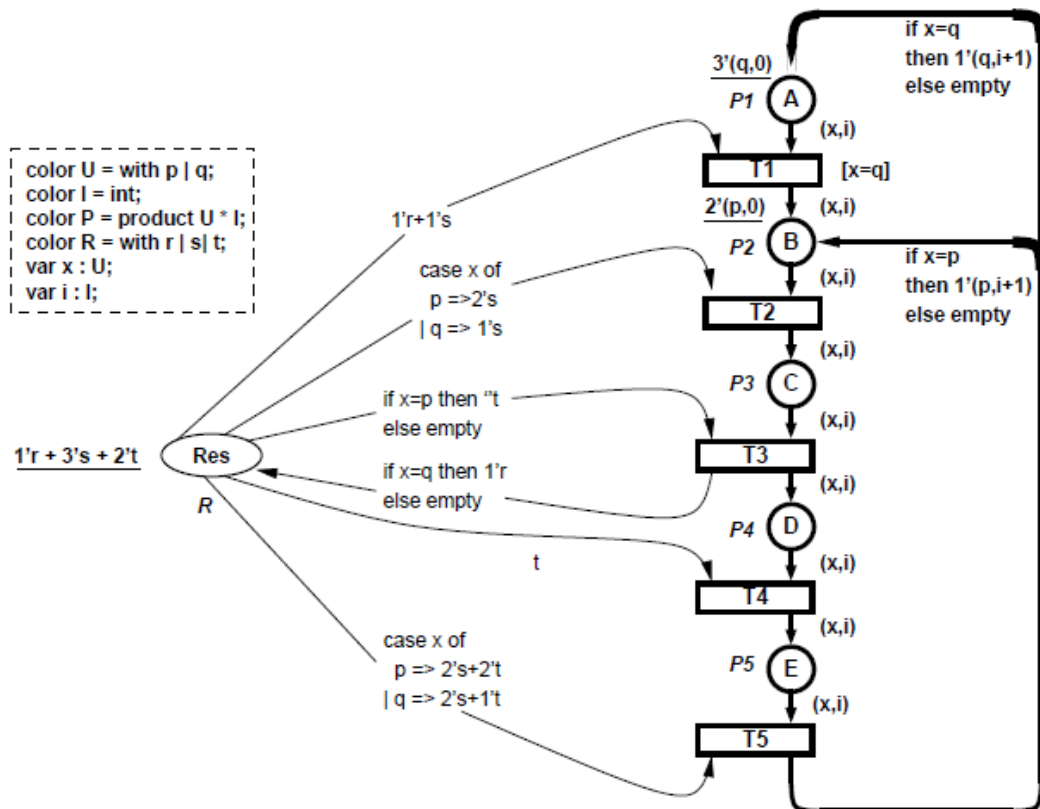
Αρχικά το χαρακτηριστικό του χρόνου εισήχθη στα Petri Nets για την αξιολόγηση της απόδοσης των μοντέλων. Το χρονισμένο PN είχε μια σειρά από καθορισμένες χρονικές καθυστερήσεις που αντισπροσώπευαν χρονικούς περιορισμούς αναφερόμενους στις μεταβάσεις του δικτύου. Με αυτόν τον τρόπο, η εκκίνηση μιας ενεργοποιημένης μετάβασης εμποδίζεται για χρόνο ίσο με τη χρονική καθυστέρηση. Αν η μετάβαση παραμένει ενεργοποιημένη και μετά τη λήξη του αντίστοιχου χρονικού διαστήματος, τότε μπορεί να εκκινήσει. Ωστόσο οι ντετερμινιστικές χρονικές καθυστερήσεις δεν ήταν κατάλληλες για όλες τις περιπτώσεις συστημάτων. Για παράδειγμα τα Ευέλικτα Συστήματα Κατεργασιών μπορεί να παρουσιάζουν τυχαία συμπεριφορά. Για τη μοντελοποίηση τέτοιων συστημάτων προτείνονται στοχαστικά δίκτυα PN. Ένα στοχαστικό δίκτυο είναι ένα δίκτυο όπου στις χρονικές καθυστερήσεις αντιστοιχίζονται τυχαίες τιμές του χρόνου[6].

Υψηλού επιπέδου δίκτυα Petri

Οι Genrich και Lautenbach [Genrich 1991] αναγνώρισαν ότι τα κλασικά PN περιέχουν επαναλαμβανόμενα στοιχεία για τη μοντελοποίηση παρόμοιων συμπεριφορών πολλαπλών αντικειμένων. Τα δίκτυα PrT (Predicate Transition Nets) ήταν αυτά που σχεδιάστηκαν πρώτα με στόχο τη μείωση αυτών των στοιχείων μέσω της συγχώνευσης τους σε ένα. Αυτό επιτυγχάνεται αντικαθιστώντας τα κλασικά με άλλου τύπου token (predicates) τα οποία αποδίδουν στην εκάστοτε θέση κάποια συγκεκριμένη σημασία. Επιπροσθέτως, τα τόξα του δικτύου σημειώνονται με εκφράσεις και οι μεταβάσεις με συναρτήσεις/συνθήκες. Η τοπολογία αυτή καθορίζει τελικά ποια token μπορούν να ενεργοποιήσουν μια μετάβαση και ποια είναι αυτά που προκύπτουν από την εκκίνησή της.

Ο Jensen το 1992 εμπνευσμένος από τα PrT-nets, εισήγαγε τα Coloured Petri nets (CP-nets ή CPN). Παρόλο που παρουσιάζονται με λίγο διαφορετικό τρόπο, τα CP-nets είναι αρκετά παρόμοια με τα PrT-nets. Το κίνητρο για τη δημιουργία των CP-nets ήταν οι δυσκολίες στην ανάλυση εξαιτίας της αντιστοίχισης των μεταβλητών που εμφανίζονταν στις εκφράσεις των τόξων. Στα CP-nets οι δυσκολίες αυτές αποφεύγονται με την αντικατάσταση των εκφράσεων των τόξων με συναρτήσεις.

Σε ένα κλασικό PN τα token έχουν δυαδική φύση (δηλώνουν την παρουσία ή την απουσία token). Ένα CPN είναι ένα δίκτυο Petri στο οποίο τα token είναι τιμές ενός τύπου δεδομένων. Οι τιμές αυτές ονομάζονται colours (χρώματα). Τα colours μπορεί να είναι τιμές τύπων δεδομένων ή σύνθετων τύπων δεδομένων όπως λίστες. Η δομή και η λειτουργία των μοντέλων CPN γίνεται περισσότερο αντιληπτή από το επόμενο παράδειγμα που περιγράφει την κατανομή πόρων σε ένα σύστημα[6].



Παράδειγμα μοντέλου Coloured petri δικτύου, Πηγή : [6]

Στο πλαίσιο πάνω αριστερά φαίνονται όλοι οι τύποι των θέσεων του PN και οι τιμές οι οποίες μπορεί να πάρουν. Τα σύνολα (coloursets) U, I, R είναι απλοί τύποι ενώ το P είναι ένας σύνθετος τύπος.

Δίπλα σε κάθε θέση σημειώνεται το όνομα του συνόλου τιμών που μπορεί να πάρει κάθε θέση ($R, P1, P2, P3, P4, P5$) ενώ τα υπογραμμισμένα στοιχεία δείχνουν την αρχική σήμανση των θέσεων.

Στο μοντέλο του παραδείγματος υπάρχουν δύο τύποι διαδικασιών p και q . Στην αρχική κατάσταση του συστήματος τρεις q διαδικασίες βρίσκονται στη θέση και τρεις p διαδικασίες στη θέση B . Επίσης στη θέση R βρίσκονται: ένας πόρος r , τρεις πόροι s και δύο πόροι t . Οι διαδικασίες p και q εξελίσσονται περνώντας από τις θέσεις A, B, C, D, E βάσει των επιγραφών των τόξων. Οι επιγραφές αυτές καθορίζουν πόσοι πόροι κάθε τύπου απαιτεί μία διαδικασία καθώς περνά από τη μία φάση στην άλλη. Για παραδειγμα, η επιγραφή $\text{case } x \text{ of } p \Rightarrow 2's \mid q \Rightarrow 1's$ (στο τόξο από τη θέση Res στη μετάβαση $T2$) δηλώνει ότι μια p -διαδικασία απαιτεί δύο s -πόρους για να περάσει στη θέση (φάση) C , ενώ μια q -διαδικασία απαιτεί έναν s -πόρο. Η παράμετρος x αναφέρεται στη μεταβλητή x του στοιχείου (x,i) του τόξου που αφήνει τη θέση B και εισέρχεται στην ίδια μετάβαση.

Επίσης και οι μεταβάσεις είναι δυνατόν να περικλείουν συνθήκες ενεργοποίησης οι οποίες εκφράζονται ως εκφράσεις Boolean. Για παράδειγμα, η μετάβαση $T1$ η οποία σημειώνεται με την επιγραφή $[x=q]$ επιτρέπει να περάσουν μόνο τα tokens που αναφέρονται σε διαδικασίες q [6],[11].

Ιεραρχικά δίκτυα Petri

Η ενσωμάτωση τεχνικών ιεραρχικής δόμησης μέσα στα δίκτυα Petri περιλαμβάνει συνήθως αντικατάσταση των στοιχείων (θέσεων ή των μεταβάσεων) του δικτύου με ένα πιο χαμηλότερου επιπέδου δίκτυο. Ένα τέτοιο υπο-δίκτυο παρουσιάζει μια πιο αναλυτική και λεπτομερή απεικόνιση για την εργασία που περιγράφεται από το στοιχείο που αντικαθιστά. Με αυτόν τον τρόπο, τα μοντέλα Petri Net μπορούν να δομηθούν ως ένα σύνολο επιπέδων. Τα επίπεδα αυτά αλληλεπιδρούν μεταξύ τους μέσα από ένα σύνολο κατάλληλα ορισμένων διεπιφανειών ώστε να μπορούν διαχειριστούν μεγαλύτερες εφαρμογές [6].

2.4 Βιβλιογραφική μελέτη εφαρμογών με δίκτυα Petri

Στην ενότητα αυτή γίνεται αρχικά μια σύντομη αναφορά σε περιπτώσιολογικές μελέτες και παραδείγματα εφαρμογών μοντελοποίησης με δίκτυα Petri που σχετίζονται κυρίως με το αντικείμενο της παρούσας διπλωματικής:

- **A Modeling Method for Flexible Manufacturing Systems based on Coloured Petri Nets**

Σε αυτή την εργασία ο Armin Zimmermann μοντελοποιεί ένα ευέλικτο συστήμα κατεργασιών με διαφορετικές παραλλαγές των δικτύων Petri. Το σύστημα αποτελείται από δύο μηχανές κατεργασίας, ένα ρομπότ και δύο ταινιόδρομους για τη διακίνηση του υλικού στο σύστημα από το οποίο παράγονται δύο τύποι τεμαχίων.

Αρχικά το σύστημα μοντελοποιείται με γενικευμένα στοχαστικά δίκτυα Petri (GSPN). Στην περίπτωση αυτή, η ίδια εργασία που γίνεται στους δύο διαφορετικούς τύπους τεμαχίων πρέπει να παριστάνεται με δύο ξεχωριστές θέσεις. Αποδεικνύεται ότι η συγκεκριμένη μοντελοποίηση οδηγεί σε αρκετά αναλυτικό και σχετικά ογκώδες και δυσνόητο μοντέλο.

Στη συνέχεια το ίδιο σύστημα μοντελοποιείται με “χρωματιστά” γενικευμένα στοχαστικά GS-CPN. Σε αυτό το μοντέλο τα χρωματισμένα tokens δηλώνουν τον τύπο και τη φάση επεξεργασίας των τεμαχίων ενώ υπάρχουν και κάποια μη-χρωματισμένα tokens. Το μοντέλο που προκύπτει εμφανίζεται απλούστερο και πιο ευανάγνωστο. Ωστόσο απαιτεί τη δήλωση των συνόλων χρωμάτων των θέσεων, των μεταβλητών και των συνθηκών ενεργοποίησης των μεταβάσεων.

- **Petri nets and manufacturing systems : An examples-driven tour**

Οι L.Recalde, M. Silva, J. Ezpeleta, E.Teruel παρουσιάζουν και αναλύουν παραδείγματα βιομηχανικών συστημάτων παραγωγής που έχουν μοντελοποιηθεί με δίκτυα Petri.

Το πρώτο παράδειγμα αφορά ένα κύτταρο κατεργασιών και συναρμολόγησης. Δύο διαφορετικά τεμάχια Α,Β φτάνουν, με τη βοήθεια ταινιόδρομων, στις μηχανές Μ1 και Μ2 αντίστοιχα προς επεξεργασία. Η διάταξη διαθέτει βιομηχανικό ρομπότ για τη φόρτωση/εκφόρτωση των τεμαχίων στις/από τις Μ1 και Μ2 αλλά και τη φόρτωση των επεξεργασμένων τεμαχίων στη Μ3 όπου παράγεται το τελικό τεμάχιο. Λόγω της μεγάλης σχετικά συχνότητας βλαβών που παρουσιάζει η Μ1, τοποθετείται ενδιάμεσος αποθηκευτικός χώρος (buffer) που επιτρέπει, για κάποιο διάστημα, την ανεξαρτησία του κύκλου εργασιών. Σχολιάζεται διαγραμματικά η απόδοση του κυττάρου σε σχέση με τη συχνότητα βλαβών για διαφορετικές χωρητικότητες του buffer. Στη συνέχεια, για σταθερή χωρητικότητα του buffer, αναζητάται ο βέλτιστος έλεγχος των ελεγχόμενων μεταβάσεων μέσω διαγραμμάτων Gant όπου αποτυπώνεται η διάρκεια και η διακοχή των εργασιών.

Στη συνέχεια γίνεται αναφορά στη μοντελοποίηση κλασικών στρατηγικών διαχείρισης της παραγωγής όπως kanban ,JIT με δίκτυα Petri και στα χρήσιμα δεδομένα που προκύπτουν σχετικά με την απόδοση των μοντελοποιημένων συστημάτων.

Το επόμενο παράδειγμα αφορά τη μοντελοποίηση γραμμής παραγωγής σε αυτοκινητοβιομηχανία με δίκτυα Coloured Petri . Και στα δύο παραδείγματα η γραμμή παραγωγής αποτελείται από πολλούς σταθμούς εργασιών που λειτουργούν με παρόμοιο τρόπο και ένα μεταφορικό σύστημα μονής κατεύθυνσης που μεταφέρει τα 'σώματα' των αυτοκινήτων σε ένα σταθμό εργασίας/συναρμολόγησης. Το ολοκληρωμένο μοντέλο του συστήματος προκύπτει από τη συγχώνευση των μεταβάσεων από τα υπομοντέλα για τους σταθμούς εργασίας και το μεταφορικό σύστημα

Ακολουθεί ένα ανάλογο παράδειγμα δικτύου Coloured Petri για τη μοντελοποίηση γραμμής παραγωγής συσκευών μικροκυμάτων. Παράλληλα παρουσιάζονται για τα παραπάνω παραδείγματα τεχνικές αποφυγής deadlocks αλλά και η αξιολόγηση διαφορετικών τεχνικών προγραμματισμού των μοντελοποιημένων συστημάτων.

- **Application of design and control tools in a multirobot cell**

Οι Jaejin Jang, Pyung-Hoi Koo και Shimon Y.NOF ασχολήθηκαν με τον έλεγχο ενός βιομηχανικού κυττάρου αποτελούμενο από δύο ρομπότ, ένα ελεγκτή, ταινιόδρομο και δύο μηχανές επεξεργασίας. Για τον προγραμματισμό του κυττάρου χρησιμοποιήθηκαν δύο διαφορετικά σχήματα ελέγχου: σύγχρονο και ασύγχρονο. Στο πρώτο περιορίζεται σημαντικά ο αριθμός των σημάτων επικοινωνίας μεταξύ των στοιχείων του κυττάρου αλλά απαιτείται η εκτίμηση των χρόνων κινήσεων των ρομπότ εκ των προτέρων. Το δεύτερο έχει αυξημένη πολυπλοκότητα αλλά είναι πιο ευέλικτο και αξιόπιστο σε απρόβλεπτες συμπεριφορές του συστήματος. Παράλληλα κατασκευάζεται το αντίστοιχο δίκτυο Petri και το δένδρο προσβασιμότητας για να επιβεβαιωθεί η απουσία deadlocks που μπορεί να 'κρύβει' ένα ασύγχρονο μοντέλο ελέγχου.

- **A Coloured Petri Net Model for Automated Storage and Retrieval Systems Serviced by rail-Guided vehicles : a Control Perspective**

Σε αυτό το Paper παρουσιάζεται η μοντελοποίηση ενός αυτόματου συστήματος διαχείρισης αποθήκης (AS/RS) με χρονισμένα «χρωματιστά» δίκτυα Petri. Το συγκεκριμένο σύστημα

που αναλύεται εδώ αποτελείται από παράλληλα διατεταγμένα ράφια και συστήματα διακίνησης τεμαχίων από τους σταθμούς παραλαβής και των θέσεων αποθήκευσης (ή αντίστροφα) όπως ταινιόδρομους, γερανούς, οχήματα RGVs.

Ο προγραμματισμός του ελέγχου στο σύστημα αυτό πραγματοποιείται σε δύο επίπεδα : στο 1^ο επίπεδο ορίζονται η τακτική για την αποδοτική αποθήκευση/ ανάκτηση των τεμαχίων αλλά και το πότε θα δρομολογείται κάθε επόμενη διαδικασία. Το δεύτερο επίπεδο αφορά την αξιολόγηση κάθε επόμενης εργασίας ώστε να αποφεύγονται οι συγκρούσεις και οι καταστάσεις deadlock.

Στο συγκεκριμένο CTPN μοντέλο οι θέσεις αντιστοιχούν σε όλες τις πραγματικές θέσεις που μπορεί να βρίσκεται το τεμάχιο είτε όταν είναι αποθηκευμένο σε ράφι είτε κατά τη μεταφορά του. Τα κουπόνι δηλώνει είτε μια διαδικασία που βρίσκεται σε εξέλιξη είτε την παρουσία πόρων, γερανού ή RGV, στη συγκεκριμένη θέση. Τα χρώματα που αποδίδονται στα κουπόνια αντιπροσωπεύουν τις εργασίες αποθήκευσης ανάκτησης που ανατίθενται στο σύστημα ενώ οι μεταβάσεις τα ελεγχόμενα γεγονότα μέσω των οποίων μεταβάλλονται οι καταστάσεις του συστήματος.

Η δυναμική του CTPN περιγράφεται από τον πίνακα συμβάντων $C(ri,t) = Post(ri,t) - Pre(ri,t)$. Στο απόσπασμα 4 δίνονται οι κανόνες βάσει των οποίων ορίζονται τα μητρώα των συναρτήσεων εισόδου $Pre(ri,t)$ και συναρτήσεων εξόδου $Post(ri,t)$ για το δεδομένο σύστημα αλλά και οι συνθήκες κάτω από τις οποίες, τα χρωματισμένα tokens ενεργοποιούν κάθε μεταβάση.

Στο τελευταίο απόσπασμα της μελέτης παρουσιάζεται η εφαρμογή και αξιολόγηση της μοντελοποίησης στην πλατφόρμα Stateflow του Matlab με βάση τη ρυθμαπόδοση του συστήματος, το χρόνο ολοκλήρωσης όλων των εργασιών και το βαθμό χρήσης των πόρων. Τα αποτελέσματα λαμβάνονται για τρεις διαφορετικές τακτικές προγραμματισμού αλλά και για τις περιπτώσεις χρήσης ενός/δύο οχημάτων RGVs.

2.5 Μοντελοποίηση συστήματος της παρούσας εργασίας με δίκτυα Petri

Σε πρώτη φάση ορίζονται οι θέσεις που απαιτούνται για την πλήρη μοντελοποίηση του συστήματος που μελετάται στην παρούσα εργασία. Οι θέσεις αυτές θα πρέπει να περιλαμβάνουν :

- όλες τις εργασίες του συστήματος οι οποίες διακρίνονται κατα βάση σε:
 - εργασίες φόρτωσης/εκφόρτωσης εκτελούμενες από το ρομπότ οι οποίες παρουσιάστηκαν αναλυτικά στο προηγούμενο κεφάλαιο.

Παρακάτω φαίνεται για πόσα και ποια τεμάχια εκτελείται καθεμία από αυτές τις εργασίες:

- μεταφορά τεμαχίου από την αρχική αποθήκη εισόδου στον τόρνο (A,B,C)
- μεταφορά τεμαχίου από την αρχική αποθήκη εισόδου στη φρέζα (D,E,F)
- μεταφορά τεμαχίου από τον τόρνο στην τελική αποθήκη (A,C,E)

- μεταφορά τεμαχίου από τη φρέζα στην τελική αποθήκη (D,F,B)
- μεταφορά τεμαχίου από τον τόρνο στη φρέζα (B,C,F)
- μεταφορά τεμαχίου από τη φρέζα στον τόρνο (E,F,C)
- μεταφορά τεμαχίου από τον τόρνο στην ενδιάμεση αποθήκη (B,C,F)
- μεταφορά τεμαχίου από τη φρέζα στην ενδιάμεση αποθήκη (E,F,C)
- μεταφορά τεμαχίου από την ενδιάμεση αποθήκη στη φρέζα (B,C,F)
- μεταφορά τεμαχίου από την ενδιάμεση αποθήκη στον τόρνο (E,F,C)

Αν χρησιμοποιηθεί μία θέση στο PN για κάθε εργασία μεταφοράς σε κάθε τεμάχιο απαιτούνται 30 θέσεις.

- εργασίες κατεργασίας των τεμαχίων εκτελούμενες από τις μηχανές κατεργασίας οι οποίες παρουσιάστηκαν αναλυτικά στο προηγούμενο κεφάλαιο. Συνολικά εκτελούνται 12 διαφορετικές κατεργασίες (6 στον τόρνο και 6 στη φρέζα). Αν χρησιμοποιηθεί μία θέση στο PN για κάθε κατεργασία απαιτούνται 12 διαφορετικές θέσεις
- τους πόρους του συστήματος (κέντρο τόννευσης, φρέζα και ρομπότ). Αν χρησιμοποιηθεί μία θέση στο PN για κάθε πόρο του συστήματος απαιτούνται 3 διαφορετικές θέσεις.
- τους αποθηκευτικούς χώρους (αρχική, ενδιάμεσες και τελική αποθήκη τεμαχίων). Αν χρησιμοποιηθεί μία θέση στο PN για κάθε αποθηκευτικό χώρο του συστήματος απαιτούνται 4 διαφορετικές θέσεις.

Επομένως, αν όλοι οι πόροι, οι αποθήκες και οι εργασίες για κάθε τεμάχιο μοντελοποιηθούν ξεχωριστά, το PN θα έχει τουλάχιστον 49 θέσεις. Ο αριθμός αυτός είναι ιδιαίτερα μεγάλος και σίγουρα θα οδηγήσει σε ένα ογκώδες και πολύπλοκο μοντέλο Petri. Η πολυπλοκότητα αυτή αναμένεται ακόμα μεγαλύτερη μετά την πρόσθεση των μεταβάσεων αλλά και των τόξων που θα ορίζουν τις αλληλεπιδράσεις μεταξύ αυτών.

Σε αυτό το σημείο και με βάση την προηγούμενη βιβλιογραφία (και ιδιαίτερα των πηγών [4] και [7]) γίνεται εμφανής η ανάγκη χρήσης δικτύων Coloured Petri. Όπως και στα παραδείγματα [4],[7] και [8] έτσι και στο σύστημα κατεργασιών της παρούσας εργασίας υπάρχουν εργασίες αλλά και πόροι που λειτουργούν με τον ίδιο τρόπο. Τα στοιχεία αυτά μπορούν να συγχωνευτούν σε πολύ λιγότερα ορίζοντας παράλληλα τα απαραίτητα coloursets. Έτσι, οι θέσεις του μοντέλου ρίζονται εκ νέου ως εξής :

- κάθε διαφορετική κίνηση του ρομπότ αντιπροσωπεύεται από μία θέση ανεξάρτητα από τον τύπο τεμαχίου και τη μηχανή κατεργασίας. Με αυτόν τον τρόπο προκύπτουν οι παρακάτω πέντε διαφορετικές θέσεις στο CPN:
 - μεταφορά τεμαχίου από την αρχική αποθήκη στη μηχανή κατεργασίας

- μεταφορά τεμαχίου από τη μηχανή κατεργασίας στην τελική αποθήκη
 - μεταφορά τεμαχίου από τη μία μηχανή κατεργασίας στην άλλη
 - μεταφορά τεμαχίου από τη μηχανή κατεργασίας στην ενδιάμεση αποθήκη
 - μεταφορά τεμαχίου από την ενδιάμεση αποθήκη στη μηχανή κατεργασίας
-
- όλες οι κατεργασίες συγχωνεύονται σε μία θέση CPN
 - οι δύο ενδιάμεσες αποθήκες συγχωνεύονται σε μία θέση CPN
 - οι δύο μηχανές κατεργασίας συγχωνεύονται σε μία θέση CPN
 - το ρομπότ αντιπροσωπείται από μία θέση CPN

Άρα η μοντελοποίηση με Coloured Petri δίκτυα μπορεί να γίνει με 11 θέσεις αντί για 49 που απαιτούν τα απλά Petri. Από την άλλη πλευρά η χρήση των Coloured Petri συνεπάγεται την εισαγωγή :

- coloursets που θα καθορίζουν τους τύπους και τις τιμές των tokens κάθε θέσης
- συγκεκριμένων συνθηκών/συναρτήσεων τόξων και μεταβάσεων με καθορισμένη σύνταξη που εξαρτάται εν μέρει και από το εργαλείο λογισμικού που θα χρησιμοποιηθεί

Ο ορισμός των coloursets και των υπόλοιπων στοιχείων του μοντέλου CPN γίνεται στο Κεφάλαιο 4 όπου εξηγείται λεπτομερώς όλη η διαδικασία κατάστρωσής του.

Για τη μοντελοποίηση του μελετούμενου συστήματος επιλέχθηκε το πρόγραμμα CPN tools που παρουσιάζεται αναλυτικά στο επόμενο κεφάλαιο.

Κεφάλαιο 3 : Το πρόγραμμα CPN tools

3.1 Εισαγωγή

Τα Coloured Petri Nets (CPNs) είναι μια ιδιαίτερη κατηγορία δικτύων Petri που χρησιμοποιείται για τη μοντελοποίηση και την επαλήθευση συστημάτων τα οποία χαρακτηρίζονται από παραλληλες / ταυτόχρονες λειτουργίες, συγχρονισμό και επικοινωνία. Η CPN είναι μια γλώσσα διακριτών γεγονότων που συνδυάζει τη θεωρία των δικτύων Petri με τη λειτουργική γλώσσα προγραμματισμού Standard ML. Το μοντέλο CPN ενός συστήματος είναι ένα εκτελέσιμο μοντέλο το οποίο αναπαριστά τις καταστάσεις του συστήματος (θέσεις) και τα γεγονότα (μεταβάσεις) που αλλάζουν την κατάσταση αυτού. Με τη γλώσσα CPN μπορεί να μοντελοποιηθεί μια μεγάλη γκάμα συστημάτων όπως πρωτόκολλα επικοινωνίας, δίκτυα δεδομένων, διανεμημένοι αλγόριθμοι, embedded συστήματα.

Το CPN tools είναι ένα υπολογιστικό εργαλείο κατάστρωσης και ανάλυσης μοντέλων CPN. Χρησιμοποιώντας το συγκεκριμένο εργαλείο είναι δυνατή η διερεύνηση της συμπεριφοράς και των ιδιοτήτων του μοντελοποιημένου συστήματος. Η διάδραση του χρήστη με το πρόγραμμα CPN tools βασίζεται στον άμεσο χειρισμό του γραφήματος του CPN μοντέλου μέσω μεθόδων αλληλεπίδρασης όπως παλέτες εργαλείων και marking menus.

Αρχικά, ο χρήστης θα πρέπει να συνθέσει το γράφημα του μοντέλου από τα βασικά δομικά στοιχεία χρησιμοποιώντας την εργαλειοθήκη του CPN tools και εισάγοντας τις απαραίτητες μεταβλητές, συνθήκες και συναρτήσεις. Στη συνέχεια μέσω της προσομοίωσης ενός μοντέλου CPN, μπορεί να εξετάσει και να ελέγξει διάφορα σενάρια τη λειτουργίας του μοντελοποιημένου συστήματος. Η εξακρίβωση των ιδιοτήτων του μοντέλου, που αποτελεί και το πιο ουσιαστικό κομμάτι του συγκεκριμένου εργαλείου, γίνεται εφαρμόζοντας μεθόδους state space που θα αναλυθούν στη συνέχεια.

Όπως και δε όλα τα δίκτυα Petri έτσι και εδώ τα βασικά δομικά στοιχεία ενός μοντέλου CPN είναι οι θέσεις οι μεταβάσεις και τα τόξα. Το σύνολο των θέσεων και οι μεταβάσεις καλούνται κόμβοι οι οποίοι σε συνδυασμό με τα τόξα δίνουν τη δομή του δικτύου. Βασική διαφορά με τα απλά Petri είναι η δυνατότητα δημιουργίας Coloursets δηλαδή συνόλων token διαφορετικού τύπου. Η χρήση των coloursets στα Coloured Petri Nets είναι ανάλογη της χρήσης των types στις γλώσσες προγραμματισμού. Τα coloursets καθορίζουν τις πιθανές τιμές των tokens όπως τα types καθορίζουν τις πιθανές τιμές των μεταβλητών. Η κατάσταση του μοντελοποιημένου συστήματος καθορίζεται από τις θέσεις. Μία θέση μπορεί να σημειωθεί με ένα ή περισσότερα tokens και κάθε token έχει χαρακτηριστική τιμή. Ο αριθμός των token και ο τύπος τους καθορίζουν την κατάσταση στην οποία βρίσκεται το σύστημα ανά πάσα στιγμή. Αυτό καλείται σήμανση (marking) του μοντέλου CPN.

Το πρόγραμμα CPN tools δίνει τη δυνατότητα να δηλώσουμε τα παρακάτω είδη colourset :

Απλά coloursets:

Ακέραιοι (Integers)

Λέξεις (Strings)

Μεταβλητές Boole (Boolean variables)

Απαριθμητικά σύνολα (Enumeration coloursets)

Δομημένα coloursets:

Γινόμενα (Products)

Καταχωρήσεις (Records)

Ενότητες (Unions)

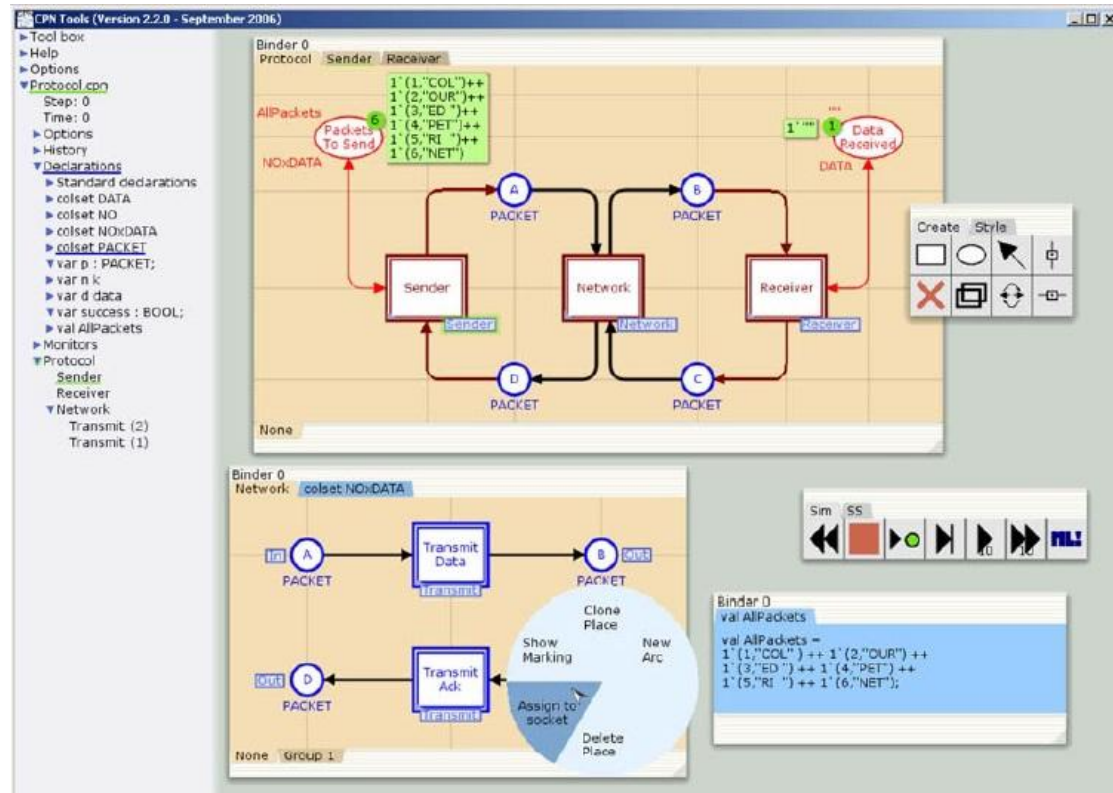
Λίστες (Lists)

Υποσύνολα (Subsets)

Για τη συγκεκριμένη εργασία χρησιμοποιήθηκαν Integer Colourset , Enumeration coloursets και Products τα οποία αποτελούνται από απαριθμημένες αλφαριθμητικές τιμές.

3.2 Γραφικό περιβάλλον χρήστη προγράμματος

Στην παρακάτω εικόνα απεικονίζεται το γραφικό περιβάλλον του CPN tools. Το τετράγωνο αριστερά είναι ο πίνακας περιεχομένων στον οποίο περιλαμβάνεται το tool box. Εκεί περιέχονται εργαλεία για τη δημιουργία των βασικών στοιχείων του μοντέλου cphn αλλά και εργαλεία για τη διάταξη και εμφάνιση των αντικειμένων. Επίσης ο πίνακας περιεχομένων διαθέτει μια γενική εικόνα του μοντέλου που τρέχει τη δεδομένη στιγμή στο πρόγραμμα. Σε αυτό φαίνεται το όνομα ,οι μεταβλητές και τα colourset (declarations), οι ενότητες και η ιεραρχική δομή του τρέχοντος μοντέλου.



Γραφικό περιβάλλον CPN tools

Το υπόλοιπο μέρος της οθόνης αποτελεί το workspace πάνω στο οποίο μπορεί να βρίσκονται είτε στοιχεία των μοντέλων cph είτε παλέτες εργαλείων. Και τα δύο καλούνται φάκελοι (binders). Στην παραπάνω εικόνα βρίσκονται 5 φάκελοι (τετραγωνικά παράθυρα). Κάθε φάκελος μπορεί να περιέχει περισσότερες από μία ενότητες στις οποίες ο χρήστης μπορεί να έχει πρόσβαση πατώντας στις ταμπέλες που βρίσκονται στο πάνω μέρος των φακέλων.

Επίσης εμφανίζονται και κυκλικά μενού όπως αυτό που φαίνεται στην παραπάνω εικόνα. Τα κυκλικά μενού δίνουν στο χρήστη τη δυνατότητα επιλογής μεταξύ των δυνατών λειτουργιών ενός αντικειμένου στην τρέχουσα φάση. Τα περισσότερα εργαλεία που είναι διαθέσιμα στα κυκλικά μενού είναι διαθέσιμα και στις παλέτες εργαλείων.

Παλέτα Net

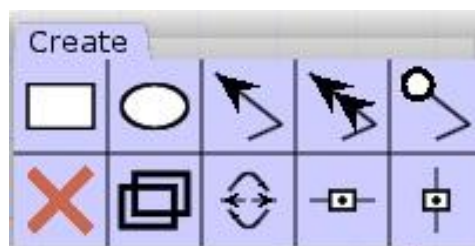
Η παλέτα Net περιέχει εργαλεία για τη δημιουργία/αποθήκευση νέων μοντέλων ή τη φόρτωση / αποθήκευση ήδη υπάρχοντων. Όταν δημιουργείται ή φορτώνεται ένα μοντέλο τα γενικά στοιχεία του εμφανίζονται αυτόματα στον πίνακα περιεχομένων



Παλέτα διαχείρισης μοντέλων στο CPN tools

Παλέτα Create

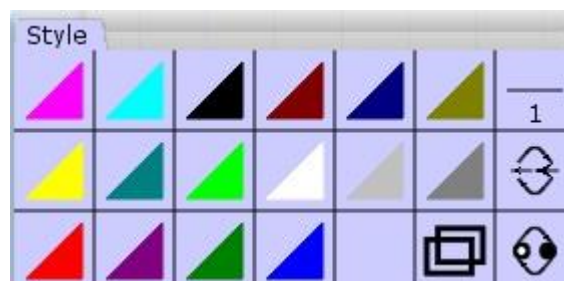
Η παλέτα Create χρησιμοποιείται για την εισαγωγή/αντιγραφή/διαμόρφωση/διαγραφή των βασικών δομικών στοιχείων του cpn (θέσεων, μεταβάσεων, τόξων). Επίσης υπάρχει η επιλογή εισαγωγής κατευθυντήριων γραμμών στο γράφημα για την οργάνωση της διάταξης των δομικών στοιχείων του μοντέλου.



Παλέτα διαχείρισης δομικών στοιχείων στο CPN tools

Παλέτα Style

Η παλέτα Style χρησιμοποιείται για τη διαμόρφωση των δομικών στοιχείων του μοντέλου cpn (πάχη γραμμών, χρώματα)



Παλέτα διαμόρφωσης δομικών στοιχείων στο CPN tools

Status bubble

Είναι ένα bubble που εμφανίζεται περιστασιακά στο κάτω μέρος του πίνακα περιεχομένων. Το status bubble ειδοποιεί τον χρήστη για την κατάσταση του μοντέλου μέσω του χρώματος που έχει κάθε φορά. Το πράσινο εκφράζει ότι μία λειτουργία ολοκληρώθηκε επιτυχώς, το κόκκινο σημαίνει σφάλμα ενώ το μοβ δείχνει ότι εκτελείται μεγάλης διάρκειας προσομοίωση. Παράλληλα με το χρώμα εμφανίζεται και ένα σύντομο μήνυμα επεξήγησης.



Μήνυμα status bubble στο CPN tools

3.3 Ορισμός θέσεων μεταβάσεων και τόξων στο CPN tools

Σε αυτήν την ενότητα περιγράφονται συνοπτικά τα χαρακτηριστικά των βασικότερων δομικών στοιχείων των Coloured Petri και ο τρόπος με τον οποίο δηλώνονται στο συγκεκριμένο πρόγραμμα. Θα γίνουν όμως πιο κατανοητά στο επόμενο κεφάλαιο που θα παρουσιαστεί αναλυτικά ο τρόπος κατάστρωσης του μοντέλου για το σύστημα το οποίο μελετάται.

➤ Εισαγωγή θέσης στο CPN tools

Μία θέση στο CPN tools έχει κυκλικό ή ελλειψοειδές σχήμα και χαρακτηρίζεται από τρία πεδία τα οποία συμπληρώνονται από το χρήστη:

- Το όνομα της θέσης. Συμβατικά, γράφουμε τα ονόματα των θέσεων μέσα στους κύκλους. Τα ονόματα αυτά δεν επηρεάζουν τη δομή και τη λειτουργία του μοντέλου αλλά καθιστούν το μοντέλο πιο ευανάγνωστο.
- Το όνομα του συνόλου (colourset) το οποίο υποδεικνύει το είδος των tokens που μπορεί να έχει η συγκεκριμένη θέση.

Το όνομα του συνόλου καθώς και το σύνολο των στοιχείων που περιέχει θα πρέπει να δηλώνονται από πριν στο πρόγραμμα με καθορισμένη σύνταξη. Η δήλωση αυτή γίνεται στην ενότητα Declarations που βρίσκεται αριστερά στο overview του μοντέλου. Εκεί υπάρχουν πάντα και κάποια colourset ως προεπιλογή από το πρόγραμμα. Πατώντας δεξί κλικ στο μενού αριστερά και στη συνέχεια επιλέγοντας New Declaration εισάγεται ένα νέο colourset.

Δηλώσεις colourset

Η εισαγωγή νέου colourset γίνεται πατώντας δεξί κλικ στο μενού αριστερά και στη συνέχεια επιλέγοντας New declaration. Για να δηλώσουμε ένα καινούριο colourset γράφουμε:

colset = with|....|....|.... όπου δίπλα στο colset πρέπει να γράψουμε το όνομα του και δεξιά τις τιμές που μπορεί να πάρει χωριζόμενες από κάθετο.

Παραδείγματα:

colset U = with p|q; (δημιουργείται colourset με όνομα U και στοιχεία p και q)

colset I = int; (δημιουργείται colourset με όνομα I και στοιχεία ακέραιους αριθμούς)

Επίσης μπορούμε να δηλώσουμε ένα σύνθετο σύνολο στοιχείων ως συνδυασμό άλλων δηλωμένων συνόλων ως εξής :

colset = product*....*....*....*.... όπου δίπλα στο colset γράφουμε το όνομα του συνόλου και δεξιά τα σύνολα από τα οποία αποτελείται. Τα αντίστοιχα tokens δεν έχουν μόνο ένα στοιχείο κάθε φορά αλλά αποτελούνται από περισσότερα στοιχεία (όσα και τα σύνολα από τα οποία αποτελείται το σύνθετο colourset)

Παράδειγμα:

colset P = product U*I; (δημιουργείται colourset με όνομα P και στοιχεία που ανήκουν στα σύνολα U και I)

Το αντίστοιχο token θα περιέχει δύο δεδομένα : το πρώτο θα ανήκει στο σύνολο U και το δεύτερο στο σύνολο I .

Για παράδειγμα τα tokens (q,2) , (p,1), (q,5) ανήκουν στο σύνολο P.

- Η αρχική σήμανση η οποία υποδεικνύει πόσα και ποια tokens έχει αρχικά η συγκεκριμένη θέση . Όταν θέλουμε η αρχική σήμανση να έχει παραπάνω από ένα token χρησιμοποιείται ο παρακάτω τρόπος σύνταξης όπου διαφορετικά tokens χωρίζονται με ++ και μπροστά από κάθε τύπο token ο αριθμός αυτών :

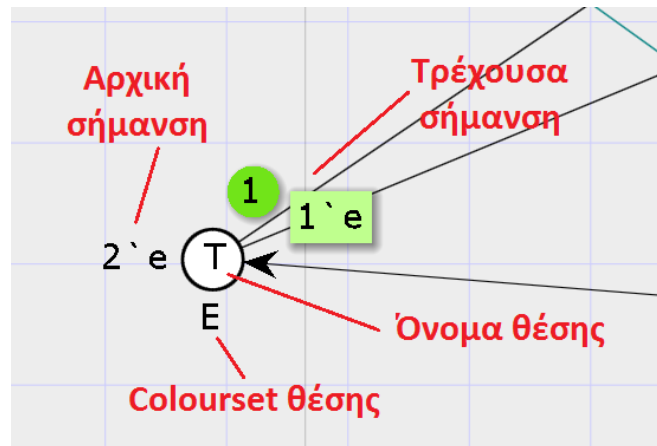
....'.....++'.....++....'.....++....'.....

Παράδειγμα: 2` (p,1) ++1`(q,3) ++4'(q,5)

Το πεδίο της αρχικής σήμανση σε μία θέση δεν είναι υποχρεωτικό να συμπληρωθεί. Δηλαδή, αν μια θέση δε θέλουμε να διαθέτει tokens κατά την εκκίνηση του μοντελου αφήνουμε το συγκεκριμένο πεδίο κενό.

Επίσης κατά τη ροή του προγράμματος εμφανίζεται δίπλα σε κάθε θέση και η τρέχουσα σήμανση και συγκεκριμένα :

- ο συνολικός αριθμός των tokens σε πράσινο κυκλάκι
- οι τιμές των tokens και ο συνολικός αριθμός κάθε τύπου σε πράσινο τετράγωνο



Πεδία που συμπληρώνονται κατά την εισαγωγή μίας θέσης και τρέχουσα σήμανση στο CPN tools

Πατώντας TAB εναλλάσσονται τα διαφορετικά πεδία που ορίζουν μία θέση (το ίδιο συμβαίνει και με τη μετάβαση).

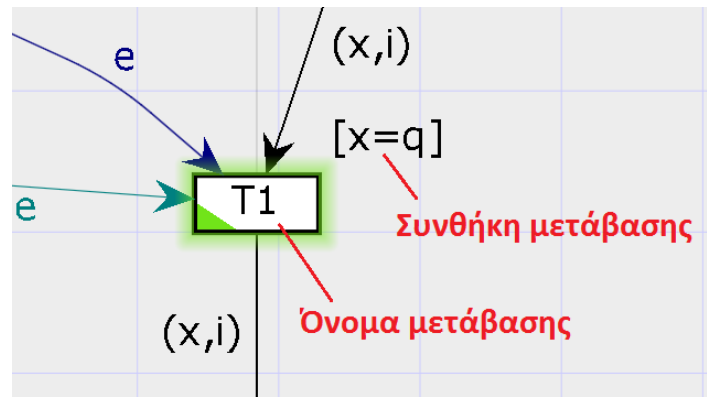
➤ Εισαγωγή μετάβασης στο CPN tools

Οι μεταβάσεις έχουν τετραγωνικό σχήμα και χαρακτηρίζονται από δύο πεδία τα οποία συμπληρώνονται από το χρήστη:

- Το όνομα της μετάβασης που συμβατικά γράφεται στο εσωτερικό των τετραγώνων . Όπως και τα ονόματα των θέσεων του μοντέλου δεν επηρεάζουν τη δομή και τη λειτουργία του.
- Η συνθήκη(guard) η οποία είναι μια έκφραση Boolean .Πριν προστεθεί μια συνθήκη σε μια μετάβαση εμφανίζεται προκαθορισμένα το σύμβολο []. Οι συνθήκες εισάγονται στις μεταβάσεις όταν θέλουμε να θέσουμε περιορισμούς στη διέλευση των tokens. Οι συνθήκες αυτές αναφέρονται στις μεταβλητές των εισερχόμενων τόξων και περιέχουν ένα ή περισσότερα από τα παρακάτω σύμβολα :

= <> <= >= < > andalso orelse

Βέβαια, αν δεν απαιτείται κάποια περιοριστική συνθήκη σε μια μετάβαση, το συγκεκριμένο πεδίο δε συμπληρώνεται.



Πεδία που συμπληρώνονται κατά την εισαγωγή μίας μετάβασης στο CPN tools

➤ Εισαγωγή τόξου στο CPN tools

Τα τόξα χαρακτηρίζονται από μία επιγραφή η οποία μπορεί να είναι

- μία μεταβλητή η οποία θα πρέπει να αντιστοιχεί στο σύνολο στοιχείων της θέσης με την οποία συνδέεται το συγκεκριμένο τόξο. Διαφορετικά εμφανίζεται σφάλμα. Όπως και τα coloursets, έτσι και οι μεταβλητές θα πρέπει να έχουν δηλωθεί προηγουμένως στην ενότητα Declarations που βρίσκεται αριστερά στο overview του μοντέλου.

Δηλώσεις μεταβλητών :

Για να δηλώσουμε μια μεταβλητή που ανήκει σε ένα colourset γράφουμε :

var : όπου αριστερά εισάγεται το όνομα της μεταβλητής και δεξιά το όνομα του colourset στο οποίο θέλουμε να αναφέρεται η μεταβλητή.

Παράδειγμα: var r : U (η μεταβλητή r αντιστοιχεί στο coloursets U)

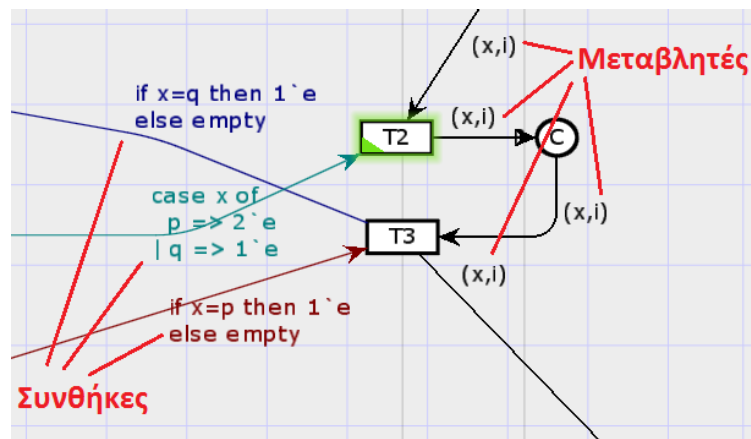
- μια συνθήκη η οποία υποδεικνύει εκτός από τον τύπο των tokens, τον αριθμό τους και υπό ποιες συνθήκες μπορούν να 'περάσουν' από τη συγκεκριμένη μετάβαση. Οι συνθήκες αυτές έχουν συγκεκριμένο τρόπο σύνταξης και διαφέρουν για την περίπτωση τόξων εισόδου και εξόδου.

Συνθήκη για τα τόξα εισόδου σε μια μεταβάση:

Ένα τόξο που ξεκινά από μια θέση και καταλήγει σε μια μεταβαση μπορεί να έχει μία συνθήκη η οποία συσχετίζει τα token της συγκεκριμένης θέσης με αυτά των υπολοίπων θέσεων που εισέρχονται στην ίδια μετάβαση. Η συνθήκη κάθε τόξου καθορίζει τελικά πόσα και ποια tokens θα αποσπαστούν από την αντίστοιχη θέση εισόδου.

Συνθήκη για τα τόξα εξόδου από μια μεταβάση :

Ένα τόξο που ξεκινά από μια μετάβαση και καταλήγει σε μια θέση μπορεί να έχει μία συνθήκη η οποία καθορίζει τον αριθμό και το είδος των tokens που θα τοποθετηθούν στη συγκεκριμένη θέση ανάλογα με τα tokens που εισέρχονται στη μετάβαση.

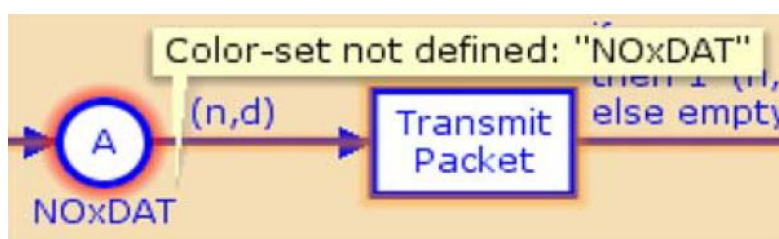


Μεταβλητές και συνθήκες των τόξων στο CPN tools

Έλεγχος σύνταξης και εντοπισμός σφαλμάτων

Το CPN tools εκτελεί έλεγχο στη σύνταξη των coloursets, των μεταβλητών, των συναρτήσεων και των συνθηκών (inscriptions) σε ένα μοντέλο και παράλληλα παράγει και τον κώδικα προσομοίωσης. Ο κώδικας αυτός περιέχει τις συναρτήσεις που δίνουν το σύνολο των ενεργοποιημένων μεταβάσεων για μια δεδομένη σήμανση και τον υπολογισμό της σήμανσης στην οποία φτάνει το σύστημα μετά την εκκίνηση μιας ενεργοποιημένης μετάβασης.

Σε περίπτωση σφάλματος εμφανίζεται μήνυμα δίπλα στο στοιχείο που προκαλεί το σφάλμα όπως φαίνεται στο παράδειγμα της παρακάτω εικόνας



Μήνυμα συντακτικού σφάλματος στο CPN tools

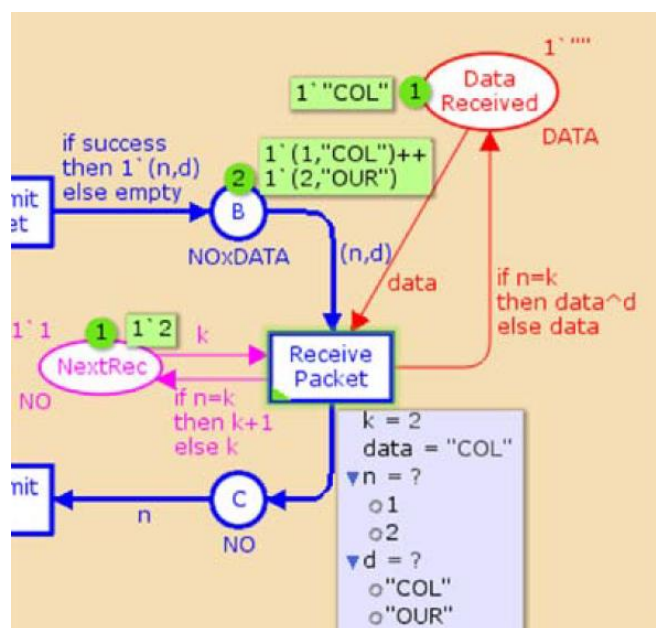
Κάποια στοιχεία δεν ελέγχονται μέχρι να οριστούν πλήρως. Για παράδειγμα μια θέση δεν ελέγχεται από το πρόγραμμα μέχρι να της αποδοθεί ένα colourset ενώ μια μετάβαση δεν ελέγχεται μέχρι να αποδοθούν μεταβλητές ή συνθήκες σε όλα τα τόξα εισόδου/εξόδου σε/από αυτήν.

3.4 Προσομοίωση στο CPN tools

Στο CPN tools η προσομοίωση μπορεί να γίνει είτε διαδραστικά με το χρήστη είτε αυτόματα. Η διαδραστική προσομοίωση γίνεται βήμα-βήμα και επιτρέπει τη διερεύνηση διαφορετικών σεναρίων λεπτομερώς. Κατά τη διάρκεια μιας τέτοιας προσομοίωσης ο χρήστης καθορίζει το επόμενο βήμα επιλέγοντας μεταξύ των ταυτόχρονα ενεργοποιούμενων μεταβάσεων και αντιστοιχίσεων μεταβλητών. Τα αποτελέσματα που προκύπτουν σε κάθε βήμα από την επιλογή του χρήστη εμφανίζονται στη γραφική αναπαράσταση του μοντέλου. Η αυτόματη προσομοίωση είναι παρόμοια με την εκτέλεση ενός προγράμματος. Ο σκοπός είναι να προσομοιωθεί το σύστημα όσο πιο γρήγορα γίνεται και τυπικά χρησιμοποιείται για έλεγχο ή ανάλυση.

Παράλληλα κατά την προσομοίωση ενός μοντέλου στο CPN tools μπορεί να συμπεριληφθεί και η έννοια του χρόνου που απαιτείται για να εκτελεστούν οι διάφορες λειτουργίες τους συστήματος.

Ο προσομοιωτής του CPN tools εκτελεί βήματα που αποτελούνται από μία μόνο εκκίνηση μετάβασης κάθε φορά. Κατά τη διάρκεια της προσομοίωσης η ροή των tokens και η ενεργοποίηση των μεταβάσεων δίνεται στο χρήστη με ιδιαίτερα παραστατικό τρόπο όπως φαίνεται από την παρακάτω εικόνα που δίνει ένα στιγμιότυπο της προσομοίωσης. Δίπλα σε θέση εμφανίζεται ένα μικρό πράσινο κυκλάκι που δείχνει τον τρέχοντα αριθμό tokens αλλά και ένα πράσινο τετράγωνο που δείχνει ποια ακριβώς tokens περιέχει η θέση ανά πάσα στιγμή. Όταν μια μετάβαση είναι φωτισμένη με πράσινο περίγραμμα (όπως η Receive Packet της παρακάτω εικόνας) σημαίνει ότι μπορεί να ενεργοποιηθεί. Αν σε κάποια φάση της προσομοίωσης δεν υπάρχει καμία φωτισμένη μετάβαση σημαίνει ότι το σύστημα βρίσκεται σε μια νεκρή σήμανση.



Στιγμιότυπο από την προσομοίωση στο CPN tools

Τα εργαλεία που χρησιμοποιούνται για την προσομοίωση βρίσκονται στην παλέτα Sim που φαίνεται παρακάτω. Οι λειτουργίες των εργαλείων αυτών είναι κατά σειρά (από αριστερά προς τα δεξιά)



Παλέτα προσομοίωσης στο CPN tools

- Επιστροφή στην αρχική σήμανση
- Διακοπή αυτόματης προσομοίωσης που βρίσκεται σε εξέλιξη
- Εκτέλεση μίας μόνο μετάβασης η οποία θα επιλεγεί από το χρήστη μεταξύ όλων των ενεργοποιημένων μεταβάσεων
- Εκτέλεση μίας μόνο μετάβασης η οποία θα επιλεγεί τυχαία από το σύστημα μεταξύ όλων των ενεργοποιημένων μεταβάσεων
- Εκτέλεση αυτόματης προσομοίωσης όπου εμφανίζεται η τρέχουσα σήμανση σε κάθε βήμα
- Γρήγορη εκτέλεση αυτόματης προσομοίωσης χωρίς να εμφανίζεται η τρέχουσα σήμανση σε κάθε βήμα

Στην περίπτωση που η προσομοίωση του μοντέλου γίνεται βήμα βήμα διαδραστικά με το χρήστη, ο προσομοιωτής είναι αυτός που υπολογίζει για κάθε σήμανση το σύνολο των ενεργοποιούμενων μεταβάσεων. Έπειτα είναι στην ευχέρεια του χρήστη να επιλέξει μεταξύ των μεταβάσεων αυτών αλλά και των δυνατών αντιστοιχιών (bindings).

Η επιλογή μεταξύ των δυνατών αντιστοιχιών γίνεται από το γκρι τετράγωνο που εμφανίζεται δίπλα στη μετάβαση όπως αυτό της προηγούμενης εικόνας. Σε αυτό το πλαίσιο εμφανίζεται η λίστα των μεταβλητών που φτάνουν στη μετάβαση μέσω των τόξων εισόδου και οι τιμές που μπορούν να πάρουν σύμφωνα με τη δεδομένη σήμανση (δηλαδή σύμφωνα με τα tokens που βρίσκονται στις θέσεις εισόδου τη δεδομένη στιγμή).

Παράδειγμα:

Στην περίπτωση του μοντέλου της προηγούμενης εικόνας η τιμή 2 έχει ήδη αντιστοιχιστεί στη μεταβλητή k και η τιμή "COL" στη μεταβλητή $data$ αντίστοιχα. Αυτό γίνεται αυτόματα καθώς υπάρχει μόνο μία πιθανή επιλογή για αυτές τις μεταβλητές. Παραμένει όμως στο χρήστη η επιλογή να αντιστοιχίσει τιμές στις μεταβλητές n και d (αρκεί η επιλογή μόνο για μία από τις δύο).

Αφού επιλεγεί η αντιστοιχία και εκτελεστεί η επιλεγμένη μετάβαση η σήμανση του μοντέλου ανανεώνεται και εμφανίζεται στο γραφικό μοντέλο δίνοντας νέα ή νέες ενεργοποιημένες μεταβάσεις.

Η προσομοίωση σταματά μόνο όταν το σύστημα φτάσει σε μια νεκρή σήμανση (εκτός αν έχει γίνει προεπιλογή να σταματήσει μετά από συγκεκριμένο αριθμό βημάτων). Η νεκρή

σήμανση σημαίνει ότι δεν υπάρχει πλέον καμία ενεργοποιημένη μετάβαση. Λόγω του μη ντετερμινιστικού χαρακτήρα των μοντέλων cph, οι νεκρές σημάνσεις δεν εμφανίζονται πάντα κατά την προσομοίωση. Γι' αυτό και η προσομοίωση δεν μπορεί να χρησιμοποιηθεί ως κριτήριο για τις ιδιότητες του μοντελοποιημένου συστήματος. Αυτό είναι κάτι που καθορίζεται από την ανάλυση state space που θα αναλυθεί στην επόμενη ενότητα.

Ωστόσο, αποτελεί ένα πολύ χρήσιμο και παραστατικό εργαλείο για τον έλεγχο και εντοπισμό σφαλμάτων. Επιπλέον, σε περίπτωση που ο χρήστης επιθυμεί να επιθεωρήσει τις σημάνσεις και τις αντιστοιχίες που συνέβησαν σε μια προσομοίωση μπορεί να πάρει την αναφορά προσομοίωσης (simulation report). Σε αυτήν την αναφορά εμφανίζεται η λίστα όλων των βημάτων που ακολουθήθηκαν σε μία προσομοίωση.

3.5 Ανάλυση χώρου καταστάσεων (state space) στο CPN tools

Η σύνταξη και η σημασιολογία της γλώσσας μοντελοποίησης CPN ορίζονται με μαθηματικό τρόπο. Αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί για να αποδείξει την ύπαρξη συγκεκριμένων επιθυμητών ιδιοτήτων του συστήματος ή να εγγυηθεί την απουσία κάποιων άλλων ανεπιθύμητων ιδιοτήτων.

Η εξακρίβωση των ιδιοτήτων αυτών υποστηρίζεται από ένα σύνολο state space μεθόδων. Η βασική ιδέα πίσω από τις μεθόδους state space είναι ο υπολογισμός όλων των δυνατών καταστάσεων του συστήματος αλλά και τον τρόπο και σειρά διαδοχής μεταξύ των διαφορετικών καταστάσεων. Όλα αυτά μπορούν να αναπαρασταθούν σε ένα γράφημα όπου οι κόμβοι αντιπροσωπεύουν τις καταστάσεις και οι σύνδεσμοι αντιπροσωπεύουν τα γεγονότα τα οποία μεσολαβούν. Ο χώρος καταστάσεων (state space) υπολογίζεται εντελώς αυτόματα. Από το state space είναι δυνατόν να απαντηθούν βασικά και πολύ ουσιαστικά ζητήματα που αφορούν τη συμπεριφορά ενός συστήματος όπως :

- η απουσία κολλημάτων (deadlocks)
- η δυνατότητα να φτάσει το σύστημα σε μία δεδομένη κατάσταση

Όπως προαναφέρθηκε, η προσομοίωση είναι κατάλληλη για τον έλεγχο της λειτουργίας του μοντέλου και τον εντοπισμό σφαλμάτων. Όμως ακόμα και αν εκτελέσουμε πολλές προσομοιώσεις που δείχνουν ότι το μοντέλο λειτουργεί σωστά και φτάνει στην επιθυμητή κατάσταση, δεν μπορούμε να είμαστε απόλυτα σίγουροι ότι οι προσομοιώσεις έχουν καλύψει όλες τις περιπτώσεις.

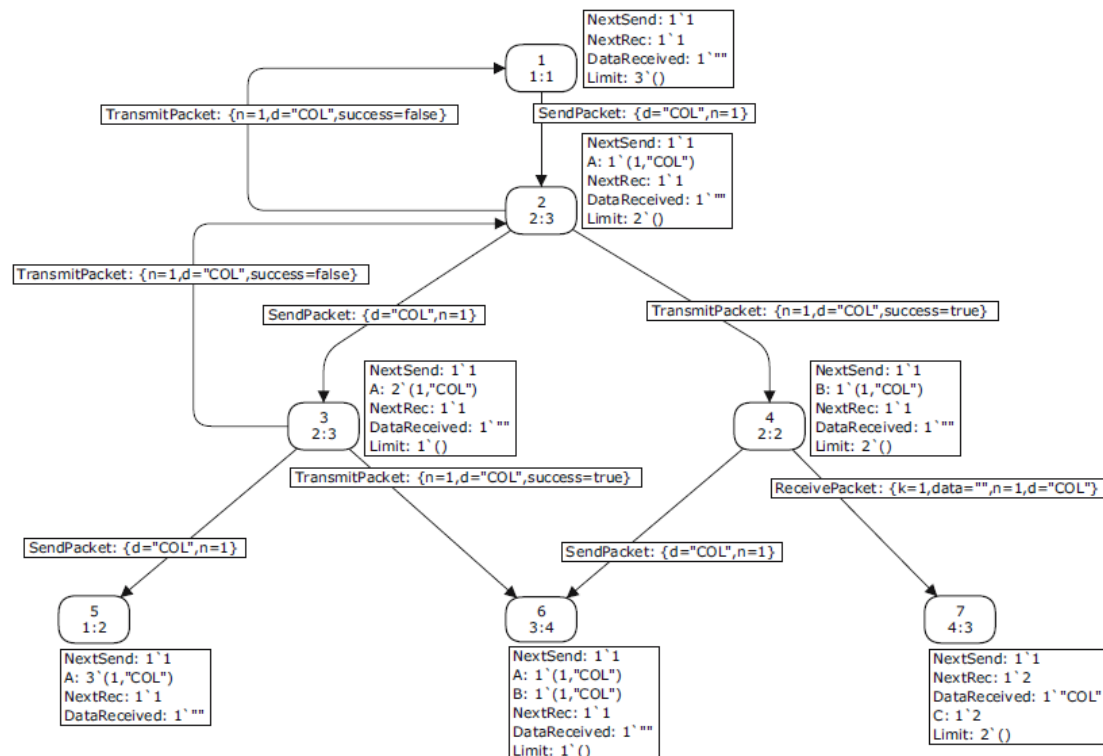
Πλήρης χώρος καταστάσεων (full state space)

Ενας πλήρης χώρος καταστάσεων δίνει όλες τις πιθανές καταστάσεις ενός μοντέλου cph. Ένας τέτοιος χώρος απεικονίζεται ως γράφημα με δενδροειδή μορφή στο οποίο υπάρχει

έναν κόμβος για κάθε δυνατή σήμανση και ένας σύνδεσμος που ενώνει δύο κόμβους για κάθε δυνατή αντιστοίχιση (binding).

Ένας σύνδεσμος σημειωμένος με (t,b) από έναν κόμβο που αναπαριστά τη σήμανση M_1 σε έναν κόμβο που αναπαριστά τη σήμανση M_2 υπάρχει αν και μόνο αν η αντιστοίχιση (t,b) ενεργοποιείται από τη M_1 και η εκτέλεση του (t,b) από τη M_1 οδηγεί στη σήμανση M_2 .

Στην παρακάτω εικόνα φαίνεται το αρχικό τμήμα από το χώρο καταστάσεων του μοντέλου crn . Το τμήμα αυτό δημιουργήθηκε χρησιμοποιώντας την υποστήριξη για την οπτικοποίηση του state space στο cpn tools.

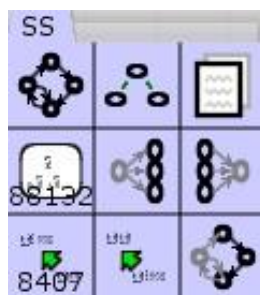


Τμήμα χώρου καταστάσεων μοντέλου στο CPN tools ,Πηγή: [9]

Κάθε κόμβος χαρακτηρίζεται από 3 ακεραίους. Ο πάνω είναι ο αριθμός του κόμβου και οι δύο ακέραιοι που χωρίζονται από : είναι ο αριθμός των προηγούμενων (pedecessors) και επόμενων (successors) κόμβων. Σε κάθε κόμβο (κατάσταση) του συστήματος αντιστοιχεί ένα τετράγωνο (node descriptor) που δίνει πληροφορίες για τη σήμανση των θέσεων. Συγκεκριμένα, δίνει τη λίστα των θέσεων με μη μηδενική σήμανση.

Γενικά ο χώρος κατάστασης είναι πολύ μεγάλος για να απεικονιστεί σε ένα σχήμα. Ενδεικτικά αναφέρεται ότι ο χώρος κατάστασης του μοντέλου της προηγούμενης εικόνας αποτελείται από 13.215 κόμβους και 52.784 συνδέσμους. Ωστόσο, ένα τμήμα του χώρου καταστάσεων μπορεί να χρησιμοποιηθεί για την ανάλυση ενός περιορισμένου αριθμού προσβάσιμων σημάνσεων από μια συγκεκριμένη σήμανση που έχει επιλέξει ο χρήστης.

Ο χώρος καταστάσεων υπολογίζεται εντελώς αυτόματα χρησιμοποιώντας τα εργαλεία της παλέτας state space που φαίνεται παρακάτω.



Παλέτα ανάλυσης χώρου καταστάσεων (state space) στο CPN tools

Συγκεκριμένα τα εργαλεία αυτά είναι (από αριστερά προς δεξιά και από την πάνω προς την κάτω σειρά) :

- **Υπολογισμός του χώρου καταστάσεων (state space)**

Το CPN tools υπολογίζει όλο το χώρο καταστάσεων και αποθηκεύει το γράφημα που αντιστοιχεί στο χώρο καταστάσεων στην εσωτερική μνήμη.

- **Υπολογισμός γραφήματος SCC**

Η δημιουργία του full state space ακολουθείται από τη δημιουργία του Strongly Connected Component Graph (SCC-graph) το οποίο προκύπτει από τη δομή του γραφήματος του χώρου καταστάσεων. Οι κόμβοι του SCC-graph είναι υπο-γραφήματα του χώρου καταστάσεων τα οποία συνθέτουν οι κόμβοι που ικανοποιούν την εξής συνθήκη :

Δύο κόμβοι του χώρου καταστάσεων ανήκουν σε ένα SCC graph αν είναι αμοιβαία προσβάσιμοι δηλαδή υπάρχει αλληλουχία εκκινήσεων μεταβάσεων που να οδηγεί από τον πρώτο κόμβο στο δεύτερο και αντίστροφα.

Η διαφορά μεταξύ του αριθμού των κόμβων στο χώρο καταστάσεων και στο SCC γράφημα συνεπάγεται την ύπαρξη κυκλικής συμπεριφοράς του μοντέλου. Αυτό σημαίνει ότι μπορεί να έχουμε άπειρο αριθμό εκτελέσεων αλληλουχιών και το μοντέλο να μη φτάνει ποτέ στην επιθυμητή κατάσταση

- **Αποθήκευση αναφοράς state space**

Το πρώτο βήμα κατά την ανάλυση του χώρου καταστάσεων είναι να ζητήσουμε την αποθήκευση της αναφοράς state space. Η αναφορά αυτή αποθηκεύεται ως αρχείο .txt με την εντολή Save report της παλέτας του state space και παρέχει τις βασικές πληροφορίες για το μέγεθος και τις ιδιότητες συμπεριφοράς του μοντέλου cph οι οποίες είναι κατά σειρά :

- **Μέγεθος και χρόνος δημιουργίας state space**
- **Μέγεθος και χρόνος δημιουργίας SCC graph**

State Space Statistics

State Space	Scc Graph
Nodes: 13215	Nodes: 5013
Arcs: 52784	Arcs: 37312
Secs: 53	Secs: 2
Status: Full	

Βασικά στοιχεία του χώρου καταστάσεων του μοντέλου από το αρχείο .txt που αποθηκεύει η εντολή Save report στο CPN tools, Πηγή: [9]

- **Boundedness properties**
- Best upper integer bound: Εμφανίζεται για κάθε θέση ο μέγιστος αριθμός tokens μπορεί να κρατήσει για όλες τις δυνατές σημάνσεις
- Best upper integer bound : Εμφανίζεται για κάθε θέση ο ελάχιστος αριθμός tokens μπορεί να κρατήσει για όλες τις δυνατές σημάνσεις

Best Integers Bounds	Upper	Lower
PacketsToSend	6	6
DataReceived	1	1
NextSend, NextRec	1	1
A, B, C, D	3	0
Limit	3	0

Μέγιστος και ελάχιστος αριθμός tokens κάθε θέσης στο χώρο καταστάσεων του μοντέλου από το αρχείο .txt που αποθηκεύει η εντολή Save report στο CPN tools, Πηγή: [9]

- Best upper multi-set bound : Εμφανίζεται για κάθε θέση και για κάθε τύπο που ανήκει στο αντίστοιχο colourset ο ελάχιστος αριθμός tokens για όλες τις δυνατές σημάνσεις
- Best lower multi-set bound : Εμφανίζεται για κάθε θέση και για κάθε τύπο που ανήκει στο αντίστοιχο colourset ο μέγιστος αριθμός tokens για όλες τις δυνατές σημάνσεις.

Best Upper Multi-set Bounds

PacketsToSend	1'(1,"COL")+1'(2,"OUR")+ 1'(3,"ED ") +1'(4,"PET")+ 1'(5,"RI ") +1'(6,"NET")
DataReceived	1'""+1' "COL"+1' "COLOUR"+ 1' "COLOURED "++ 1' "COLOURED PET"+ 1' "COLOURED PETRI "++ 1' "COLOURED PETRI NET"
NextSend, Nextrec	1'1++1'2++1'3++1'4++1'5++ 1'6++1'7
A, B	3'(1,"COL")+3'(2,"OUR")+ 3'(3,"ED ") +3'(4,"PET")+ 3'(5,"RI ") +3'(6,"NET")
C, D	3'2++3'3++3'4++3'5++3'6++3'7
Limit	3'()

Best Lower Multi-set Bounds

PacketsToSend	1'(1,"COL")+1'(2,"OUR")+ 1'(3,"ED ") +1'(4,"PET")+ 1'(5,"RI ") +1'(6,"NET")
DataReceived	empty
NextSend, NextRec	empty
A, B, C, D	empty
Limit	empty

Μέγιστο και ελάχιστο πολυσύνολο *tokens* κάθε θέσης στο χώρο καταστάσεων του μοντέλου από το αρχείο *.txt* που αποθηκεύει η εντολή *Save report* στο *CPN tools*, Πηγή: [9]

Παράδειγμα : ο μέγιστος αριθμός *tokens* που μπορεί να πάρει είναι 3 (πίνακας Best Integer Bounds) και ο μέγιστος αριθμός *tokens* τύπου 7 είναι 3 (πίνακας Best Upper Multiset Bounds) (3'2++3'3++3'4++3'5++3'6++3'7)

▪ **Home properties**

Μία σήμανση *home* είναι μια σήμανση προσβάσιμη από κάθε δυνατή σήμανση. Αυτό σημαίνει ότι αποκλείεται να έχουμε μια αλληλουχία εκκινήσεων μεταβάσεων που να μην μπορεί να επεκταθεί και να φτάσει τη σήμανση *home*.

Home Properties

Home Markings: [4868]

Ιδιότητες home από την ανάλυση του χώρου καταστάσεων του μοντέλου όπως φαίνεται στο αρχείο .txt που αποθηκεύει η εντολή Save report στο CPN tools, Πηγή: [9]

Στον παραπάνω πίνακα φαίνεται ότι στο σύστημα υπάρχει μία μόνο σήμανση home η οποία αντιστοιχεί στον κόμβο 4868. Ζητώντας από το CPN tools να εμφανίσει τη σήμανση του κόμβου 4868 φαίνεται ότι η σήμανση αυτή αντιστοιχεί στην κατάσταση όπου το μοντελοποιημένο σύστημα έχει ολοκληρώσει τις διαδικασίες που είναι προγραμματισμένο να εκτελέσει. Συνεπώς, το σύστημα φτάνει πάντοτε στο επιθυμητό αποτέλεσμα από οποιαδήποτε κατάσταση μπορεί να βρεθεί.

▪ **Liveness properties**

Μια νεκρή σήμανση (dead marking) είναι μια σήμανση για την οποία δεν υπάρχει καμία ενεργοποιημένη μετάβαση. Όπως φαίνεται και παρακάτω, το σύστημα έχει μόνο μία νεκρή σήμανση που αντιστοιχεί στον κόμβο 4868 δηλαδή στο μοντελοποιημένο σύστημα υπάρχει μόνο μία κατάσταση στην οποία σταματά. Όπως προαναφέρθηκε, ο κόμβος 4868 αντιστοιχεί στην κατάσταση όπου το μοντελοποιημένο σύστημα έχει ολοκληρώσει τις διαδικασίες που είναι προγραμματισμένο να εκτελέσει. Συνεπώς η μόνη κατάσταση στην οποία μπορεί να σύστημα να σταματήσει είναι η επιθυμητή κατάσταση.

Μια μετάβαση είναι ζωντανή (live transition) αν από κάθε προσβάσιμη σήμανση υπάρχει πάντα μια αλληλουχία εκκινήσεων μεταβάσεων που περιέχει τη συγκεκριμένη μετάβαση. Όταν υπάρχει έστω και μία νεκρή σήμανση στο σύστημα, δεν μπορούν να υπάρχουν ζωντανές μεταβάσεις καθώς καμία μετάβαση δεν μπορεί να ενεργοποιηθεί από τη νεκρή μετάβαση.

Liveness Properties

Dead Markings: [4868]
Dead Transitions: None
Live Transitions: None

Ιδιότητες 'ζωτικότητας' από την ανάλυση του χώρου καταστάσεων του μοντέλου όπως φαίνονται στο αρχείο .txt που αποθηκεύει η εντολή Save report στο CPN tools, Πηγή: [9]

Μια νεκρή μετάβαση (dead transition) είναι μια μετάβαση που δεν μπορεί να ενεργοποιηθεί από καμία δυνατή σήμανση. Η απουσία νεκρών μεταβάσεων σημαίνει ότι όλες οι μεταβάσεις του ενεργοποιημένου συστήματος μπορούν να ενεργοποιηθούν τουλάχιστον μία φορά.

- **Εμφάνιση συγκεκριμένου κόμβου με αριθμό που δίνεται από το χρήστη**

Ο χρήστης πληκτρολογεί πάνω στο συγκεκριμένο εργαλείο τον αριθμό του κόμβου που επιθυμεί. Στη συνέχεια πατώντας δεξί κλικ πάνω στο εργαλείο παίρνει τον κόμβο και τον τοποθετεί πάνω σε μια σελίδα του workspace ξαναπατώντας δεξί κλικ.

- **Εμφάνιση των κόμβων που διαδέχονται ένα συγκεκριμένο κόμβο σε ένα state space**

Ο χρήστης πατάει δεξί κλικ πάνω στο συγκεκριμένο εργαλείο και στη συνέχεια επιλέγει έναν κόμβο που βρίσκεται πάνω στο workspace. Αμέσως εμφανίζονται όλοι οι επόμενοι κόμβοι που είναι δυνατόν να ενεργοποιηθούν.

- **Εμφάνιση των κόμβων που προηγούνται ενός συγκεκριμένου κόμβου σε ένα state space**

Ο χρήστης πατάει δεξί κλικ πάνω στο συγκεκριμένο εργαλείο και στη συνέχεια επιλέγει έναν κόμβο που βρίσκεται πάνω στο workspace. Αμέσως εμφανίζονται όλοι οι προηγούμενοι κόμβοι από τους οποίους είναι δυνατόν να ενεργοποιηθεί ο συγκεκριμένος.

- **State space to sim**

Ο χρήστης πληκτρολογεί πάνω στο συγκεκριμένο εργαλείο τον αριθμό του κόμβου που επιθυμεί. Στη συνέχεια πατώντας δεξί κλικ πάνω στο εργαλείο παίρνει τον κόμβο και τον τοποθετεί πάνω στη σελίδα του workspace όπου βρίσκεται στο μοντέλο ξαναπατώντας δεξί κλικ. Αμέσως το μοντέλο αποκτά τη σήμανση που αντιστοιχεί στον επιλεγμένο κόμβο.

- **Sim to State space**

Σε οποιαδήποτε φάση της προσομοίωσης ο χρήστης μπορεί να δει την αντίστοιχη σήμανση. Αυτό γίνεται πατώντας δεξί κλικ πάνω στο συγκεκριμένο εργαλείο, στη συνέχεια πατώντας πάνω στο μοντέλο και τέλος πάνω στο workspace. Αμέσως εμφανίζεται ο κόμβος που αντιστοιχεί στη δεδομένη φάση της προσομοίωσης.

- **Εμφάνιση τμήματος του SSC graph σύμφωνα με έκφραση που δηλώνεται με βοηθητικό κείμενο πάνω στο workspace**

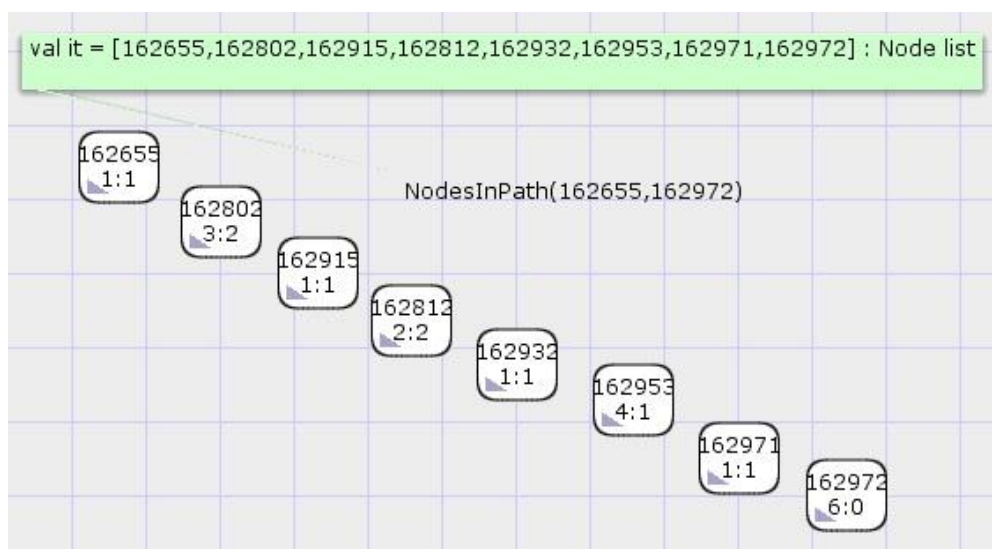
Ο χρήστης μπορεί να δει τους κόμβους ή τα τόξα που μεσολαβούν μεταξύ δύο συγκεκριμένων σημάνσεων. Πριν χρησιμοποιηθεί το συγκεκριμένο εργαλείο, ο χρήστης θα πρέπει να εισάγει στο workspace μια βοηθητική έκφραση. Πατώντας δεξί κλικ πάνω

στο workspace και έπειτα επιλέγοντας Create Aux Text από το κυκλικό μενού, εμφανίζεται πάνω στο workspace ένα πεδίο στο οποίο εισάγεται η βοηθητική έκφραση.

Για να εμφανιστούν ενδιάμεσοι κόμβοι κλικάρουμε NodesInPath στο βοηθητικό κείμενο και δύο προεπιλεγμένους κόμβους, όπως φαίνεται στο παρακάτω παράδειγμα.

Με δεξίκλικ και Evaluate ML πάνω στο βοηθητικό κείμενο δίνεται η αλληλουχία των κόμβων μεταξύ των 162655 έως 162972 (Node list).

Κλικάροντας με το εργαλείο **Display a partial SS graph according to the expression specified in the target Aux** πάνω στο βοηθητικό κείμενο, εμφανίζονται και οι αντίστοιχοι κόμβοι.



Εμφάνιση κόμβων μεταξύ δύο προεπιλεγμένων μετά την ανάλυση χώρου καταστάσεων στο CPN tools, Πηγή: [9]

Οι λειτουργίες όλων εργαλείων του state space που περιγράφηκαν παραπάνω γίνονται πιο σαφείς κατά την ανάλυση του μοντέλου του υπό μελέτη συστήματος στο επόμενο κεφάλαιο.

Κεφάλαιο 4: Μοντελοποίηση Ευέλικτου Συστήματος Κατεργασιών με το πρόγραμμα CPN tools

4.1 Δομή του μοντέλου –θέσεις/μεταβάσεις

Για τη μοντελοποίηση του ευέλικτου συστήματος κατεργασιών που μελετάται στην παρούσα εργασία επιλέχθηκε το πρόγραμμα CPN tools το οποίο παρουσιάστηκε στο Κεφάλαιο 3.

Από την πρώτη φάση κατάστρωσης του μοντέλου στόχος ήταν να χρησιμοποιηθούν τόσες θέσεις όσες και οι διαφορετικές φάσεις στις οποίες μπορεί να βρεθεί το σύστημα. Αποφεύγοντας τη χρήση παραπάνω θέσεων από τις απαιτούμενες το μοντέλο που προκύπτει είναι πιο απλό και ευανάγνωστο διευκολύνοντας τη μελέτη των ιδιοτήτων του συστήματος. Παράλληλα, με αυτόν τον τρόπο διευκολύνεται και η μετατροπή του μοντέλου σε γλώσσα plc, όπως φαίνεται στο επόμενο κεφάλαιο, έτσι ώστε να μη δημιουργηθούν περιττές βοηθητικές μνήμες.

Έτσι, για τις 6 διαφορετικές φάσεις λειτουργίας του συστήματος (ανεξάρτητα από τον τύπο του τεμαχίου και από τη μηχανή επεξεργασίας) δημιουργήθηκαν 6 διαφορετικές θέσεις. Οι 5 από αυτές αντιστοιχούν σε διαφορετικές κινήσεις του ρομπότ ενώ η 6^η στη φάση κατεργασίας σε τόρνο/φρέζα.

Επίσης χρησιμοποιήθηκαν 4 θέσεις για τη διαθεσιμότητα ρομπότ, τόρνου/ φρέζας και buffer.

Η μη διαθεσιμότητα ως συνθήκη :

Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι στο πρόγραμμα CPN tools δεν υπάρχει η δυνατότητα χρήσης αποτρεπτικών κλάδων (inhibitor arcs) σε θέσεις που αντιστοιχούν σε colourset με περισσότερα από ένα στοιχεία. Δηλαδή για αυτές τις θέσεις δεν μπορεί να εκφραστεί η μη διαθεσιμότητα token ως συνθήκη.

Για να μπορεί να εκφραστεί η μη διαθεσιμότητα ενός token χρησιμοποιείται “Anti-place” . Αυτή είναι μια θέση που δημιουργείται όπως ακριβώς και οι υπόλοιπες στο πρόγραμμα CPN αλλά συνδέεται με τέτοιο τρόπο στο δίκτυο έτσι ώστε να παρέχει κάθε φορά την αντίθετη πληροφορία από μια συγκεκριμένη θέση [http://cpntools.org/documentation/tasks/editing/constructs/anti-placeslimit_places].

Για το δεδομένο σύστημα το στοιχείο “Anti-place” χρησιμοποιείται για να εκφράσει τη μη διαθεσιμότητα των δύο μηχανών. Η θέση αυτή ονομάζεται **Anti Machine availability** στο μοντέλο και έχει αρχική σήμανση 0 και μέγιστο όριο tokens : 1` AntiM+1` AntiT

Για να δημιουργηθεί η **Anti Machine availability** θα πρέπει σε κάθε μετάβαση που η **Machine availability** έχει τόξο εισόδου η **Anti Machine availability** να έχει τόξο εξόδου και αντίστροφα.

Προφανώς ,το περιεχόμενο της θέσης **Anti Machine availability** θαπρέπει να βρίσκεται πάντοτε σε συμφωνία με το περιεχόμενο της **Machine availability** όπως φαίνεται στον παρακάτω πίνακα . Δηλαδή οποιαδήποτε στιγμή το σύστημα μπορεί να βρίσκεται σε μία από τις παρακάτω περιπτώσεις :

Θέση : Machine availability	Θέση : Anti Machine availability	Κατάσταση συστήματος
1` M	1` AntiT	Φρέζα διαθέσιμη –τόρνος μη διαθέσιμος
1` T	1` AntiM	Τόρνος διαθέσιμος- φρέζα μη διαθέσιμη
1` M+1`T	0	Φρέζα, τόρνος διαθέσιμα
0	1` AntiM+1` AntiT	Καμία μηχανή διαθέσιμη

Ακόμα χρησιμοποιήθηκαν 3 θέσεις για την αρχική, τελική και ενδιάμεση αποθήκη αντίστοιχα. Στον παρακάτω πίνακα φαίνονται όλες οι θέσεις που χρησιμοποιήθηκαν για τη μοντελοποίηση του ευέλικτου συστήματος κατεργασιών.

Θέσεις του μοντέλου ΕΣΚ στο CPN tools	
Unload Stock_I - Load Mi Processing	το ρομπότ ξεφορτώνει το τεμάχιο από την αποθήκη εισόδου και το φορτώνει στη μηχανή κατεργασίας
Unload from Mi- Load Stock_F	το ρομπότ ξεφορτώνει το τεμάχιο από τη μηχανή κατεργασίας και το φορτώνει στην τελική αποθήκη
Unload from Mi- Load Mi+1	το ρομπότ ξεφορτώνει το τεμάχιο από την προηγούμενη και το φορτώνει στην επόμενη μηχανή κατεργασίας
Unload from Mi- Load Buffer	το ρομπότ ξεφορτώνει το τεμάχιο από την προηγούμενη μηχανή κατεργασίας και το φορτώνει στην ενδιάμεση αποθήκη
Unload Buffer - Load Mi+1	το ρομπότ ξεφορτώνει το τεμάχιο από την ενδιάμεση αποθήκη και το φορτώνει στην επόμενη μηχανή κατεργασίας
Mi processing	Η μηχανή κατεργάζεται το τεμάχιο
Stock_I	Αρχική αποθήκη

Stock_F	Τελική αποθήκη
Buffer	Ενδιάμεση αποθήκη
Robot availability	Διαθεσιμότητα/Μη διαθεσιμότητα ρομπότ
Machine availability	Διαθεσιμότητα μηχανής κατεργασίας
Anti Machine availability	Μη διαθεσιμότητα μηχανής κατεργασίας

Για τη μοντελοποίηση του υπό μελέτη συστήματος χρησιμοποιήθηκαν 10 συνολικά μεταβάσεις οι οποίες συγκεντρώνονται στον παρακατω πίνακα :

Μεταβάσεις του μοντέλου ΕΣΚ στο CPN tools	
Start unloading new part from Stock_I	Ξεκινάει η εκφόρτωση νέου τεμαχίου από την αρχική αποθήκη
Finish loading new part to Mi - Start processing	Ολοκληρώνεται η εκφόρτωση νέου τεμαχίου – Ξεκινά η πρώτη κατεργασία
Finish processing - Start unloading finished from Mi	Ολοκληρώνεται η κατεργασία - Ξεκινά η εκφόρτωση του έτοιμου τεμαχίου από τη μηχανή κατεργασίας
Finish loading finished part to Stock_F	Ολοκληρώνεται η φόρτωση του έτοιμου τεμαχίου στην τελική αποθήκη
Finish processing - Start unloading part from Mi (to Mi+1)	Ολοκληρώνεται η κατεργασία – Ξεκινά η εκφόρτωση του τεμαχίου από την προηγούμενη στην επόμενη μηχανή ατεργασίας
Finish loading part to Mi+1 - Start processing	Ολοκληρώνεται η φόρτωση του τεμαχίου στην επόμενη μηχανή κατεργασίας – Ξεκινά η κατεργασία
Finish processing - Start unloading part from Mi (to Buffer)	Ολοκληρώνεται η κατεργασία – Ξεκινά η εκφόρτωση του τεμαχίου από τη μηχανή κατεργασίας στην ενδιάμεση αποθήκη
Finish loading part to Buffer	Ολοκληρώνεται η φόρτωση του τεμαχίου στην ενδιάμεση αποθήκη
Start unloading part from Buffer	Ξεκινά η εκφόρτωση του αποθηκευμένου τεμαχίου από την ενδιάμεση αποθήκη
Finish loading stocked part to Mi+1 - Start processing	Ολοκληρώνεται η φόρτωση του αποθηκευμένου τεμαχίου στην επόμενη μηχανή κατεργασίας – Ξεκινά η κατεργασία

4.2 Δημιουργία colourset και μεταβλητών

Καταρχήν , θα πρέπει να παραστήσουμε το τεμάχιο ως token μέσα στο σύστημα . Το τεμάχιο μέσα στο σύστημα δεν είναι πλέον ένα κανονικό token όπως στα απλά δίκτυα Petri αλλά ένα στοιχείο που χαρακτηρίζεται από 4 μεταβλητές **(p,r,i,n)** :

p : process plan τεμαχίου

r : επόμενη μηχανή (resource) στην οποία θα κατεργαστεί το τεμάχιο

i : αύξων αριθμός κατεργασίας στο process plan

n : συνολικός αριθμός κατεργασιών του συγκεκριμένου process plan

Παράδειγμα: το token (C,mp,1,3) είναι ένα τεμάχιο C που απαιτεί 3 συνολικά κατεργασίες, έχει ήδη υποστεί τόννευση και η επόμενη κατεργασία είναι στη φρέζα.

Οι παραπάνω μεταβλητές ανήκουν στα coloursets P, R, I, I αντίστοιχα και θα πρέπει να δηλωθούν μέσα στο πρόγραμμα ως εξής :

var p: P

var r: R

var i: I

var n: I

Colour Sets και τύποι token:

colset P: with A|B|C|D|E|F (αντιστοιχούν στους διαφορετικούς τύπους τεμαχίου)

colset R: with mp|tp (το mp αντιστοιχεί στη φρέζα και το tp στον τόρνο)

colset I: int with 0..3 (οι ακέραιοι αριθμοί 0,1,2,3 είναι αυτοί που χρησιμοποιούνται για να δηλώσουν τον τρέχοντα αριθμό και συνολικό αριθμό κατεργασιών του τεμαχίου μέσα στο σύστημα.

Εκτός από τα τεμάχια, πρέπει να παραστήσουμε μέσα στο σύστημα και τη διαθεσιμότητα των μηχανών κατεργασίας, των buffers και του ρομπότ. Έτσι δημιουργήσαμε και τα παρακάτω σύνολα :

colset M: with M|T (το M αντιστοιχεί στη φρέζα και το T στον τόρνο)

(αρχική σήμανση : 1`M++1`T)

colset antiM: with antiM|antiT (το M αντιστοιχεί στη φρέζα και το T στον τόρνο)

(αρχική σήμανση : 1`antiM++1`antiT)

colset Buffer: with Bm|Bt (αρχική σήμανση: 4`Bm++4`Bt)

colset Rav = with robot

```
▼ colset P = with A|B|C|D|E|F;  
▼ colset R= with mp|tp;  
▼ colset I = int with 0..3;  
▼ colset Rav= with robot;  
▼ colset M = with M|T;  
▼ colset antiM = with antiM|antiT;  
▼ colset Buffer=with Bm|Bt;  
▼ colset S= product P*R*I*I;  
▼ var p: P;  
▼ var r: R;  
▼ var i: I;  
▼ var n:I;  
▼ var m : M;
```

*Δήλωση μεταβλητών και coloursets του
μοντέλου FMS στο CPN tools*

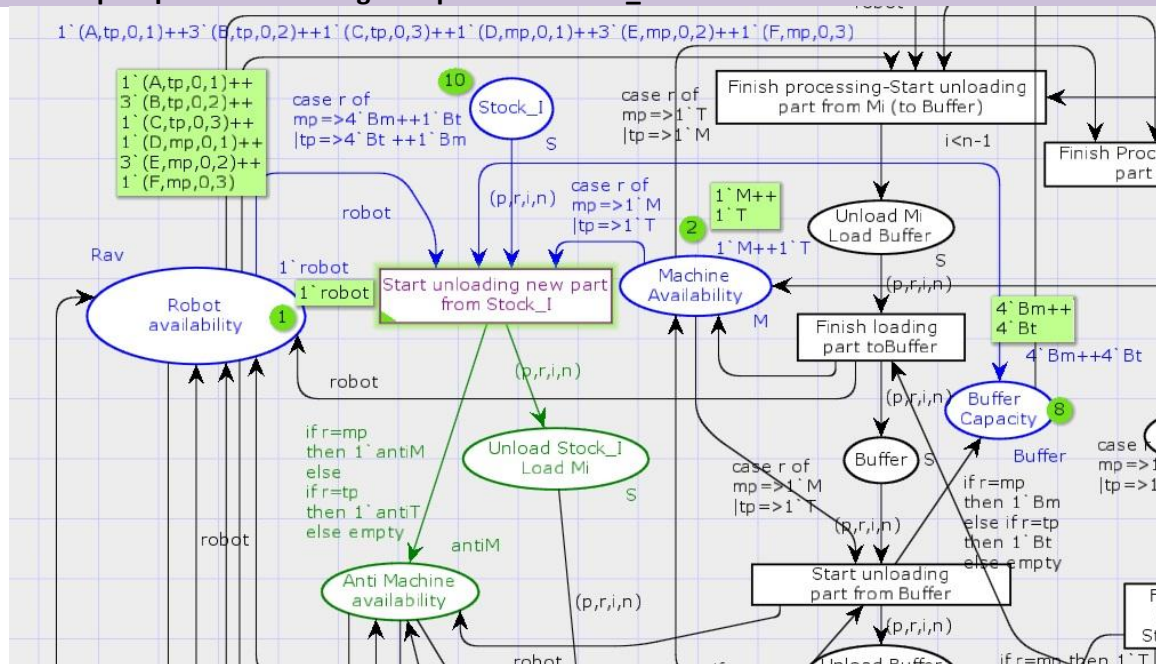
4.3 Εισαγωγή συνθηκών στα τόξα και τις μεταβάσεις του μοντέλου

Σε αυτήν την ενότητα θα παρουσιαστούν αναλυτικά οι συνθήκες που φέρουν τα τόξα και οι μεταβάσεις του μοντέλου. Δεδομένου ότι το τελικό μοντέλο ήταν αρκετά πολύπλοκο, σε κάθε μετάβαση που παρουσιάζεται παρακάτω έχουν σημειωθεί με μπλέ χρώμα τα τόξα εισόδου και με πράσινο τα τόξα εξόδου.

Τα τόξα εισόδου δείχνουν ποια στοιχεία απαιτούνται στις θέσεις εισόδου προκειμένου να ξεκινήσει η διαδικασία που περιγράφει η μετάβαση. Τα τόξα εξόδου υποδεικνύουν το αποτέλεσμα ενεργοποίησης της μετάβασης στις αντίστοιχες θέσεις εξόδου. Επιπλέον σε κάποιες μεταβάσεις χρησιμοποιούνται περιοριστικές συνθήκες οι οποίες ελέγχουν τη ροή των tokens.

Στις παρακάτω εικόνες του μοντέλου φαίνονται η τρέχουσα σήμανση κάθε θέσης (πράσινα κουτάκια) αλλά και ο τρέχων αριθμός των tokens σε κάθε θέση (πράσινα κυκλάκια)

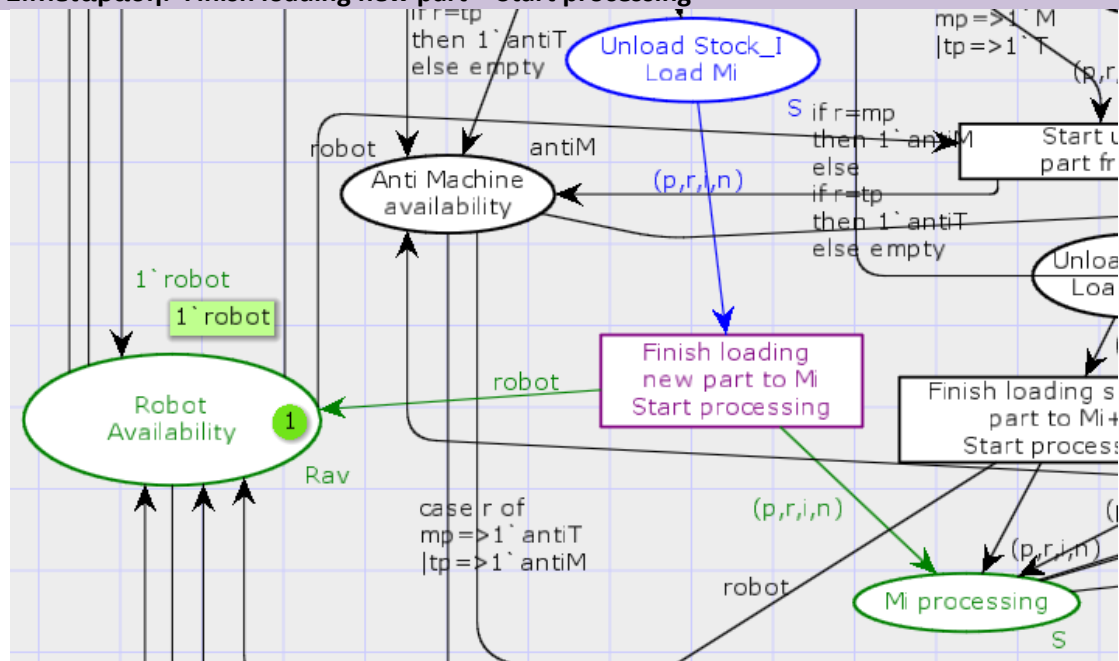
1. Μετάβαση: Start unloading new part from Stock_I



Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Robot availability	Rav	1`robot (αρχικά το ρομπότ είναι διαθέσιμο)	robot	Απαιτείται το ρομπότ
Machine availability	M	1`M++1`T (αρχικά και οι δύο μηχανές κατεργασίας είναι διαθέσιμες)	case r of mp=> 4`M   tp=> 4`T	Η διαδικασία mp(milling) απαιτεί έναν πόρο M (milling machine) ενώ η διαδικασία tp (turning) απαιτεί έναν πόρο T (turning machine)
Buffer Capacity	Buffer	4`Bm++4`Bt (αρχικά όλες οι θέσεις των buffers είναι άδειες)	case r of mp=>4`Bm+1`Bt tp=>4`Bt++1`Bm	Η διαδικασία mp(milling) απαιτεί 4 πόρους Bm* και έναν πόρο Bt **ενώ η διαδικασία tp(turning) απαιτεί 4 πόρους Bt και έναν πόρο Bm *αυτό γίνεται για να μην μπορεί να εισέρχεται νέο τεμάχιο στο σύστημα αν δεν αδειάσει πρώτα το ενδιάμεσο αντίστοιχο buffer ** αυτό γίνεται για να υπάρχει μια κενή θέση στο buffer της επόμενης κατεργασίας σε περίπτωση που χρειαστεί ενδιάμεση αποθήκευση
Stock_I	S	1`(A,tp,0,1)++3`(B,tp,0,2)++ 1`(C,tp,0,3)++1`(D,mp,0,1) ++ 3`(E, mp,0,2)++ 1`(F,mp,0,3)	(p,r,i,n)	Απαιτείται token (τεμάχιο) στην αρχική αποθήκη

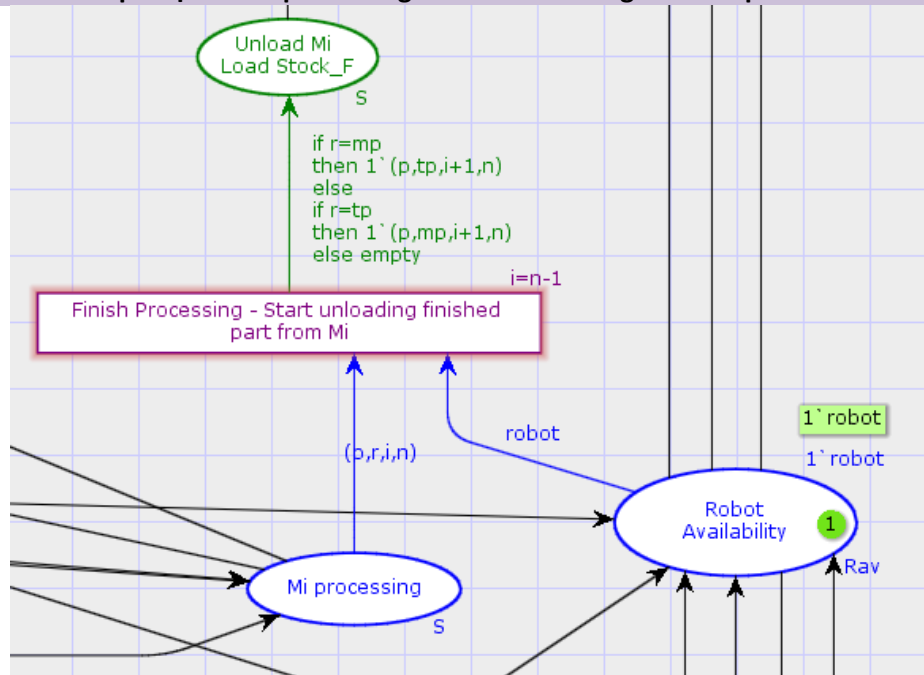
Unload Stock_I Load Mi	S	----	(p,r,i,n)	Η θέση αποκτά ένα token που αντιπροσωπεύει το τεμάχιο σε φάση μεταφοράς από την αρχική αποθήκη στη μηχανή κατεργασίας
Anti Machine availability	antiM	----	If r=mp then 1'AntiM else if r=tp then 1'AntiT else empty	Αν το τεμάχιο πρόκειται να υποστεί επεξεργασία mp(milling) δίνει στη συγκεκριμένη θέση έναν πόρο AntiM ενώ αν το τεμάχιο πρόκειται να υποστεί επεξεργασία tp (turning) δίνει στη συγκεκριμένη θέση έναν πόρο AntiT

2.Μετάβαση: Finish loading new part - Start processing



Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Unload Stock_I Load Mi	S	----	(p,r,i,n)	Απαιτείται token (τεμάχιο) που να βρίσκεται σε φάση μεταφοράς από την αρχική αποθήκη στη μηχανή κατεργασίας
Robot availability	Rav	1'robot	robot	Απελευθερώνεται το ρομπότ
Mi processing	S	---	(p,r,i,n)	Η θέση αποκτά ένα token (τεμάχιο) σε φάση κατεργασίας

3. Μετάβαση: Finish processing – Start unloading finished part from Mi

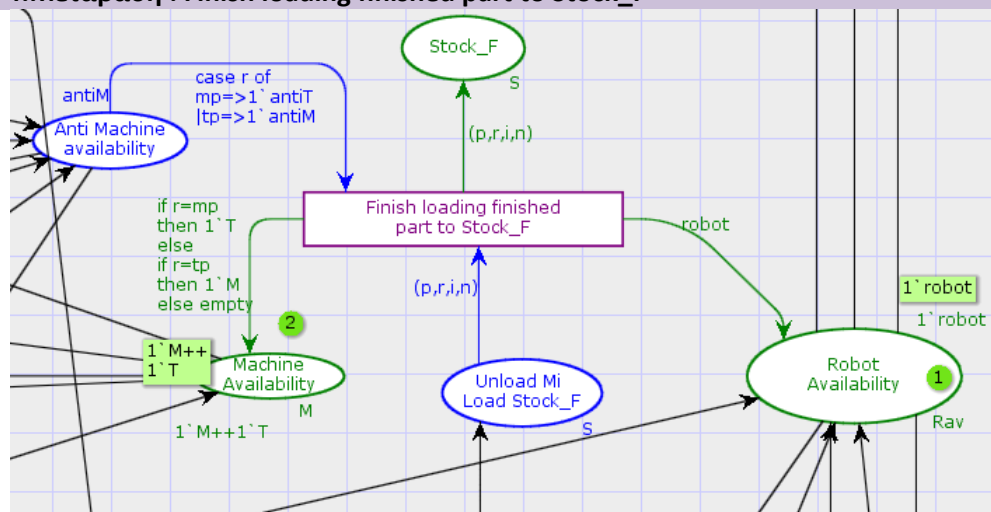


Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Mi Processing	S	----	(p,r,i,n)	Απαιτείται token (τεμάχιο) που να βρίσκεται σε φάση κατεργασίας
Robot availability	Rav	1^*robot	robot	Απαιτείται το ρομπότ
Unload Mi Load Stock_F	S	----	if $r=tp$ then $1^*(p,tp,i+1,n)$ else if $r=mp$ then $1^*(p,mp,i+1,n)$ else empty	Η θέση αποκτά ένα token (τεμάχιο) σε φάση μεταφοράς από τη μηχανή κατεργασίας στην τελική αποθήκη

Περιοριστική συνθήκη μετάβασης $i=n-1$

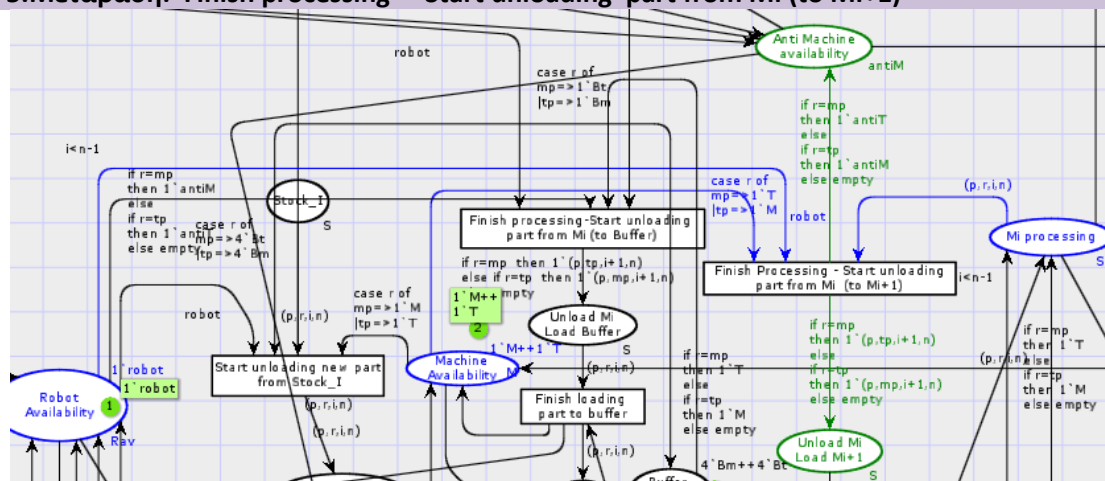
Από τη συγκεκριμένη μετάβαση πρέπει να περνά ένα token/τεμάχιο το οποίο έχει ολοκληρώσει όλες τις κατεργασίες που υπήρχαν στο αντίστοιχο process plan δηλαδή όταν ο αύξων αριθμός της κατεργασίας i γίνει ίσος με το συνολικό αριθμό n των κατεργασιών του process plan. Όμως επειδή η αλλαγή από i σε $i+1$ δεν μπορούσε να γίνει σε προηγούμενο βήμα (και παράλληλα αποφεύχθηκε η χρήση περισσότερων θέσεων από τις απαραίτητες) χρησιμοποιήθηκε τελικά η συνθήκη $i=n-1$.

4.Μετάβαση : Finish loading finished part to Stock_F



Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Unload Mi Load Stock_F	S	----	(p,r,i,n)	Απαιτείται token (τεμάχιο) που να βρίσκεται στη φάση μεταφοράς από τη μηχανή κατεργασίας στην τελική αποθήκη
Anti Machine availability	AntiM	----	case r of mp=> 1'AntiM tp=>1'AntiT	Αν τελευταία επεξεργασία του τεμαχίου ήταν mp(milling) δεσμεύεται από τη συγκεκριμένη θέση ένας πόρος AntiM ενώ αν η τελευταία επεξεργασία του τεμαχίου ήταν tp (turning) δεσμεύεται από τη συγκεκριμένη θέση ένας πόρος AntiT
Stock_F	S		(p,r,i,n)	Η θέση αποκτά ένα token (τεμάχιο) που αποθηκεύεται στην τελική αποθήκη
Machine availability	M	----	if r=mp then 1'M else if r=tp then 1'T	Αν τελευταία επεξεργασία του τεμαχίου ήταν mp(milling) ελευθερώνεται ένας πόρος M ενώ αν η τελευταία επεξεργασία του τεμαχίου ήταν tp (turning) ελευθερώνεται ένας πόρος T
Robot availability	Rav	----	1'robot	Ελευθερώνεται το ρομπότ

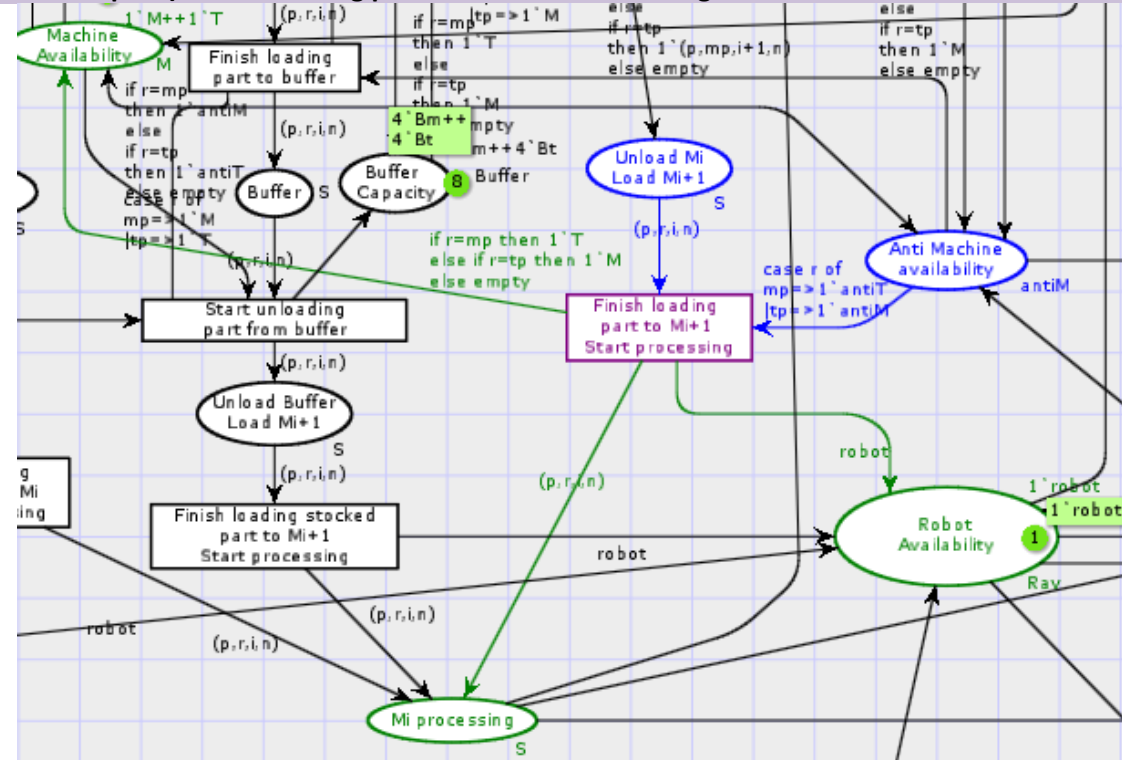
5.Μεταβαση: Finish processing – Start unloading part from Mi (to Mi+1)



Προκειμένου να αποφευχθεί η προσθήκη περισσότερων θέσεων στο μοντέλο (ο αύξων αριθμός κατεργασίας i γίνεται $i+1$ στο επόμενο στάδιο) και δεδομένου πάντα ότι οι κατεργασίες σε κάθε τεμάχιο γίνονται στις 2 μηχανές κατεργασίας εναλλακτικά χρησιμοποιείται ως κριτήριο η προηγούμενη μηχανή κατεργασίας για να δεσμευτεί η επόμενη.

Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Robot availability	Rav	1`robot (αρχικά το ρομπότ είναι διαθέσιμο)	robot	Απαιτείται το ρομπότ
Machine availability	M	1`M++1`T (αρχικά και οι δύο μηχανές κατεργασίας είναι διαθέσιμες)	case r of mp=>1`T   tp=>1`M	Αν η προηγούμενη κατεργασία ήταν mp(milling) απαιτείται ένας πόρος T (turning machine) ενώ αν ήταν tp (turning) απαιτείται ένας πόρος M (milling machine) **Χρησιμοποιείται ως κριτήριο η προηγούμενη μηχανή κατεργασίας για να δεσμευτεί η επόμενη
Mi processing	S	----	(p,r,i,n)	Απαιτείται token (τεμάχιο) που να βρίσκεται σε φάση κατεργασίας
Unload Mi Load Mi+1	S	----	if r=tp then 1`(p,tp,i+1,n) else if r=mp then 1`(p,mp,i+1,n) else empty	Η θέση αποκτά ένα token (τεμάχιο) σε φάση μεταφοράς από την προηγούμενη στην επόμενη μηχανή κατεργασίας
Anti Machine availability	antiM	----	If r=mp then 1`AntiT else if r=tp then 1`AntiM else empty	Αν η προηγούμενη κατεργασία ήταν mp(milling) ελευθερώνεται ένας πόρος AntiT (turning machine) ενώ αν ήταν tp (turning) ελευθερώνεται ένας πόρος AntiM **Χρησιμοποιείται ως κριτήριο η προηγούμενη μηχανή κατεργασίας για να δεσμευτεί η επόμενη

6. Μετάβαση : Finish loading part to Mi - Start Processing

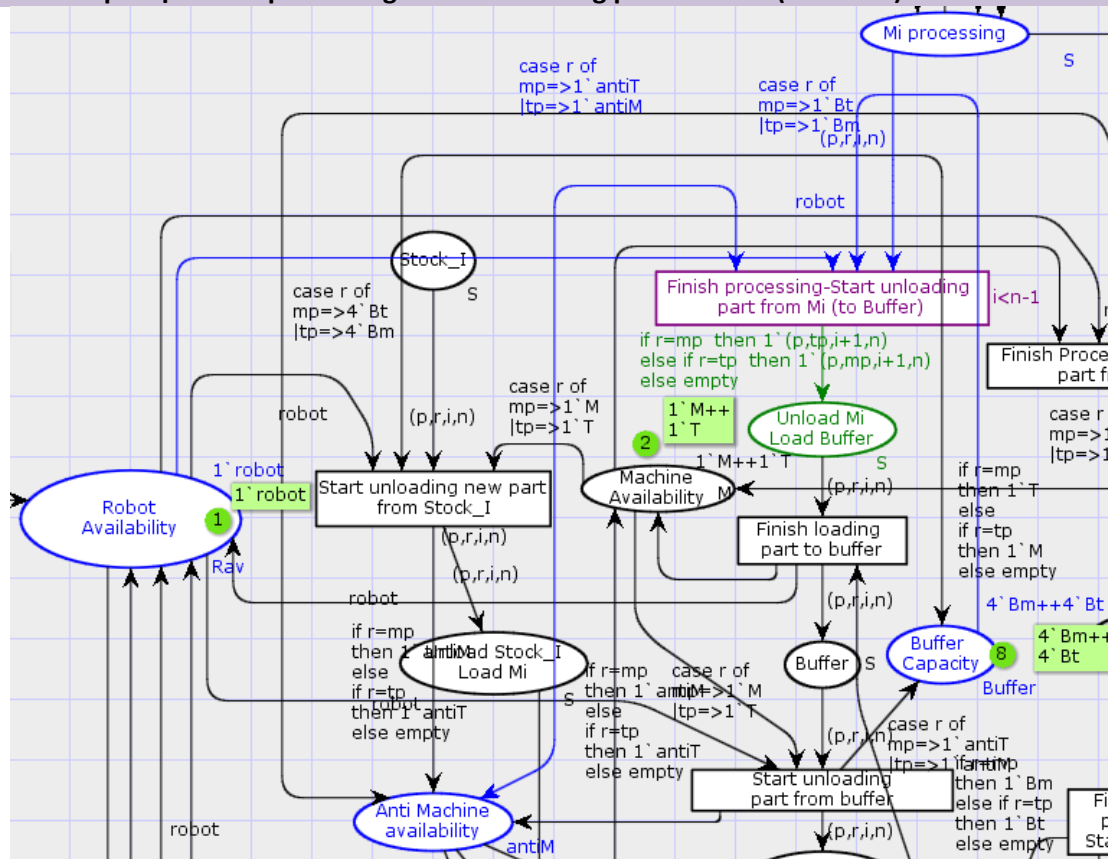


Σε αυτή τη μετάβαση θα πρέπει να ελευθερωθεί η προηγούμενη μηχανή κατεργασίας. Όμως επειδή ο τρέχων αριθμός της κατεργασίας έχει αλλάξει από i σε $i+1$ και δεδομένου και εδώ ότι πάντα οι κατεργασίες γίνονται στις 2 μηχανές εναλλακτικά, οι συνθήκες των τόξων χρησιμοποιούν ως κριτήριο την τρέχουσα μηχανή κατεργασίας για να ελευθερωθεί η προηγούμενη.

Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Unload Mi Load Mi+1	S	----	(p,r,i,n)	Απαιτείται τεμάχιο που να βρίσκεται στη φάση μεταφοράς από την προηγούμενη στην επόμενη μηχανή κατεργασίας
Anti Machine availability	AntiM	----	case r of $mp \Rightarrow 1'AntiT$ $tp \Rightarrow 1'AntiM$	Αν η επόμενη επεξεργασία του τεμαχίου είναι mp (milling) δεσμεύεται από τη συγκεκριμένη θέση ένας πόρος AntiT ενώ αν η επόμενη επεξεργασία του τεμαχίου ήταν tp (turning) δεσμεύεται από τη συγκεκριμένη θέση ένας πόρος AntiT **Χρησιμοποιείται ως κριτήριο η τρέχουσα μηχανή κατεργασίας για να απελευθερωθεί η προηγούμενη
Mi processing	S	----	(p,r,i,n)	Η θέση αποκτά ένα token (τεμάχιο) σε φάση κατεργασίας
Machine availability	M	----	if $r=mp$ then $1'T$ else if $r=tp$ then $1'M$	Αν η επόμενη επεξεργασία του τεμαχίου είναι mp (milling) ελευθερώνεται ένας πόρος T ενώ αν η ήταν tp (turning) ελευθερώνεται ένας πόρος M

				**Χρησιμοποιείται ως κριτήριο η τρέχουσα μηχανή κατεργασίας για να απελευθερωθεί η προηγούμενη
Robot availability	Rav	----	robot	Ελευθερώνεται το ρομπότ

7. Μετάβαση : Finish processing - Start unloading part from Mi (to Buffer)

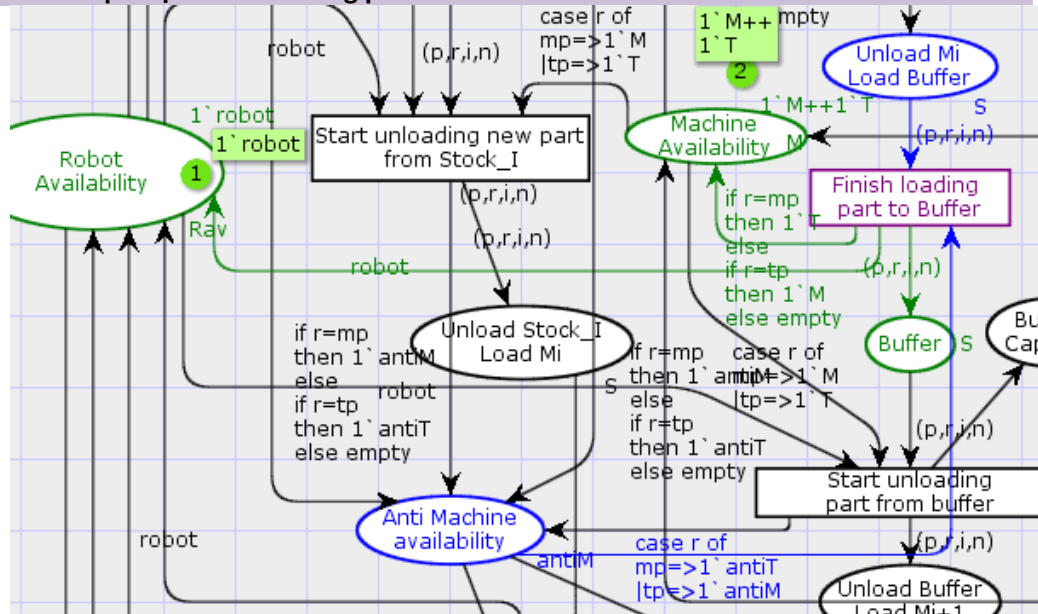


Και εδώ για τους ίδιους λόγους που προαναφέρθηκαν στη μετάβαση **Finish processing – Start unloading part from Mi (to Mi+1)** χρησιμοποιείται ως κριτήριο η προηγούμενη μηχανή κατεργασίας για να δεσμευτεί η επόμενη.

Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Mi processing	S	----	(p,r,i,n)	Απαιτείται token (τεμαχίο) που να βρίσκεται σε φάση κατεργασίας
Robot availability	Rav	1'robot	robot	Απαιτείται το ρομπότ
Anti Machine availability	M	1'M++1'T	Case r of mp=>1'AntiT tp=>1'AntiM	Αν η τρέχουσα κατεργασία του τεμαχίου είναι mp(milling) δεσμεύεται από τη συγκεκριμένη θέση ένας πόρος AntiT ενώ αν είναι tp(turning) δεσμεύεται από τη συγκεκριμένη θέση ένας πόρος AntiT **Χρησιμοποιείται ως κριτήριο η

				προηγούμενη μηχανή κατεργασίας για να κριθεί αν θα αποθηκευτεί το τεμάχιο στο Buffer
Buffer Capacity	Buffer	$4 \cdot B_m ++ 4 \cdot B_t$ (αρχικά όλες οι θέσεις των ενδιάμεσων buffers είναι άδειες)	$\text{case } r \text{ of}$ $mp \Rightarrow 1 \cdot B_t \mid$ $tp \Rightarrow 1 \cdot B_m$	Αν η προηγούμενη μηχανή κατεργασίας mp τότε η επόμενη είναι tp άρα απαιτείται μια κενή θέση στο Buffer της tp Ενώ αν η προηγούμενη μηχανή κατεργασίας tp τότε η επόμενη είναι mp άρα απαιτείται μια κενή θέση στο Buffer της mp **Χρησιμοποιείται ως κριτήριο η προηγούμενη μηχανή κατεργασίας για να δεσμευτεί θέση στο buffer που αντιστοιχεί στην επόμενη κατεργασία
Unload Mi Load Buffer	S	----	(p, r, i, n)	Απαιτείται (token) τεμάχιο που να βρίσκεται σε φάση μεταφοράς από τη μηχανή κατεργασίας στο Buffer

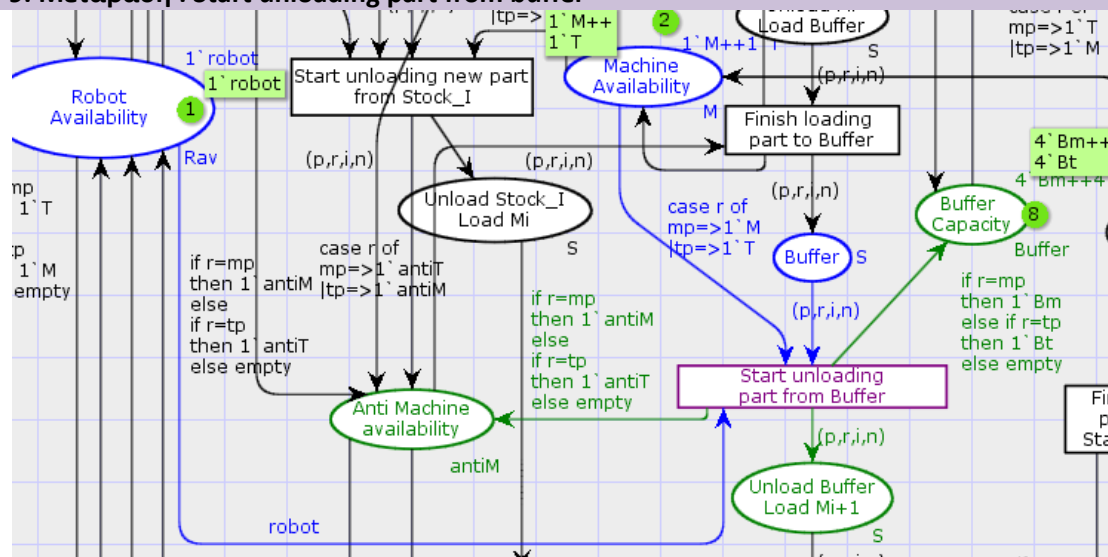
8. Μετάβαση : Finish loading part to Buffer



Και εδώ για τους ίδιους λόγους που προαναφέρθηκαν στη μετάβαση **Finish loading part to Mi - Start Processing** οι συνθήκες των τόζων χρησιμοποιούν ως κριτήριο την τρεχουσα μηχανή κατεργασίας για να ελευθερωθεί η προηγούμενη.

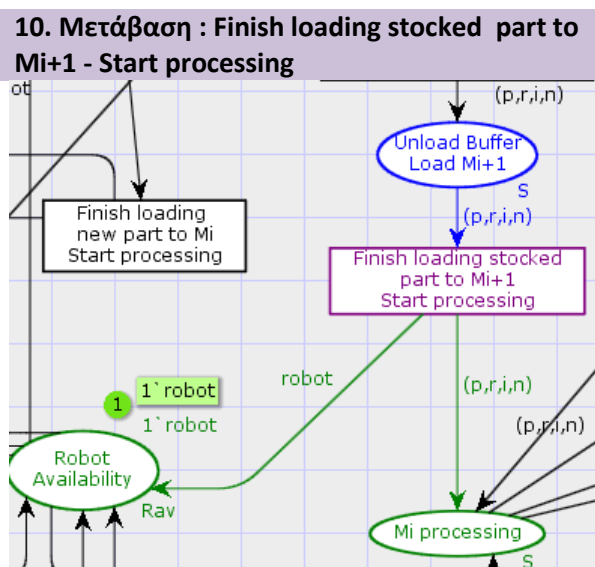
Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Unload Mi Load Buffer	S	----	(p,r,i,n)	Απαιτείται τεμάχιο που να βρίσκεται σε φάση μεταφοράς από την προηγούμενη μηχανή κατεργασίας στο Buffer
Anti Machine availability	AntiM	----	Case r of mp=> 1'AntiT tp=>1'AntiM	Αν η επόμενη επεξεργασία του τεμαχίου είναι mp(milling) δεσμεύεται ένας πόρος AntiT ενώ αν είναι tp (turning) δεσμεύεται ένας πόρος AntiM **Χρησιμοποιείται ως κριτήριο η τρέχουσα μηχανή κατεργασίας για να απελευθερωθεί η προηγούμενη
Buffer	S	----	(p,r,i,n)	Η θέση αποκτά ένα token (τεμάχιο) αποθηκευμένο στην ενδιάμεση αποθήκη
Machine availability	M	----	if r=mp then 1'T else if r=tp then 1'M	Αν η επόμενη επεξεργασία του τεμαχίου είναι mp(milling) ελευθερώνεται ένας πόρος T ενώ αν η είναι tp (turning) ελευθερώνεται ένας πόρος M **Χρησιμοποιείται ως κριτήριο η τρέχουσα μηχανή κατεργασίας για να απελευθερωθεί η προηγούμενη
Robot availability	Rav	----	robot	Ελευθερώνεται το ρομπότ

9. Μετάβαση : Start unloading part from buffer



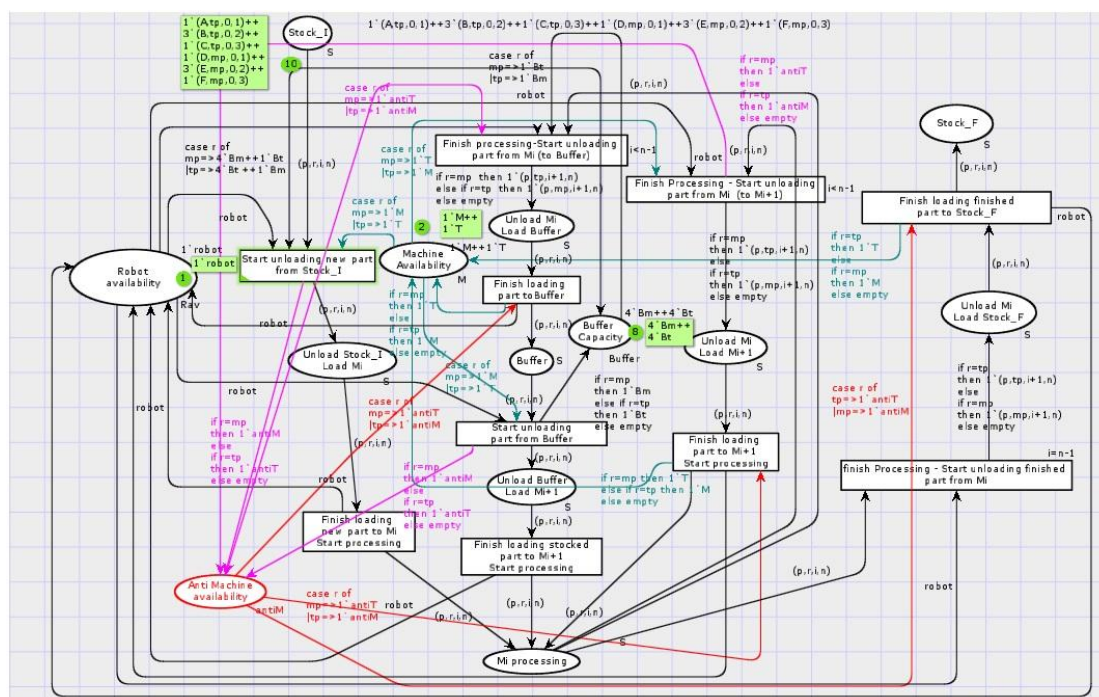
Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Buffer	S	----	(p,r,i,n)	Απαιτείται (token) τεμάχιο που να βρίσκεται στο Buffer
Robot availability	Rav	1'robot	robot	Απαιτείται το ρομπότ

Machine availability	M	$1'M++1'T$	case r of mp=>1'M tp=>1'T	Αν η επόμενη κατεργασία είναι mp(milling) απαιτείται ένας πόρος M (milling machine) ενώ αν είναι tp (turning) απαιτείται ένας πόρος T (turning machine)
Unload Buffer Load Mi+1	S	----	(p,r,i,n)	Η θέση αποκτά ένα token (τεμάχιο) που βρίσκεται σε φάση μεταφοράς από το Buffer στην επόμενη μηχανή κατεργασίας
Buffer Capacity	Buffer	$4'Bm++4'Bt$	If r=mp then 1'Bm else if r=tp then 1'Bt else empty	Αν η επόμενη κατεργασία είναι mp(milling) ελευθερώνεται ένας πόρος Bm (ελευθερώνεται μία θέση στο αντίστοιχο buffer) ενώ αν η επόμενη κατεργασία είναι tp(milling) ελευθερώνεται ένας πόρος Bt
Anti Machine availability	antiM	----	If r=mp then 1'AntiM else if r=tp then 1'AntiT else empty	Αν η επόμενη κατεργασία είναι mp(milling) ελευθερώνεται ένας πόρος AntiM ενώ αν η επόμενη κατεργασία είναι tp(milling) ελευθερώνεται ένας πόρος AntiT



Όνομα θέσης	Colourset θέσης	Αρχική σήμανση	Συνθήκη τόξου	Επεξήγηση
Unload Buffer Load Mi+1	S	----	(p,r,i,n)	Απαιτείται token (τεμάχιο) που να βρίσκεται σε φάση μεταφοράς από το Buffer στην επόμενη μηχανή κατεργασίας
Robot availability	Rav	1' robot	robot	Ελευθερώνεται το ρομπότ
Mi processing	S	----	(p,r,i,n)	Η θέση αποκτά ένα token (τεμάχιο) σε φάση κατεργασίας

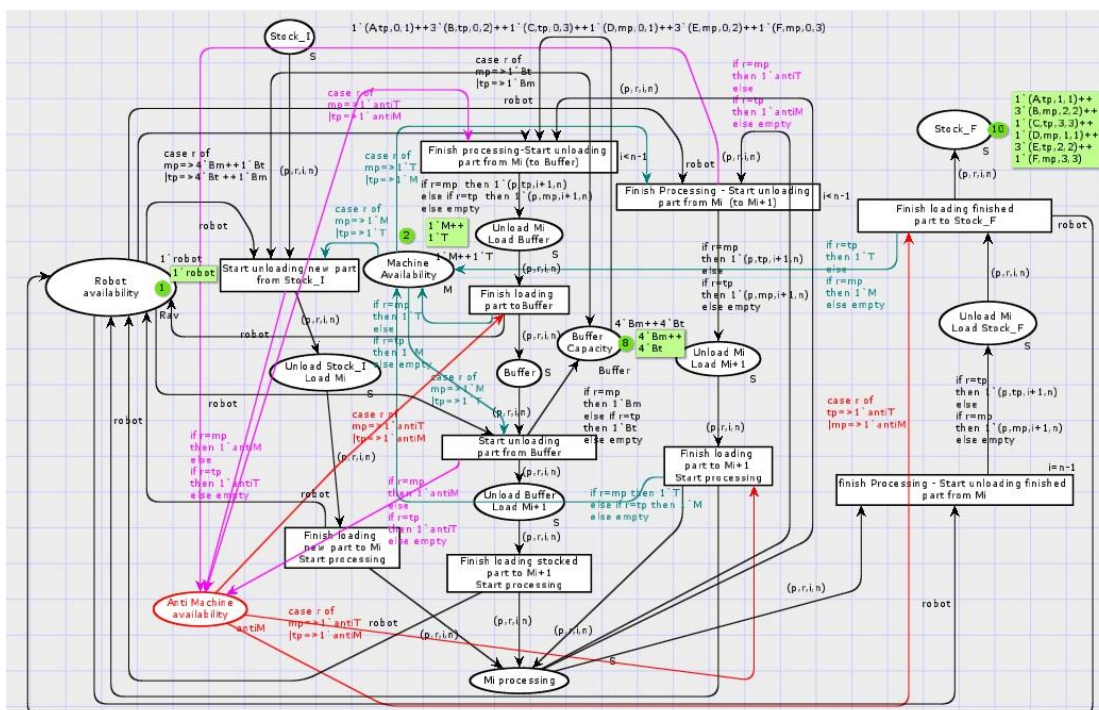
Το πλήρες μοντελοποιημένο σύστημα παρουσιάζεται παρακάτω :



Πλήρες μοντελοποιημένο σύστημα στο CPN tools

4.4 Προσομοίωση μοντέλου

Χρησιμοποιώντας τα εργαλεία της παλέτας Sim ,τα οποία αναλύθηκαν στο προηγούμενο κεφάλαιο, μπορούμε να κάνουμε προσομοίωση του συστήματος με διάφορους τρόπους είτε βήμα –βήμα είτε αυτόματα. Σε όλες τις περιπτώσεις η τελική σήμανση του μοντέλου είναι αυτή που φαίνεται παρακάτω :



Η τελική σήμανση , όπως αποδεικνύεται και από το αποτέλεσμα της προσομοίωσης , είναι όλα τα κατεργασμένα τεμάχια να καταλήξουν στην τελική αποθήκη Stock_F.

Επίσης , όπως είναι αναμενόμενο, η αρχική αποθήκη και τα Buffers έχουν αδειάσει και οι μηχανές κατεργασίας και το ρομπότ είναι διαθέσιμα.

4.5 Ανάλυση χώρου καταστάσεων (State space)

Όπως επισημάνθηκε και στο προηγούμενο κεφάλαιο , η προσομοίωση δεν είναι αρκετή για να εξακριβωθούν οι ιδιότητες του συστήματος. Αυτό μπορεί να γίνει μόνο με τον υπολογισμό του χώρου καταστάσεων ο οποίος βασίζεται στη θεωρία των δικτύων Petri.

Χρησιμοποιώντας τα δύο πρώτα από τα εργαλεία της παλέτας Sim ,τα οποία αναλύθηκαν στο προηγούμενο κεφάλαιο, το πρόγραμμα υπολογίζει το χώρο καταστάσεων και το SCC γράφημα.

Στη συνέχεια με το εργαλείο Save report σώζεται μια αναφορά της state space ανάλυσης σε αρχείο .txt. Ανοίγοντας το αρχείο αυτό παίρνουμε τα παρακάτω δεδομένα:

```
Statistics
-----
State Space
Nodes: 162972
Arcs: 225188
Secs: 584
Status: Full Scc Graph
Nodes: 162972
Arcs: 225188
Secs: 6
```

Βασικά στοιχεία του χώρου καταστάσεων του μοντέλου FMS από το αρχείο .txt που αποθηκεύει η εντολή Save report στο CPN tools

Όπως φαίνεται από τα αποτελέσματα που έδωσε η αναφορά, ο χρόνος υπολογισμού του χώρου καταστάσεων είναι 584 seconds . Παρατηρείται ακόμα ότι ο αριθμός των κόμβων του χώρου κατάστασης είναι ίσος με τον αριθμό των κόμβων του SSC γραφήματος. Αυτό επιβεβαιώνει ότι το σύστημα δεν παρουσιάζει κυκλική συμπεριφορά.

Από τα boundness properties μπορεί να ελεγχθεί ο αριθμός και ο τύπος των tokens κάθε θέσης .

Boundedness Properties

Best Integer Bounds	Upper	Lower
Plan_1'Anti_Machine_availability	2	0
Plan_1'Buffer	5	0
Plan_1'Buffer_Capacity	8	3
Plan_1'Machine_Availability	2	0
Plan_1'Mi_processing	2	0
Plan_1'Robot_availability	1	0
Plan_1'Stock_F	10	0
Plan_1'Stock_I	10	0
Plan_1'Unload_Buffer_Load_Mi	1	0
Plan_1'Unload_Mi_Load_Buffer	1	0
Plan_1'Unload_Mi_Load_Mi	1	0
Plan_1'Unload_Mi_Load_Stock_F	1	0
Plan_1'Unload_Stock_I_Load_Mi	1	0

Μέγιστος και ελάχιστος αριθμός tokens κάθε θέσης στο χώρο καταστάσεων του μοντέλου FMS από το αρχείο .txt που αποθηκεύει η εντολή Save report στο CPN tools

Best Upper Multi-set Bounds

Plan_1'Anti_Machine_availability	1`antiM++1`antiT
Plan_1'Buffer	3`(B,mp,1,2)++1`(C,mp,1,3)++1`(C,tp,2,3)++ 3`(E,tp,1,2)++1`(F,mp,2,3)++1`(F,tp,1,3)
Plan_1'Buffer_Capacity	4`Bm++4`Bt
Plan_1'Machine_Availability	1`M++1`T
Plan_1'Mi_processing	1`(A,tp,0,1)++1`(B,mp,1,2)++1`(B,tp,0,2)++ 1`(C,mp,1,3)++1`(C,tp,0,3)++1`(C,tp,2,3)++ 1`(D,mp,0,1)++1`(E,mp,0,2)++1`(E,tp,1,2)++ 1`(F,mp,0,3)++1`(F,mp,2,3)++1`(F,tp,1,3)
Plan_1'Robot_availability	1`robot
Plan_1'Stock_F	1`(A,tp,1,1)++3`(B,mp,2,2)++1`(C,tp,3,3)++ 1`(D,mp,1,1)++3`(E,tp,2,2)++1`(F,mp,3,3)
Plan_1'Stock_I	1`(A,tp,0,1)++3`(B,tp,0,2)++1`(C,tp,0,3)++ 1`(D,mp,0,1)++3`(E,mp,0,2)++1`(F,mp,0,3)
Plan_1'Unload_Buffer_Load_Mi	1`(B,mp,1,2)++1`(C,mp,1,3)++1`(C,tp,2,3)++ 1`(E,tp,1,2)++1`(F,mp,2,3)++1`(F,tp,1,3)
Plan_1'Unload_Mi_Load_Buffer	1`(B,mp,1,2)++1`(C,mp,1,3)++1`(C,tp,2,3)++ 1`(E,tp,1,2)++1`(F,mp,2,3)++1`(F,tp,1,3)
Plan_1'Unload_Mi_Load_Mi	1`(B,mp,1,2)++1`(C,mp,1,3)++1`(C,tp,2,3)++ 1`(E,tp,1,2)++1`(F,mp,2,3)++1`(F,tp,1,3)
Plan_1'Unload_Mi_Load_Stock_F	1`(A,tp,1,1)++1`(B,mp,2,2)++1`(C,tp,3,3)++ 1`(D,mp,1,1)++1`(E,tp,2,2)++1`(F,mp,3,3)
Plan_1'Unload_Stock_I_Load_Mi	1`(A,tp,0,1)++1`(B,tp,0,2)++1`(C,tp,0,3)++ 1`(D,mp,0,1)++1`(E,mp,0,2)++1`(F,mp,0,3)

Best Lower Multi-set Bounds

Plan_1'Anti_Machine_availability	empty
Plan_1'Buffer	empty
Plan_1'Buffer_Capacity	empty
Plan_1'Machine_Availability	empty
Plan_1'Mi_processing	empty
Plan_1'Robot_availability	empty
Plan_1'Stock_F	empty
Plan_1'Stock_I	empty
Plan_1'Unload_Buffer_Load_Mi	empty
Plan_1'Unload_Mi_Load_Buffer	empty
Plan_1'Unload_Mi_Load_Mi	empty
Plan_1'Unload_Mi_Load_Stock_F	empty
Plan_1'Unload_Stock_I_Load_Mi	empty

Μέγιστο και ελάχιστο πολυσύνολο tokens κάθε θέσης στο χώρο καταστάσεων του μοντέλου FMS από το αρχείο .txt που αποθηκεύει η εντολή Save report στο CPN tools

Παρατηρείται ότι οι θέσεις που αντιστοιχούν σε φάση μεταφοράς τεμαχίου από το ρομπότ έχουν άνω όριο το 1 (εφόσον υπάρχει ένα ρομπότ στο σύστημα) ενώ η θέση Mi processing έχει άνω όριο 2 (μέχρι δύο τεμάχια είναι δυνατόν να κατεργάζονται κάθε στιγμή αφού οι μηχανές κατεργασίες είναι 2). Επίσης οι θέσεις Machine availability και Anti-Machine availability έχουν άνω όριο 2 (2 μηχανές διαθέσιμες ή μη διαθέσιμες αντίστοιχα) ενώ το ρομπότ έχει άνω όριο το 1.

Από τα liveness properties φαίνεται ότι υπάρχει μόνο μία νεκρή σήμανση που αντιστοιχεί στον κόμβο 162972 και μάλιστα συμπίπτει με την επίσης μοναδική σήμανση home που αντιστοιχεί στον ίδιο κόμβο.

Home Properties

Home Markings [162972]
Liveness Properties

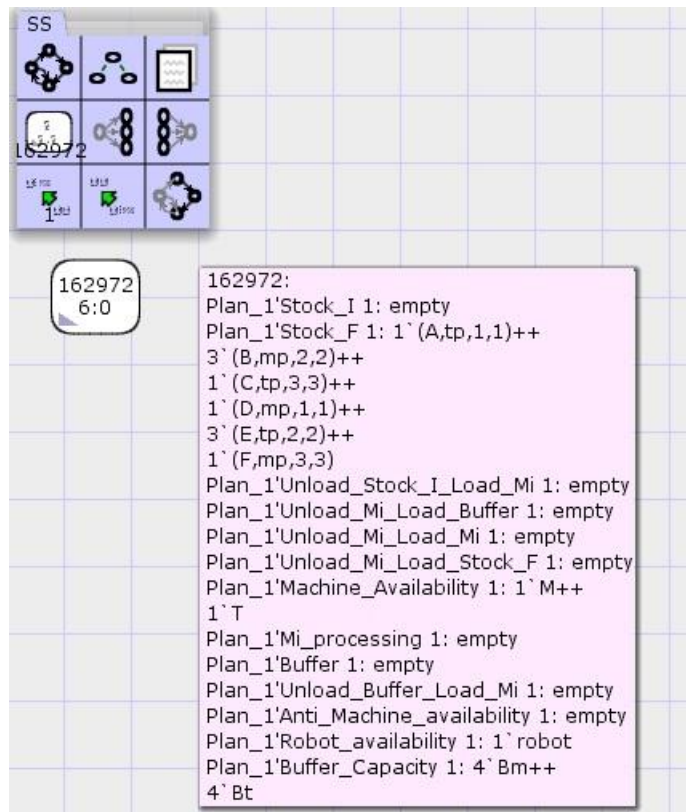
Dead Markings [162972]
Dead Transition Instances None
Live Transition Instances None

Fairness Properties

No infinite occurrence sequences.

Ιδιότητες home και ιδιότητες 'ζωτικότητας' από την ανάλυση του χώρου καταστάσεων του μοντέλου FMS από το αρχείο .txt που αποθηκεύει η εντολή Save report στο CPN tools

Γράφοντας τον αριθμό 162972 του συγκεκριμένου κόμβου πάνω στο 4^ο εργαλείο της παλέτας state space και κλικάροντας το εργαλείο στο workspace εμφανίζεται η σήμανση που αντιστοιχεί στον κόμβο :

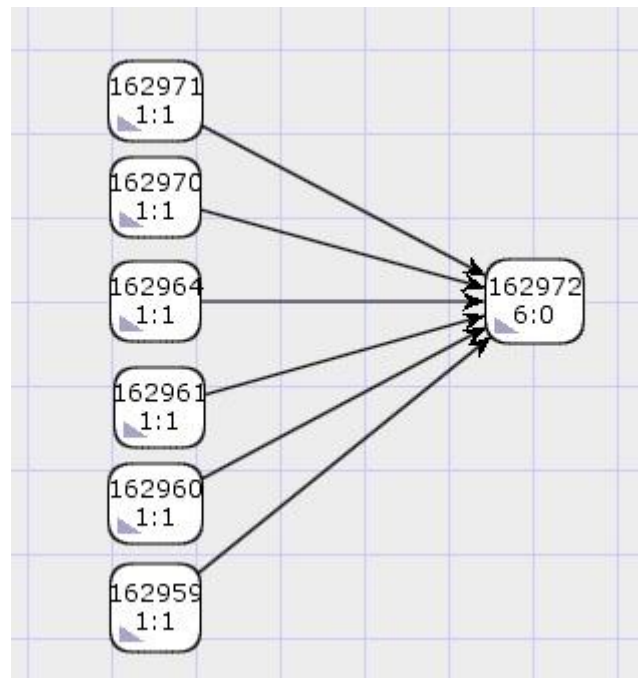


Εμφάνιση σήμανσης που αντιστοιχεί στον κόμβο 162972 του μοντέλου μετά την ανάλυση χώρου καταστάσεων στο CPN tools

Όπως φαίνεται η νεκρή σήμανση αλλά και η σήμανση home αντιστοιχούν στην κατάσταση του συστήματος όπου όλα τα κατεργασμένα πλέον τεμάχια έχουν φτάσει στην τελική αποθήκη προορισμού. Συνεπώς, η ανάλυση state space επιβεβαιώνει ότι το μοντελοποιημένο σύστημα φτάνει πάντοτε στην επιθυμητή κατάσταση και μάλιστα από οποιαδήποτε κατάσταση και αν βρίσκεται. Έτσι επιβεβαιώνεται η σωστή λειτουργία του συστήματος αλλά και η απουσία κολλημάτων (deadlocks) που ήταν και ο απώτερος στόχος.

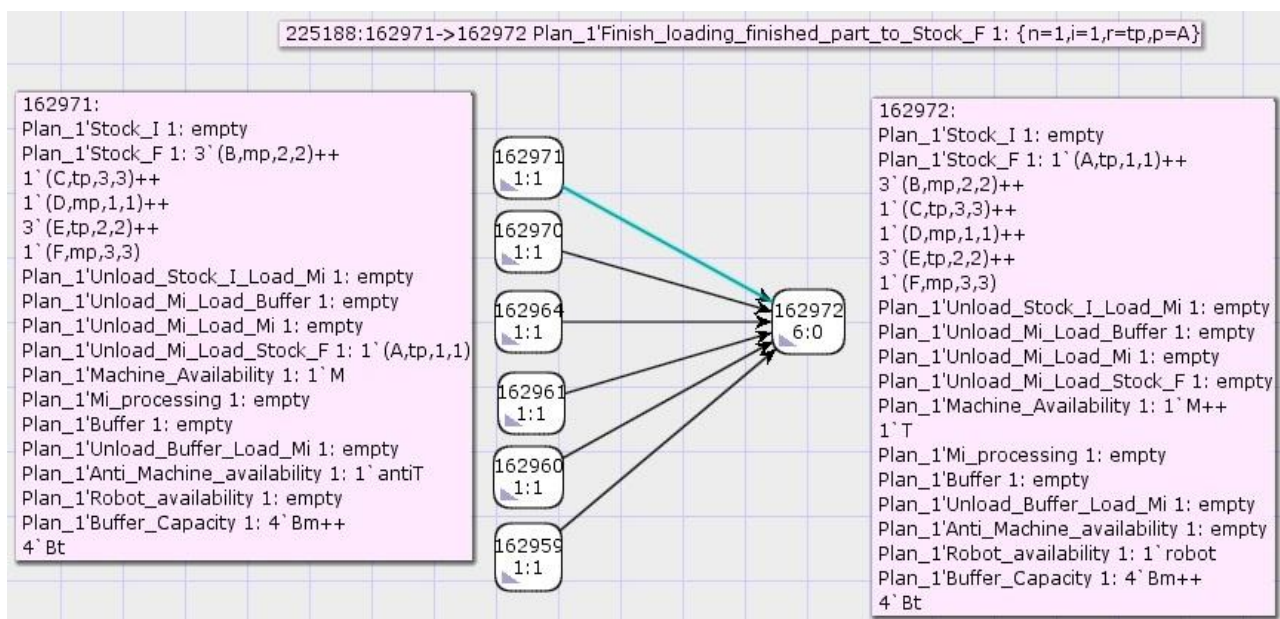
Οι ιδιότητες ζωτικότητας μας πληροφορούν για την απουσία νεκρών μεταβάσεων όπως είναι επιθυμητό καθώς αν υπήρχαν μεταβάσεις που δεν ενεργοποιούνται ποτέ μέσα στο σύστημα θα ήταν περιττές. Ακόμα μας πληροφορούν και για την απουσία ζωντανών μεταβάσεων όπως είναι αναμενόμενο καθώς υπάρχει νεκρή σήμανση στο σύστημα.

Έχοντας τη σήμανση 162971 στο workspace και χρησιμοποιώντας τα εργαλεία **display successors to this node** βλέπουμε όλους τους κόμβους (σημάνσεις) που προηγούνται από τη νεκρή σήμανση.



Εμφάνιση σήμανσεων που προηγούνται της νεκρής σήμανσης στο μοντέλο FMS μετά την ανάλυση χώρου καταστάσεων στο CPN tools

Αν χρησιμοποιήσουμε το **display predecessors to this node** επιβεβαιώνουμε ότι η νεκρή σήμανση δεν ακολουθείται από καμία επόμενη κατάσταση που είναι και το αναμενόμενο εξ ορισμού. Κλικάροντας πάνω στους συνδέσμους μεταξύ των κόμβων μπορούμε να δούμε τις αντιστοιχίσεις μεταβλητών με τις τιμές των tokens που οδηγούν από τις σημάνσεις αυτές στην τελική νεκρή σήμανση.



Αναλυτική παρουσίαση κόμβων και συνδέσμων στο μοντέλο FMS μετά την ανάλυση χώρου καταστάσεων στο CPN tools

Όπως φαίνεται στην παραπάνω εικόνα από το workspace, στον κόμβο(σήμανση) 162971 υπάρχει ένα token (A,tp,1,1) στη θέση **Unload_Mi_Load_Stock_F** ενώ παράλληλα η μηχανή κατεργασίας T και το ρομπότ δεν είναι διαθέσιμα. Ο σύνδεσμος 225188 δείχνει την ενεργοποίηση της μετάβασης **Finish_loading_finshed_part_to_Stock_F** μέσω της αντιστοίχισης των μεταβλητών n,i,r,p στις τιμές του token 1,1,tp,A. Η ενεργοποίηση αυτή οδηγεί στον κόμβο 162972 όπου αυτό το τελευταίο token καταλαμβάνει τη θέση Stock_F ενώ το ρομπότ και η μηχανή κατεργασίας T ελευθερώνονται.

Η παραπάνω αναλυτική παρουσίαση των κόμβων και συνδέσμων του state space είναι ιδιαίτερα χρήσιμη για τον εντοπισμό σφαλμάτων. Σε διάφορες φάσεις της κατάστρωσης και ελέγχου του μοντέλου FMS παρουσιάστηκαν περισσότερες από μία νεκρές σημάνσεις λόγω προγραμματιστικών σφαλμάτων ή παραλλείψεων.

Ο αριθμός του κόμβου μιας νεκρής σήμανσης που δίνεται από την αναφορά σε συνδυασμό με τα εργαλεία του state space δίνουν αναλυτικά στοιχεία για το πως οδηγήθηκε το σύστημα σε μία νεκρή σήμανση. Με αυτόν τον τρόπο διορθώθηκαν τα διάφορα deadlocks που εμφανίστηκαν στη φάση του σχεδιασμού ώστε τελικά το σύστημα να φτάνει πάντοτε στην τελική επιθυμητή κατάσταση.

Κεφάλαιο 5: Μετατροπή δικτύων Petri σε διαγράμματα ladder

5.1 Γενικά

5.1.1 Εισαγωγή στους Προγραμματιζόμενους λογικούς ελεγκτές

Οι προγραμματιζόμενοι λογικοί ελεγκτές ή PLC (Programmable Logic Controllers) ανήκουν στην ευρύτερη κατηγορία των ψηφιακών Υπολογιστικών Συστημάτων. Χρησιμοποιούνται για να ελέγχουν μηχανές και διεργασίες όπου απαιτείται να γίνονται αυτόματες λειτουργίες: κατ' εξοχήν στη βιομηχανία αλλά και σε κτιριακές εγκαταστάσεις, στη ναυτιλία, σε μεγάλα έργα όπως σήραγγες ή σταθμούς παραγωγής ενέργειας, στον έλεγχο κυκλοφορίας οχημάτων, σε συστήματα ανελκυστήρων και σε πολλούς άλλους τομείς εφαρμογών.

Η καθιέρωση των PLC προέκυψε τόσο από τα πλεονεκτήματα που συγκεντρώνουν σε σχέση με τον κλασικό αυτοματισμό όσο και από τις δυνατότητες επικοινωνίας και δικτύωσης τους με άλλες μονάδες π.χ υπολογιστές, μηχανές, άλλα PLC ή και συστήματα π.χ αποθήκης, λογιστηρίου. Τα κύρια πλεονεκτήματα που χαρακτηρίζουν τους ελεγκτές PLC είναι :

- μικρό μέγεθος,στιβαρότητα/αντοχή κατάλληλη για βιομηχανικές εφαρμογές, απλότητα
- Ευελιξία στην τροποποίηση λειτουργίας των αυτοματισμών
- Εύκολη επέκταση αλλάζοντας το πρόγραμμα ή τοποθετώντας νέες μονάδες εισόδων/εξόδων
- Ενσωματωμένες αυτόματες λειτουργίες διάγνωσης σφαλμάτων
- Ενσωματωμένη, άμεσα διαθέσιμη τεκμηρίωση (επεξηγηματικά σχόλια, παρατηρήσεις κ.λ.π.)
- Εύκολη, πιο γρήγορη και σχεδόν άνευ κόστους αναπαραγωγή όμοιων εφαρμογών

5.1.2 Βασική αρχή λειτουργίας των PLC

Τα κύρια μέρη από τα οποία αποτελείται ένα PLC είναι :

- η κεντρική μονάδα επεξεργασίας CPU
- οι μονάδες εισόδων/εξόδων
- η μονάδα τροφοδοσίας

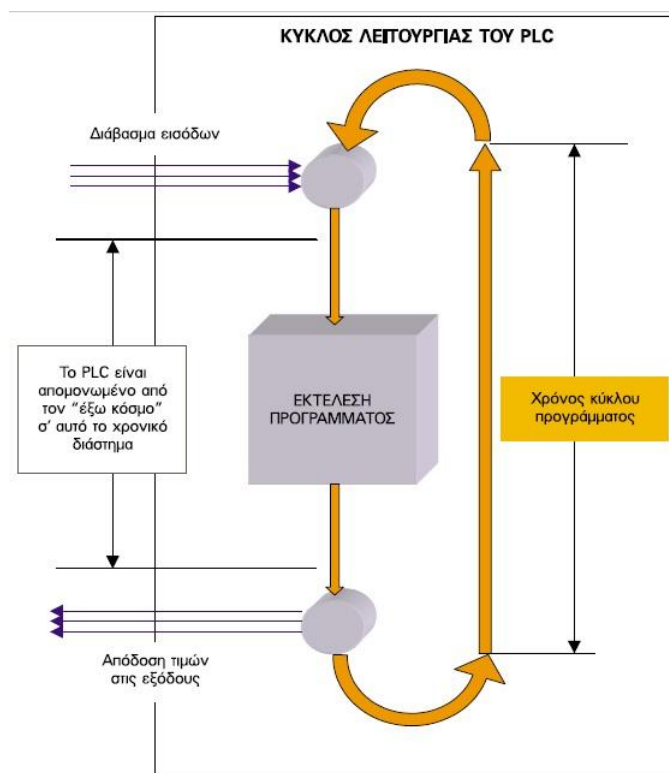
Τα PLC, όπως όλα τα ψηφιακά υπολογιστικά συστήματα, διαχειρίζονται και αποθηκεύουν πληροφορίες με τη μορφή δύο διαφορετικών καταστάσεων : ON και OFF (ή αλλιώς με τη μορφή του “λογικού” 1 και 0). Οι πληροφορίες δηλαδή αποτελούνται από “δυαδικά ψηφία” (Bits). Τα Bits σαν στοιχεία πληροφορίας χρησιμοποιούνται είτε αυτόνομα

(αναπαριστώντας όπως είπαμε τις καταστάσεις ON ή OFF) ή χρησιμοποιούνται σε συνδυασμό με συγκεκριμένη κωδικοποίηση αναπαριστώντας αριθμούς.

Ένα PLC μπορεί να “αντιληφθεί” και να επεξεργασθεί πληροφορίες είτε σε ψηφιακή μορφή (0-1 ή ON-OFF): π.χ. “πατημένο” ή “μη πατημένο” μπουτόν, τερματοδιακόπτης, κλειστή ή ανοιχτή επαφή κ.λ.π. ή σε αναλογική μορφή: πληροφορία από σύστημα μέτρησης στάθμης, θερμοκρασίας, πίεσης, βάρους κ.λ.π. Παρ’ όλα αυτά η CPU που είναι ένα ολοκληρωμένο ψηφιακό κύκλωμα (microchip) “αντιλαμβάνεται” μόνο ψηφιακά δυαδικά σήματα “ON” – “OFF” ή λογικά 1 και 0.

Κύκλος λειτουργίας PLC

Αρχικά το PLC διαβάζει τις εισόδους και τις αποθηκεύει σε μια ειδική περιοχή μνήμης που ονομάζεται εικόνα εισόδων. Στη συνέχεια εκτελείται το πρόγραμμα, συνδυάζοντας τις εντολές του προγράμματος και την κατάσταση των εισόδων. Μετά, το PLC εκτελεί εσωτερικές διαγνωστικές λειτουργίες και λειτουργίες επικοινωνιών. Τέλος, αποδίδει τα αποτελέσματα στις εξόδους. Σε έναν κύκλο λειτουργίας του PLC υπάρχει ένα χρονικό διάστημα στο οποίο το πρόγραμμα «τρέχει» με βάση τις εισόδους που έχει διαβάσει και δεν έχει καμία επαφή με τις αλλαγές που γίνονται στις εισόδους ή στις εξόδους. Βεβαίως, ο χρόνος αυτός είναι τόσο μικρός, που στις περισσότερες εφαρμογές δεν προλαβαίνουν να αλλάξουν οι είσοδοι και οι έξοδοι ή να επηρεαστεί η λειτουργία του προγράμματος από αυτές τις αλλαγές. Ενδεικτικά αναφέρεται ότι στα πιο αργά PLC ο χρόνος κύκλου του δεν ξεπερνά τα μερικά εκατοντάδες milliseconds (ένας σχετικά αργός χρόνος κύκλου είναι 300ms). Με αυτό το τρόπο ολοκληρώνεται ένας κύκλος λειτουργίας και η διαδικασία επαναλαμβάνεται συνεχώς όσο το PLC βρίσκεται σε κατάσταση RUN.



Απεικόνιση κύκλου λειτουργίας PLC

Ο χρόνος που απαιτείται για να εκτελεστεί ένας πλήρης κύκλος λειτουργίας του PLC λέγεται χρόνος κύκλου και εξαρτάται από:

- την ταχύτητα της CPU
- από το πλήθος των εισόδων και εξόδων
- από τον αριθμό και το είδος εντολών του προγράμματος
- από τον όγκο των επικοινωνιών που πρέπει να υλοποιηθούν

5.1.3 Προγραμματισμός PLC

Για τη λειτουργία ενός PLC απαιτείται πέρα από το hardware, και το software μέσω του οποίου δημιουργείται το πρόγραμμα που εκτελεί τους αυτοματισμούς. Για τη δημιουργία του προγράμματος χρησιμοποιείται κάποια συσκευή προγραμματισμού. Η συσκευή προγραμματισμού είναι τελείως ξεχωριστή συσκευή από τον λογικό ελεγκτή και μπορεί να είναι :προγραμματιστής χειρός, συσκευή προγραμματισμού, ή προσωπικός υπολογιστής (PC).

Το πρόγραμμα γράφεται σε γλώσσες προγραμματισμού (συγκεκριμένα συμβολικές γλώσσες ή διαγράμματα), οι οποίες έχουν καθορισθεί και τυποποιηθεί από το πρότυπο IEC1131-3. Οι γλώσσες και το λογισμικό προγραμματισμού διαφέρουν από εταιρεία σε εταιρεία (διατηρούν ωστόσο πολλά κοινά στοιχεία). Οι κύριες γλώσσες που χρησιμοποιούνται για τον προγραμματισμό των PLC είναι:

- Ladder .Η γλώσσα λογικών διαγραμμάτων τύπου σκάλας ή γλώσσα ηλεκτρολογικών γραφικών, μοιάζει με το διάγραμμα ενός κυκλώματος. Ο χρήστης χρησιμοποιεί σύμβολα όπως επαφές και πηνία και ήταν η γλώσσα που προωθήθηκε στην αρχή των PLC διότι ήταν πιο φιλική στους μηχανικούς που είχαν συνηθίσει να δουλεύουν πάνω σε ένα ηλεκτρολογικό σχέδιο. Η γλώσσα Ladder δεν χρησιμοποιεί την ευρωπαϊκή προτυποποίηση στο σχεδιασμό των ηλεκτρικών επαφών, αλλά την αμερικάνικη.
- STL (statement list) Η γλώσσα αυτή αποτελείται από εντολές και μοιάζει με γλώσσα μηχανής Assembly. Για αυτό το λόγο προτιμάται από χρήστες που είναι συνηθισμένοι σε άλλες γλώσσες προγραμματισμού. Η STL έχει εξελιχθεί αρκετά σε δυνατότητες προγραμματισμού και ο προγραμματισμός μπορεί να γίνει τόσο ελεύθερα με την STL που μερικές φορές είναι δύσκολο για κάποιον να παρακολουθήσει τη ροή του προγράμματος.
- FBD (function block diagram) Η γλώσσα FBD ή γλώσσα λογικών γράφων, είναι γραφική σαν τη Ladder, αλλά χρησιμοποιεί «κουτιά», όπου κάθε κουτί αντιπροσωπεύει κάποια επιμέρους λειτουργία και τελικά το αντίστοιχο λογικό κύκλωμα.

Αφού δημιουργηθεί το πρόγραμμα σε μία από τις συσκευές προγραμματισμού που προαναφέρθηκαν, ο χρήστης το εγκαθιστά στο PLC και όταν αυτό βρεθεί σε κατάσταση λειτουργίας run, πραγματοποιείται η εκτέλεση του προγράμματος.

5.1.4 Σύνδεση θεωρίας δικτύων Petri με τη λογική των PLC

Κατά τη σχεδίαση ελέγχου συστημάτων διακριτών γεγονότων είναι πιθανόν να αγνοηθούν σημαντικά σημεία αλληλεπίδρασης οδηγώντας σε εσφαλμένη λειτουργία κατά την εκτέλεσή τους. Ιδιαίτερα σε περιπτώσεις που χαρακτηρίζονται από κατανομή πόρων, συγκρούσεις, αμοιβαίο αποκλεισμό και γενικά αυξημένη πολυπλοκότητα είναι πολύ σημαντικό να παρέχονται μέθοδοι που επιτρέπουν τον έλεγχο και την επιδιόρθωση πριν από τη εφαρμογή του προγράμματος.

Το λογισμικό των σύγχρονων PLC, παρόλο που διαθέτει ενσωματωμένες αυτόματες διαγνωστικές και προσομοιωτικές λειτουργίες για την αντιμετώπιση σφαλμάτων, δε παρέχει κάποιο εργαλείο για τον υπολογισμό των ιδιοτήτων συστημάτων διακριτών γεγονότων και την απεικόνιση όλων των δυνατών καταστάσεων ενός συστήματος [9].

Έχει παρατηρηθεί ότι ένα πρόγραμμα σε γλώσσα PLC μπορεί να μετατραπεί σε ένα δίκτυο Petri και αντίστροφα. Πρόσφατες εργασίες περιλαμβάνουν την μετατροπή δικτύων Petri σε γλώσσα λογικών διαγραμμάτων για την υλοποίηση PLCs. Με αυτόν τον τρόπο, το πρόγραμμα που προκύπτει διατηρεί τις πολύ σημαντικές ιδιότητες που έχουν διερευνηθεί μέσω της προηγούμενης μορφής του και ταυτόχρονα είναι άμεσα υλοποιήσιμο σε εμπορικά διαθέσιμους ελεγκτές [17].

Συνεπώς, παρουσιάζει ενδιαφέρον η σύνδεση ενός τέτοιου εργαλείου με την υπάρχουσα τεχνολογία των PLC. Βέβαια, τα μοντέλα Petri δεν εμπεριέχουν τον τρόπο με τον οποίο ο έλεγχος αλληλεπιδρά με τους αισθητήρες ή τις εισόδους και τους επενεργητές ή εξόδους του συστήματος. Για το λόγο αυτό, η μετατροπή των μοντέλων PN σε γλώσσα PLC και η εφαρμογή τους στο πραγματικό σύστημα απαιτεί την ενσωμάτωση επιπρόσθετων πληροφοριών που αφορούν τα σήματα πραγματικού χρόνου (real-time signals) [5].

Στη συνέχεια της εργασίας πραγματοποιείται, βάσει βιβλιογραφίας, μια προσπάθεια μετάφρασης του μοντελοποιημένου CPN συστήματος, που παρουσιάστηκε στο προηγούμενο κεφάλαιο, σε γλώσσα Ladder. Κατόπιν, εξετάζεται η ορθότητα του συνολικού προγράμματος που προκύπτει για διάφορα σενάρια λειτουργίας του συστήματος χρησιμοποιώντας εργαλείο προσομοίωσης του λογισμικού *Cx-programmer* της εταιρίας *Omron*.

5.2 Βιβλιογραφική μελέτη

Στη βιβλιογραφία υπάρχουν αρκετές εργασίες που περιλαμβάνουν την μετατροπή δικτύων Petri σε λογικά διαγράμματα τύπου σκάλας (LLD) για την υλοποίηση PLCs. Παρακάτω παρουσιάζονται οι πιο σημαντικές από αυτές στις οποίες βασίστηκε και η μετατροπή του μοντέλου CPN της παρούσας εργασίας σε διάγραμμα Ladder.

Petri Nets and Ladder Logic for Fully Automating and Programmable Logic Control of Semi-Automatic Machines and Systems (*Mahmoud A. Barghash, Osama M.Abuzeid, Anas N.Al-Rabadi, Ahmad M.Jaradat*)

Οι Mahmoud A. Barghash, Osama M.Abuzeid, Anas N.Al-Rabadi, Ahmad M.Jaradat παρουσιάζουν μια περιπτωσιολογική μελέτη που αφορά τον έλεγχο ενός ημιαυτόματου βιομηχανικού συστήματος ανάμιξης. Όπως αναλύεται, κατά την αυτοματοποίηση ημιαυτόματων μηχανών, τα plc μπορεί να μην εξασφαλίζουν τον πλήρη έλεγχο των λειτουργιών. Για το λόγο αυτό εξετάζονται τρεις διαφορετικές προσεγγίσεις αναφορικά με την αλληλεπίδραση των ημιαυτόματων και αυτόματων λειτουργιών. Με τη βοήθεια ενός μοντέλου βασισμένο στα δίκτυα Petri, παρουσιάζονται επιτυχώς και οι δύο κατηγορίες λειτουργιών. Παράλληλα δημιουργείται και το αντίστοιχο διάγραμμα ladder. Με τα δύο αυτά εργαλεία αξιολογούνται οι διαφορετικές προσεγγίσεις με βάση την αξιοπιστία, το κόστος και την απόδοση του συστήματος.

Translation of Extended Petri Net Model into Ladder Diagram and Simulation with PLC (*Tomaz Perme*)

Ο Tomaz Perme ασχολείται με τη δημιουργία και προσομοίωση μοντέλων μηχανοτρονικών συστημάτων διακριτών γεγονότων. Για την παρουσίαση της μεθοδολογίας χρησιμοποιείται το παράδειγμα ελέγχου της κίνησης ενός πνευματικού κυλίνδρου διπλής διαδρομής. Η αρχική δημιουργία του μοντέλου βασίζεται στα δίκτυα Petri. Εφόσον επιβεβαιωθεί η σωστή λειτουργία και ακρίβεια του μοντέλου PN, παρουσιάζονται αναλυτικά οι κανόνες μετατροπής του μοντέλου PN σε διάγραμμα ladder και εφαρμόζονται για το συγκεκριμένο παράδειγμα. Το διάγραμμα ladder που προκύπτει προσομοιώνεται σε δύο προγραμματιστικές πλατφόρμες PLC (Cx-Simulator της Omron και 7SP LCSIM της Siemens). Κατά την προσομοίωση αυτή μπορούμε να δούμε τη μεταβολή των σημάτων εισόδου /εξόδου καθόλη τη διάρκεια της λειτουργίας του εμβόλου.

A transcription tool from Petri Net to CLP Programming Languages (*Andre Torres Ferraz de Mello, Marcelo Castro Barbosa, Diolino Jose dos Santos Filho, Paulo Eigi Miyagi, Fabricio Junqueira*)

Οι Andre Torres Ferraz de Mello, Marcelo Castro Barbosa, Diolino Jose dos Santos Filho, Paulo Eigi Miyagi, Fabricio Junqueira προτείνουν ένα εργαλείο μετατροπής των μοντέλων PN σε τρεις γλώσσες προγραμματισμού PLC του προτύπου IEC (Ladder Logic Diagram, Sequential Function Chart και Structured Text). Χρησιμοποιείται ένα παράδειγμα

προγραμματισμού ενός απλού βιομηχανικού συστήματος ανάμιξης. Το σύστημα μοντελοποιείται αρχικά ως δίκτυο Petri και πιο συγκεκριμένα στη μορφή SIPN (Signal Interpreted Petri Nets). Η μορφή αυτή αντιστοιχίζει τις μεταβλητές εισόδου (αισθητήρες, button) στις μεταβάσεις και τις μεταβλητές εξόδου (επενεργητές, λυχνίες ένδειξης) στις θέσεις του δικτύου Petri. Ακολουθώντας τη διαδικασία που περιγράφεται αναλυτικά το ίδιο μοντέλο PN μετατρέπεται στις μορφές Ladder, SFC και STL.

Petri Nets Based Design of Ladder Logic diagrams (*J-L Chirn and D.C. McFarlane*)

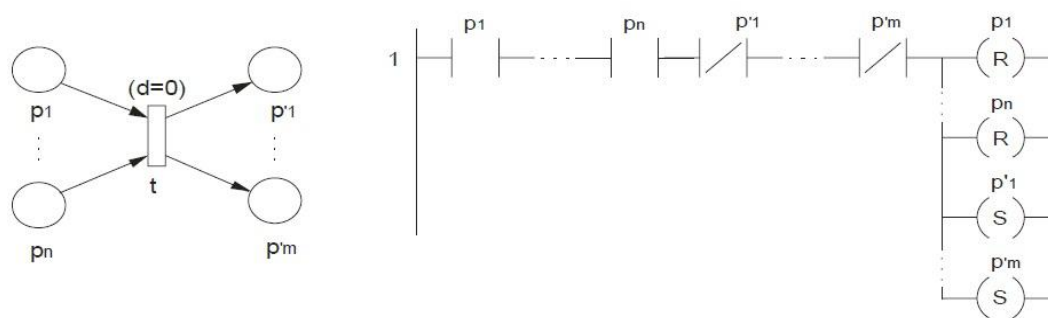
Οι J-L Chirn and D.C. McFarlane παρουσιάζουν μια συστηματική προσέγγιση σχεδιασμού διαγραμμάτων Ladder η οποία χρησιμοποιεί μια μοντελοποίηση με δίκτυα Petri. Αρχικά παρουσιάζεται μια διερευνητική μέθοδος, με βάση την οποία, ένα πρωταρχικό δίκτυο Petri απλής μορφή μετασχηματίζεται σε ένα rung ενός διαγράμματος ladder. Η μέθοδος αυτή τροποποιείται ώστε να μπορεί να εφαρμοστεί σε διάφορους τύπους PLC (horizontal /vertical scanning). Η ίδια λογική επεκτείνεται και σε πιο σύνθετες δομές PN ενώ παρουσιάζονται και κάποιες απλοποιημένες εκδοχές μετατροπής που εκφράζουν συγκεκριμένες περιπτώσεις. Στη συνέχεια, η συγκεκριμένη εργασία ασχολείται με τη σύνδεση των, βασισμένων σε δίκτυα PN, διαγραμμάτων LLD με το εξωτερικό περιβάλλον που πρόκειται να ελεγχθεί από το PLC. Οι εισοδοί στο PLC σχετίζονται με τις μεταβάσεις του PN ενώ οι εξοδοί με τις θέσεις του. Για την παρουσίαση της μεθοδολογίας μετατροπής PN-LLD χρησιμοποιείται ένα παράδειγμα σταθμού ελέγχου σε μεταφορική ταινία προϊόντων. Στο παράδειγμα αυτό παρατίθενται και πίνακες που απεικονίζουν την αντιστιχεία εισόδων-μεταβάσεων PN εξόδων-θέσεων PN. Τέλος, πραγματοποιείται μια αξιολόγηση της προτεινόμενης μεθόδου και σύγκρισή της, βάσει συγκεκριμένων κριτηρίων, με μεθόδους άλλων ερευνητών.

Automatic generation of ladder diagram with control Petri Net (*Gi Bum Lee, Han Zandong, Jin S. Lee*)

Οι Gi Bum Lee, Han Zandong, Jin S. Lee παρουσιάζουν μια αυτοματοποιημένη μέθοδο (μέσω άλγεβρας Boole) δημιουργίας διαγραμμάτων Ladder από δίκτυα Petri. Για να μπορεί να πραγματοποιηθεί σωστά η μετατροπή αυτή, θα πρέπει τα δίκτυα Petri να βρίσκονται υπό συγκεκριμένη μορφή Control Petri Nets όπως αναφέρεται. Η μορφή αυτή αντιστοιχεί σε συγκεκριμένα χαρακτηριστικά και ιδιότητες των δικτύων Petri. Η μέθοδος συσχετίζει τις λειτουργίες του συστήματος με τις θέσεις του PN και τις συνθήκες με τις μεταβάσεις του. Παρουσιάζεται μια γενική συνάρτηση άλγεβρας Boole η οποία εκφράζει τη σχέση μεταξύ θέσεων συνθηκών και γεγονότων. Η συνάρτηση αυτή εφαρμόζεται σε κάθε θέση του Petri και παράγει μια ισότητα μεταβλητών Boole από την οποία μπορεί εύκολα να προκύψει το αντίστοιχο rung του LLD. Η διαδικασία αυτή παρουσιάζεται αναλυτικά με τη βοήθεια δύο παραδειγμάτων 1) κίνηση αντικειμένου ελεγχόμενη από button και διακόπτες στάσης 2) ανάμιξη υγρών σε δεξαμενή και γέμισμα μπουκαλιών. Ακόμα παρουσιάζεται και η ενσωμάτωση χρονιστών και μετρητών στα μοντέλα PN αλλά και η μέθοδος μετασχηματισμού τους στο LLD.

5.3 Μετατροπή μοντέλου CPN παρούσας εργασίας σε διάγραμμα Ladder

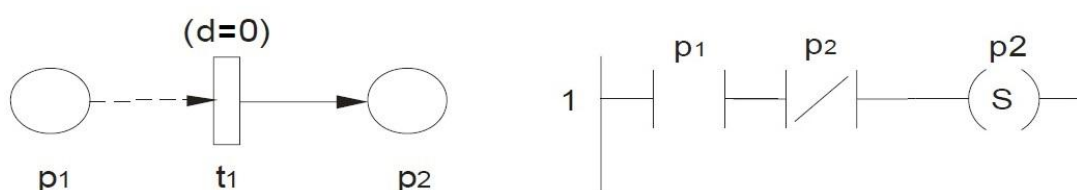
Για τη μετατροπή του διαγράμματος PN σε ladder χρησιμοποιήθηκε η μεθοδολογία που αναλύθηκε στην εργασία του Tomaž Perme. Σύμφωνα με τη συγκεκριμένη μεθοδολογία από κάθε μετάβαση του Petri Net μπορεί να προκύψει ένα ισοδύναμο rung σε ένα ladder diagram σύμφωνα με τους παρακάτω κανόνες :



Εικόνα 1 : Γενικός κανόνας μετατροπής δικτύων Petri σε διαγράμματα ladder

Πηγή: *Translation of Extended Petri Net Model into Ladder Diagram and Simulation with PLC*, Tomaž Perme

Για την περίπτωση διπλών τόξων (informational arcs) ο κανόνας παίρνει την παρακάτω μορφή. Για τη θέση που διαθέτει διπλό τόξο (θέση p1 στην παρακάτω εικόνα), αν και είναι απαραίτητο να διαθέτει token για να ενεργοποιηθεί η μετάβαση, ο αριθμός των tokens δε μεταβάλλεται μετά την ενεργοποίησή της. Για το λόγο αυτό δε θα πρέπει να υπάρχει reset για τη συγκεκριμένη θέση) [13].



Εικόνα 2 : Περίπτωση μετάφρασης διπλών τόξων του δικτύου Petri στα διαγράμματα ladder

Πηγή: *Translation of Extended Petri Net Model into Ladder Diagram and Simulation with PLC*, Tomaž Perme

Ωστόσο , οι κανόνες αυτοί αφορούν απλά μοντέλα Petri Nets. Θα μπορούσαν όμως να επεκταθούν και για την κατηγορία των Coloured Petri Nets εφόσον ο παραπάνω γενικός κανόνας εφαρμοστεί για όλες τις περιπτώσεις που ανάγονται σε κάθε μετάβαση .

Όπως αναφέρθηκε και στην ενότητα 5.1 κατά τη μετατροπή των δικτύων Petri σε διάγραμμα ladder θα πρέπει να ενσωματωθούν σήματα πραγματικού χρόνου. Αυτό απαιτεί

ορισμένες τροποποιήσεις μετά από τη μετατροπή PN-Ladder όπως αυτή παρουσιάζεται στη βιβλιογραφία. Αυτές οι τροποποιήσεις μαζί με κάποιες ακόμα προσθήκες που αφορούν την ειδική περίπτωση των CPN συνοψίζονται παρακάτω :

- Κατά τη μετατροπή έγινε προσπάθεια να αντιστοιχίζεται σε μία από τις εισερχόμενες θέσεις της μετάβασης ένα σήμα εισόδου (από το ρομπότ, τις μηχανές κατεργασίας ή την κάμερα ταυτοποίησης) το οποίο είναι καθοριστικό για τη λήξη ή την έναρξη μιας εργασίας.
- Τα στοιχεία T_{av} ,Anti_T_{av} , που υπήρχαν στο CPN μοντέλο ,δίνουν πληροφορία για το ίδιο πράγμα με αντίστροφο τρόπο. Το ίδιο ισχύει και για τα M_{av} ,Anti_M_{av}. Επομένως, τα στοιχεία Anti_T_{av} και Anti_M_{av} είναι περιττά στο διάγραμμα ladder γι' αυτό και παραλείπονται εξ' αρχής σε όλες τις παρακάτω μετατροπές.
- Για τις μεταβάσεις που συνδέονται με τη θέση **Buffer Capacity** πρέπει να χρησιμοποιηθούν βοηθητικές μνήμες στο Ladder (μία για κάθε Buffer) οι οποίες να συνδέονται με την ενεργοποίηση Counters. Αυτό εξηγείται ως εξής :

Όλες οι υπόλοιπες θέσεις στο μοντέλο του CPN έχουν αρχική σήμανση 1token για κάθε "χρώμα". Για παράδειγμα η **Robot Availability** έχει 1'R_{av} ενώ η **Machine Availability** έχει 1'M ++1'T. Αυτό σημαίνει ότι για οποιαδήποτε κατάσταση του συστήματος κάθε χρώμα θα έχει δύο καταστάσεις (1token ή 0token) . Γι' αυτό και αυτές οι καταστάσεις μπορούν να αναπαρασταθούν στο ladder ως μια μνήμη ON/OFF. Η θέση **Buffer Capacity** όμως η οποία έχει αρχική σήμανση 4'B_t++ 4'B_m έχει 5 διαφορετικές καταστάσεις για κάθε χρώμα (δηλαδή για κάθε Buffer). Πιο συγκεκριμένα, για το buffer του κέντρου τόννευσης η θέση Buffer Capacity μπορεί να έχει : 0` B<sub>t</sub>, 1` B_t , 2`B<sub>t</sub>, 3`B_t , 4`B_t. Αυτό στο ladder μπορεί να υλοποιηθεί με χρήση Counters σε συνδυασμό με εντολές σύγκρισης. Για να υπάρχει ανανέωση του counter κατά την φόρτωση και εκφόρτωση τεμαχίων στο buffer χρησιμοποιήθηκαν για όλες τις περιπτώσεις reversible counters.

Στη συγκεκριμένη εφαρμογή ο ελεγκτής θα πρέπει να 'γνωρίζει' :

- το συνολικό αριθμό τεμαχίων στα buffer_t και Buffer_m (Counters C0006 και C0007 αντίστοιχα και εντολές σύγκρισης που ενεργοποιούν τις βοηθητικές μνήμες B_t_Free, B_t_1, B_t_2 ,B_t_3, B_t_4, B_m_Free, B_m_1, B_m_2 ,B_m_3, B_m_4)
- ποια από τις τέσσερις θέσεις του buffer καταλαμβάνει κάθε τεμάχιο καθώς και τον τύπο και φάση κάθε αποθηκευμένου τεμαχίου. Για το λόγο αυτό χρησιμοποιούνται όλες οι παρακάτω βοηθητικές μνήμες που δίνουν όλους τους πιθανούς συνδυασμούς :

B_t1_E1,B_t1_F1,B_t1_C2,B_t2_E1,B_t2_F1,B_t2_C2,B_t3_E1,B_t3_F1,B_t3_C2,B_t4_E1,B_t4_F1,
B_t4_C2,

Bm1_B1,Bm1_C1,Bm1_F2, Bm2_B1,Bm2_C1,Bm2_F2, Bm3_B1,Bm3_C1,Bm3_F2,
Bm4_B1,Bm4_C1,Bm4_F2,

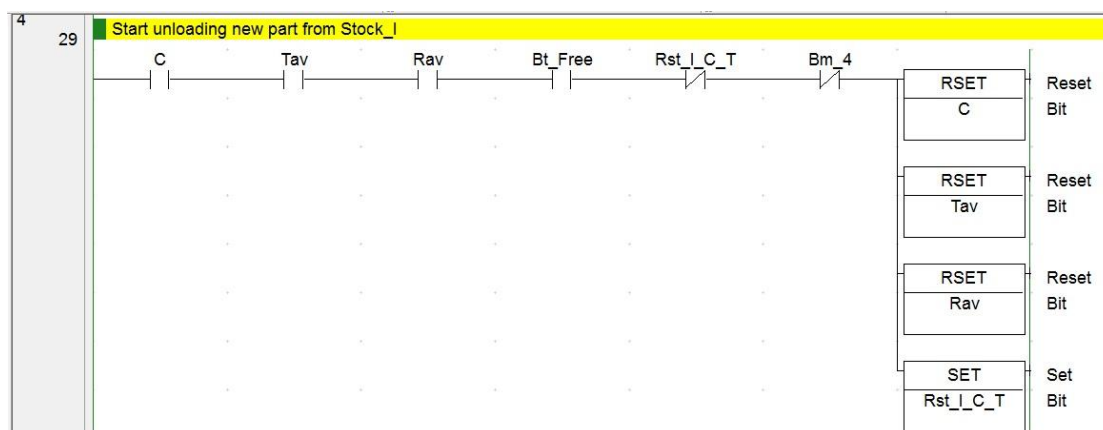
Παρακάτω παρουσιάζεται η μετατροπή κάθε μετάβασης σε rung. Το πρόγραμμα ladder δημιουργήθηκε και προσομοιώθηκε στο λογισμικό Cx-Programmer της Omron. Για κάθε μετάβαση δίνεται ένα παράδειγμα και συγκεκριμένα για το τεμάχιο C. Οι υπόλοιπες περιπτώσεις που ανάγονται στην ίδια μετάβαση γίνονται με τον ίδιο τρόπο. Στο *παράρτημα* της εργασίας δίνονται όλες οι εισοδοί, έξοδοι και εσωτερικές μνήμες με τα αντίστοιχα ονόματα και επεξηγήσεις που έχουν χρησιμοποιηθεί στο πρόγραμμα.

▪ **Μετάβαση 1 : Start unloading new part from Stock_I**

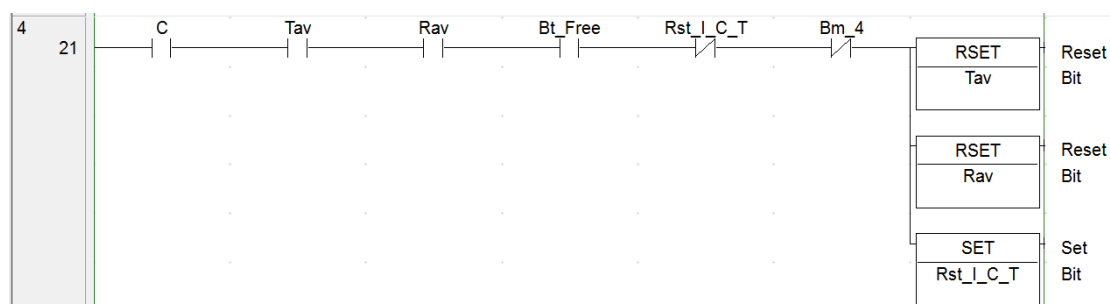
Τα δύο στοιχεία (βοηθητικές μνήμες) Bt_Free και Bm_4 σε αυτό το rung αντιστοιχούν στην πληροφορία που δίνεται στη μετάβαση από τη θέση του Buffer Capacity στο μοντέλο CPN: $\text{case } r \text{ of } tr \Rightarrow 4 \cdot Bt ++ 1 \cdot Bm$ (για να εισέλθει στο σύστημα τεμάχιο με 1^η κατεργασία τόννευση και 2^η κατεργασία φρεζάρισμα, θα πρέπει να υπάρχουν 4 κενές θέσεις στο buffer του τόννου (άδειο buffer)** και μία κενή θέση στο buffer της φρέζας σε περίπτωση που χρειαστεί στη συνέχεια ενδιάμεση αποθήκευση).

(** Υπενθυμίζεται ότι αυτό γίνεται για να δώσουμε προτεραιότητα στα τεμάχια που βρίσκονται μέσα στο σύστημα)

Για τα στοιχεία Bt_Free και Bm_4 δε χρησιμοποιείται reset καθώς στο Petri Net η αντίστοιχη θέση εισέρχεται με διπλό τόξο στη μετάβαση.



Rung μετάβασης **Start unloading new part from Stock_I** για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα



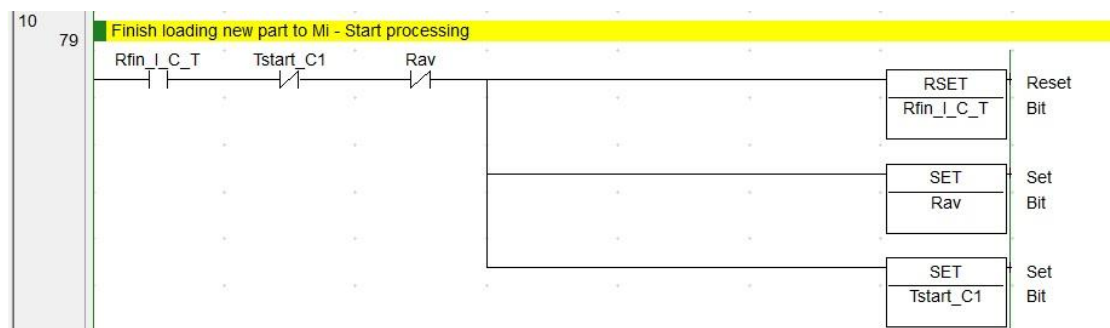
Τροποποιημένο Rung μετάβασης **Start unloading new part** για την περίπτωση του τεμαχίου C

▪ Μετάβαση 2: Finish loading new part to Mi - Start processing

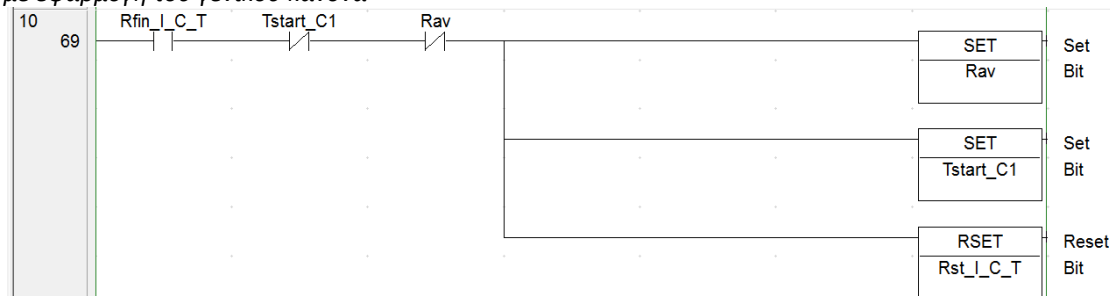
Σε αυτή τη φάση, αφού εφαρμόσουμε τον κανόνα μετατροπής διαπιστώνουμε μια σημαντική παράλληλη που δημιουργείται. Παρατηρείται ότι κατά τη ροή του προγράμματος από την μετάβαση **Start loading new part** και **Finish loading new part – Start processing** δε γίνεται reset στην εντολή **Rst_I_C_T**.

Τα τόξα εισόδου και εξόδου στη θέση **Load Mi**, αν και αναφέρονται στην ίδια λειτουργία του ρομπότ, αντιστοιχούν σε δύο διαφορετικά σήματα: το σήμα εξόδου-εντολή προς το ρομπότ για να ξεκινήσει η μεταφορά (**Rst_I_C_T**) και το σήμα εισόδου από το ρομπότ για να ειδοποιηθεί ο ελεγκτής ότι η λειτουργία έχει τελειώσει (**Rfin_I_C_T**)

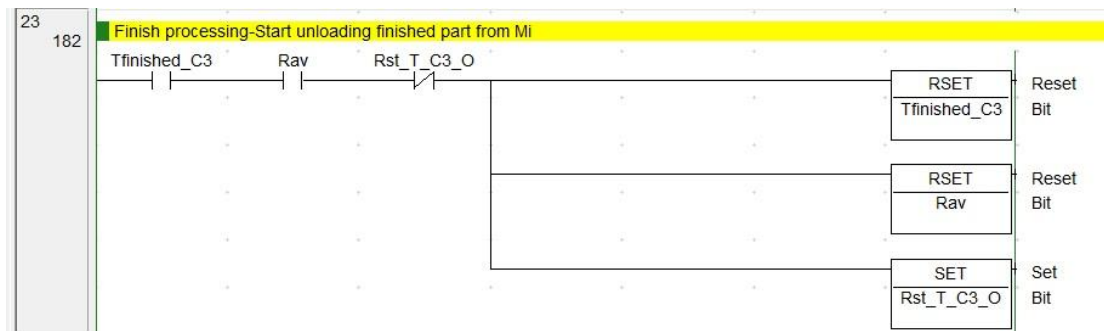
Για το λόγο αυτό βάζουμε το **Rfin_I_C_T** να κάνει reset στο **Rst_I_C_T**



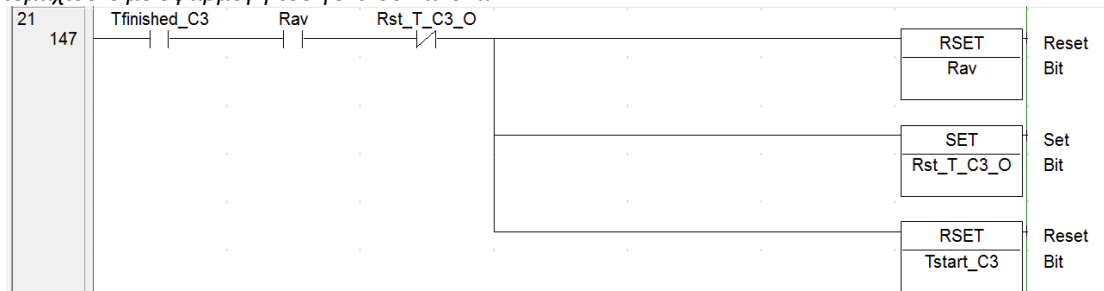
Rung μετάβασης **Finish loading new part to Mi –Start processing** για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα



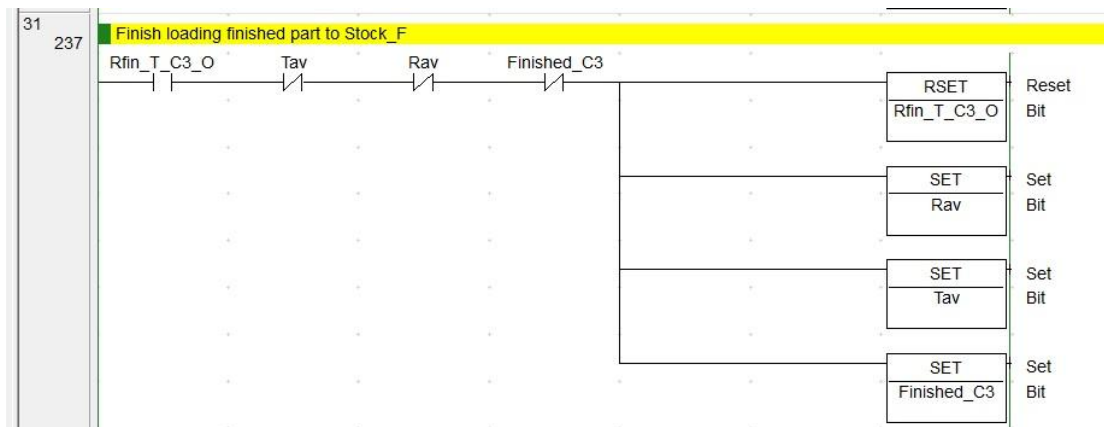
Τροποποιημένο Rung μετάβασης **Finish loading new part to Mi –Start processing** για την περίπτωση του τεμαχίου C

Μετάβαση 3 : Finish processing - Start unloading finished part from Mi

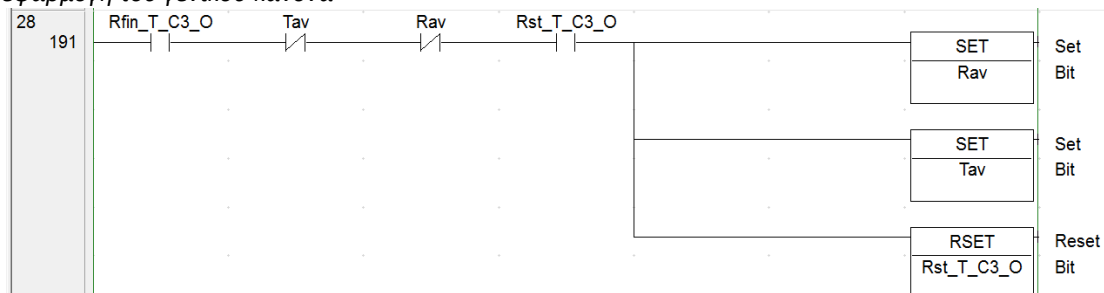
Runγ μετάβασης **Finish processing- Start unloading finished part from Mi** για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα



Τροποποιημένο Runγ μετάβασης **Finish processing- Start unloading finished part from Mi** για την περίπτωση του τεμαχίου C

Μετάβαση 4: Finish loading finished part to Stock_F

Runγ μετάβασης **Finish loading finished part to Stock_F** για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα

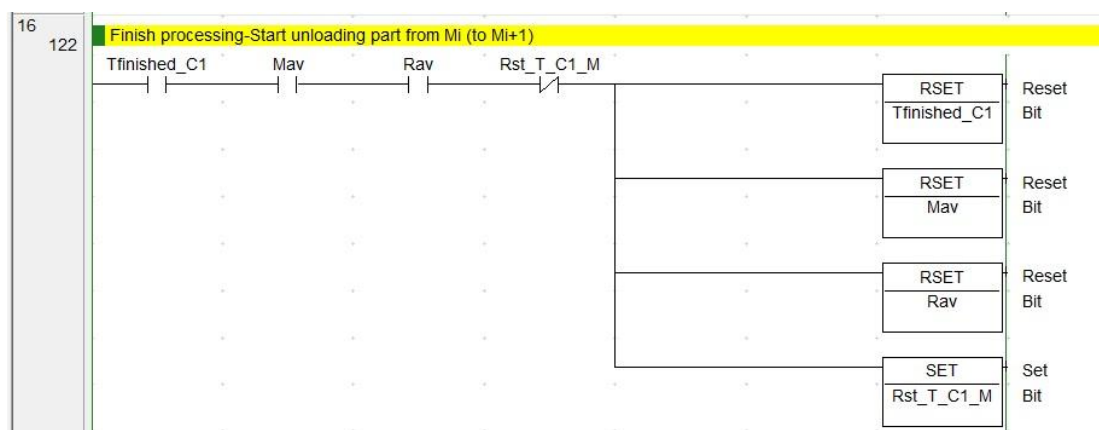


Τροποποιημένο Runγ μετάβασης **Finish loading finished part to Stock_F** για την περίπτωση του τεμαχίου C

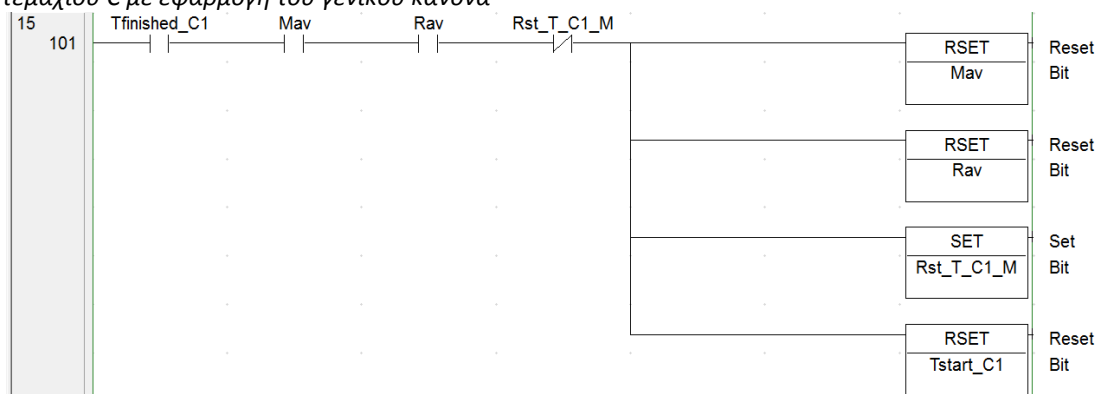
▪ Μετάβαση 5 : Finish processing - Start unloading part from Mi (to Mi+1)

Και εδώ παρατηρούμε ότι εφαρμόζοντας τη μετατροπή δε γίνεται μέσα στο πρόγραμμα reset στην εντολή **Tstart_C1** που βρίσκεται ενεργοποιημένη από το προηγούμενο στάδιο.

Στο διορθωμένο rung προσθέτουμε το reset της εντολής **Tstart_C1**

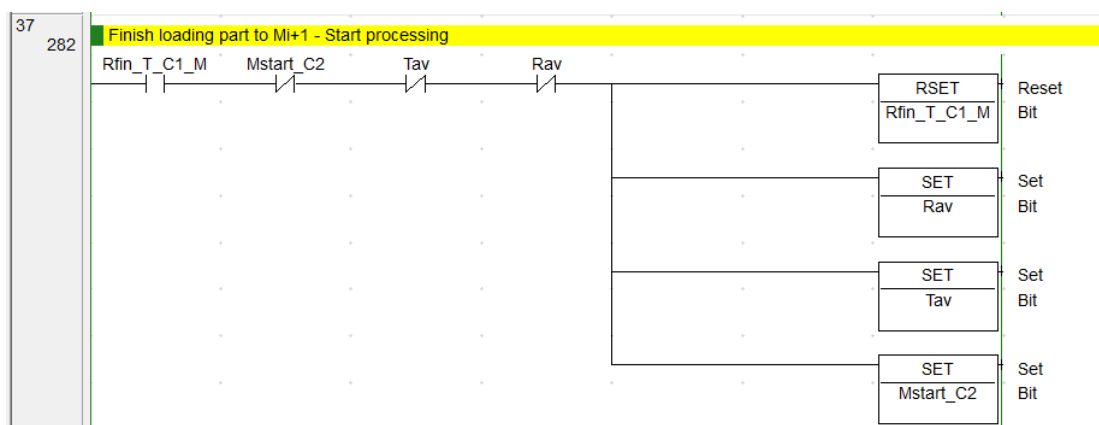


Rung μετάβασης **Finish processing - Start unloading part from Mi (to Mi+1)** για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα

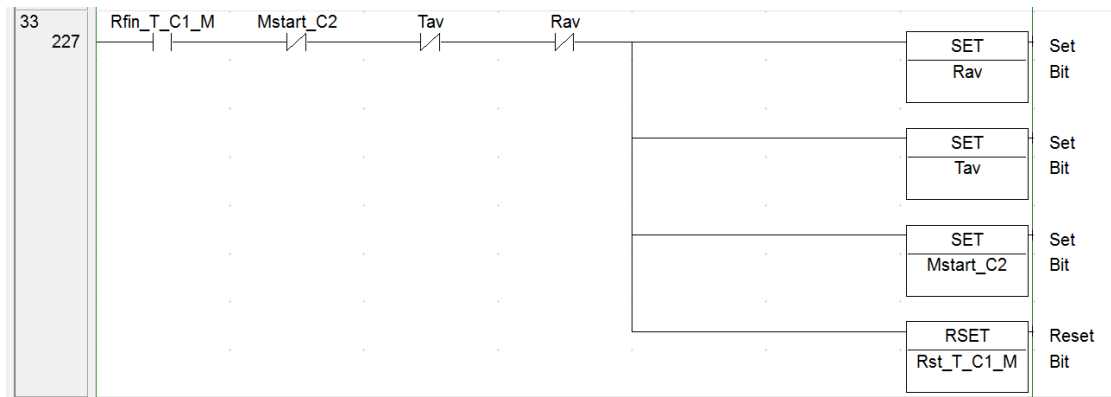


Τροποποιημένο Rung μετάβασης **Finish processing - Start unloading part from Mi (to Mi+1)** για την περίπτωση του τεμαχίου C

Μετάβαση 6 : Finish loading part to Mi+1 - Start processing

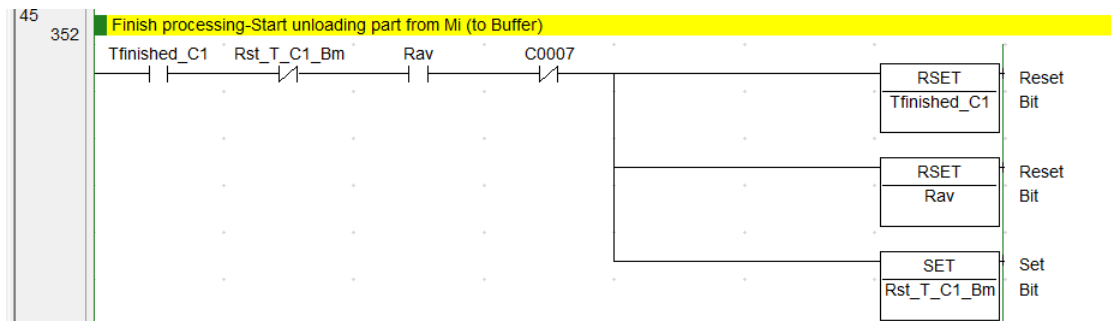


Rung μετάβασης **Finish loading part to Mi+1 - Start processing** για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα

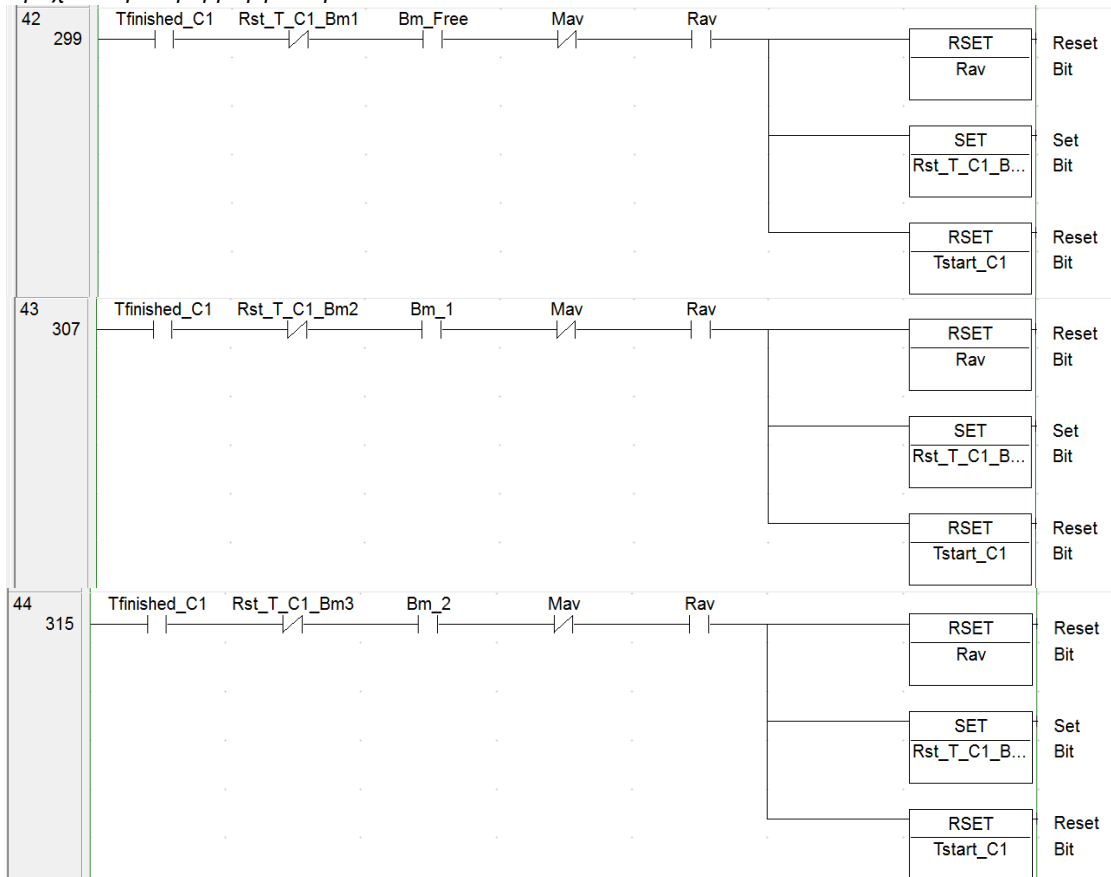


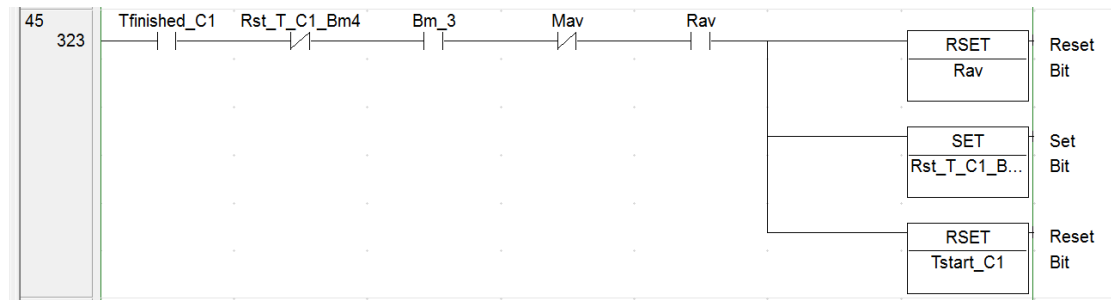
Τροποποιημένο Run μετάβασης *Finish loading part to Mi+1 - Start processing* για την περίπτωση του τεμαχίου C

Μετάβαση 7 : Finish processing - Start unloading part from Mi (to Buffer)



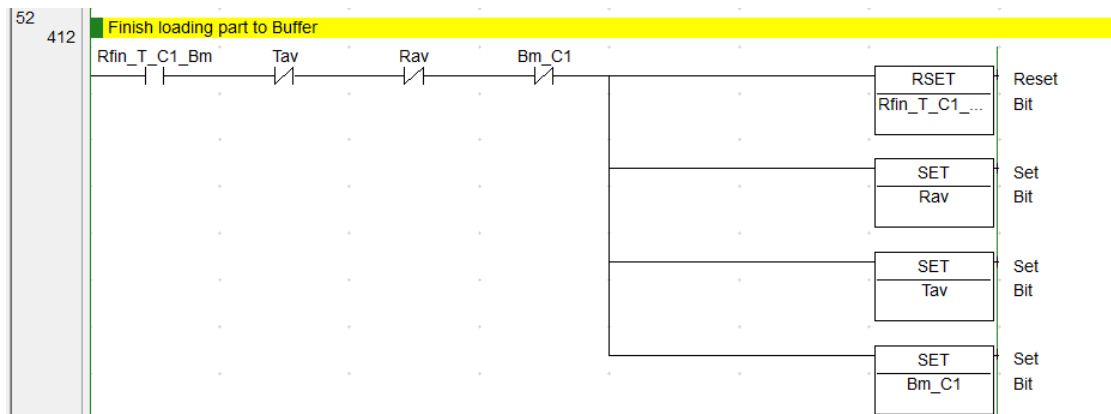
Run μετάβασης *Finish processing – Start unloading part from Mi (to Buffer)* για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα



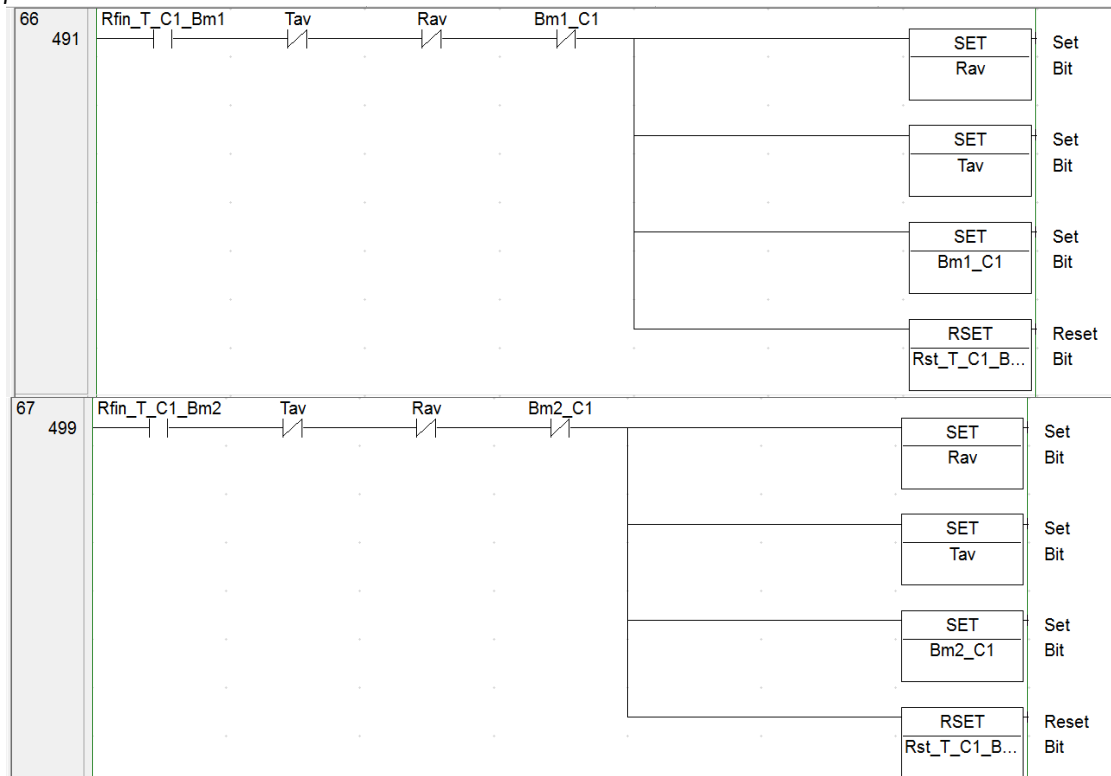


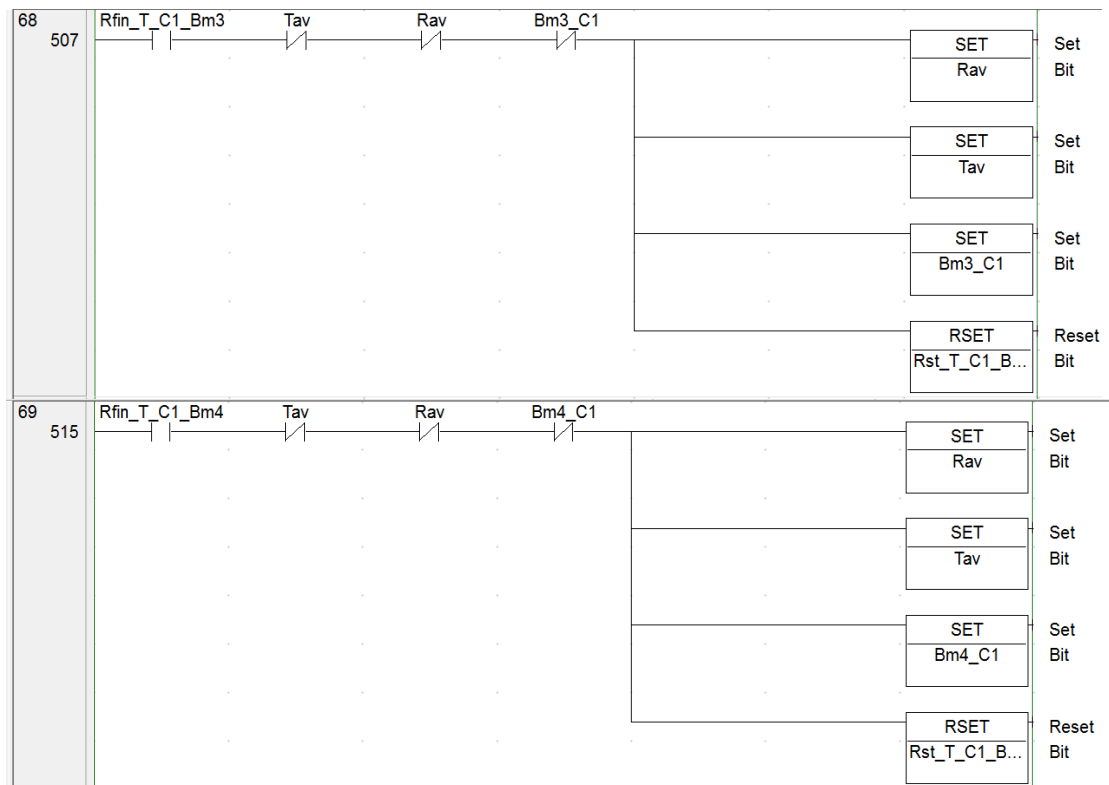
Τροποποιημένα Rungs μετάβασης **Finish processing – Start unloading part from Mi (to Buffer)** για την περίπτωση του τεμαχίου C

Μετάβαση 8: Finish loading part to Buffer



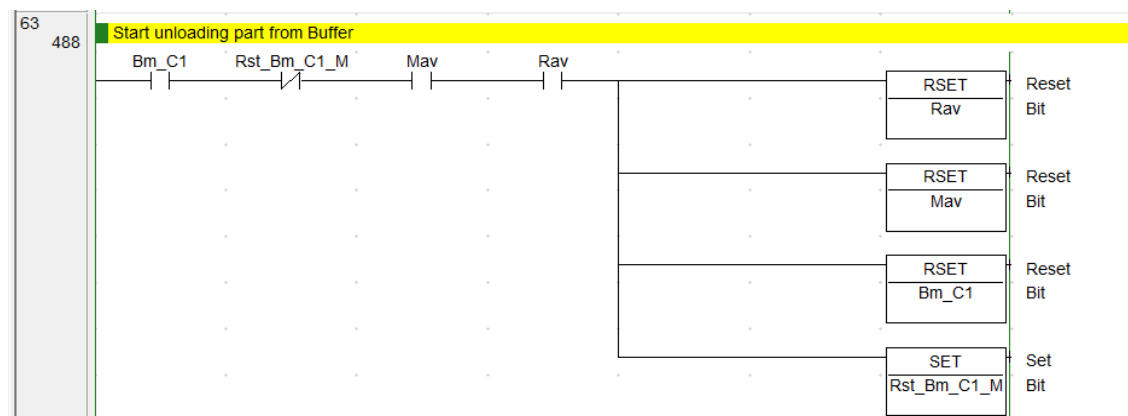
Rung μετάβασης **Finish loading part to Buffer** για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα



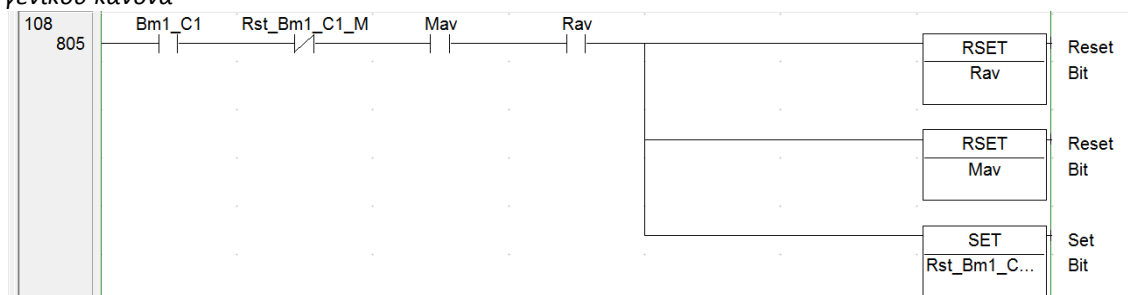


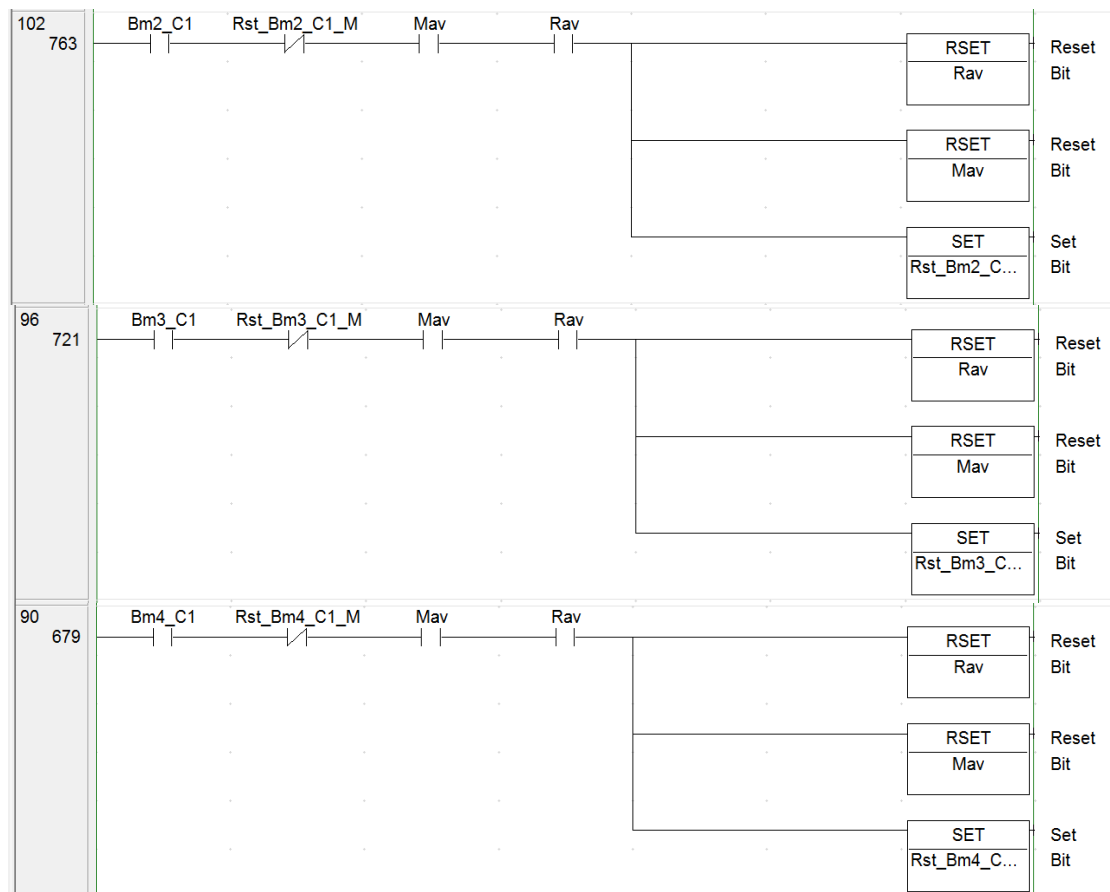
Τροποποιημένα Rungs μετάβασης *Finish loading part to Buffer* για την περίπτωση του τεμαχίου C

Μετάβαση 9: Start unloading part from Buffer



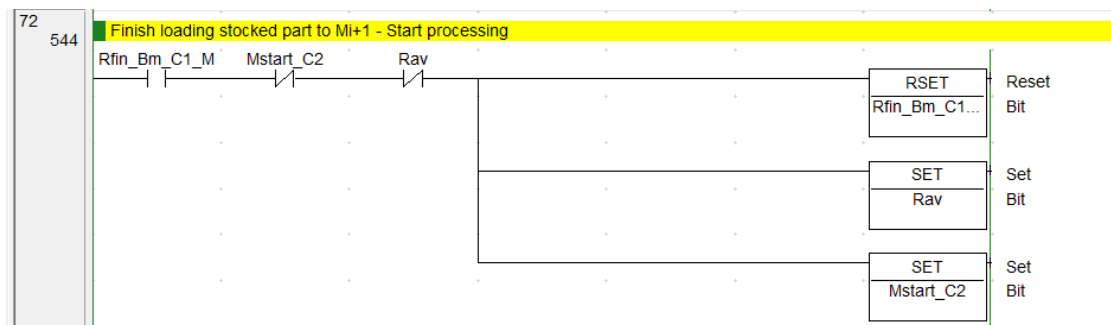
Rung μετάβασης *Start unloading part from Buffer* για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα



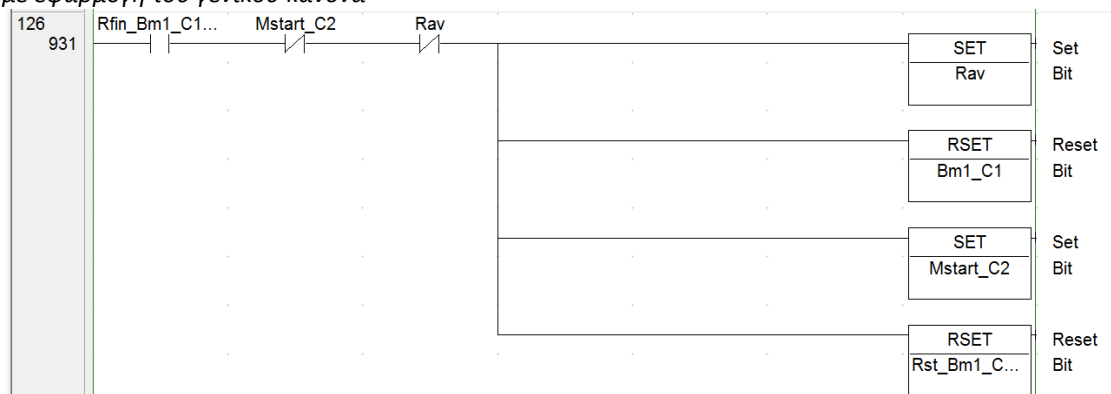


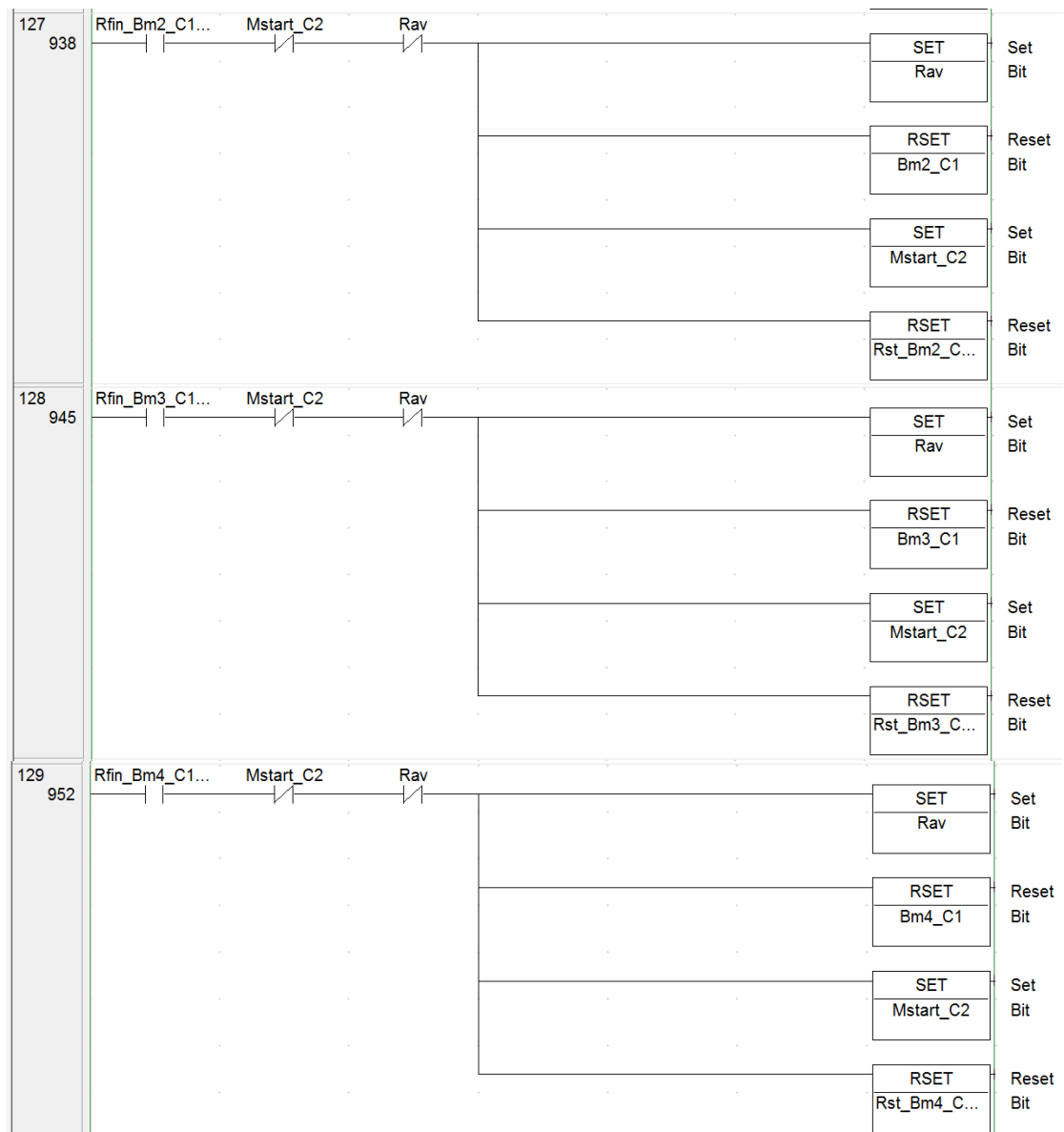
Τροποποιημένα Rungs μετάβασης *Start unloading part from Buffer* για την περίπτωση του τεμαχίου C

Μετάβαση 10: Finish loading stocked part to Mi+1 - Start processing



Rung μετάβασης *Finish loading stocked part to Mi+1-Start processing* για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα





Τροποποιημένα Rungs μετάβασης **Finish loading stocked part to Mi+1-Start processing** για την περίπτωση του τεμαχίου C με εφαρμογή του γενικού κανόνα

Όπως φαίνεται από τα παραπάνω rungs, γίνεται πολλές φορές set και reset του ίδιου στοιχείου. Αυτή είναι μια τακτική προγραμματισμού PLC που συνήθως αποφεύγεται καθώς δυσκολεύει την παρακολούθηση της κατάστασης των στοιχείων ιδιαίτερα σε μεγάλα προγράμματα. Γενικότερα, συνηθίζεται σε κάθε πηνίο ή μνήμη μέσα στο πρόγραμμα να γίνεται μία φορά set/ reset.

Όμως, στη συγκεκριμένη εφαρμογή αυτό δεν μπορεί να γίνει γιατί θα αλλάξει η σειρά των rungs μέσα στο πρόγραμμα. Ως αποτέλεσμα ο ελεγκτής, όπως φάνηκε και στις πρώτες προσπάθειες προσομοίωσης, χάνει ή 'μπερδεύει' τις καταστάσεις του συστήματος.

Έτσι διατηρήθηκε η παραπάνω μορφή του προγράμματος και σε επόμενη φάση εξετάστηκε, χρησιμοποιώντας τον On-line Simulator, αν το πρόγραμμα λειτουργεί σωστά.

5.4 Προσομοίωση προγράμματος ladder

Στην ενότητα αυτή παρουσιάζεται η προσομοίωση του τελικού προγράμματος ladder που προέκυψε από τη μετατροπή του δικτύου Petri. Για να εξετάσουμε την ορθότητα του προγράμματος επικεντρωθήκαμε κυρίως στα εξής σημεία:

- την απουσία κολλημάτων δηλαδή των καταστάσεων όπου το σύστημα δε λειτουργεί ή σταματά να λειτουργεί σωστά
- τη συνεχόμενη λειτουργία του συστήματος δηλαδή την απουσία νεκρών χρόνων ,εφόσον υπάρχουν διαθέσιμοι οι απαραίτητοι πόροι,
- την τήρηση των προτεραιοτήτων που είχαν τεθεί κατά το σχεδιασμό
- τη λογική ακολουθία των ενεργειών του συστήματος σε σχέση με τα σήματα εισόδου που του επιβάλλονται κατά την προσομοίωση

Τα σήματα εισόδου κατά την προσομοίωση δίνονται με λογική σειρά αλλά όχι σε καθορισμένα χρονικά διαστήματα στο πρόγραμμα όπως θα συνέβαινε και στο πραγματικό σύστημα όπου οι χρόνοι κατεργασιών και μεταφοράς των τεμαχίων δεν μπορούν να προβλεφθούν με ακρίβεια.

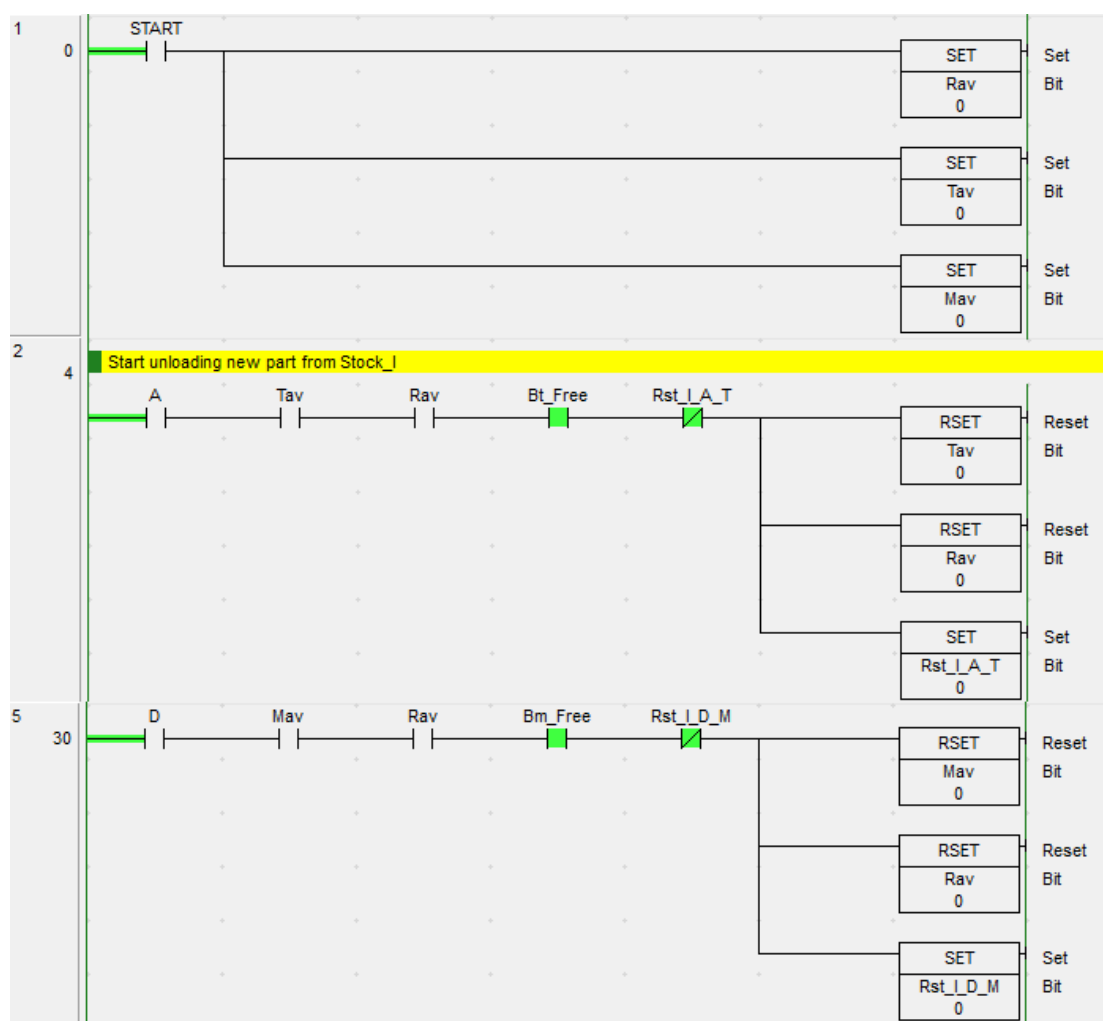
Για την παρουσίαση της προσομοίωσης αναζητήθηκε ένα σενάριο λειτουργίας που να καλύπτει όλο το φάσμα λειτουργιών του συστήματος. Κατ' αναλογία με το μοντέλο crn , στο συγκεκριμένο σενάριο λειτουργίας υπάρχουν όλες οι καταστάσεις που αντιστοιχούν στις 6 θέσεις του μοντέλου crn. Για να περάσει το σύστημα από όλες αυτές τις καταστάσεις αφήνουμε μια κατεργασία στον τόρνο να διαρκέσει/καθυστερήσει αρκετό χρονικό διάστημα ώστε να γεμίσει το αντίστοιχο buffer αναμονής των τεμαχίων (δεδομένου ότι ο τόρνος δεν είναι διαθέσιμος). Αφού γεμίσει το buffer δίνεται στο plc σήμα εισόδου ότι ολοκληρώθηκε η κατεργασία στον τόρνο. Μόλις δοθεί το σήμα αυτό ξεκινά η μεταφορά του τεμαχίου από τη μηχανή τόννευσης στη μηχανή φρεζαρίσματος. Εφόσον η μηχανή τόννευσης είναι πλέον διαθέσιμη, ξεκινά με σειρά First In Last Out η κατεργασία των τεμαχίων που βρίσκονται στο αντίστοιχο buffer αναμονής. Από κάποια στιγμή και μετά δε δίνεται σήμα ότι υπάρχει τεμάχιο στη θέση εισόδου της διάταξης. Με αυτόν τον τρόπο ελέγχεται αν το σύστημα ολοκληρώνει σωστά όλες τις κατεργασίες που απομένουν , τοποθετεί τα τεμάχια στη θέση εξόδου της διάταξης και επιστρέφει στην αρχική του κατάσταση όπου οι μηχανές κατεργασίες είναι ελεύθερες και όλες οι θέσεις των buffer αναμονής έχουν αδειάσει.

Ακολουθεί λεπτομερής παρουσίαση της προσομοίωσης του προγράμματος. Σε κάθε βήμα, το οποίο αντιστοιχεί σε ένα σήμα εισόδου, παρουσιάζονται επιλεκτικά μόνο τα rungs που δείχνουν την αλλαγή της κατάστασης του συστήματος.

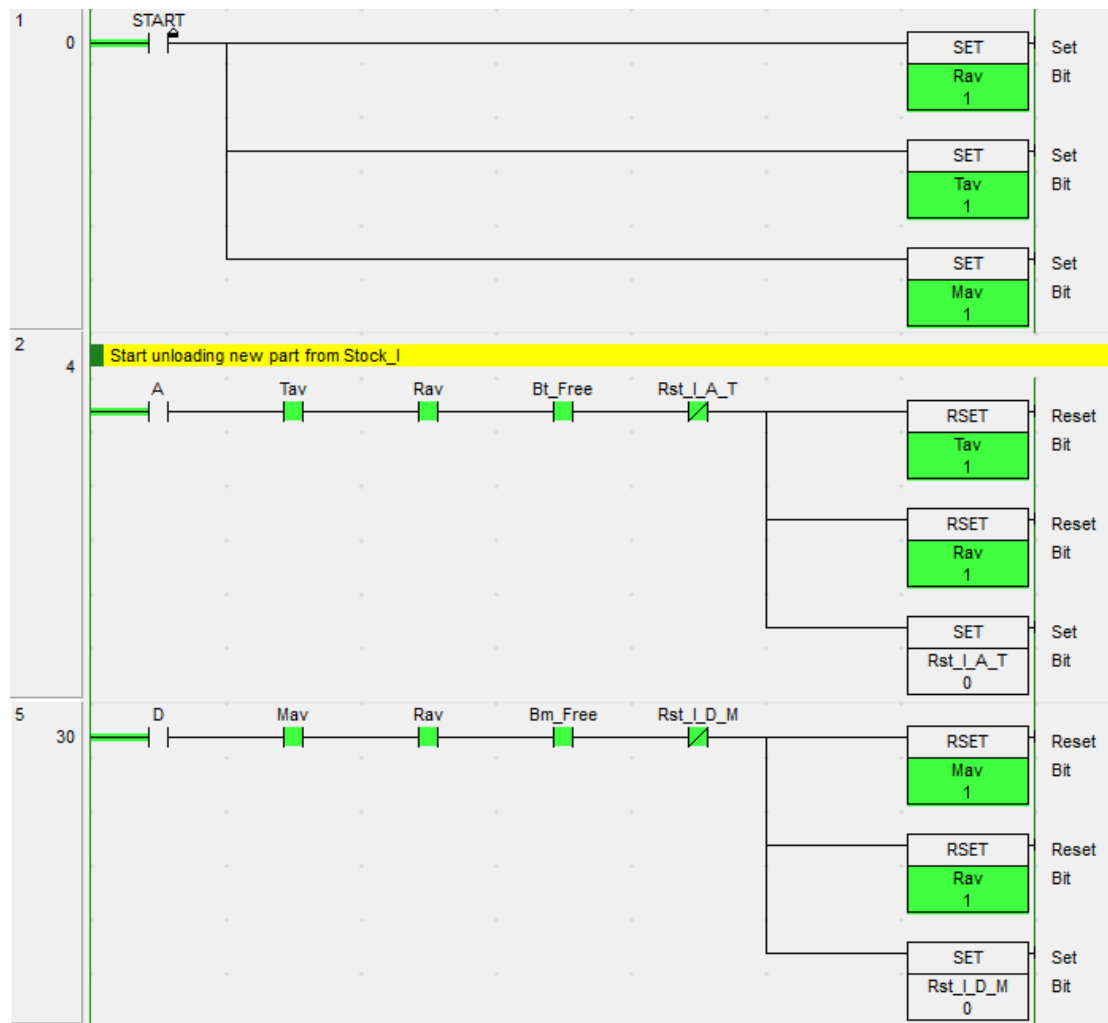
Τα τεμάχια που θα εισέρχονται στο σύστημα προς κατεργασία κατά την προσομοίωση είναι με τη σειρά : C - E - E - F - F - E . Ο λόγος που δεν επιλέχθηκαν τεμάχια με μία μόνο απαιτούμενη κατεργασία (δηλαδή τύπου A και D) είναι γιατί η προσομοίωση στοχεύει κυρίως στον έλεγχο της λειτουργίας των buffer και την τήρηση των προτεραιοτήτων. Η προσθήκη περισσότερων τεμαχίων θα αύξανε χωρίς λόγο τον όγκο της προσομοίωσης.

Τα σήματα εισόδου από τις μηχανές κατεργασίας και το ρομπότ δίνονται ως στιγμιαία σήματα. Για τη διάταξη ταυτοποίησης των εισερχόμενων τεμαχίων (A, B, C, D, E, F) θεωρούμε ότι κάθε σήμα διαρκεί μέχρι το ρομπότ να παραλάβει το τεμάχιο από τη θέση εισόδου (αυτό στο πραγματικό σύστημα θα μπορούσε να υλοποιηθεί με κάποιο φωτοκύτταρο). Μόλις γίνει αυτό, το προηγούμενο σήμα γίνεται OFF και το σήμα που αντιστοιχεί στο επόμενο τεμάχιο γίνεται ON.

Αρχικά δίνεται σήμα εισόδου start που ενεργοποιεί το plc. Όπως φαίνεται, αρχικά (μετά την επιβολή του start) οι μηχανές κατεργασίας και το ρομπότ είναι διαθέσιμα.

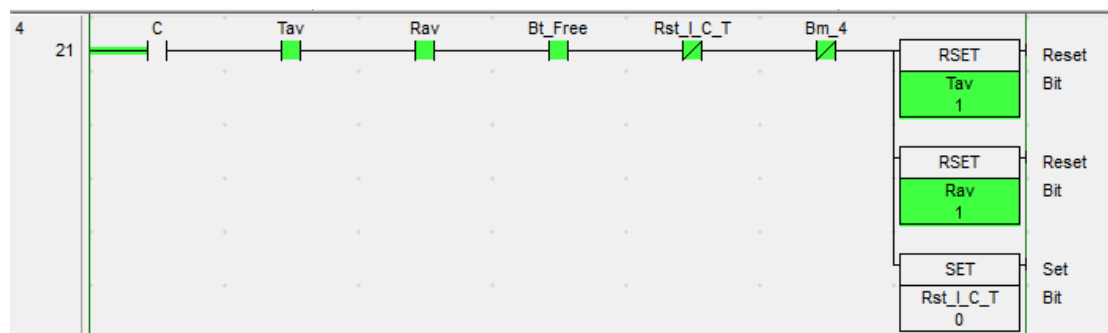


Σύστημα πριν την επιβολή του start

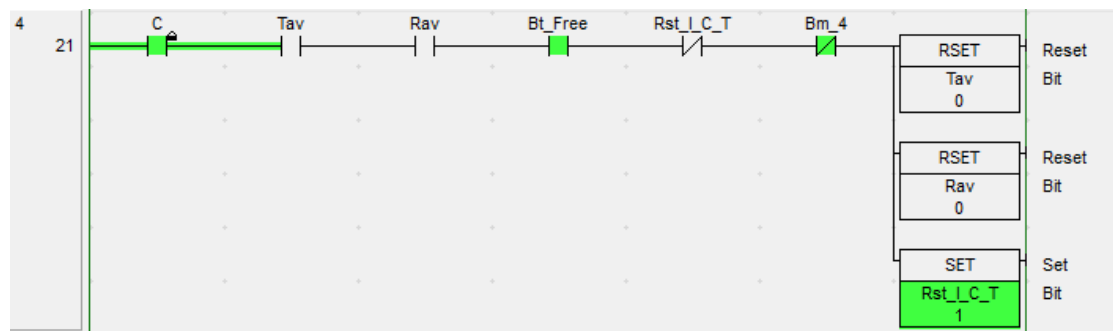


Σύστημα μετά την επιβολή του start

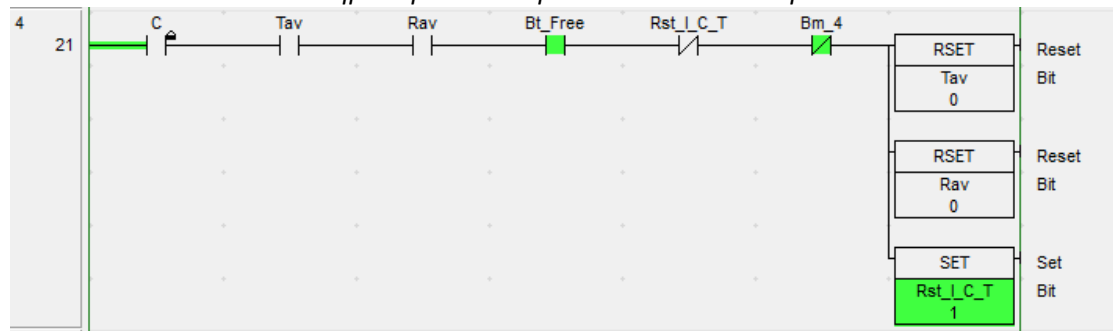
Έπειτα, θέτουμε το σήμα εισόδου C ON που σημαίνει ότι το plc λαμβάνει σήμα από τη διάταξη ταυτοποίησης ότι στη θέση εισόδου υπάρχει τεμάχιο τύπου C. Εφόσον και το ρομπότ και η μηχανή τórνευσης είναι διαθέσιμα, ξεκινάει κατευθείαν η μεταφορά-φόρτωση του τεμαχίου στον τórνο. Οπότε το plc στέλνει σήμα Rst_I_C_T στο ρομπότ.



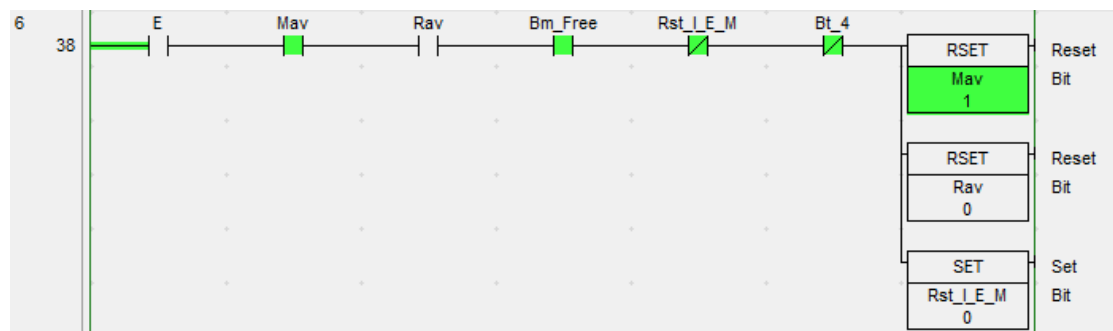
Σύστημα πριν θέσουμε το C σε κατάσταση ON



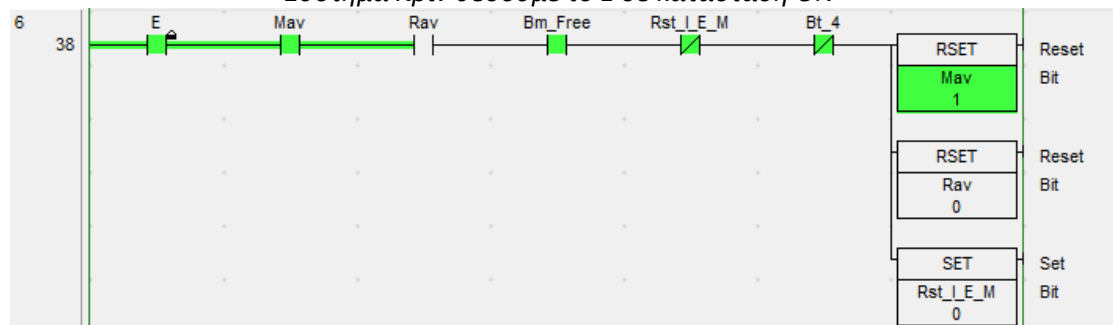
Σύστημα αφού θέσουμε το C σε κατάσταση ON



Σύστημα αφού θέσουμε το C σε κατάσταση OFF



Σύστημα πριν θέσουμε το E σε κατάσταση ON

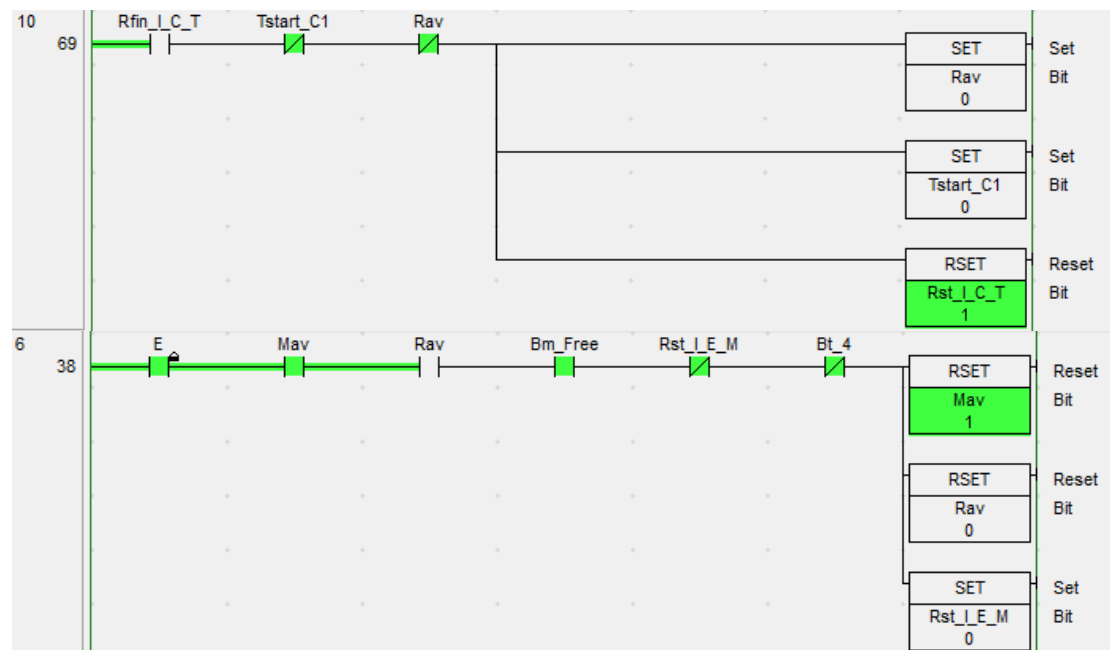


Σύστημα αφού θέσουμε το E σε κατάσταση ON

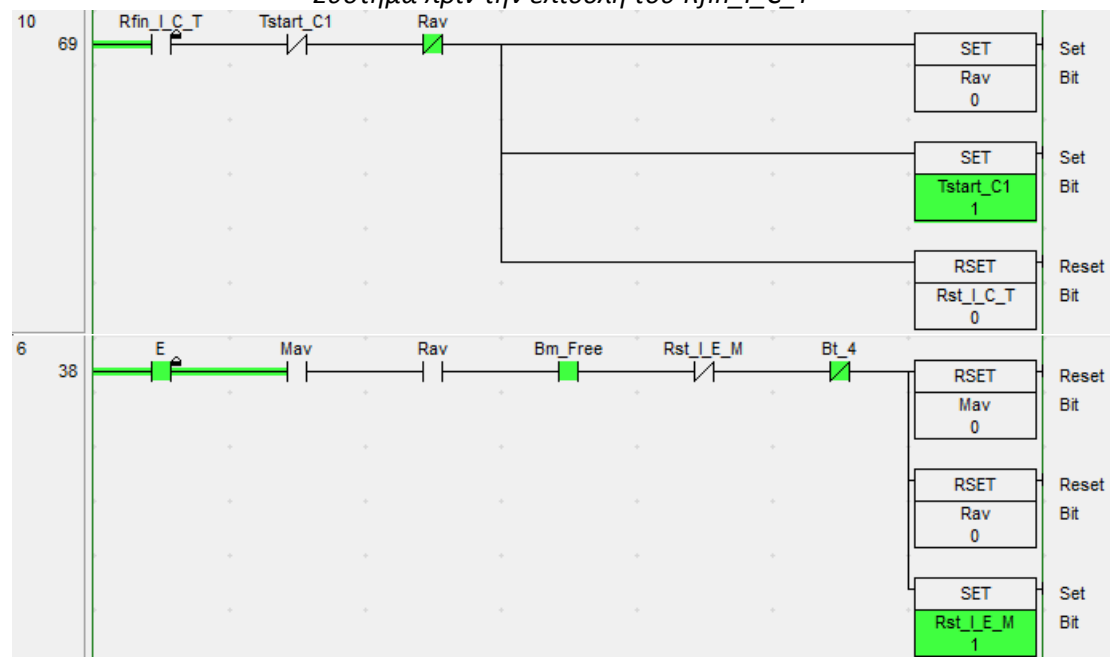
Σε αυτό το σημείο, ενώ το E βρίσκεται στη φάση ON , δηλαδή υπάρχει τεμάχιο τύπου E στη θέση εισόδου, δεν ξεκινά η μεταφορά του στη μηχανή φρεζαρίσματος (Rst_I_E_M) καθώς δεν είναι διαθέσιμο το ρομπότ. Σε όλες αυτές τις περιπτώσεις κατά την προσομοίωση αφήνουμε το σήμα εισόδου στη φάση ON μέχρι να ξεκινήσει η μεταφορά του τεμαχίου στην αντίστοιχη μηχανή. Όταν όλοι οι πόροι

(Rav, Tan, Mav) είναι διαθέσιμοι και η μεταφορά ξεκινήσει, θέτουμε το σήμα εισόδου OFF και θέτουμε το επόμενο τεμάχιο ON.

Κάποια στιγμή ολοκληρώνεται η μεταφορά-φόρτωση του τεμαχίου C στον τόρνο. Τη στιγμή αυτή το plc λαμβάνει το σήμα Rfin_I_C_T από το ρομπότ. Όπως φαίνεται παρακάτω, αμέσως μόλις ολοκληρωθεί η μεταφορά-φόρτωση του τεμαχίου, ξεκινά η κατεργασία στον τόρνο (το plc στέλνει σήμα Tstart_C1 στον ελεγκτή του τόρνου). Ταυτόχρονα όμως με την ολοκλήρωση της μεταφοράς, απελευθερώνεται και το ρομπότ οπότε το plc στέλνει σήμα Rst_I_E_M για να ξεκινήσει η μεταφορά του τεμαχίου E που περιμένει στη θέση εισόδου.

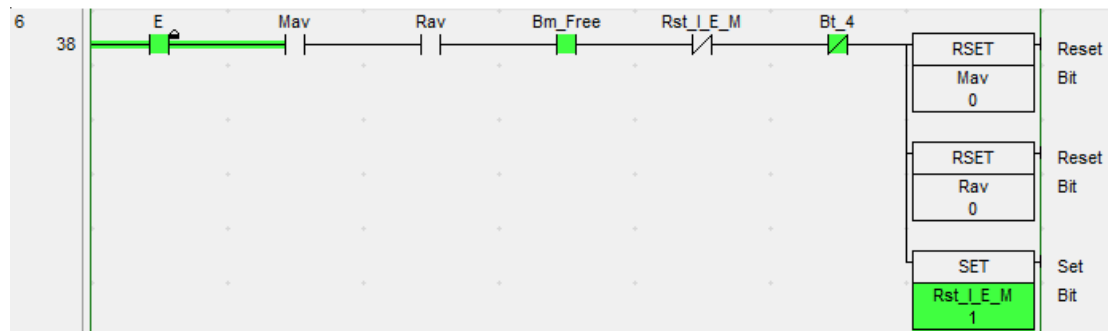


Σύστημα πριν την επιβολή του Rfin_I_C_T

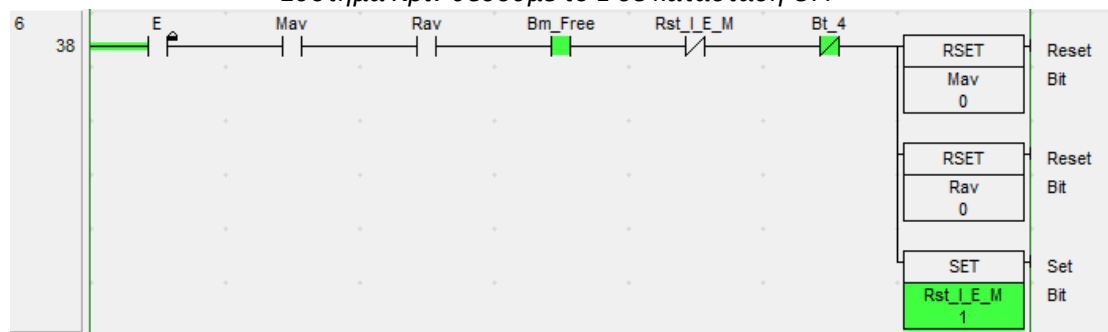


Σύστημα μετά την επιβολή του C του Rfin_I_C_T

Εφόσον ξεκίνησε η μεταφορά του τεμαχίου E από τη θέση εισόδου στη φρέζα θέτουμε το E OFF (όπως προαναφέρθηκε αυτό γίνεται αυτόματα από τη διάταξη ταυτοποίησης μόλις ξεκινήσει η μεταφορά-είσοδος του νέου τεμαχίου στο σύστημα).

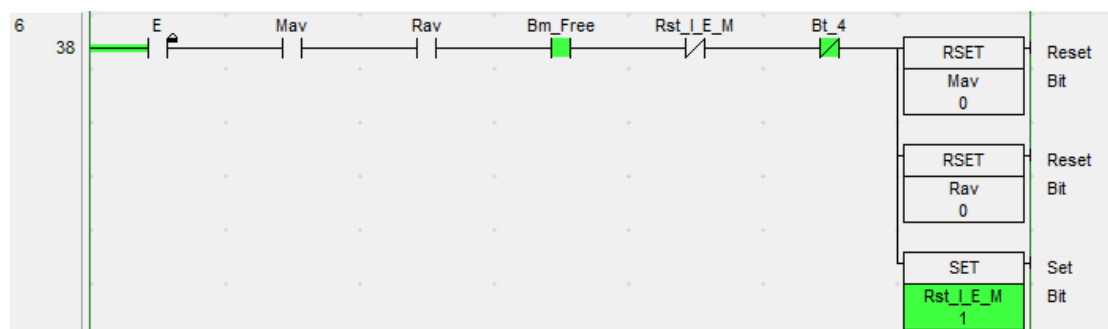


Σύστημα πριν θέσουμε το E σε κατάσταση OFF

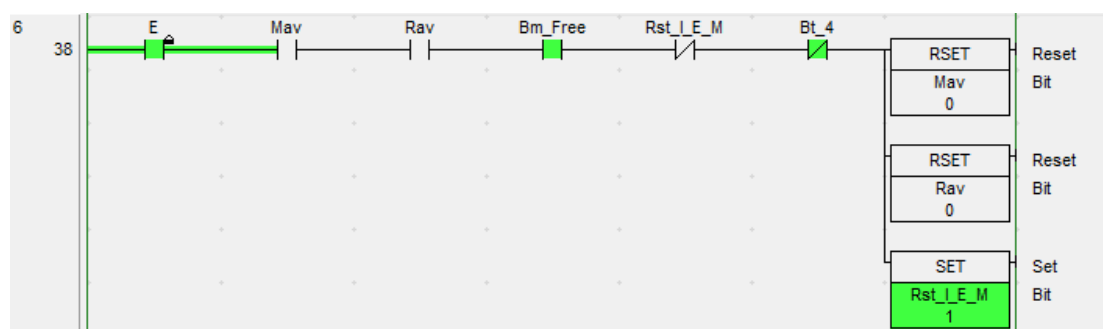


Σύστημα αφού θέσουμε το E σε κατάσταση OFF

Το επόμενο τεμάχιο προς κατεργασία είναι επίσης τύπου E. Γι' αυτό η διάταξη ταυτοποίησης στέλνει και πάλι σήμα ON στο plc μέχρι να φορτωθεί και το δεύτερο τεμάχιο τύπου E.

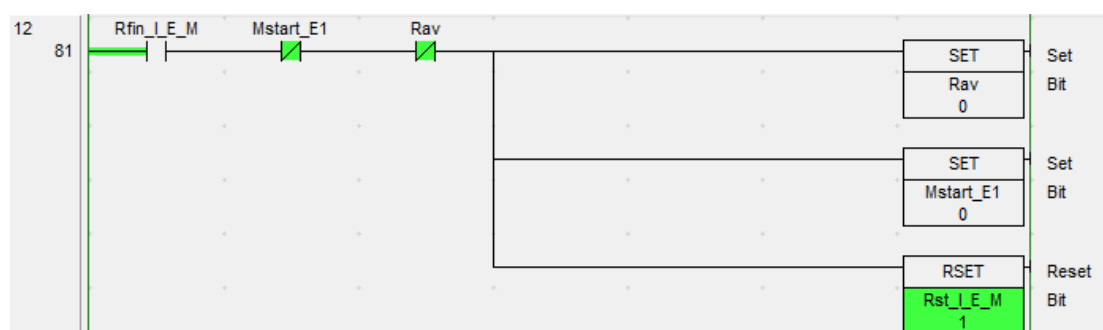


Σύστημα πριν θέσουμε το E σε κατάσταση ON

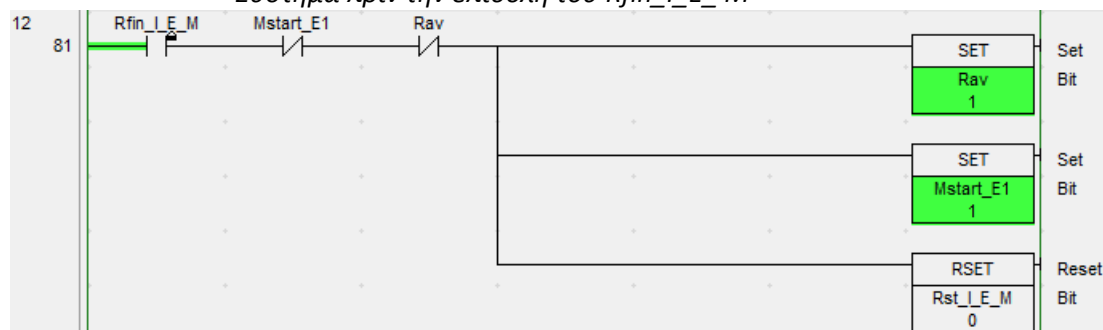


Σύστημα αφού θέσουμε το E σε κατάσταση ON

Κάποια στιγμή ολοκληρώνεται η μεταφορά-φόρτωση του τεμαχίου E στη φρέζα οπότε το plc λαμβάνει το σήμα Rfin_I_E_M από το ρομπότ. Αμέσως το plc στέλνει σήμα Mstart_E1 στον ελεγκτή της φρέζας για να ξεκινήσει η αντίστοιχη κατεργασία.

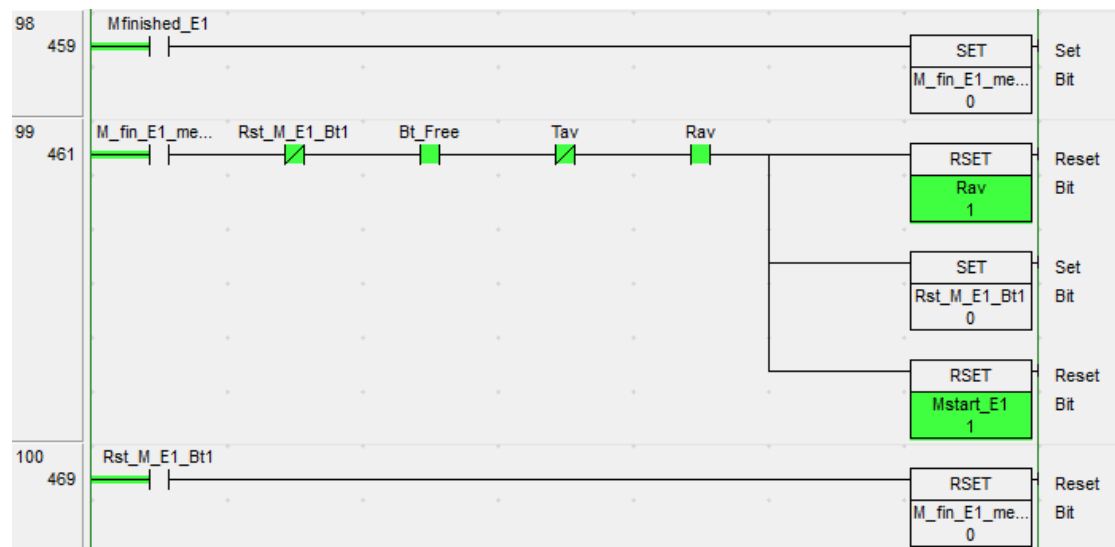


Σύστημα πριν την επιβολή του Rfin_I_E_M

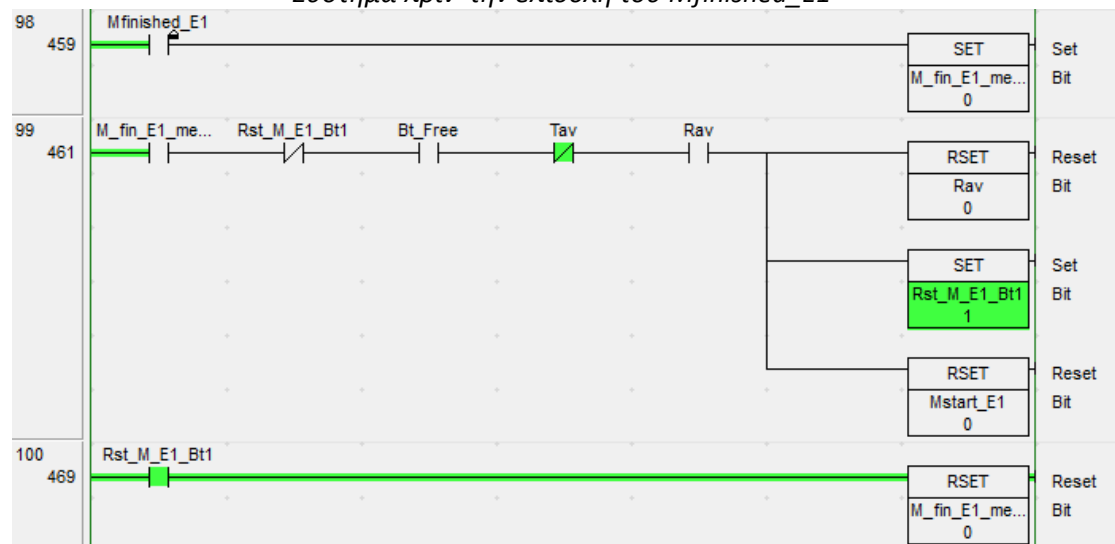


Σύστημα μετά την επιβολή του Rfin_I_E_M

Σε αυτή τη φάση καμία από τις δύο μηχανές δεν είναι διαθέσιμη. Σύμφωνα με το process plan του τεμαχίου E, μετά την κατεργασία E1 στη φρέζα απαιτείται κατεργασία E2 στον τόρνο. Επειδή όμως βρίσκεται ακόμα σε εξέλιξη η κατεργασία C1 στον τόρνο, το τεμάχιο E θα πρέπει να μεταφερθεί στο buffer αναμονής μέχρι να ελευθερωθεί ο τόρνος. Όπως φαίνεται από το παρακάτω rung, μόλις το plc λάβει σήμα Mfinished_E1 από τη φρέζα ότι η κατεργασία E1 ολοκληρώθηκε, στέλνει σήμα Rst_M_E1_Bt1 στο ρομπότ για τη μεταφορά του τεμαχίου από τη μηχανή στην πρώτη θέση του Buffer αναμονής Bt.

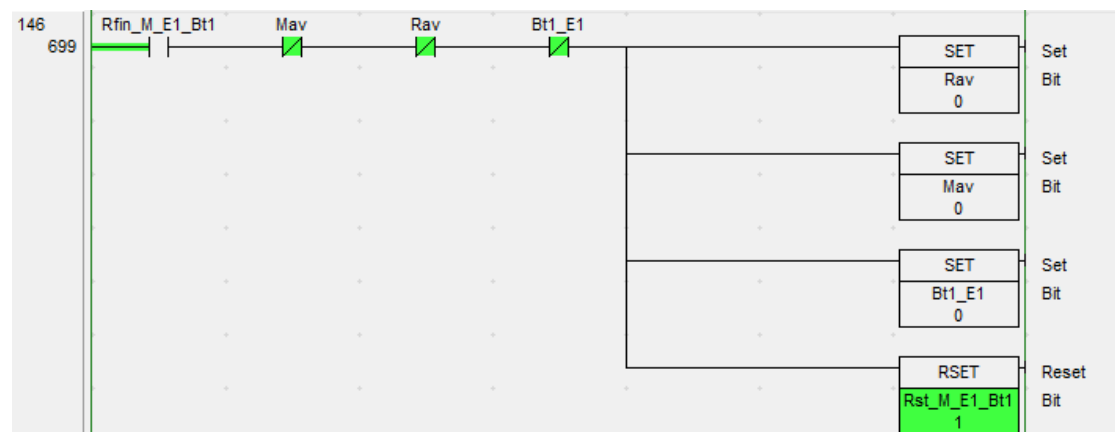


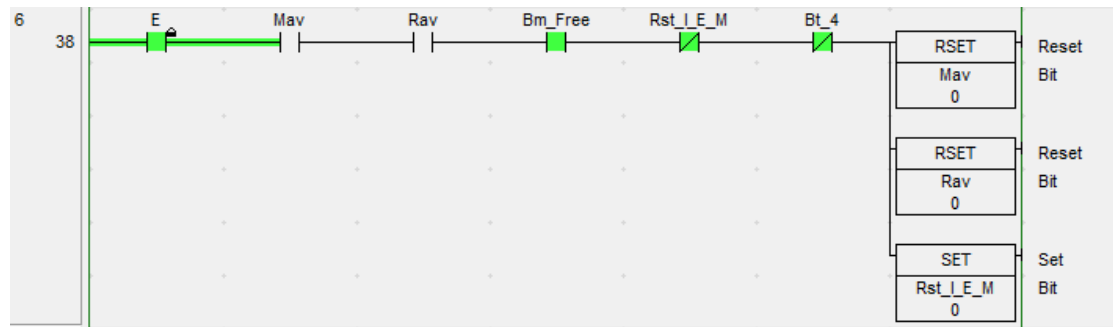
Σύστημα πριν την επιβολή του Mfinished_E1



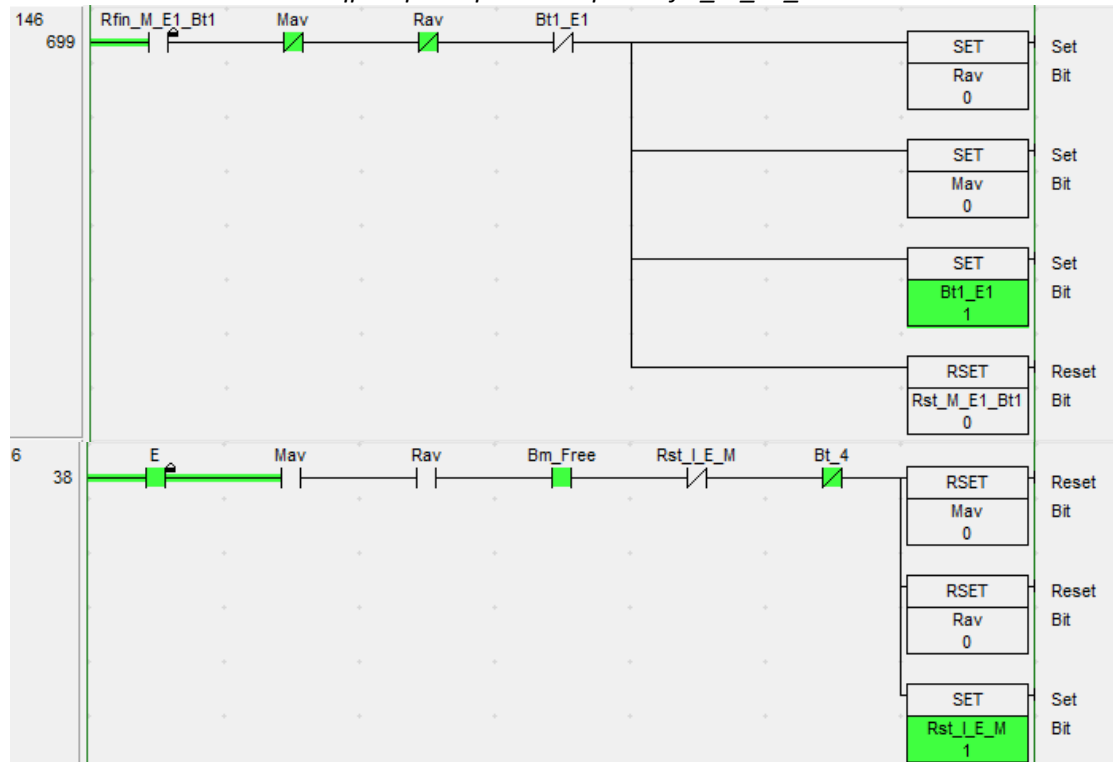
Σύστημα μετά την επιβολή του Mfinished_E1

Όταν ολοκληρωθεί η μεταφορά του τεμαχίου, το plc λαμβάνει σήμα Rfin_M_E1_Bt1 από το ρομπότ ότι ολοκληρώθηκε η συγκεκριμένη ενέργεια. Το ρομπότ γίνεται και πάλι διαθέσιμο οπότε το plc στέλνει σήμα Rst_I_E_M για τη μεταφορά του τεμαχίου E από τη θέση εισόδου στη μηχανή φρεζαρίσματος.



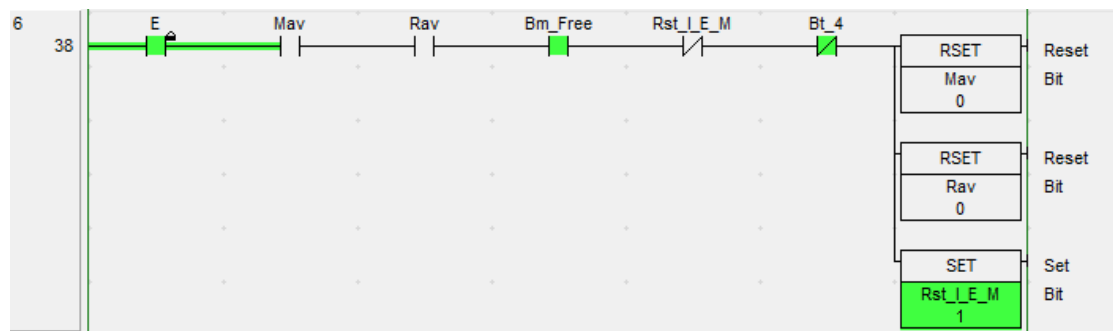


Σύστημα πριν την επιβολή του Rfin_M_E1_Bt1

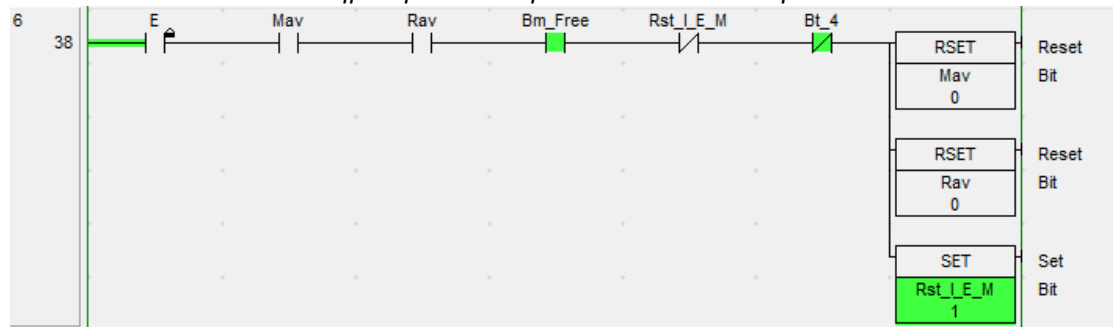


Σύστημα μετά την επιβολή του Rfin_M_E1_Bt1

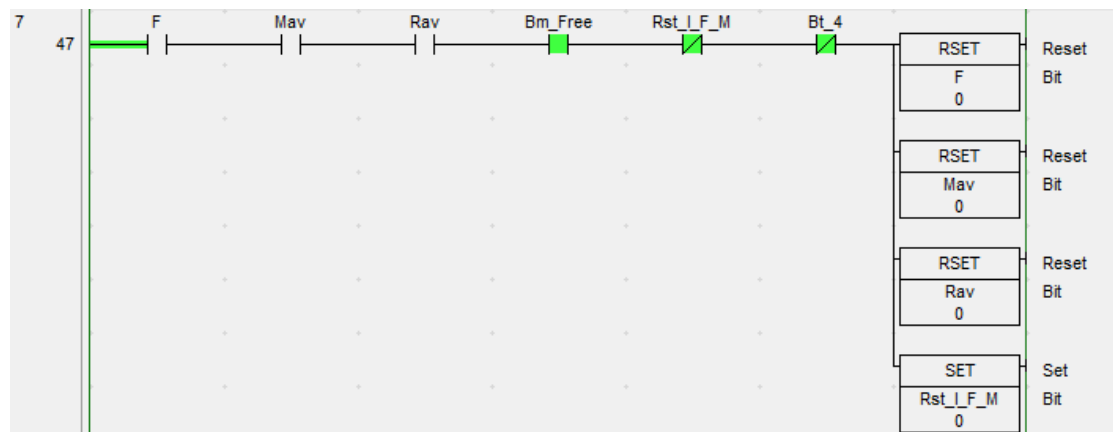
Εφόσον ξεκίνησε η μεταφορά του τεμαχίου E από τη θέση εισόδου, θέτουμε OFF το αντίστοιχο σήμα εισόδου και ON το επόμενο τεμάχιο F.



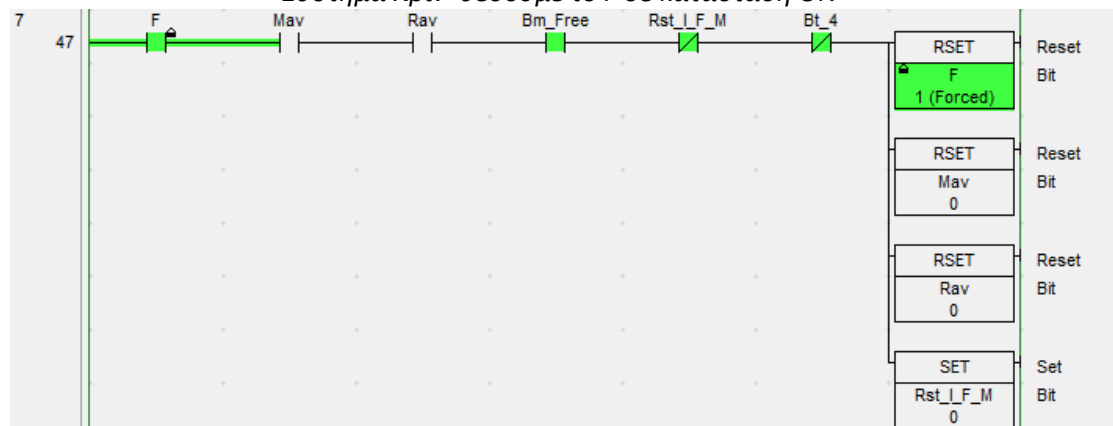
Σύστημα πριν θέσουμε το E σε κατάσταση OFF



Σύστημα αφού θέσουμε το E σε κατάσταση OFF

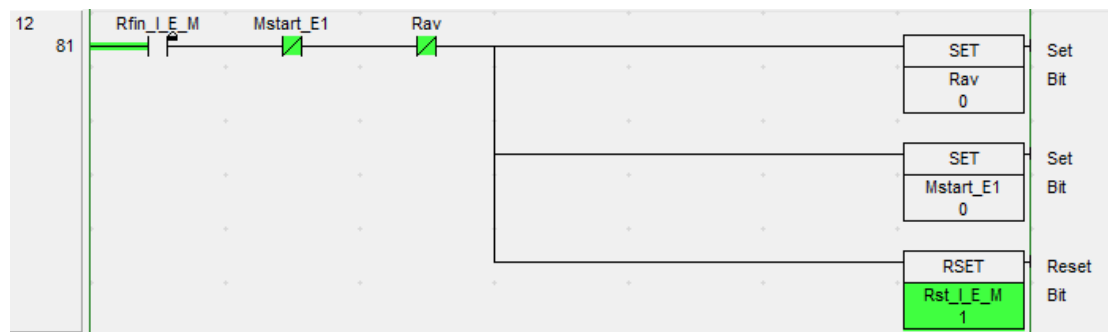


Σύστημα πριν θέσουμε το F σε κατάσταση ON

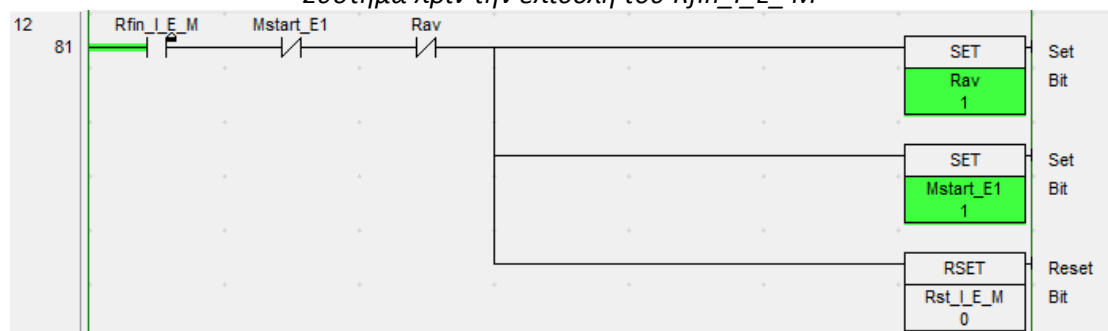


Σύστημα αφού θέσουμε το F σε κατάσταση ON

Κάποια στιγμή ολοκληρώνεται η μεταφορά-φόρτωση του τεμαχίου Ε στη φρέζα οπότε το plc λαμβάνει το σήμα Rfin_I_E_M από το ρομπότ. Αμέσως το plc στέλνει σήμα Mstart_E1 στον ελεγκτή της φρέζας για να ξεκινήσει η κατεργασία Ε1.

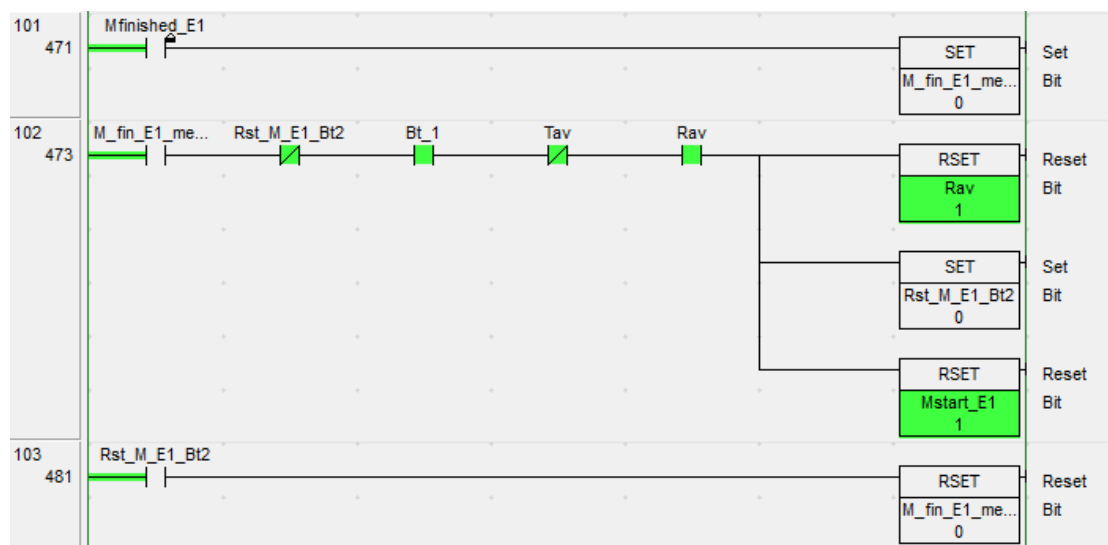


Σύστημα πριν την επιβολή του Rfin_I_E_M

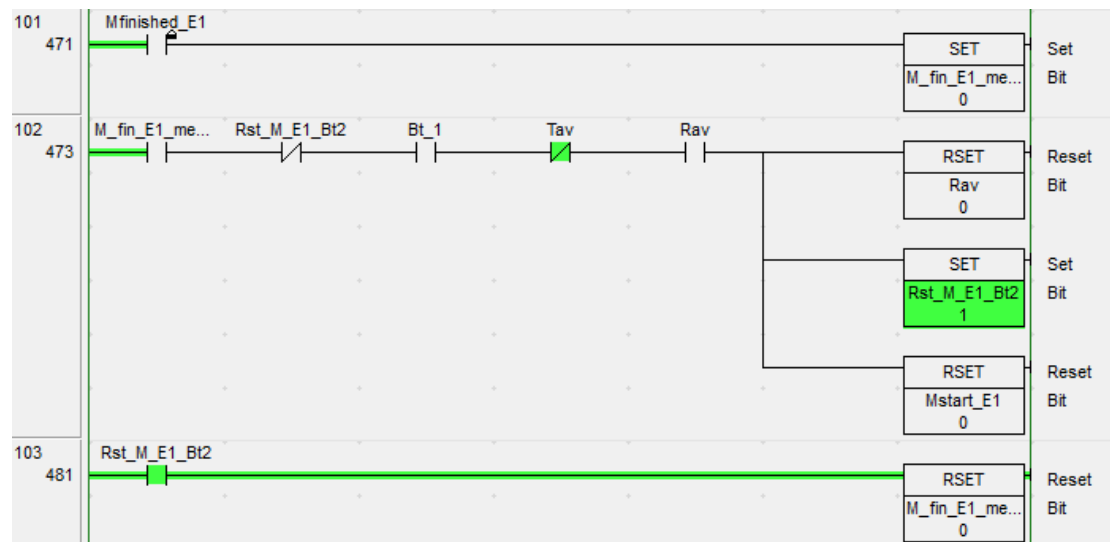


Σύστημα μετά την επιβολή του Rfin_I_E_M

Μόλις το plc λάβει σήμα Mfinished_E1 από τη φρέζα ότι η κατεργασία Ε1 ολοκληρώθηκε, στέλνει σήμα Rst_M_E1_Bt2 στο ρομπότ για τη μεταφορά του τεμαχίου από τη μηχανή στην δεύτερη θέση του Buffer αναμονής Bt (εφόσον η πρώτη είναι ήδη κατειλημμένη).

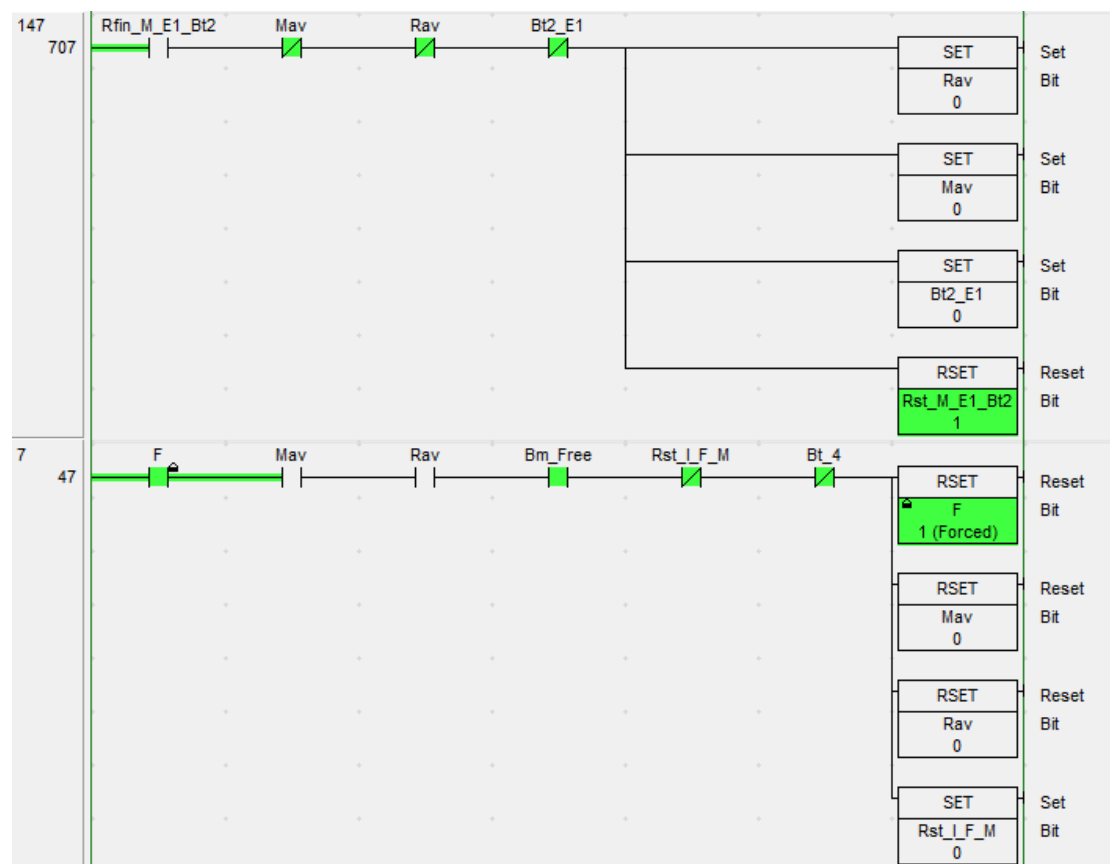


Σύστημα πριν την επιβολή του Mfinished_E1

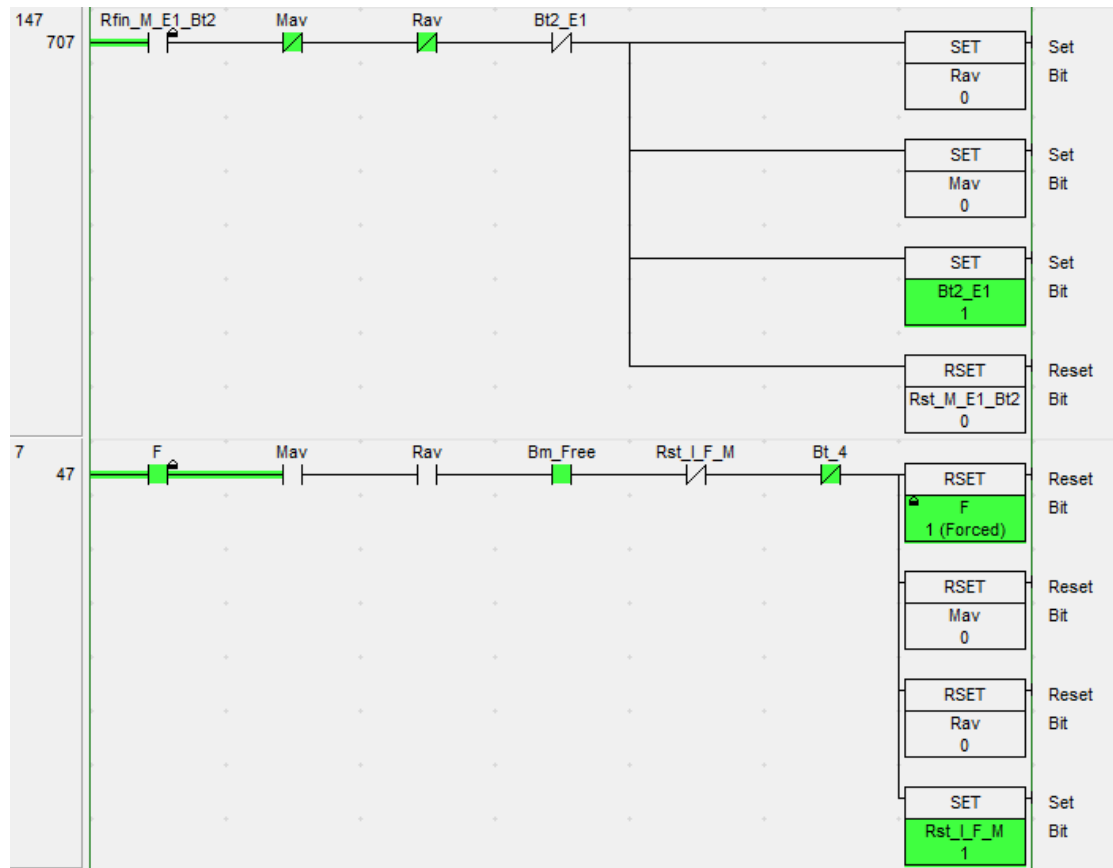


Σύστημα μετά την επιβολή του Mfinished_E1

Όταν ολοκληρωθεί η μεταφορά του τεμαχίου, το plc λαμβάνει σήμα Rfin_M_E1_Bt2 από το ρομπότ ότι ολοκληρώθηκε η συγκεκριμένη ενέργεια. Αμέσως, το ρομπότ γίνεται διαθέσιμο και το plc στέλνει σήμα Rst_I_F_M για τη μεταφορά του τεμαχίου F από τη θέση εισόδου στη φρέζα.

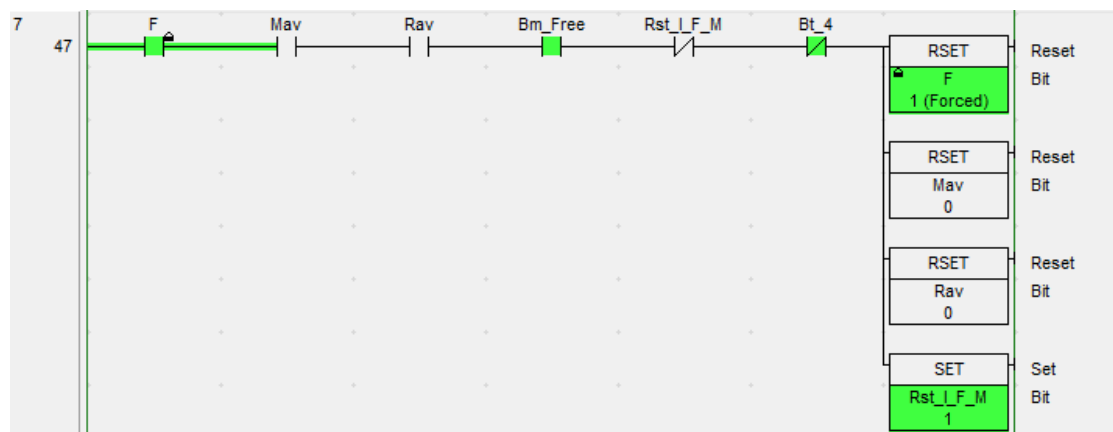


Σύστημα πριν την επιβολή του Rfin_M_E1_Bt2

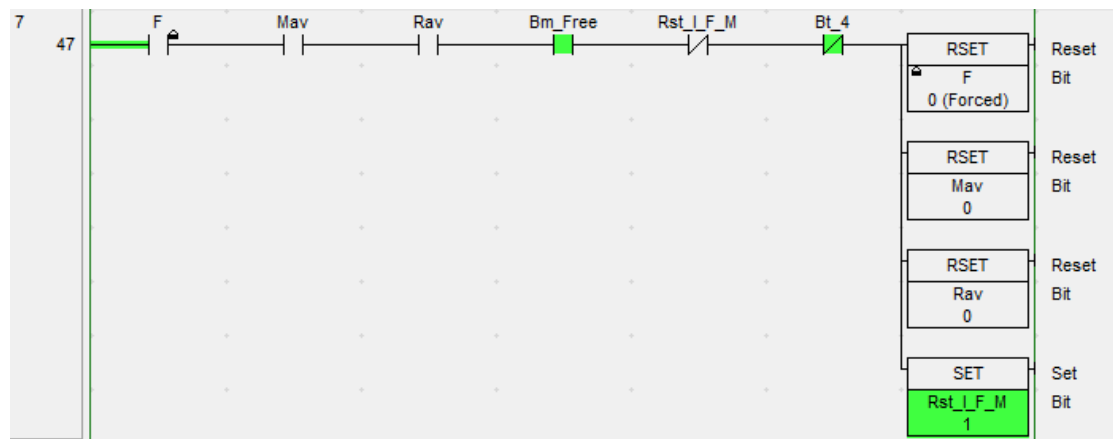


Σύστημα μετά την επιβολή του Rfin_M_E1_Bt2

Εφόσον ξεκίνησε η μεταφορά του τεμαχίου F από τη θέση εισόδου στο σύστημα, θέτουμε το F OFF.

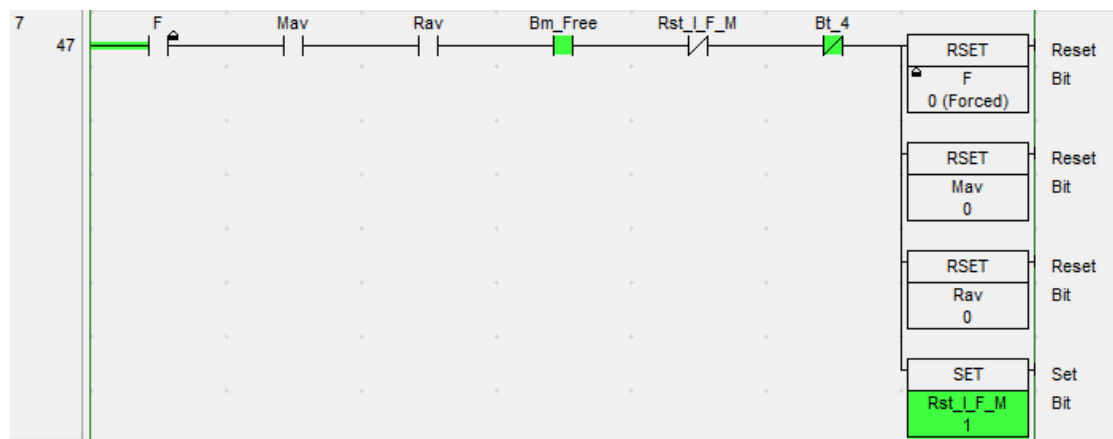


Σύστημα πριν θέσουμε το F σε κατάσταση OFF

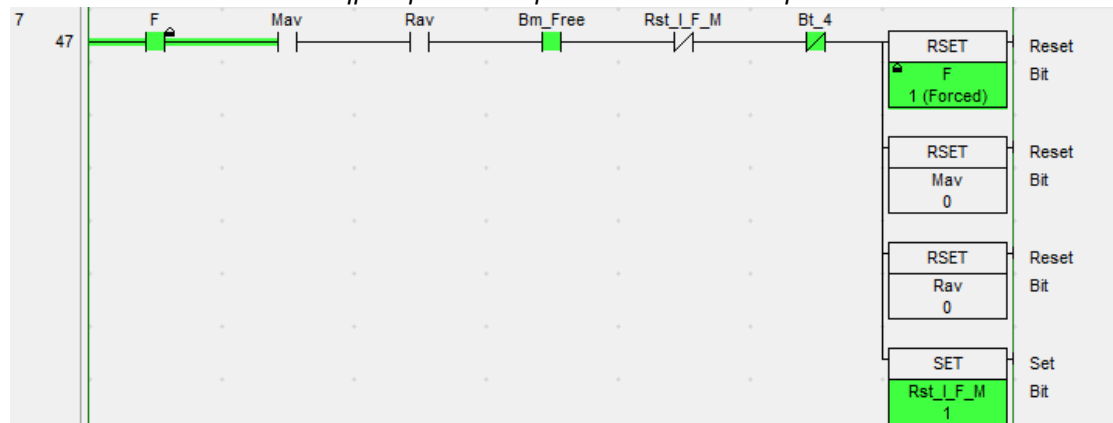


Σύστημα αφού θέσουμε το F σε κατάσταση OFF

Επειδή και το επόμενο τεμάχιο είναι τύπου F θέτουμε και πάλι το σήμα εισόδου F ON.

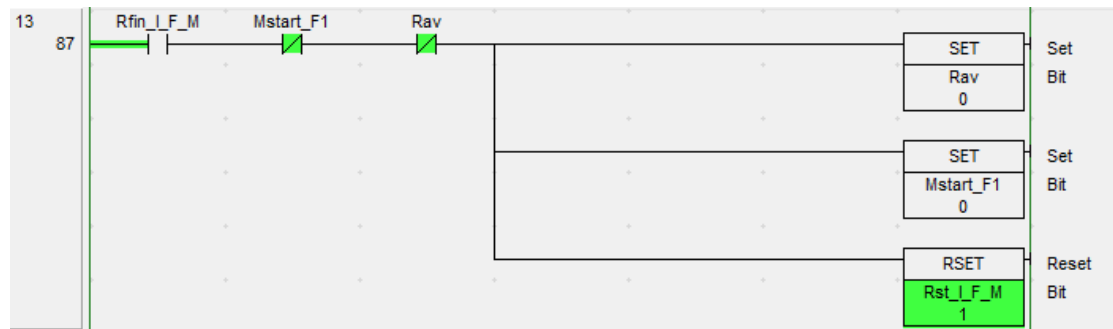


Σύστημα πριν θέσουμε το F σε κατάσταση ON

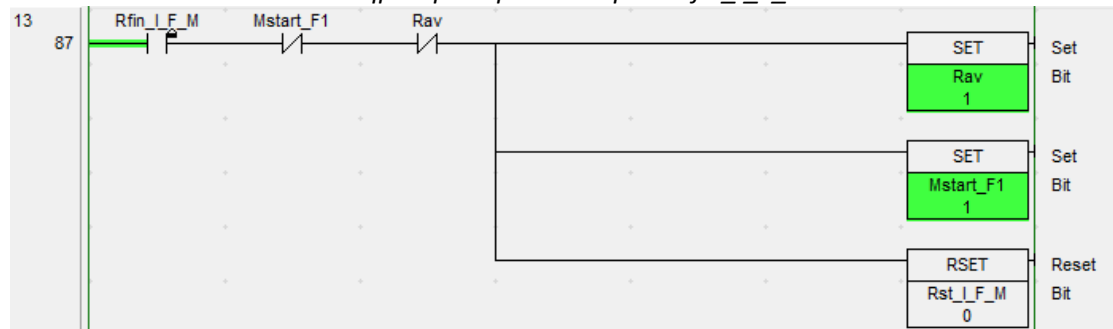


Σύστημα αφού θέσουμε το F σε κατάσταση ON

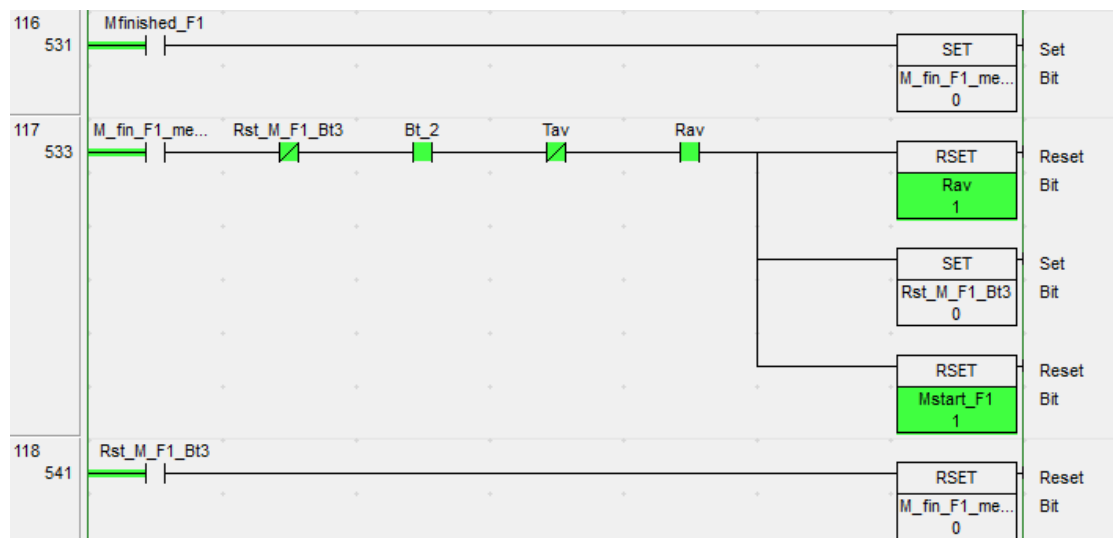
Με τον ίδιο τρόπο το πρώτο τεμάχιο F κατεργάζεται στη φρέζα και αποθηκεύεται στο buffer αναμονής εφόσον βρίσκεται ακόμη σε εξέλιξη η κατεργασία C1. Ακολουθεί ομοίως η κατεργασία και η αποθήκευση ενός ακόμη τεμαχίου F (όλα τα σήματα εισόδου στο σύστημα και οι αντίστοιχες καταστάσεις φαίνονται παρακάτω).



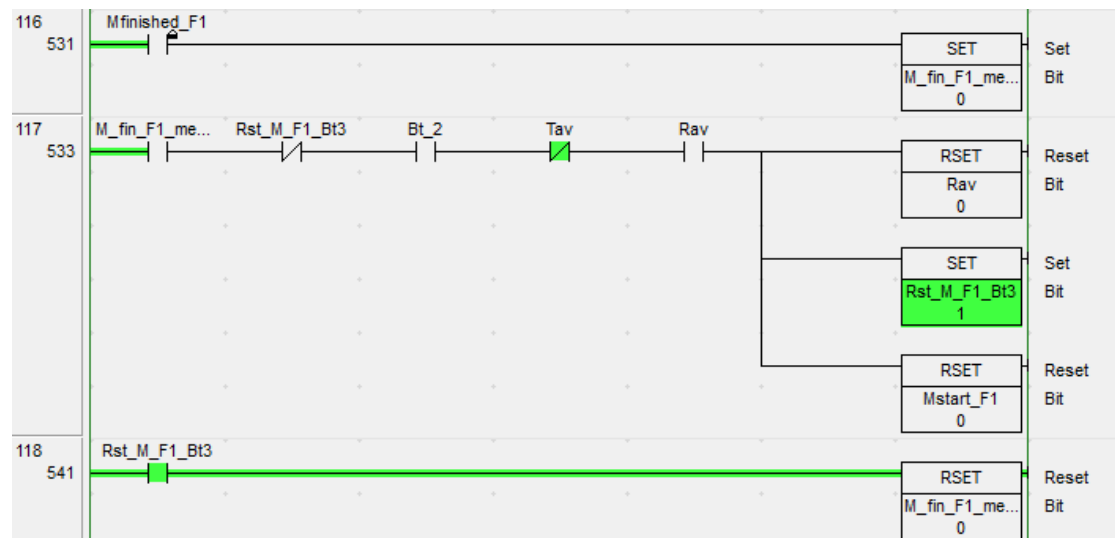
Σύστημα πριν την επιβολή του $Rfin_I_F_M$



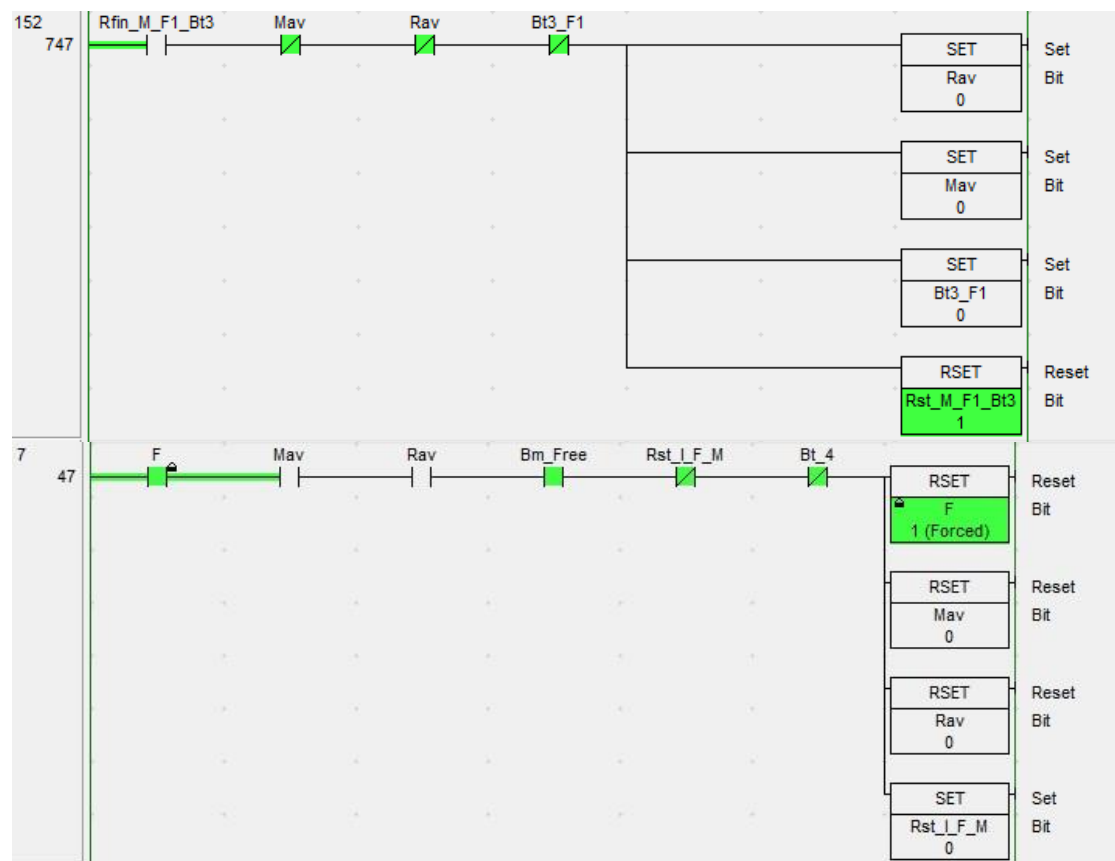
Σύστημα μετά την επιβολή του $Rfin_I_F_M$



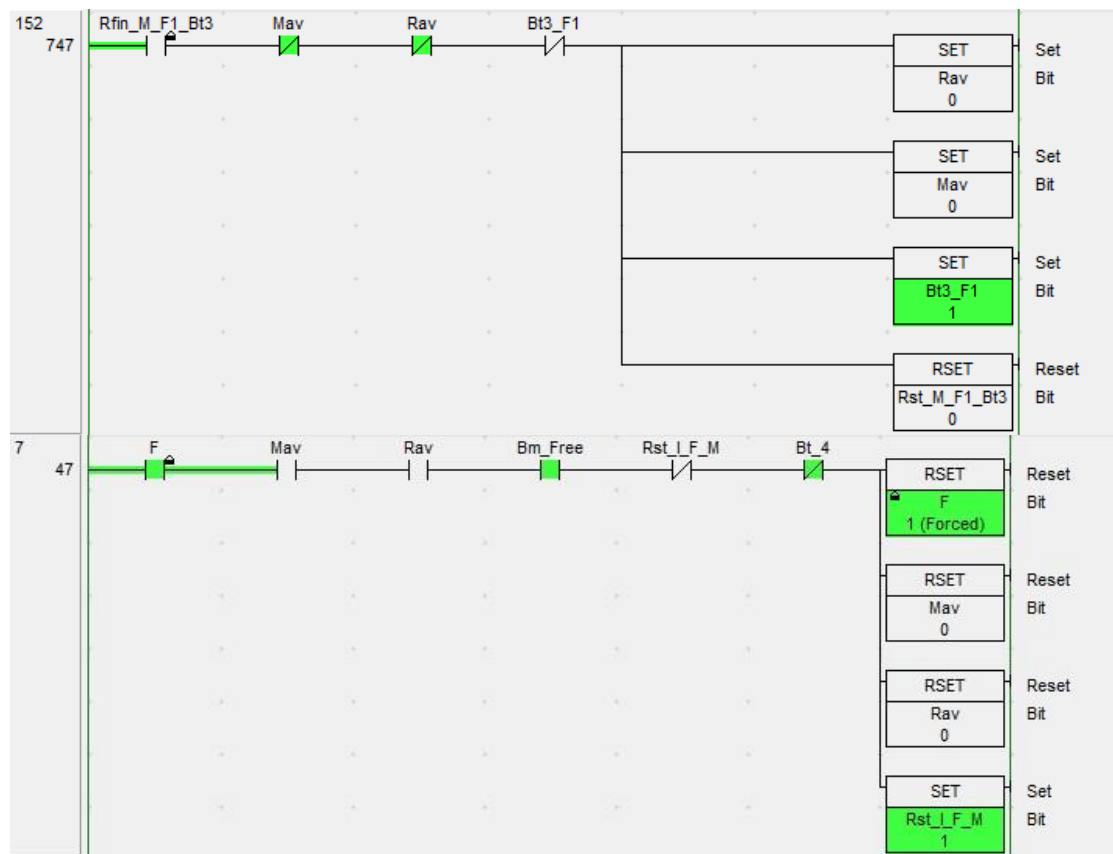
Σύστημα πριν την επιβολή του $Mfinished_F1$



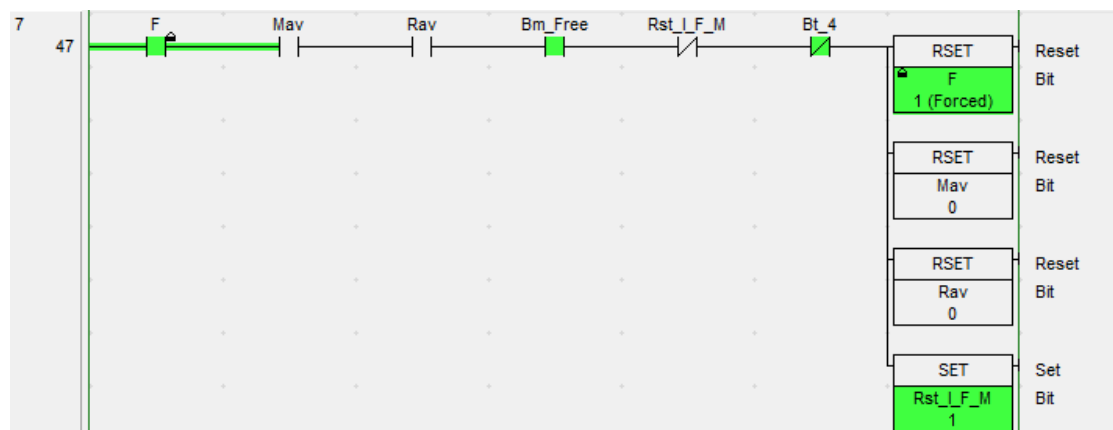
Σύστημα μετά την επιβολή του `Mfinished_F1`



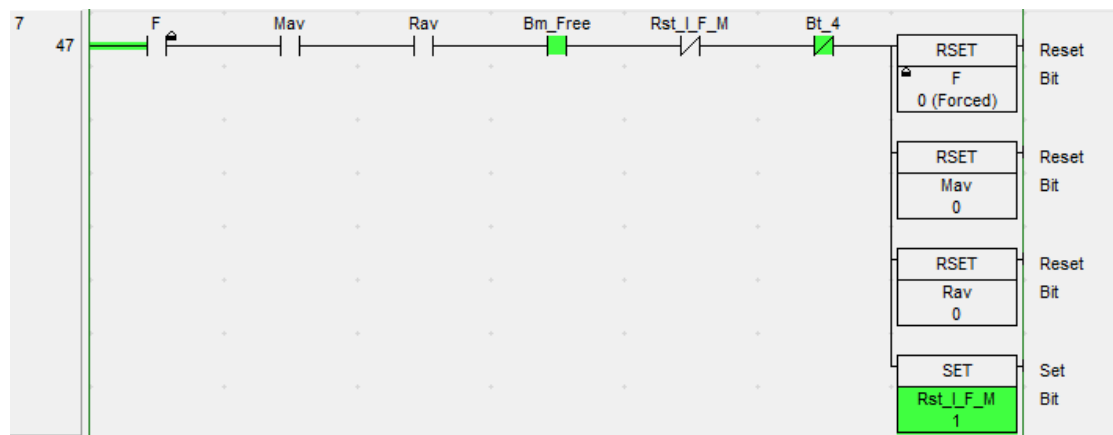
Σύστημα πριν την επιβολή του `Rfin_M_F1_Bt3`



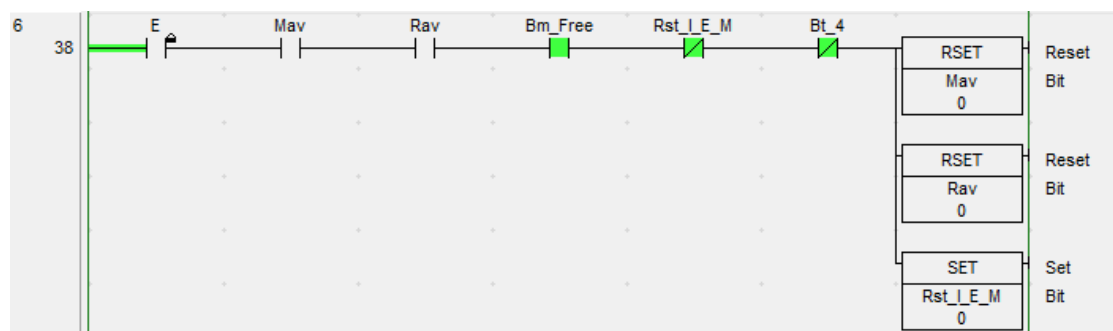
Σύστημα πριν την επιβολή του `Rfin_M_F1_Bt3`



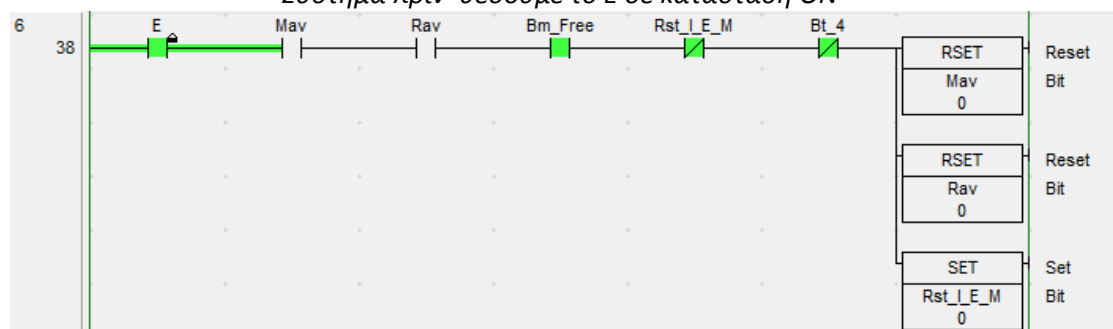
Σύστημα πριν θέσουμε το `F` σε κατάσταση `OFF`



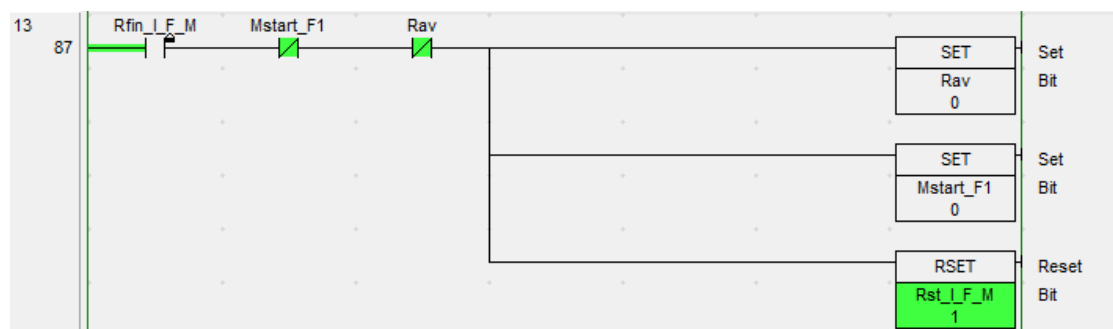
Σύστημα αφού θέσουμε το F σε κατάσταση OFF



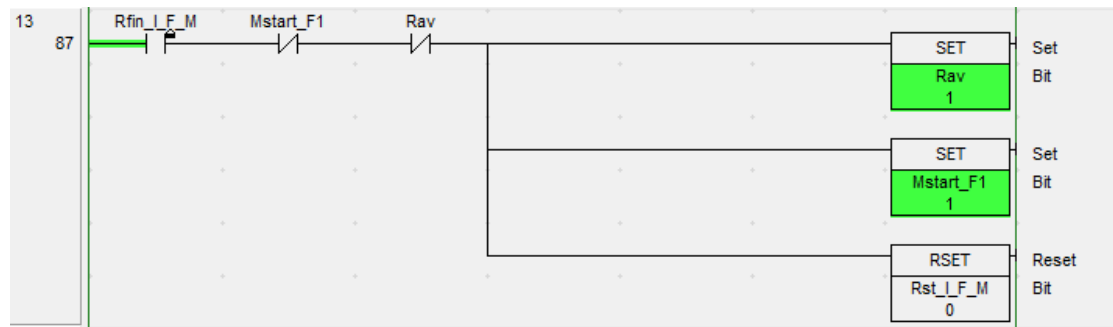
Σύστημα πριν θέσουμε το E σε κατάσταση ON



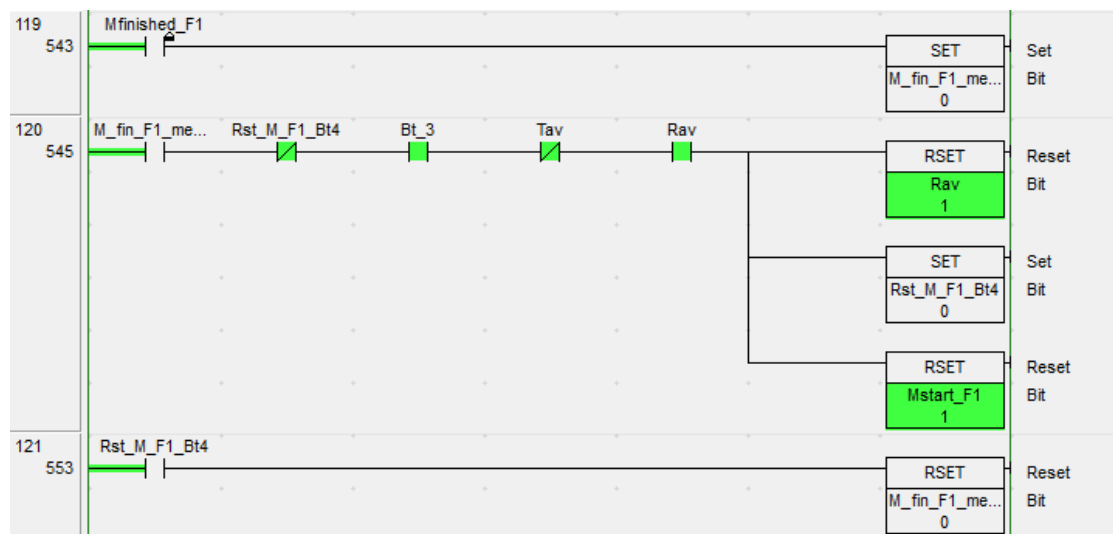
Σύστημα αφού θέσουμε το E σε κατάσταση ON



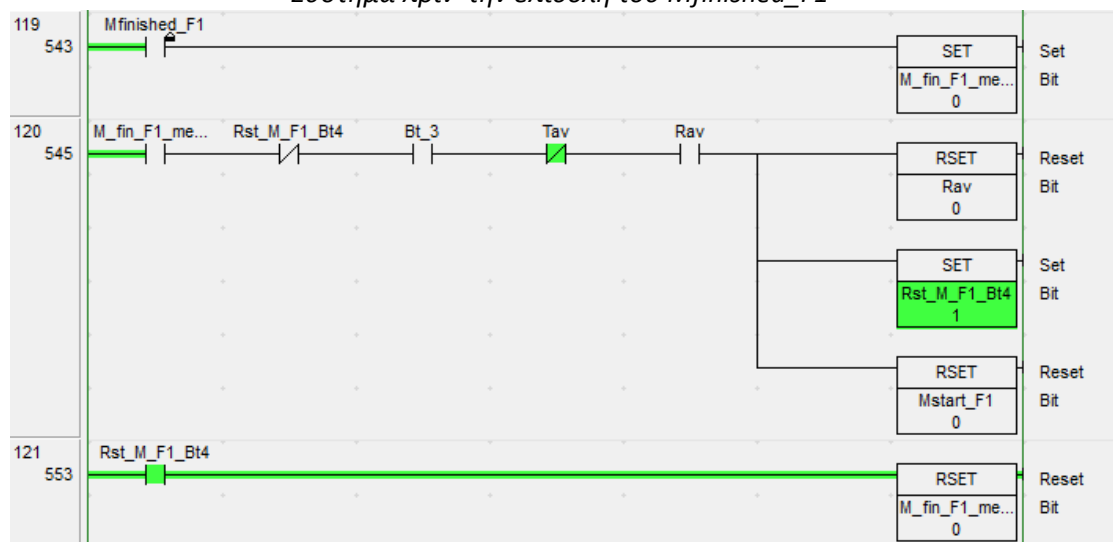
Σύστημα πριν την επιβολή του Rfin_I_F_M



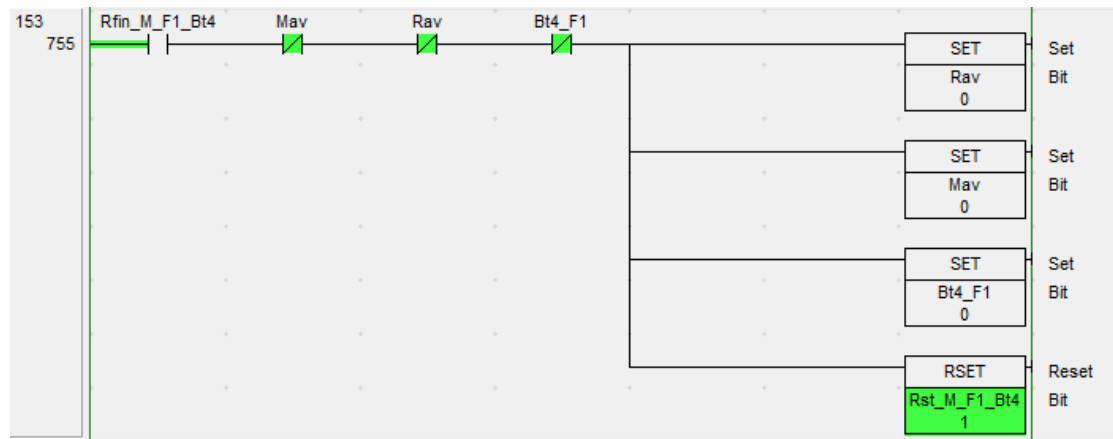
Σύστημα μετά την επιβολή του Rfin_I_F_M



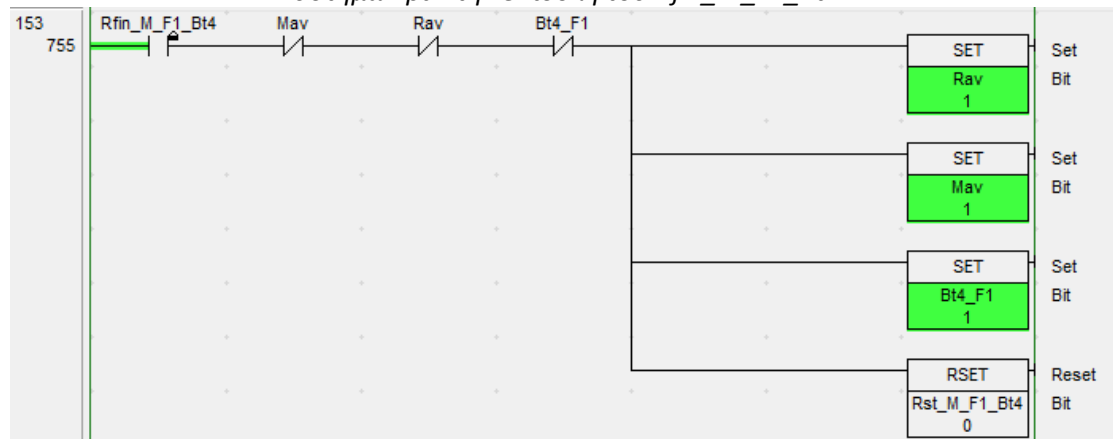
Σύστημα πριν την επιβολή του Mfinished_F1



Σύστημα μετά την επιβολή του Mfinished_F1



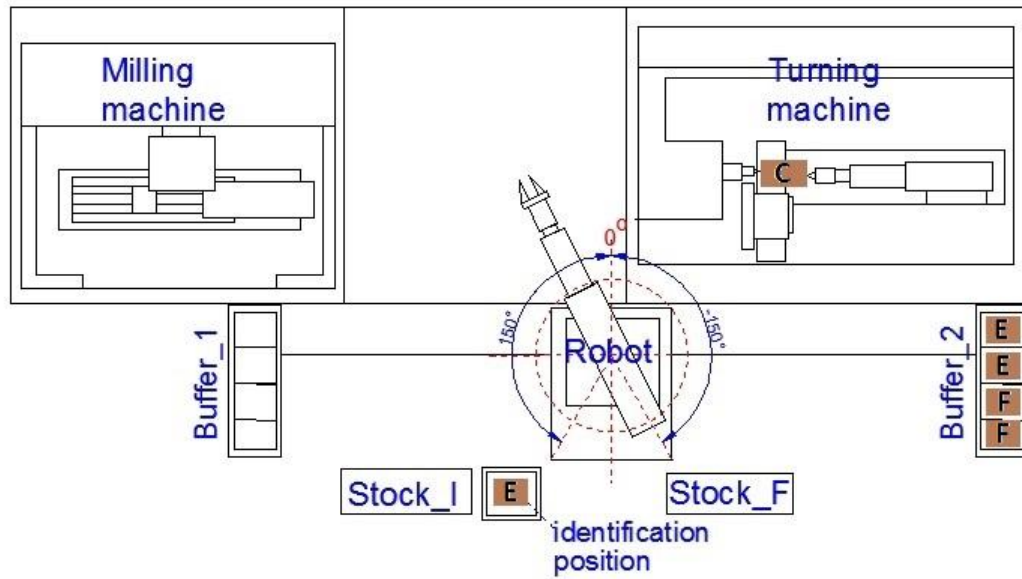
Σύστημα πριν την επιβολή του Rfin_M_F1_Bt4



Σύστημα μετά την επιβολή του Rfin_M_F1_Bt4

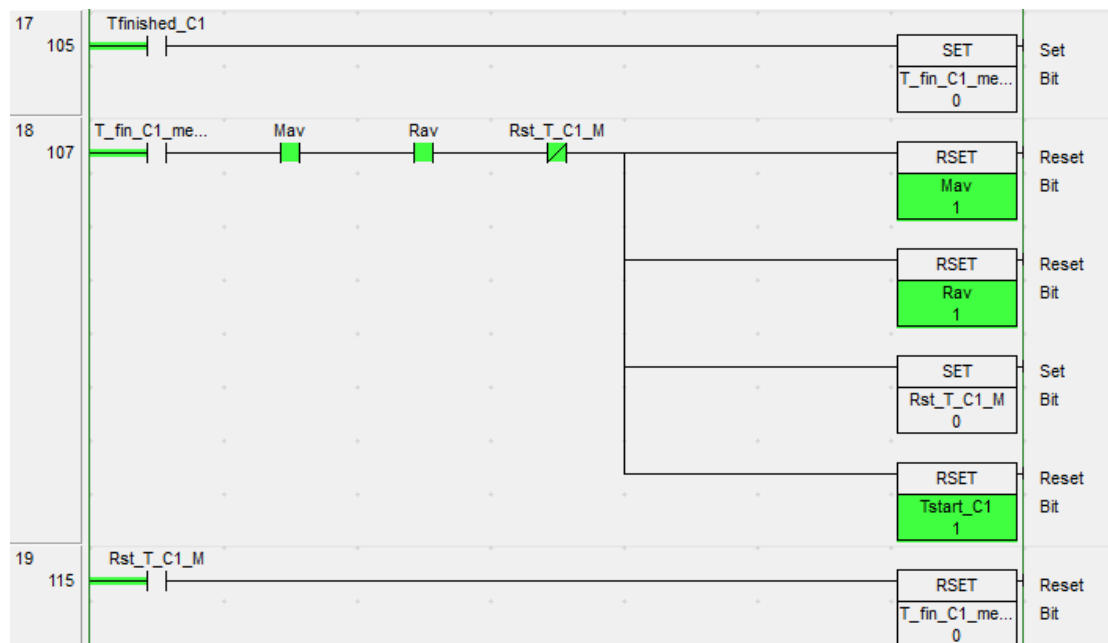
Αφού ολοκληρωθεί και η μεταφορά-αποθήκευση του τέταρτου τεμαχίου (το plc λαμβάνει σήμα Rfin_M_F1_Bt4) το buffer είναι πλήρες.

Σε αυτό το σημείο, το σύστημα δεν μπορεί να προχωρήσει σε άλλη κατεργασία ούτε να εισάγει άλλο τεμάχιο αν δεν ολοκληρωθεί η κατεργασία C1 καθώς το buffer έχει γεμίσει και η μηχανή τόννευσης δεν είναι διαθέσιμη.

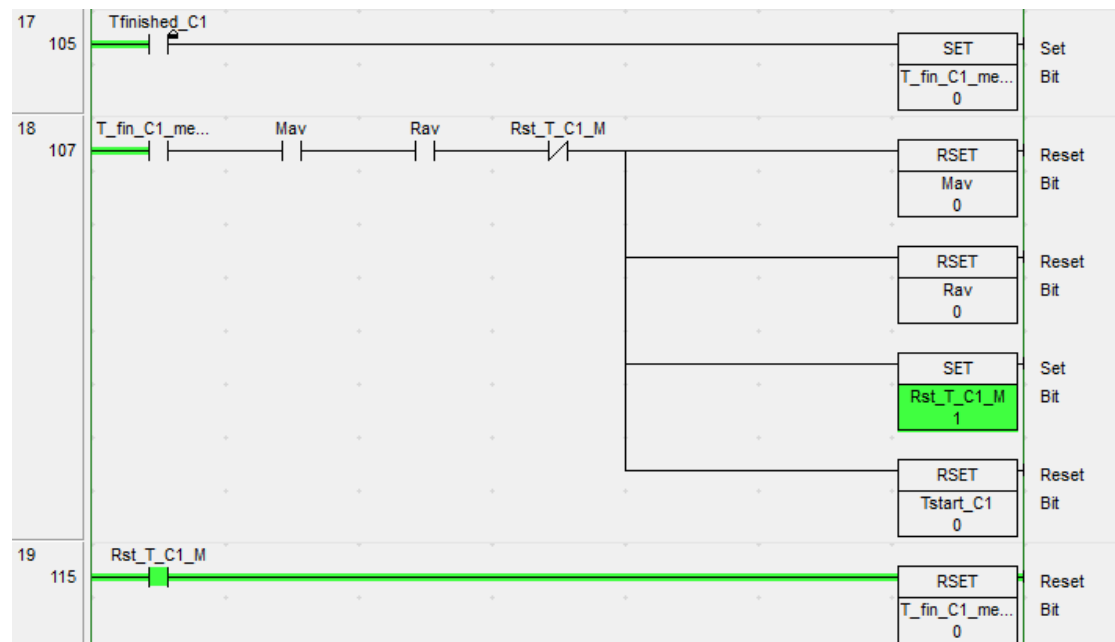


Τρέχουσα κατάσταση συστήματος

Μόλις ολοκληρωθεί τελικά η κατεργασία C1, το plc λαμβάνει σήμα Tfinished_C1 από τον ελεγκτή του τόρνου και αμέσως στέλνει σήμα Rst_T_C1_M στο ρομπότ για τη μεταφορά του τεμαχίου στη μηχανή φρεζαρίσματος.

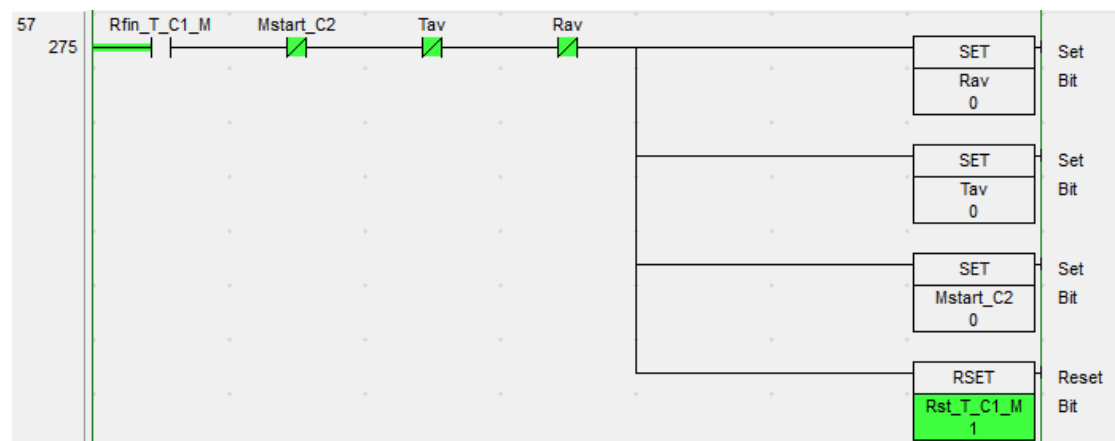


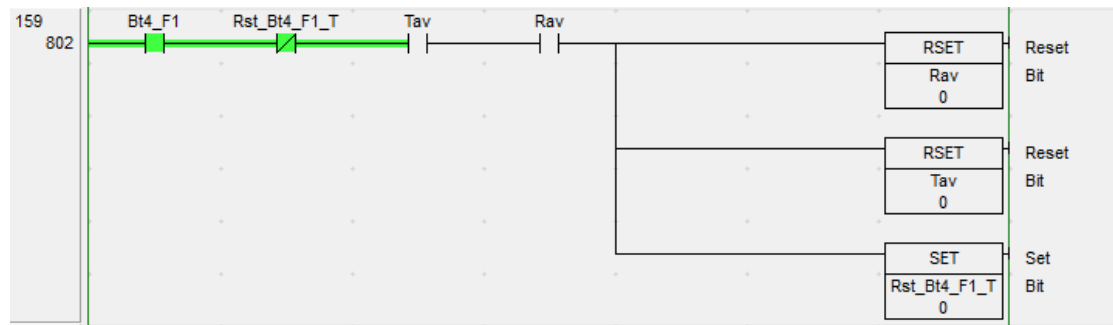
Σύστημα πριν την επιβολή του Tfinished_C1



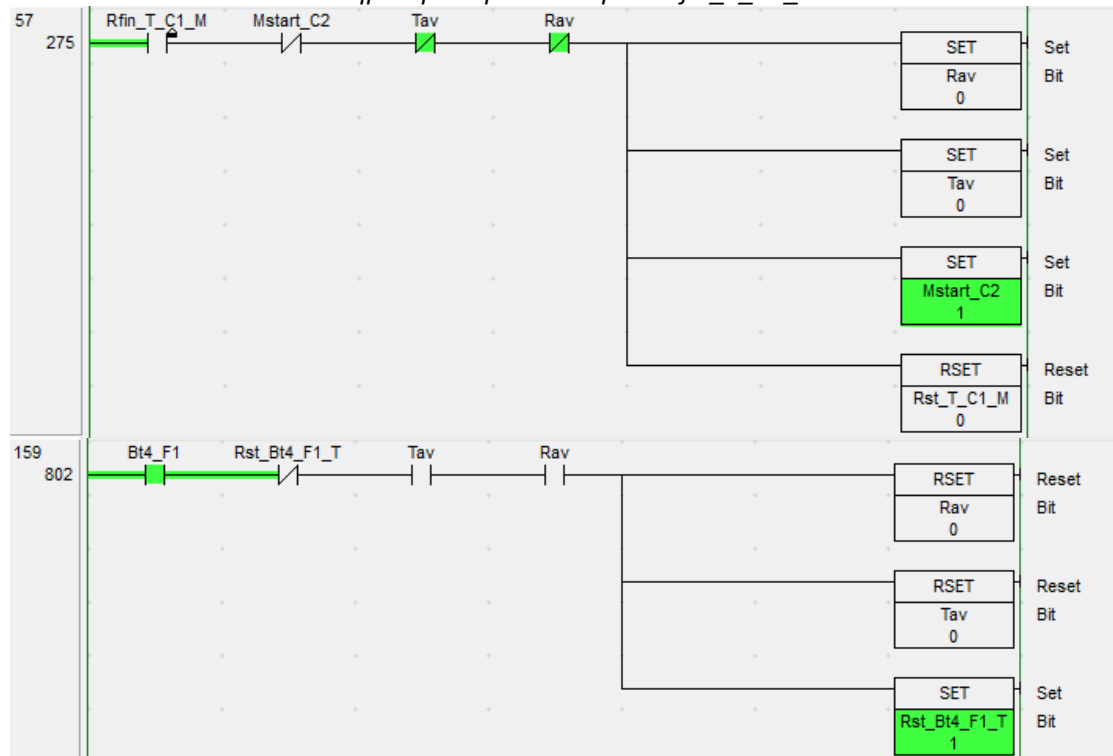
Σύστημα μετά την επιβολή του Tfinished_C1

Μόλις ολοκληρωθεί η μεταφορά του τεμαχίου C στη φρέζα το plc λαμβάνει το αντίστοιχο σήμα Rfin_T_C1_M από το ρομπότ και αμέσως στέλνει σήμα Mstart_C2 στον ελεγκτή της φρέζας να ξεκινήσει την κατεργασία. Ταυτόχρονα όμως με την ολοκλήρωση της μεταφοράς του C από τη μία μηχανή στην άλλη ελευθερώνεται και το ρομπότ. Όπως είχε μελετηθεί και κατά το σχεδιασμό του συστήματος, στη φάση αυτή το ρομπότ δεν εισάγει νέο τεμάχιο στο σύστημα αλλά παίρνει ημικατεργασμένο τεμάχιο από το buffer.



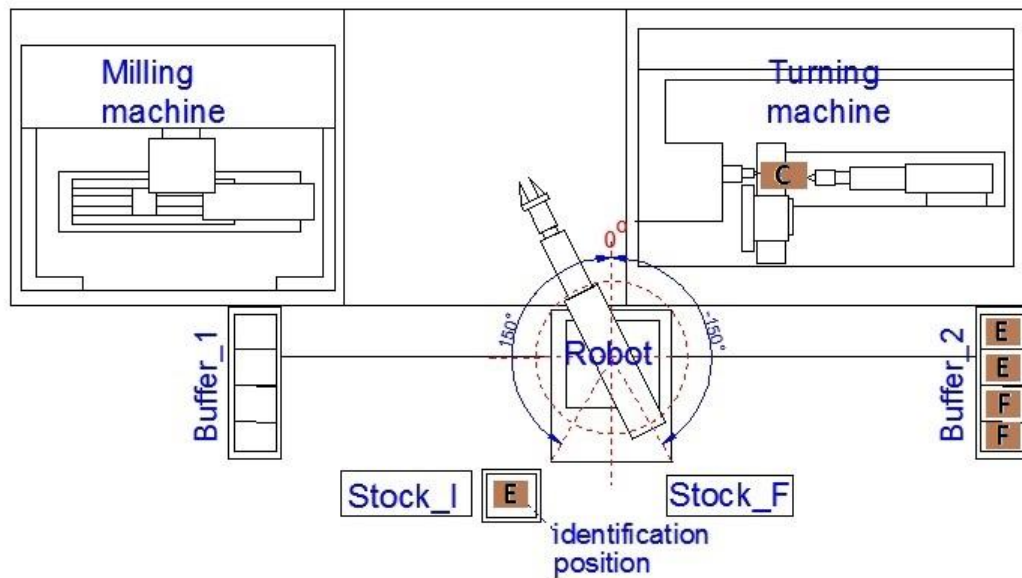


Σύστημα πριν την επιβολή του Rfin_T_C1_M

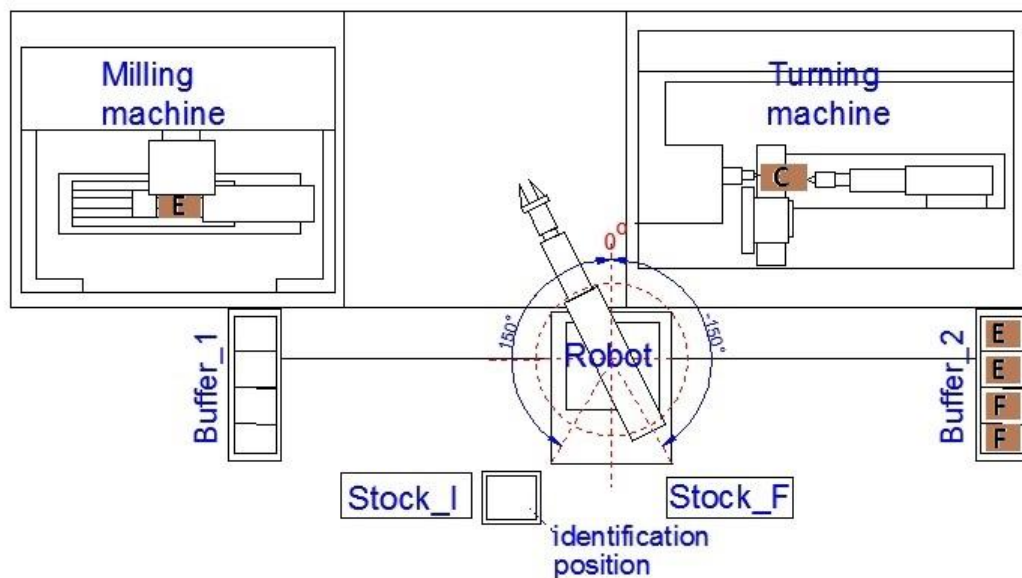


Σύστημα μετά την επιβολή του Rfin_T_C1_M

Όπως αναφέρθηκε και σε προηγούμενα κεφάλαια, με αυτόν τρόπο αποφεύγονται καταστάσεις αδιεξόδου. Το σύστημα μετά τη μεταφορά του C από τον τόρνο στη φρέζα βρίσκεται στην κατάσταση που απεικονίζεται παρακάτω. Αν σε αυτή τη φάση εισαχθεί τεμάχιο B (τόρνευση-φρεζάρισμα) και η πρώτη κατεργασία ολοκληρωθεί πριν από την C2 που βρίσκεται αυτή τη στιγμή σε εξέλιξη, το ρομπότ δε θα μπορεί να τοποθετήσει το τεμάχιο B ούτε στη φρέζα ούτε στο αντίστοιχο buffer αποθήκευσης. Επομένως το σύστημα θα έρθει σε κατάσταση αδιεξόδου.

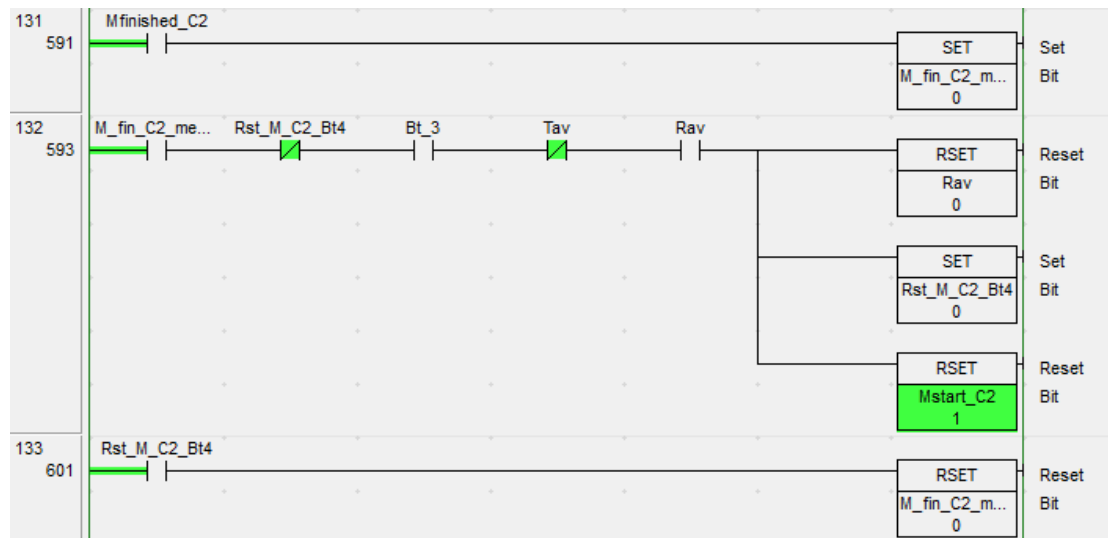


Τρέχουσα κατάσταση συστήματος

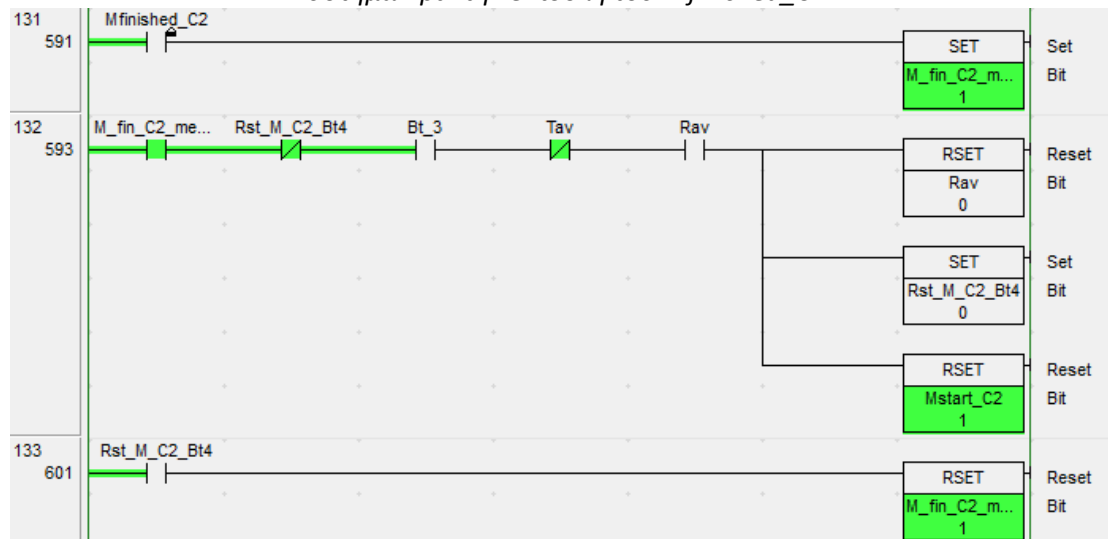


Κατάσταση συστήματος αν εισαχθεί νέο τεμάχιο - ενδεχόμενο deadlock σε περίπτωση που η κατεργασία B1 (milling machine) ολοκληρωθεί πριν την C2 (turning machine)

Κάποια στιγμή η κατεργασία C2 ολοκληρώνεται οπότε το plc λαμβάνει σήμα Mfinished_C2 από τον ελεγκτή της φρέζας. Όπως φαίνεται παρακάτω δεν ξεκινά αμέσως η μεταφορά του στον τόρνο για περαιτέρω επεξεργασία. Αυτό συμβαίνει γιατί το ρομπότ δεν έχει ακόμα αποδεσμευτεί από την προηγούμενη ενέργεια. Επειδή το σύστημα θα πρέπει να θυμάται ότι ολοκληρώθηκε μια κατεργασία δημιουργήθηκε βοηθητική μνήμη M_fin_C2_memory στην οποία γίνεται reset όταν ξεκινήσει η εκφόρτωση-μεταφορά του τεμαχίου από τη μηχανή δηλαδή από το Rst_M_C2_Bt4.

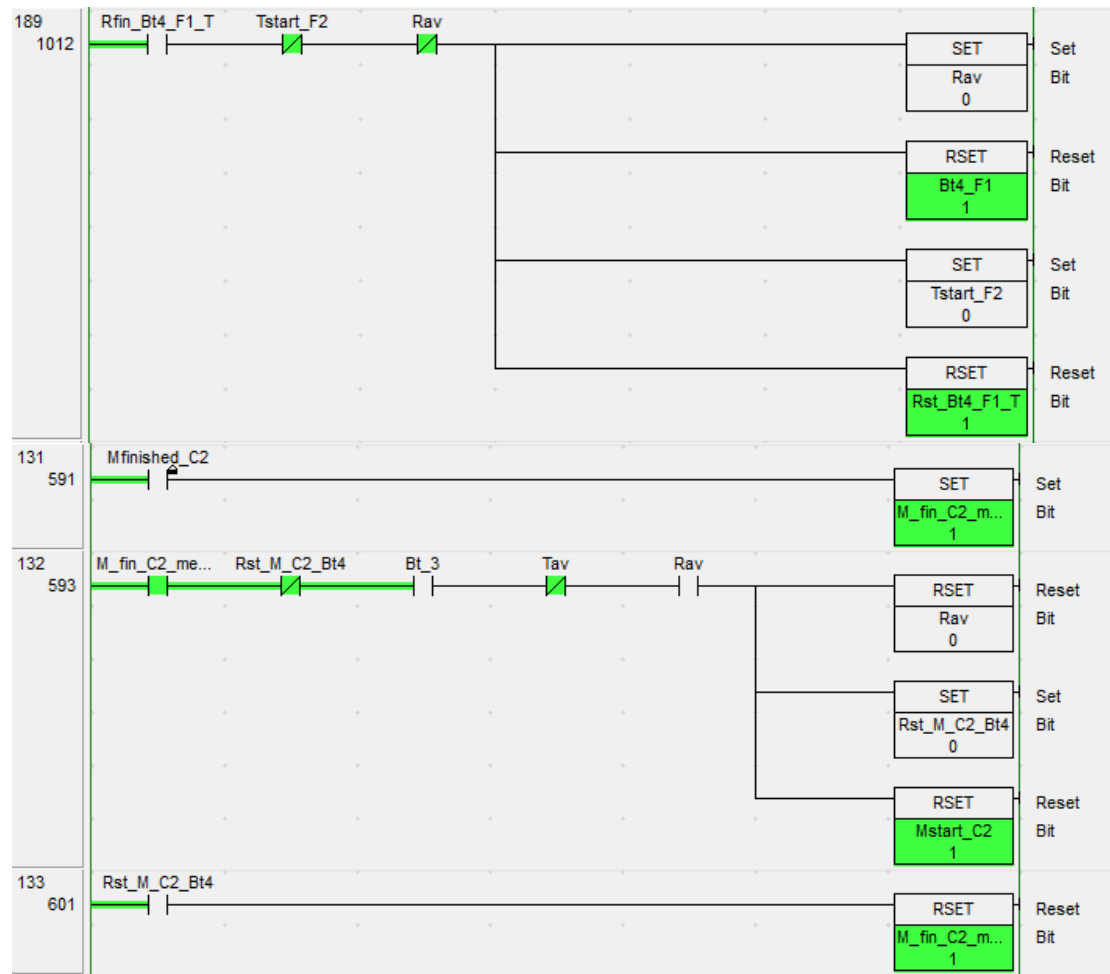


Σύστημα πριν την επιβολή του Mfinished_C2

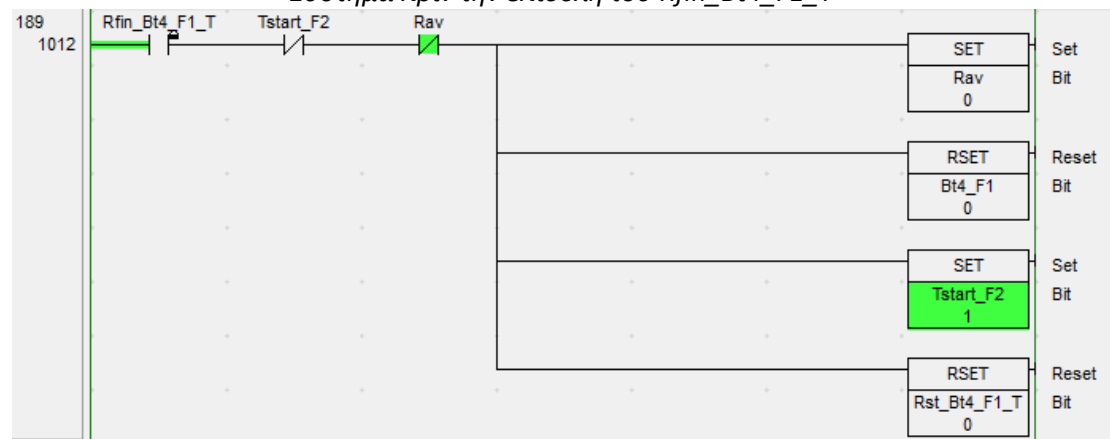


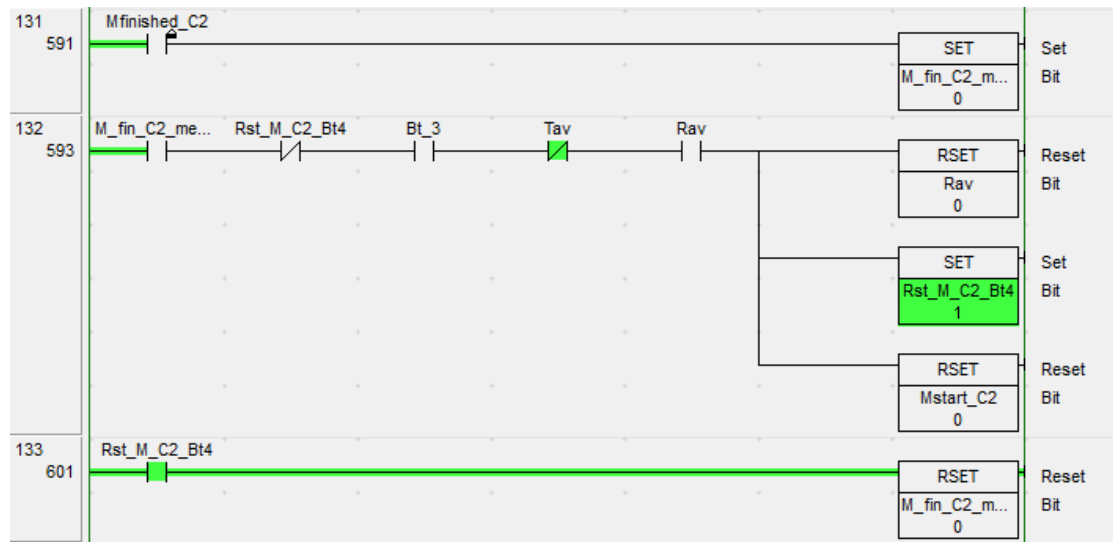
Σύστημα μετά την επιβολή του Mfinished_C2

Κάποια στιγμή ολοκληρώνεται η μεταφορά του τεμαχίου F από το buffer στον τόρνο και στέλνεται το σήμα Rfin_Bt4_F1_T στο plc. Αμέσως, το plc στέλνει σήμα Tstart_F2 στον ελεγκτή του τόρνου και ταυτόχρονα Rst_M_C_Bt4 στο ρομπότ να ξεκινήσει τη μεταφορά του τεμαχίου C από τη φρέζα στο buffer Bt.

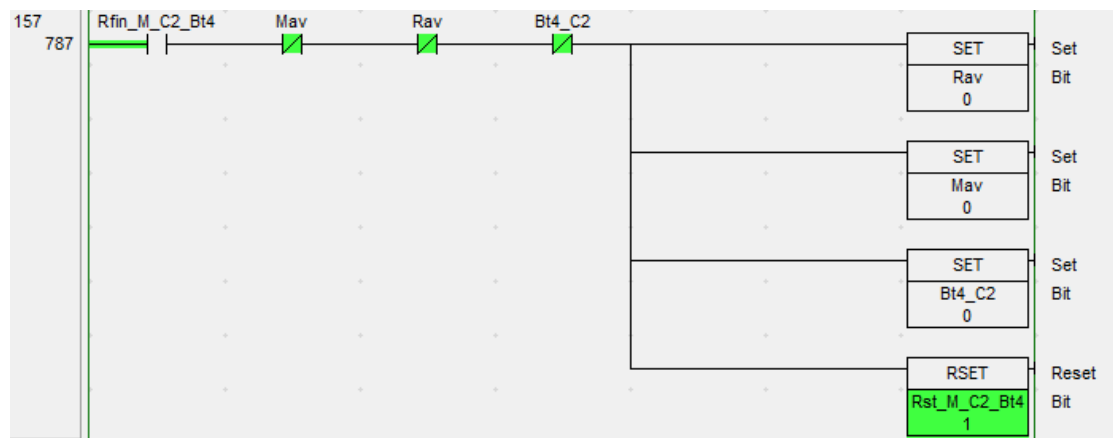
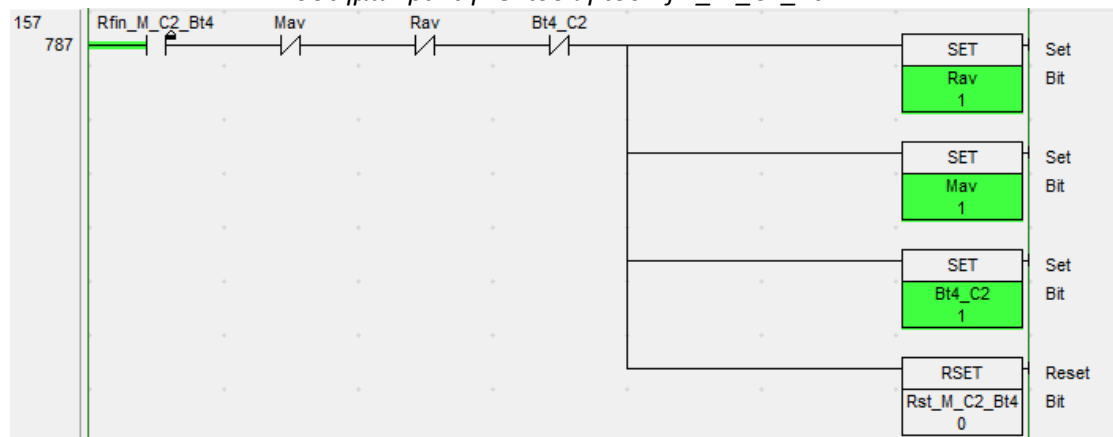


Σύστημα πριν την επιβολή του `Rfin_Bt4_F1_T`

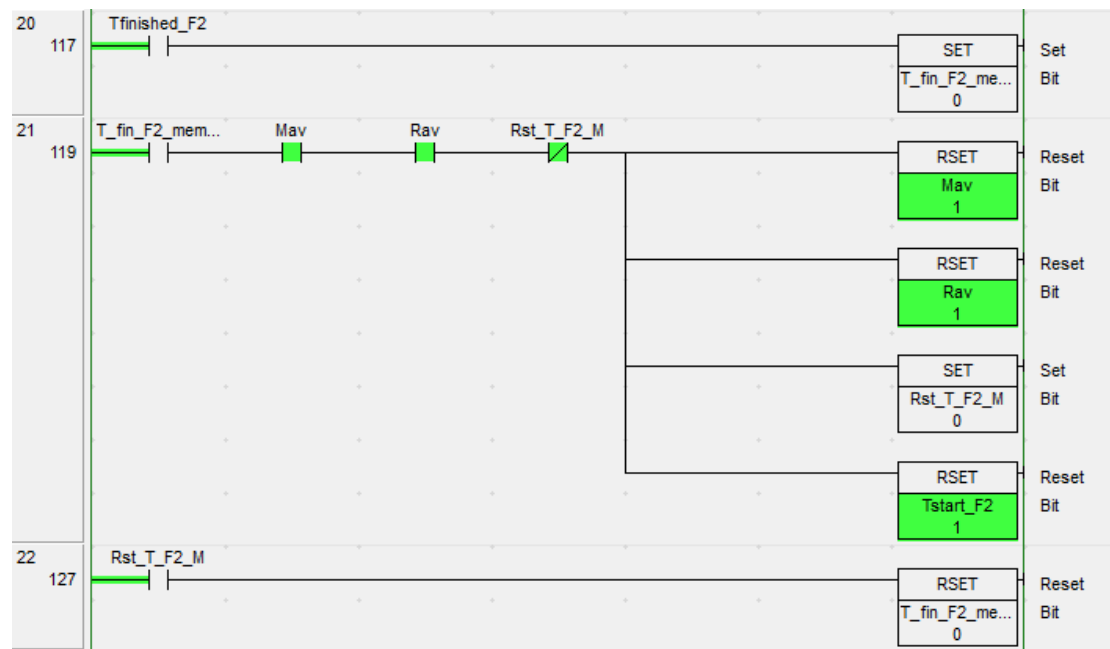


Σύστημα μετά την επιβολή του `Rfin_Bt4_F1_T`

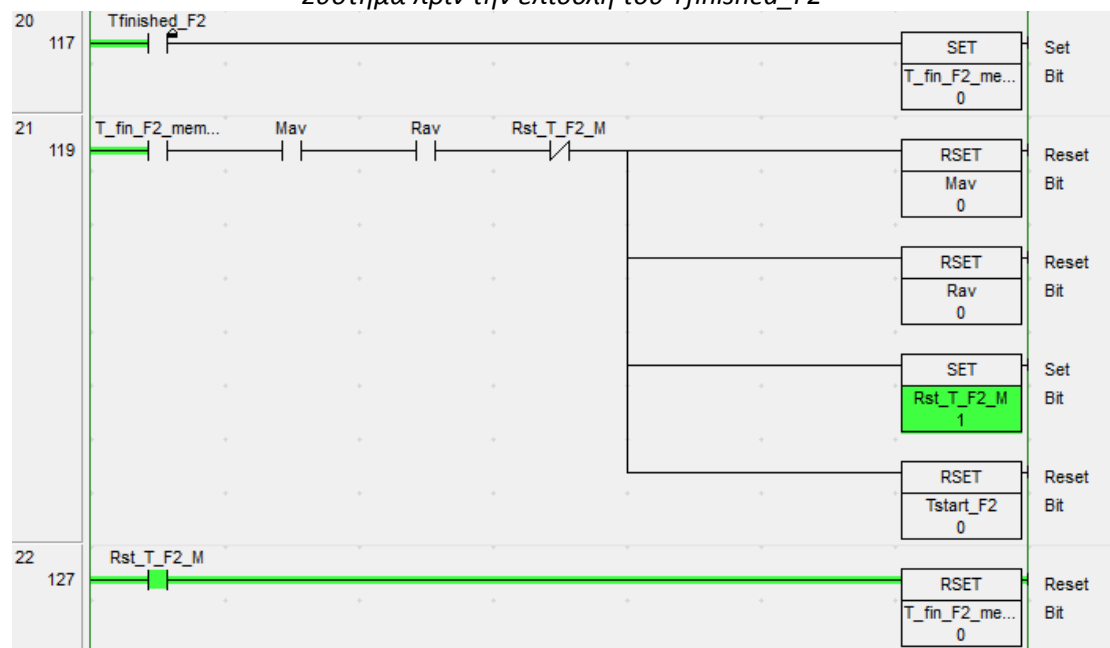
Όταν ολοκληρωθεί και η μεταφορά του τεμαχίου C, το plc λαμβάνει σήμα `Rfin_M_C2_Bt4`.

Σύστημα πριν την επιβολή του `Rfin_M_C2_Bt4`Σύστημα μετά την επιβολή του `Rfin_M_C2_Bt4`

Μόλις ολοκληρωθεί και η κατεργασία F2, το plc λαμβάνει σήμα Tfinished_F2 και αμέσως στέλνει σήμα Rst_T_F2_M για να ξεκινήσει η μεταφορά του τεμαχίου F στη φρέζα.

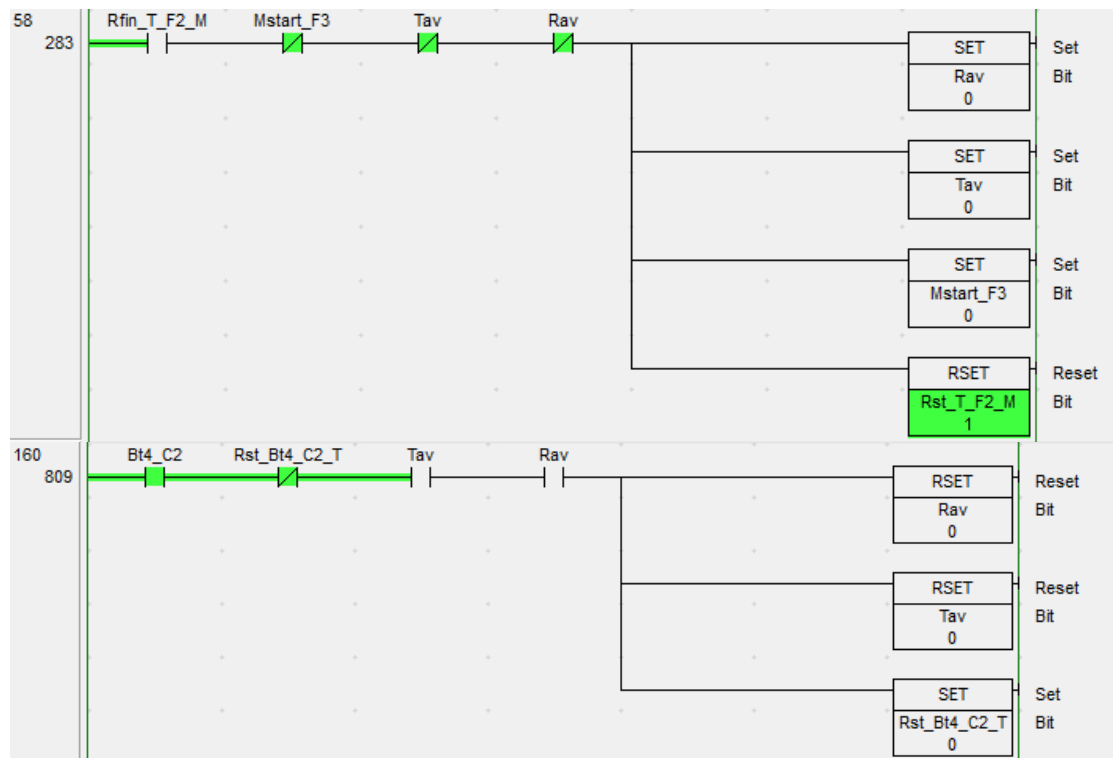


Σύστημα πριν την επιβολή του Tfinished_F2

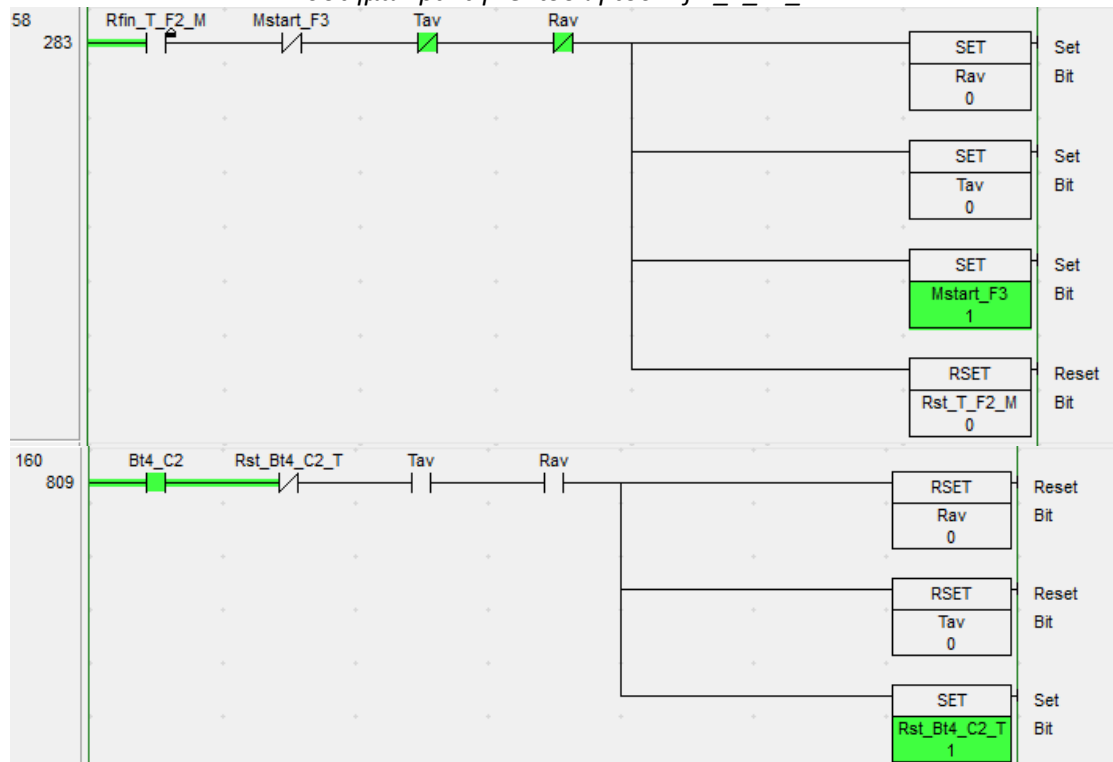


Σύστημα μετά την επιβολή του Tfinished_F2

Μόλις ολοκληρωθεί η μεταφορά-φόρτωση του F, το plc λαμβάνει σήμα Rfin_T_F2_M και αμέσως στέλνει σήμα Mstart_F3 στον ελεγκτή της φρέζας για να ξεκινήσει η κατεργασία F3. Εφόσον και το ρομπότ και ο τórνος ελευθερώνονται μετά τη μεταφορά-φόρτωση του F, ξεκινά παράλληλα και η μεταφορά του τεμαχίου C από το buffer στον τórνο (το plc στέλνει σήμα Rst_Bt4_C2_T στο ρομπότ).

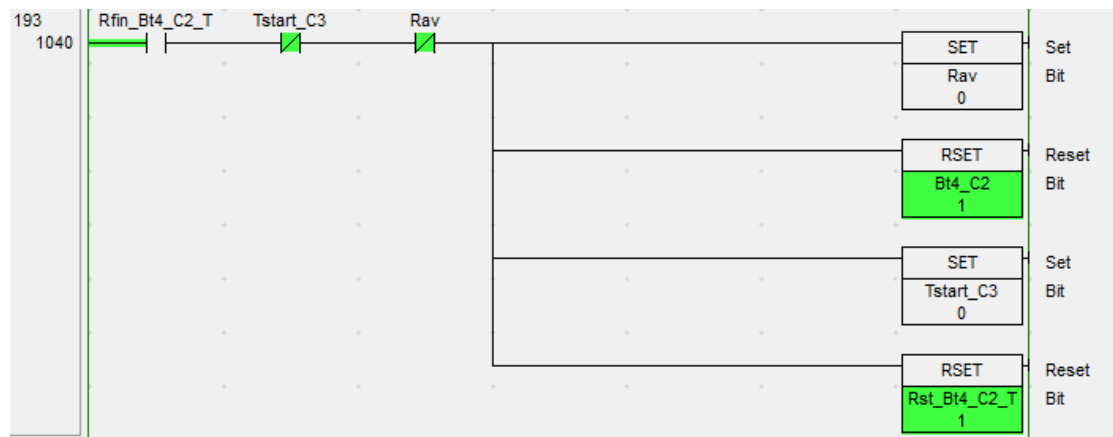


Σύστημα πριν την επιβολή του Rfin_T_F2_M

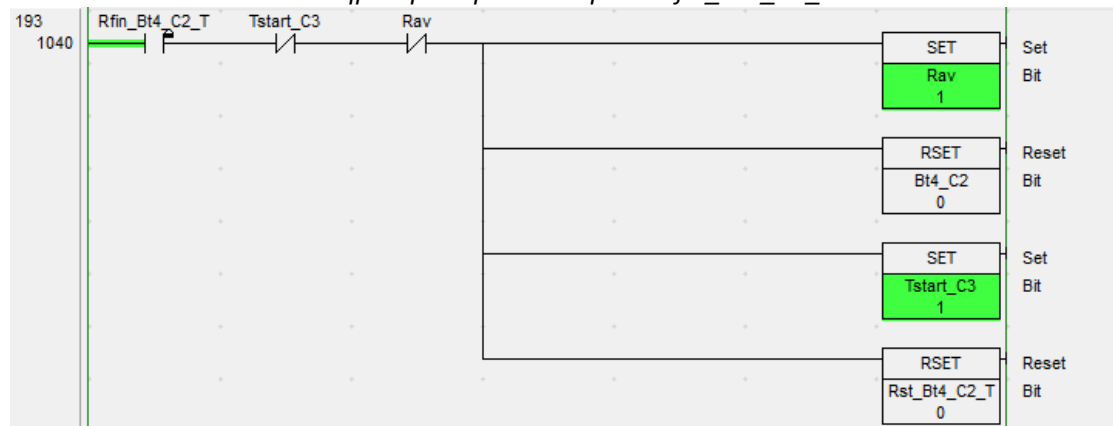


Σύστημα πριν την επιβολή του Rfin_T_F2_M

Μόλις ολοκληρωθεί η μεταφορά τεμαχίου C από το buffer στον τoρνο, το plc λαμβάνει σήμα Rfin_Bt4_C2_T από το ρομπότ και αμέσως στέλνει σήμα Tstart_C3 στον ελεγκτή του τoρνού για να ξεκινήσει η κατεργασία C3.

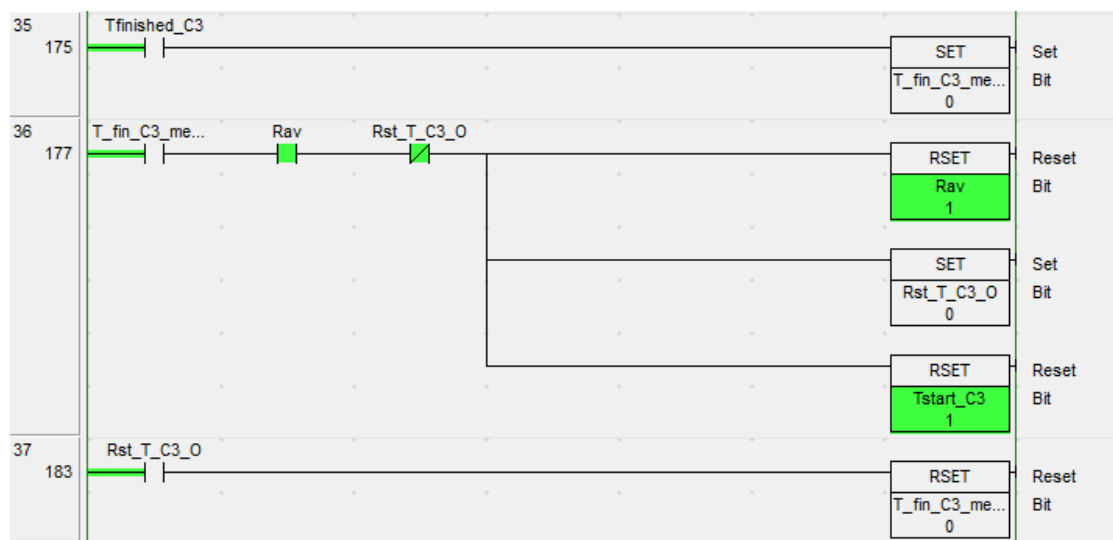


Σύστημα πριν την επιβολή του Rfin_Bt4_C2_T

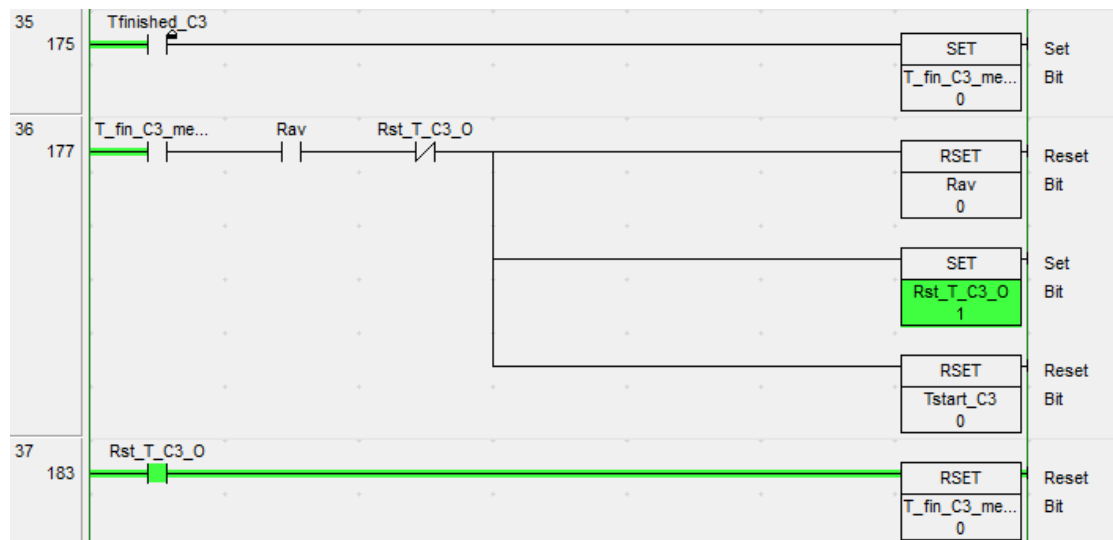


Σύστημα μετά την επιβολή του Rfin_Bt4_C2_T

Κάποια στιγμή τελειώνει η κατεργασία C3 οπότε το τεμάχιο C είναι έτοιμο. Το plc λαμβάνει σήμα Tfinished_C3. Εφόσον το ρομπότ είναι διαθέσιμο στέλνει σήμα Rst_T_C3_O στον ρομπότ για την εκφόρτωση-μεταφορά του τεμαχίου από τον τόρνο στη θέση εξόδου.

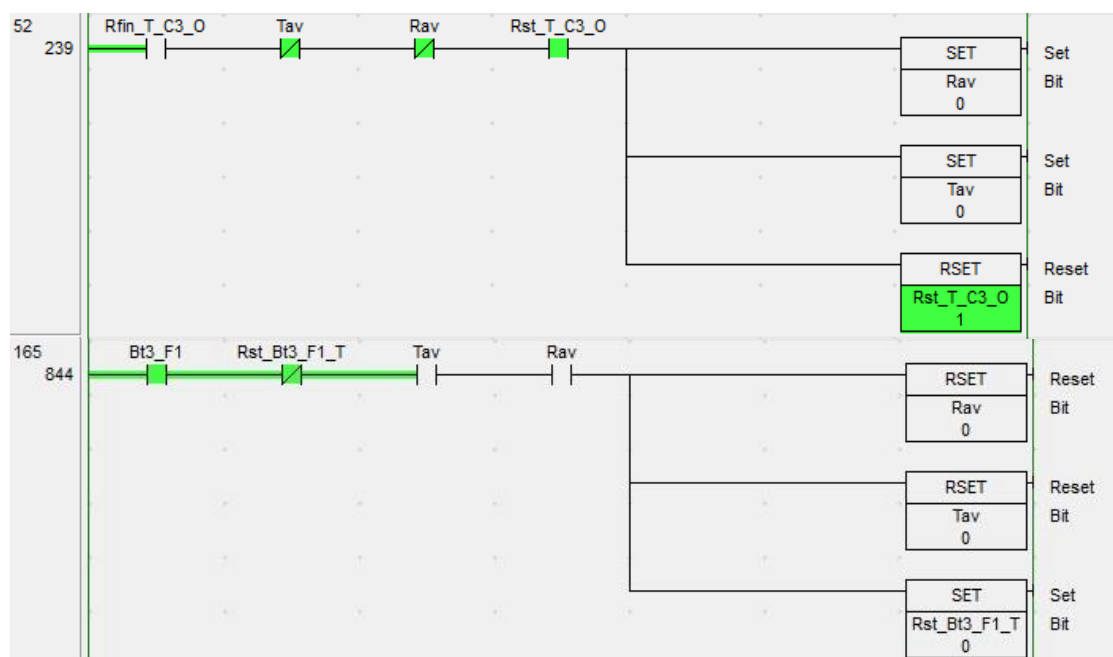


Σύστημα πριν την επιβολή του Tfinished_C3

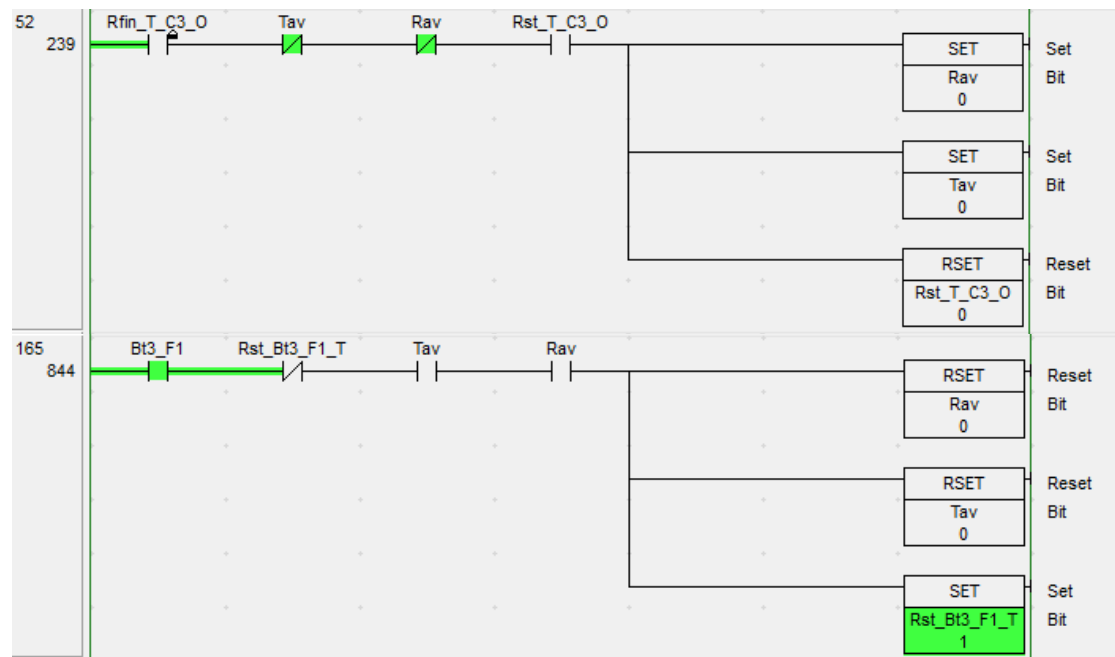


Σύστημα μετά την επιβολή του `Tfinished_C3`

Όταν ολοκληρωθεί η μεταφορά του τεμαχίου C στη θέση εξόδου, άρα και το ρομπότ ελευθερωθεί, το plc λαμβάνει σήμα `Rfin_T_C_O` και αμέσως στέλνει σήμα `Rst_Bt3_F1_T` για να ξεκινήσει η μεταφορά του τεμαχίου F από τη θέση 3 του buffer αναμονής στον τόρνο.

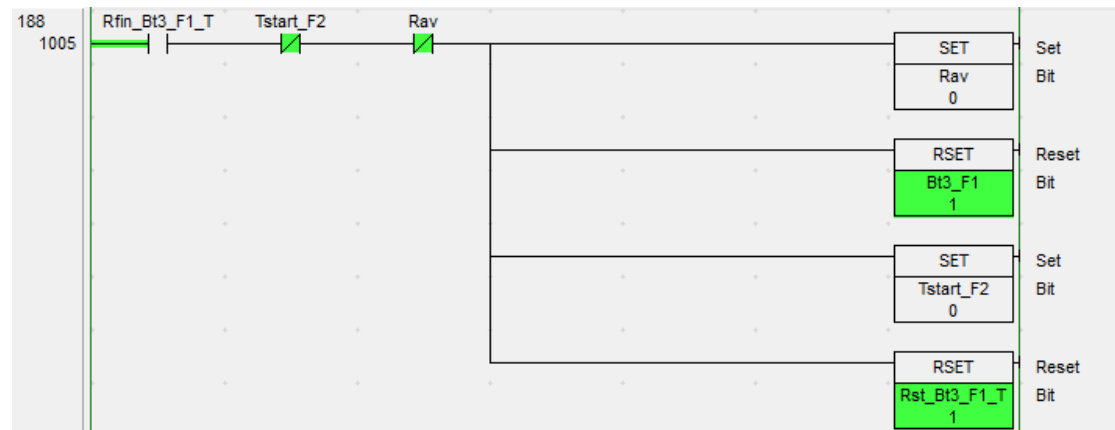


Σύστημα πριν την επιβολή του `Rfin_T_C3_O`

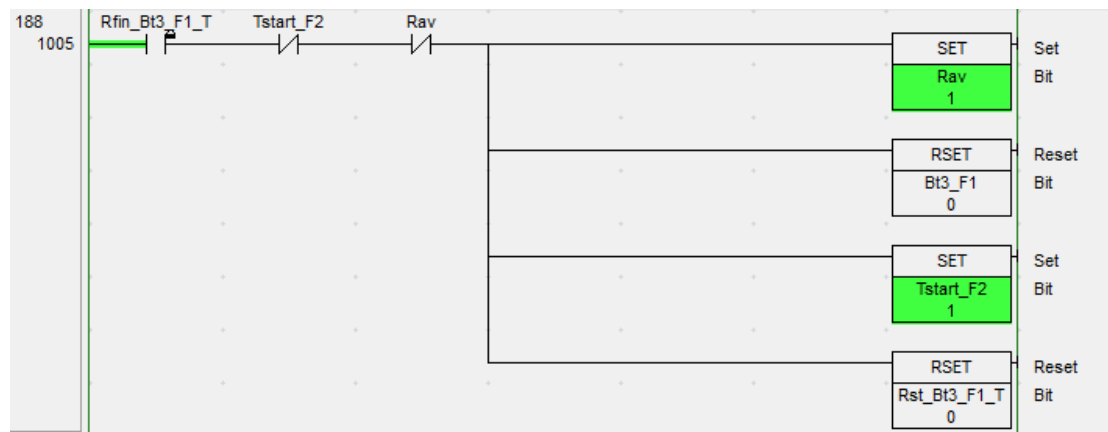


Σύστημα μετά την επιβολή του **Rfin_T_C3_O**

Αφού ολοκληρωθεί η μεταφορά, το plc λαμβάνει σήμα **Rfin_Bt3_F1_T** από το ρομπότ και αμέσως στέλνει σήμα **Tstart_F2** στον ελεγκτή του τόνου για να ξεκινήσει η κατεργασία F2.

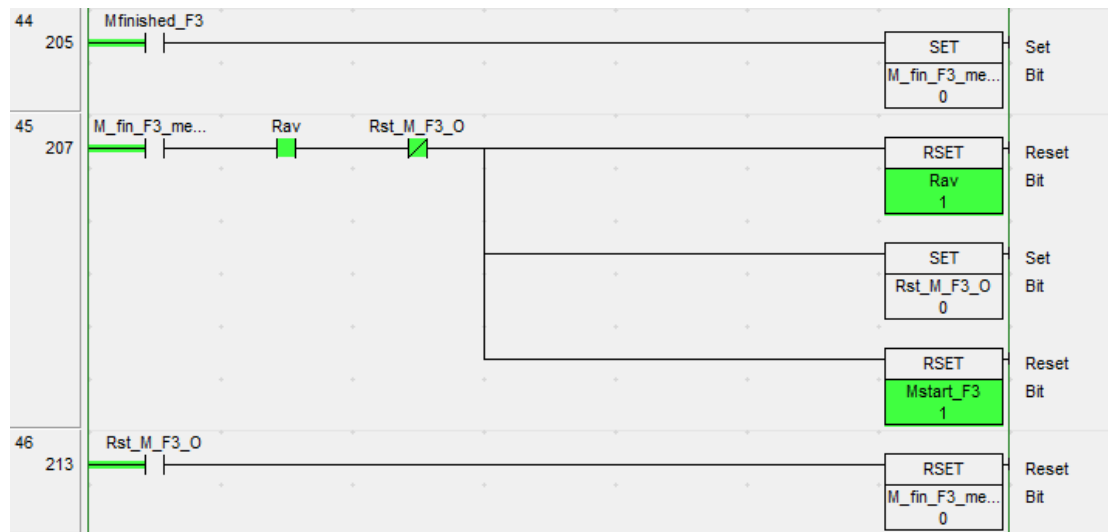


Σύστημα πριν την επιβολή του **Rfin_Bt3_F1_T**

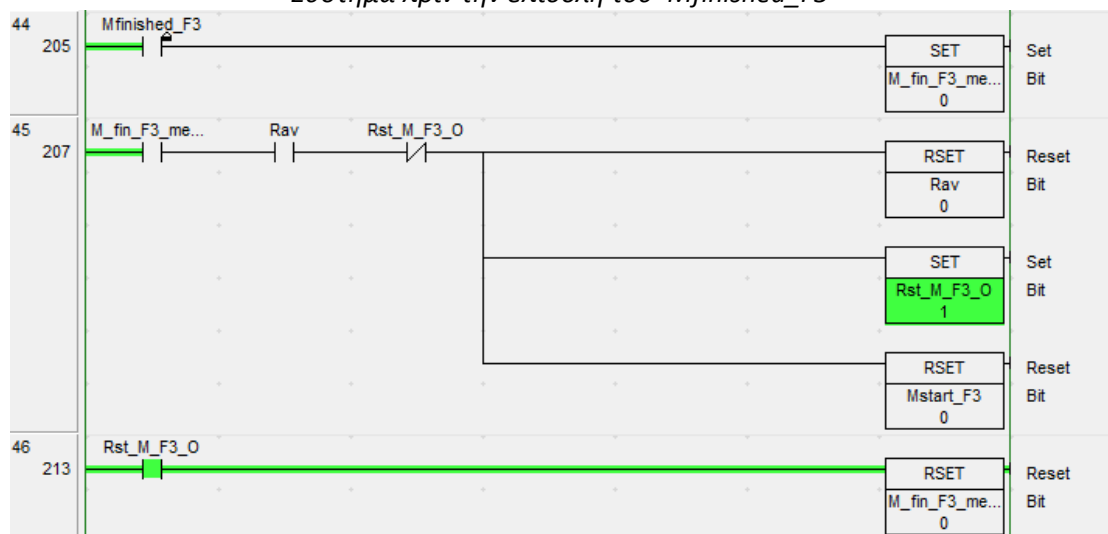


Σύστημα ματά την επιβολή του Rfin_Bt3_F1_T

Μόλις ολοκληρωθεί η κατεργασία F3, το plc λαμβάνει σήμα Mfinished_F3 από τον ελεγκτή της φρέζας και αμέσως στέλνει σήμα Rst_M_F3_O στο ρομπότ να ξεκινήσει την εκφόρτωση-μεταφορά του τεμαχίου F από τη φρέζα στη θέση εξόδου.

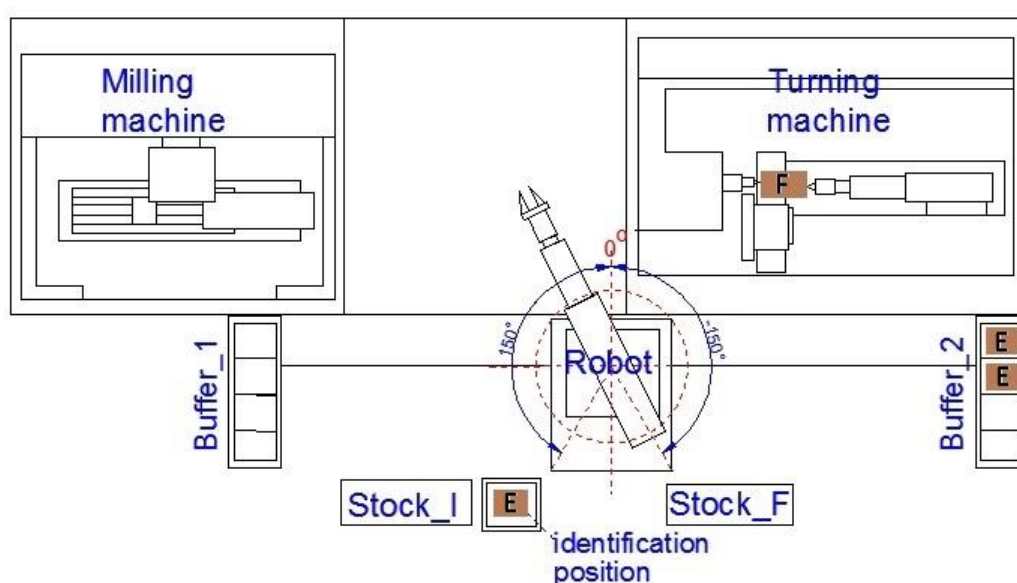


Σύστημα πριν την επιβολή του Mfinished_F3

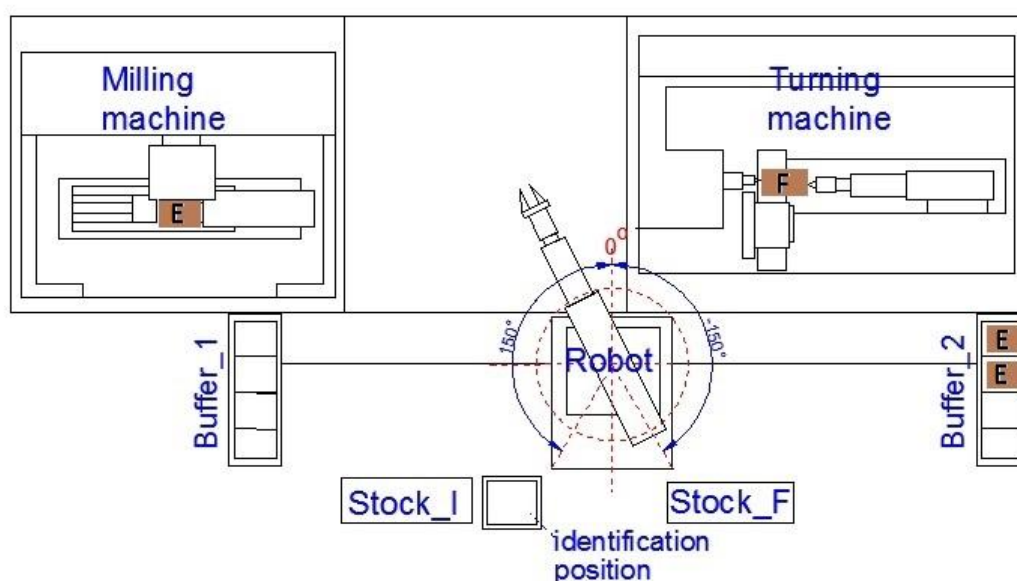


Σύστημα μετά την επιβολή του Mfinished_F3

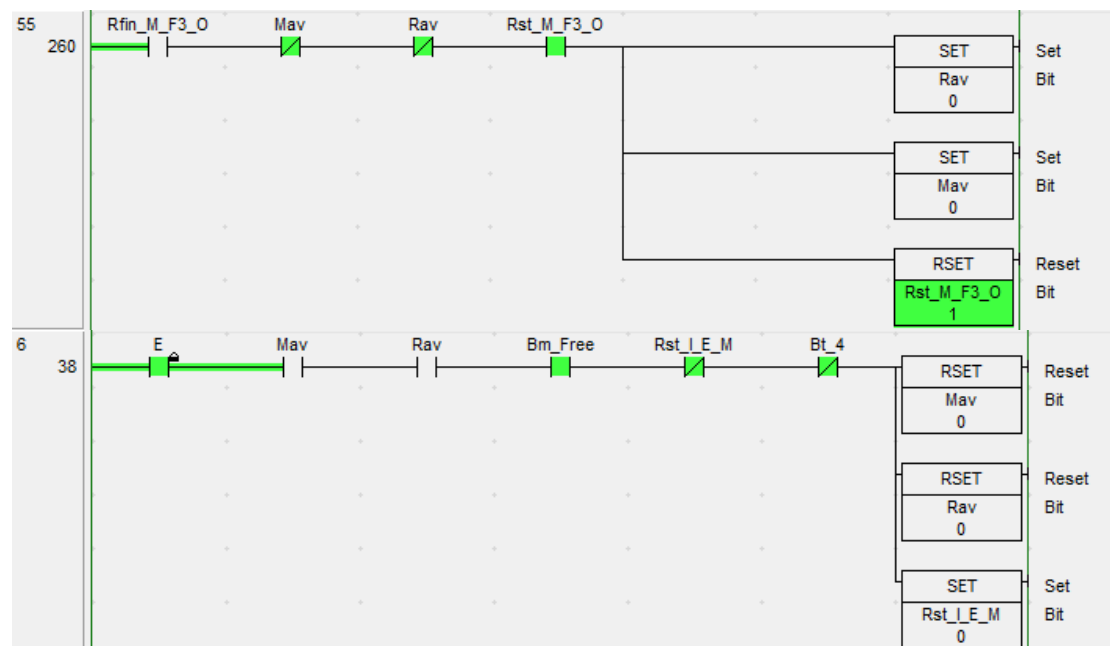
Μόλις ολοκληρωθεί η έξοδος του τεμαχίου F από το σύστημα, το plc λαμβάνει σήμα Rfin_M_F3_O από το ρομπότ και αμέσως στέλνει σήμα Rst_I_E_M στο ρομπότ για να ξεκινήσει η μεταφορά του τεμαχίου E από τη θέση εισόδου στη φρέζα. Σε αυτό το σημείο παρατηρείται ότι ενώ το buffer αναμονής δεν έχει αδειάσει, εισάγεται νέο τεμάχιο στο σύστημα. Αυτό συμβαίνει γιατί τα συγκεκριμένα τεμάχια που βρίσκονται 'σε αναμονή' απαιτούν επόμενη επεξεργασία στον τόρνο που τη δεδομένη στιγμή δεν είναι διαθέσιμος (βρίσκεται σε εξέλιξη κατεργασία F2). Ωστόσο, όπως αναφέρθηκε σε προηγούμενο βήμα, αν το buffer ήταν γεμάτο δε θα εισάγονταν νέο τεμάχιο ώστε να αποφευχθεί ενδεχόμενο κόλλημα του συστήματος (deadlock).



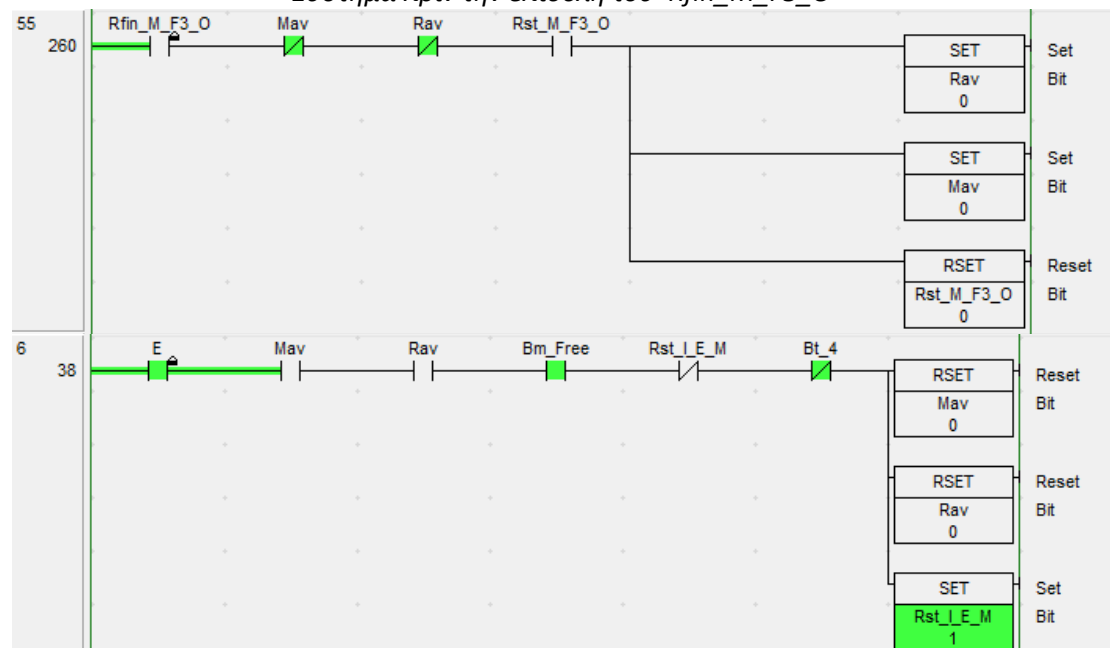
Τρέχουσα κατάσταση συστήματος



Κατάσταση συστήματος μετά την εισαγωγή του τεμαχίου E που περιμένει στη θέση εισόδου



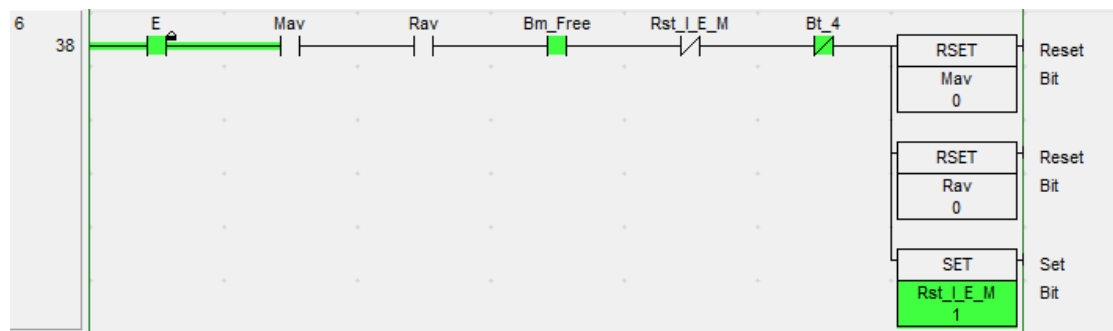
Σύστημα πριν την επιβολή του Rfin_M_F3_O



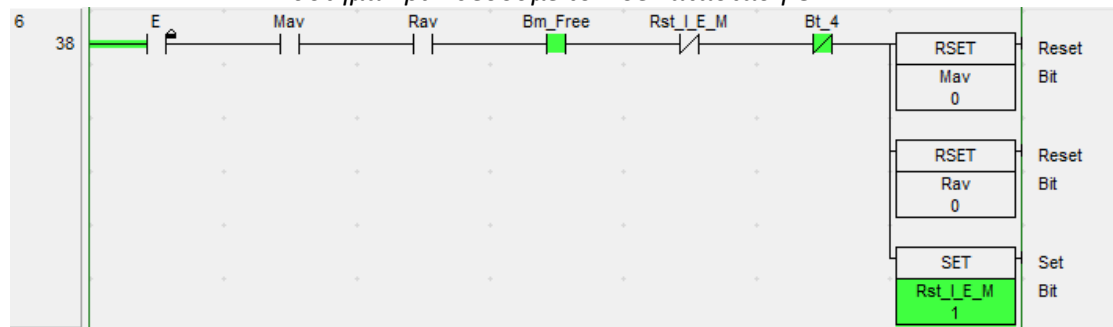
Σύστημα μετά την επιβολή του Rfin_M_F3_O

Σε αυτό το σημείο θέτουμε OFF το τεμάχιο E το οποίο ήταν το τελευταίο προς επεξεργασία. Από αυτό το σημείο και μετά, όπως έχει μελετηθεί κατά το σχεδιασμό, θα πρέπει να εκτελεστούν με τη σειρά όλες οι κατεργασίες που απομένουν στο process plan κάθε τεμαχίου που βρίσκεται μέσα στο σύστημα και τα έτοιμα τεμάχια να οδηγούνται στη θέση εξόδου.

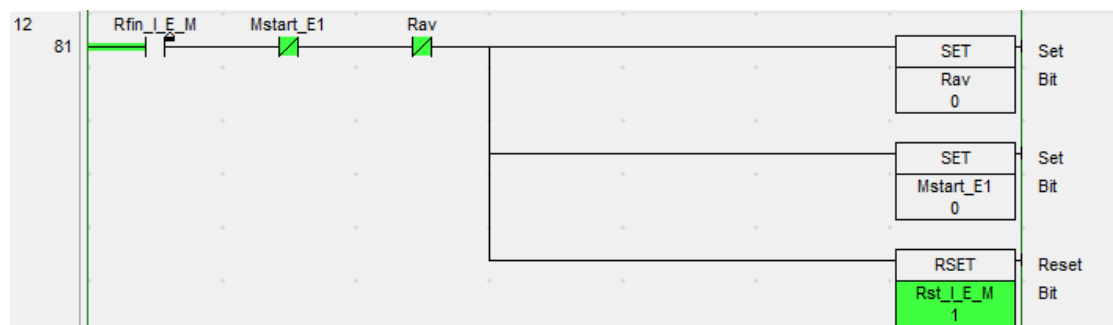
Παρακάτω παρουσιάζονται όλα τα σήματα εισόδου και οι αλλαγές στην κατάσταση του συστήματος μέχρι να εξέλθουν όλα τα τεμάχια από αυτό.



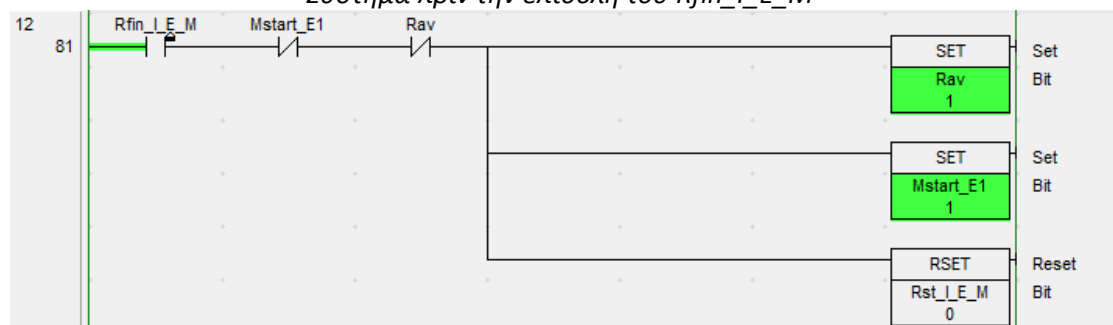
Σύστημα πριν θέσουμε το E σε κατάσταση OFF



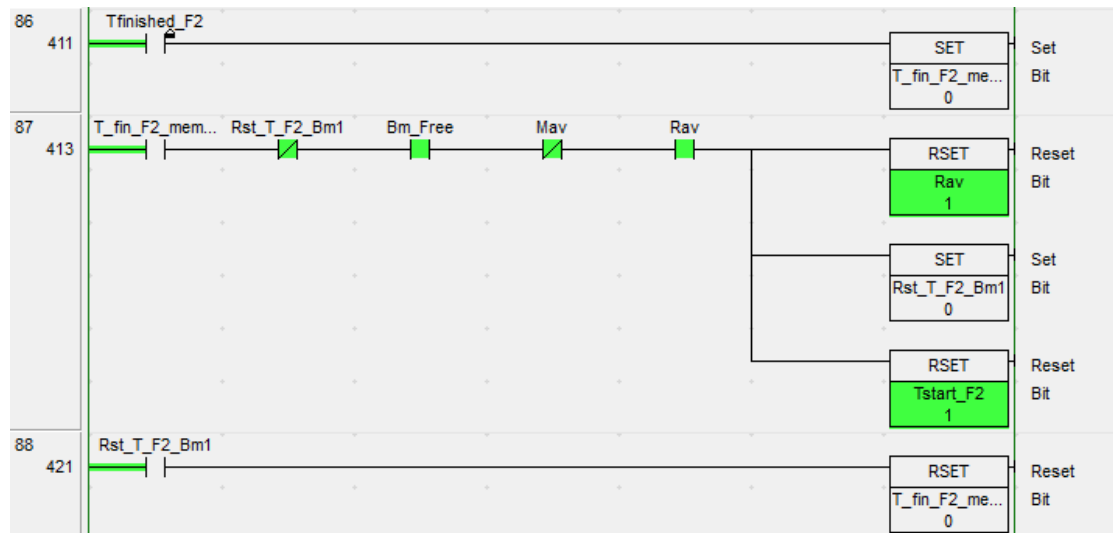
Σύστημα αφού θέσουμε το E σε κατάσταση OFF



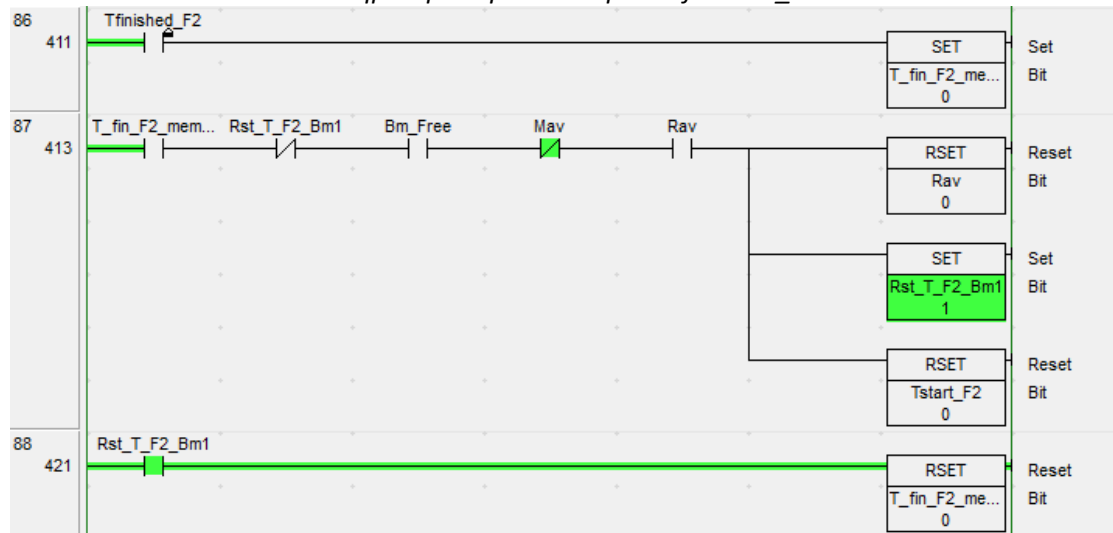
Σύστημα πριν την επιβολή του Rfin_I_E_M



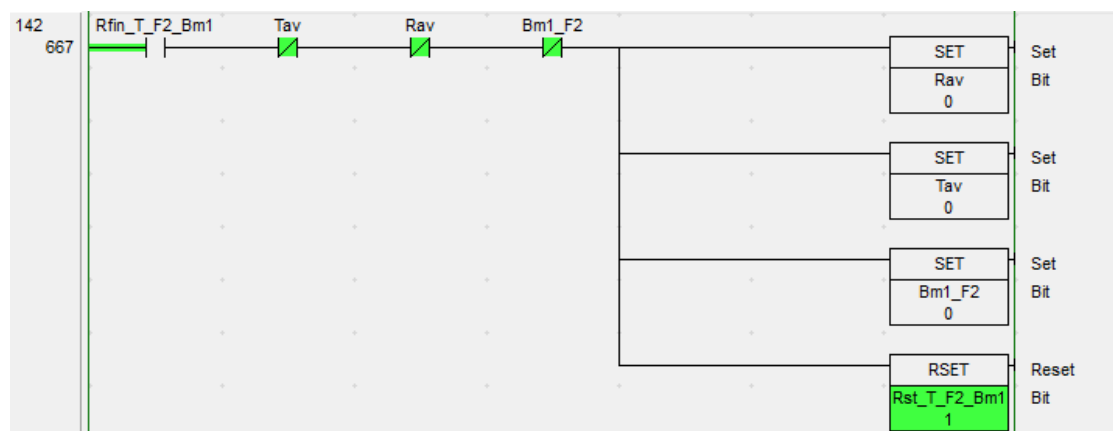
Σύστημα μετά την επιβολή του Rfin_I_E_M

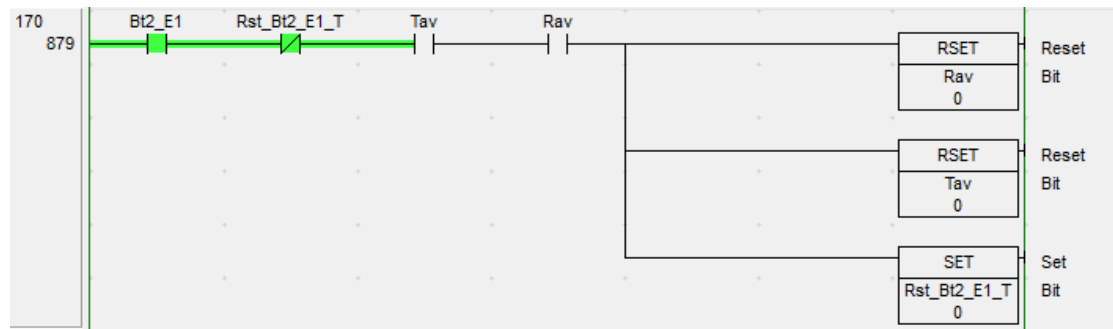


Σύστημα πριν την επιβολή του Tfinished_F2

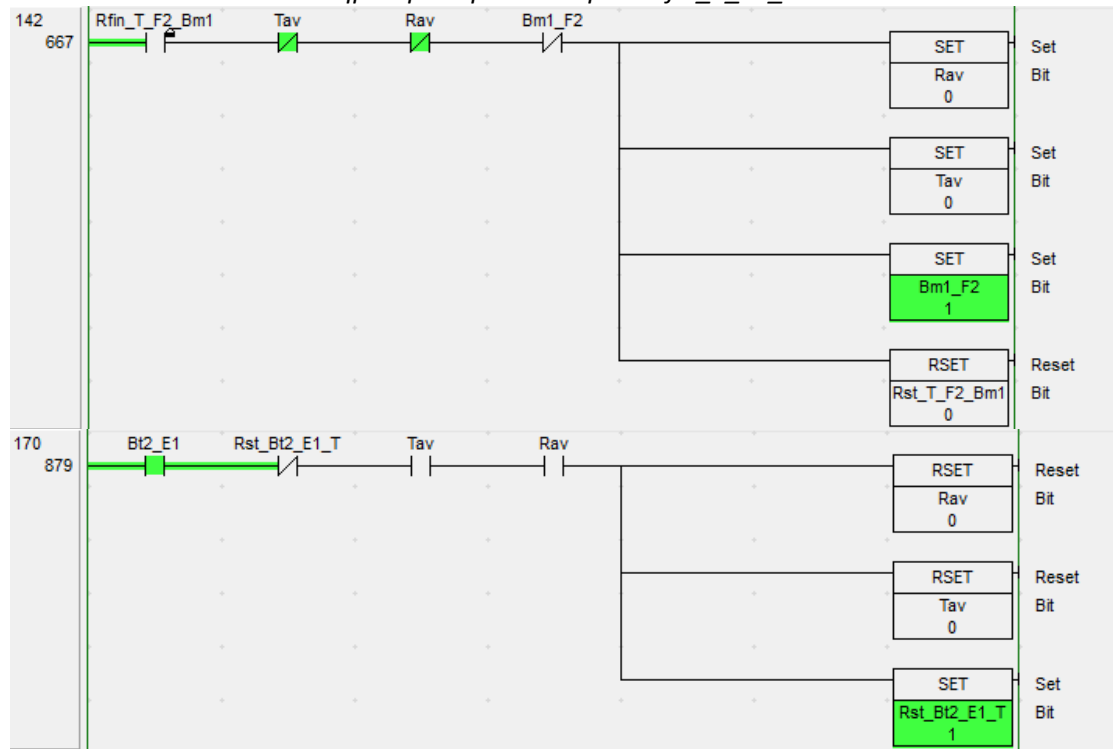


Σύστημα μετά την επιβολή του Tfinished_F2

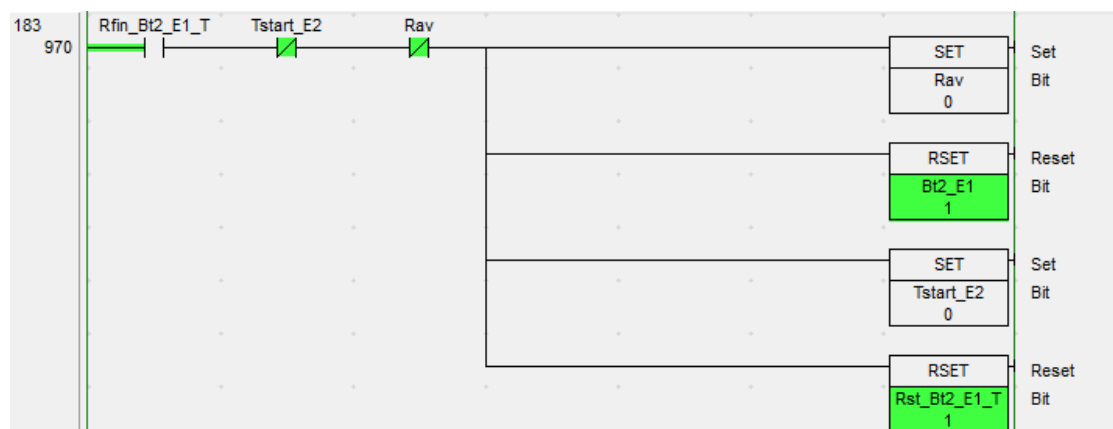




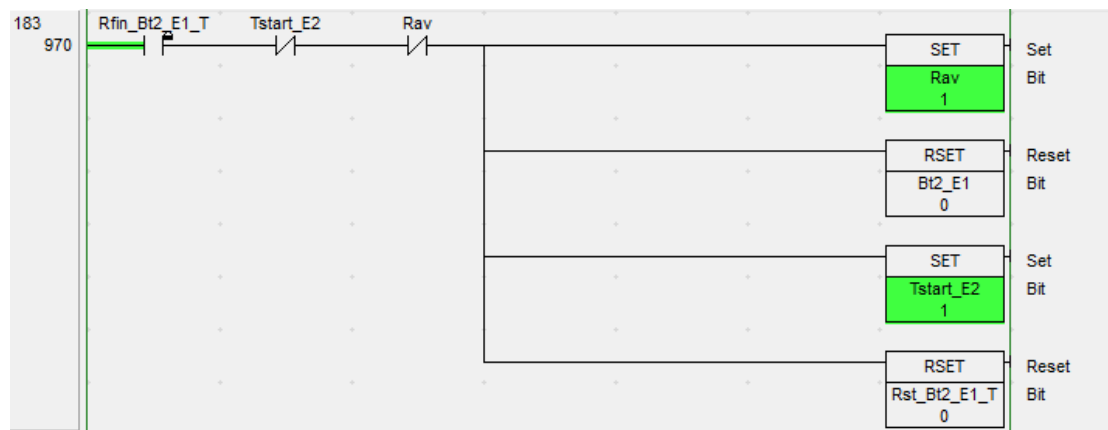
Σύστημα πριν την επιβολή του Rfin_T_F2_Bm1



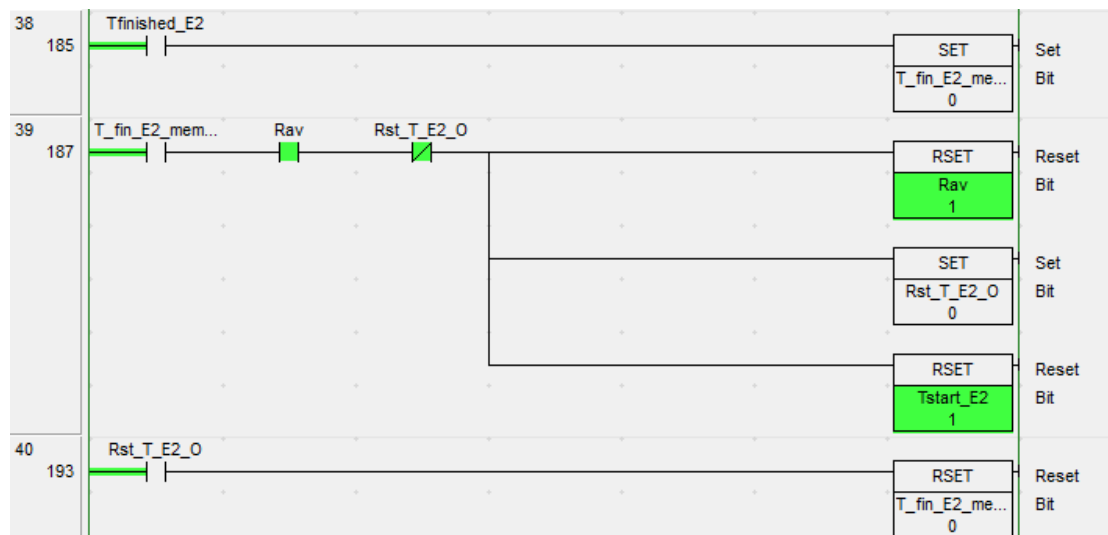
Σύστημα μετά την επιβολή του Rfin_T_F2_Bm1



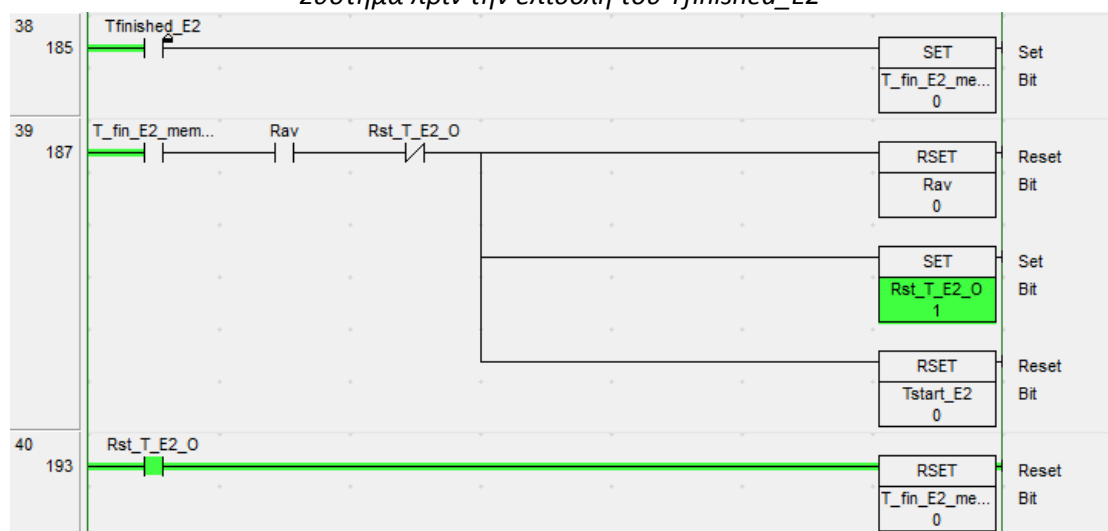
Σύστημα πριν την επιβολή του Rfin_Bt2_E1_T



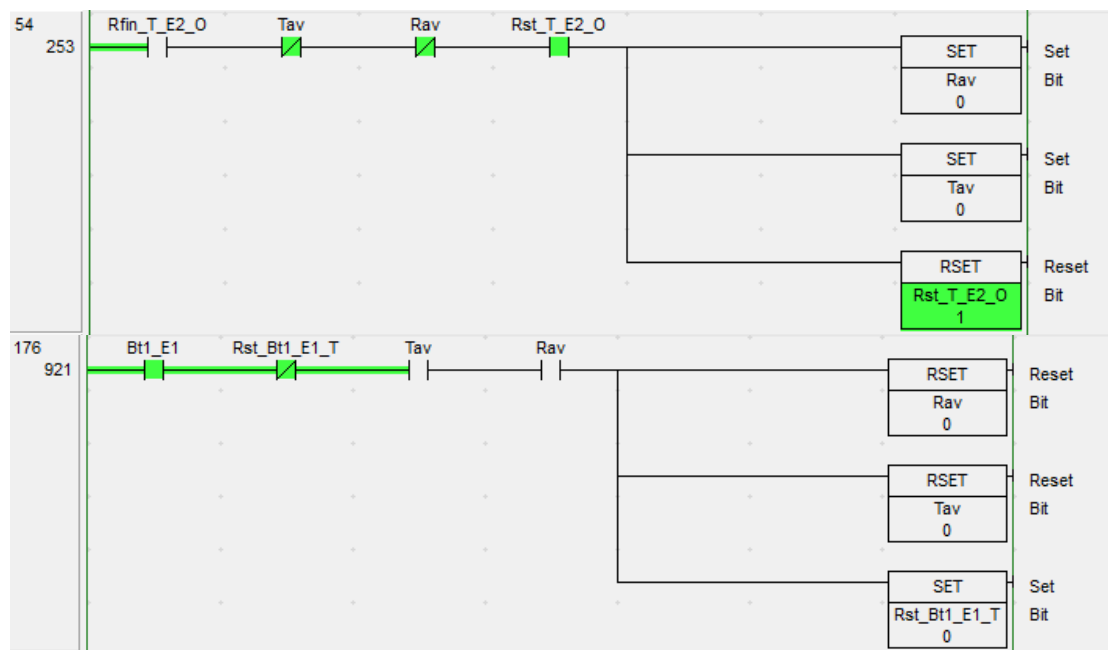
Σύστημα μετά την επιβολή του Rfin_Bt2_E1_T



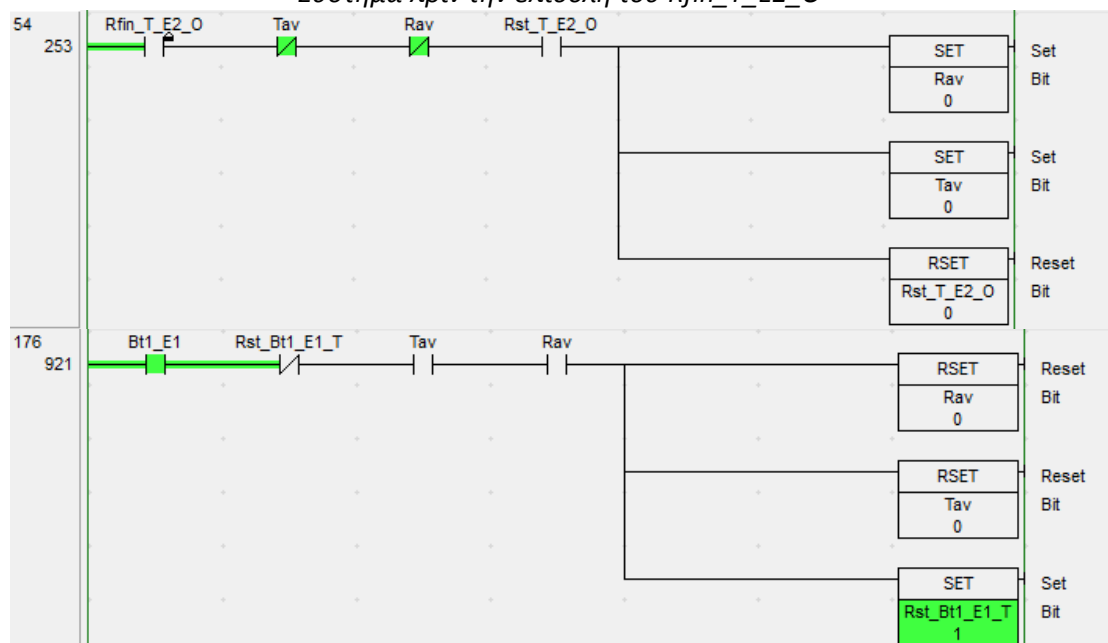
Σύστημα πριν την επιβολή του Tfinished_E2



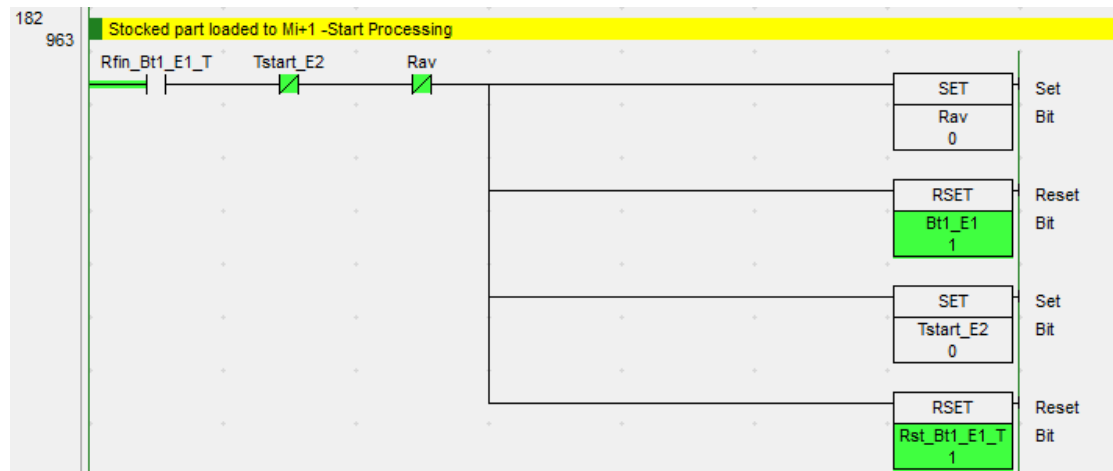
Σύστημα μετά την επιβολή του Tfinished_E2



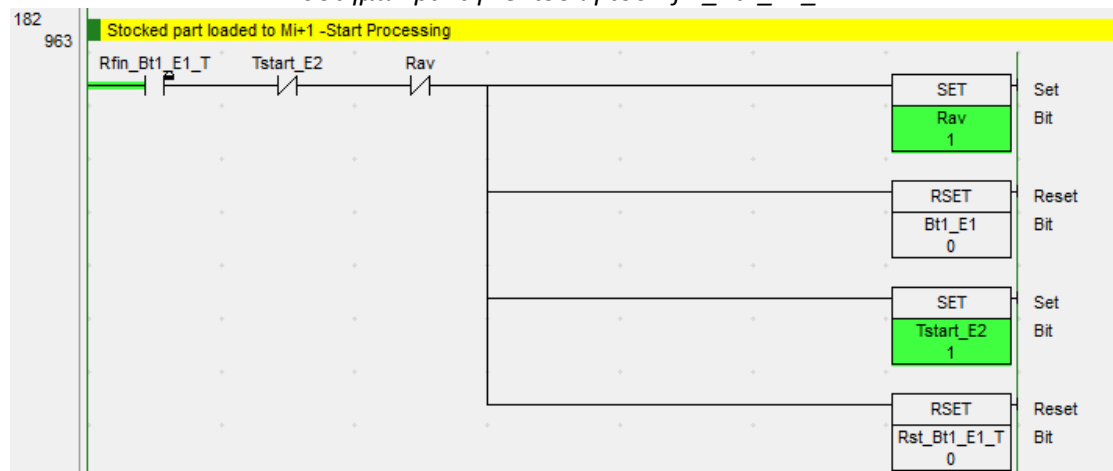
Σύστημα πριν την επιβολή του $Rfin_T_E2_O$



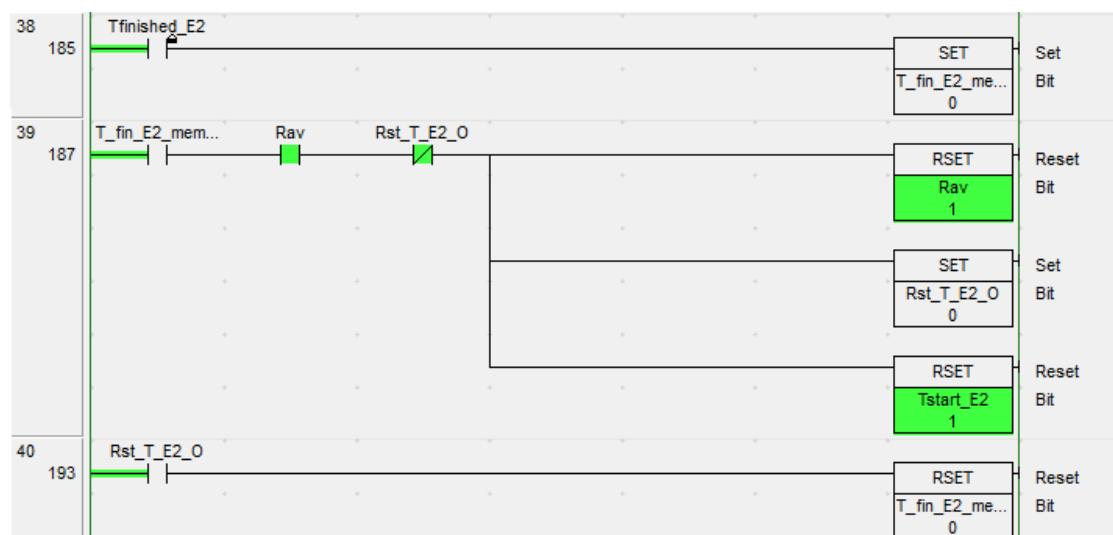
Σύστημα μετά την επιβολή του $Rfin_T_E2_O$



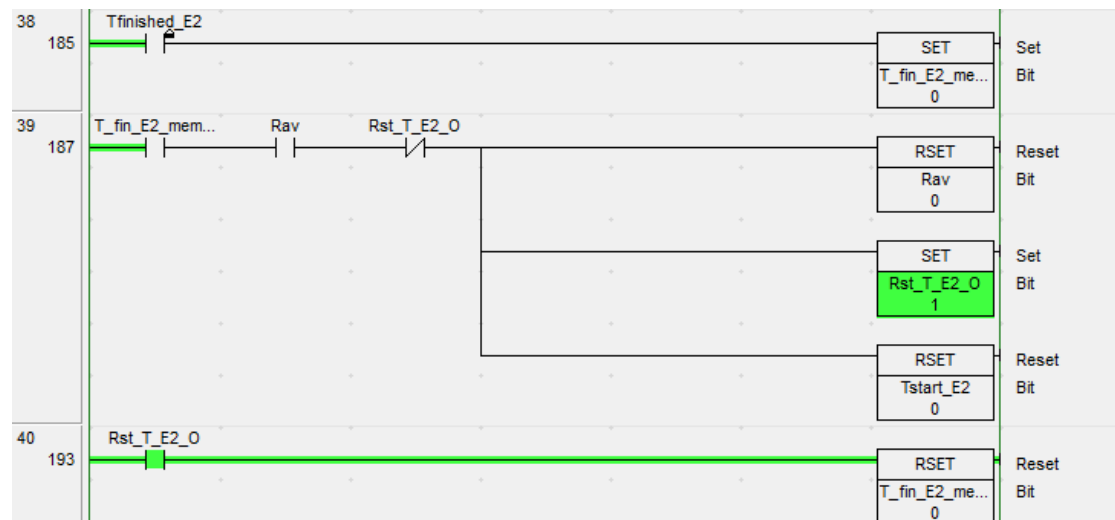
Σύστημα πριν την επιβολή του Rfin_Bt1_E1_T



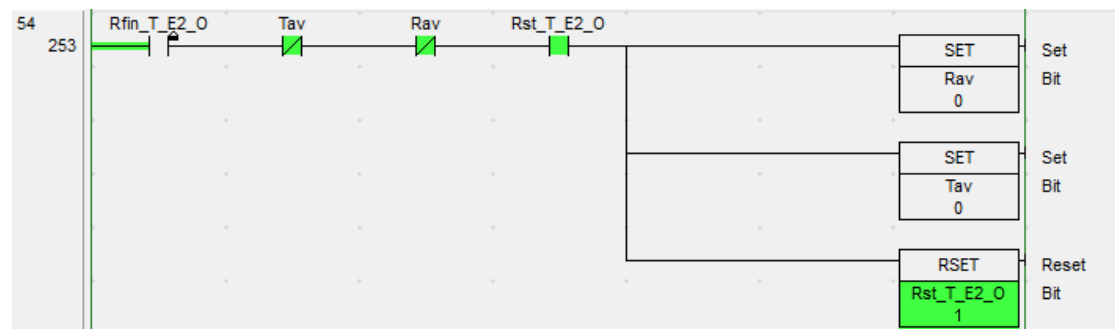
Σύστημα μετά την επιβολή του Rfin_Bt1_E1_T



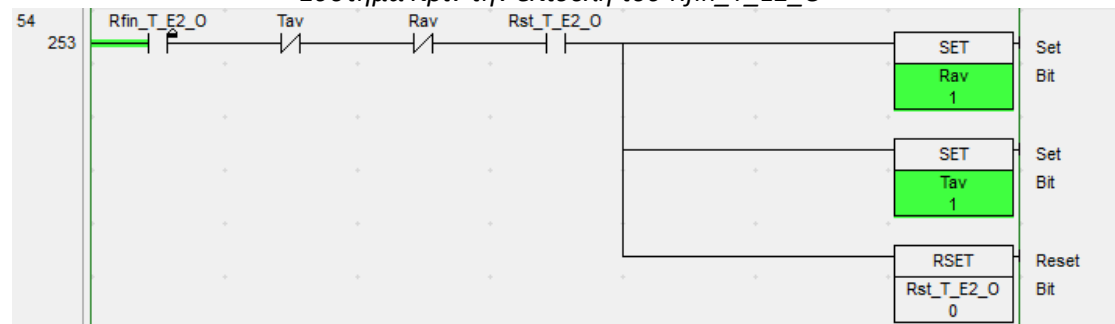
Σύστημα πριν την επιβολή του Tfinished_E2



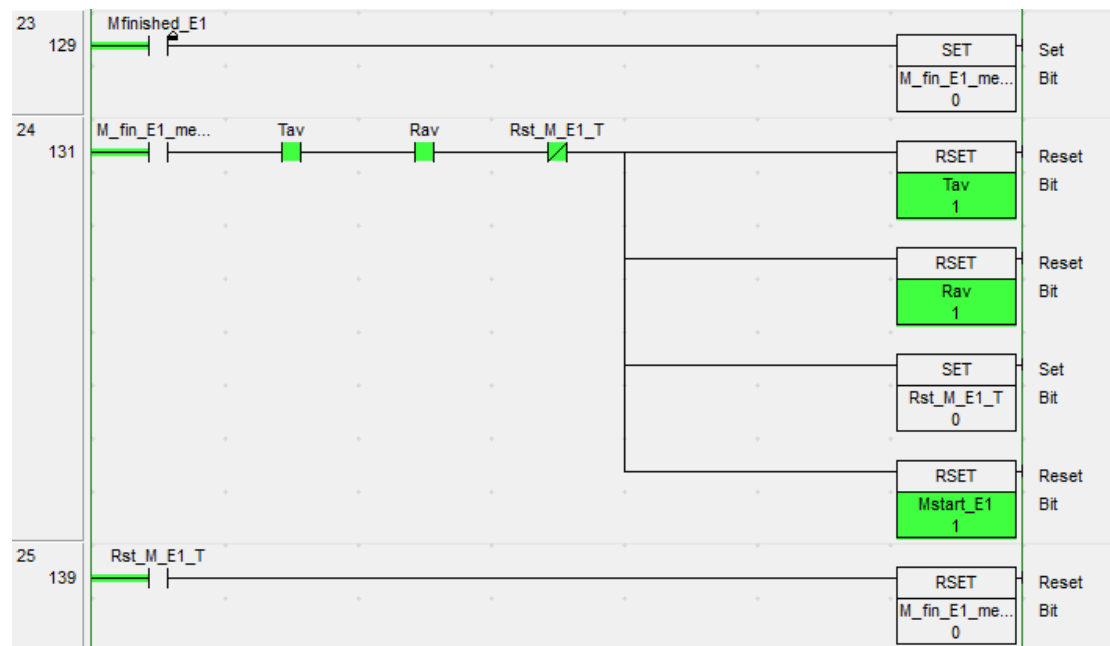
Σύστημα μετά την επιβολή του Tfinished_E2



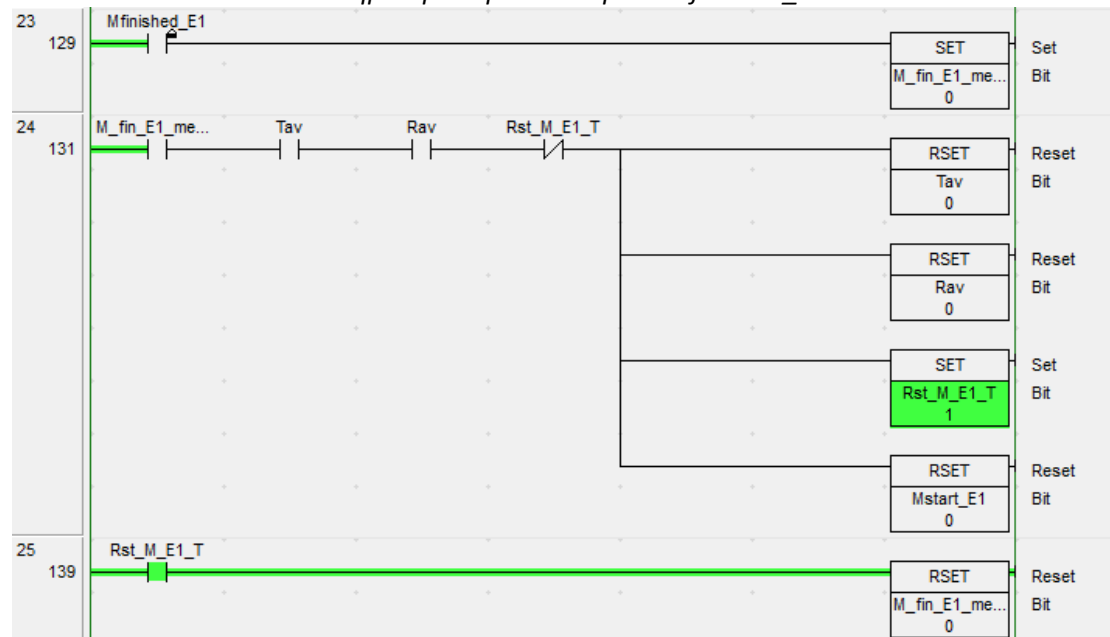
Σύστημα πριν την επιβολή του Rfin_T_E2_O



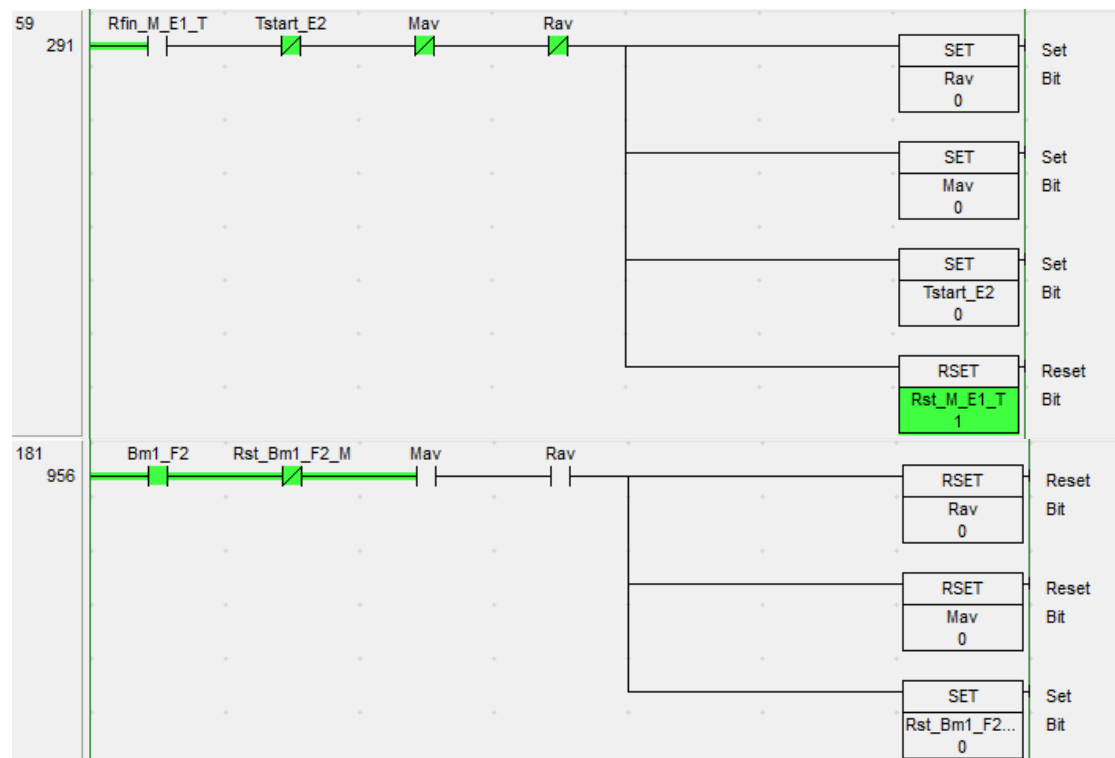
Σύστημα μετά την επιβολή του Rfin_T_E2_O



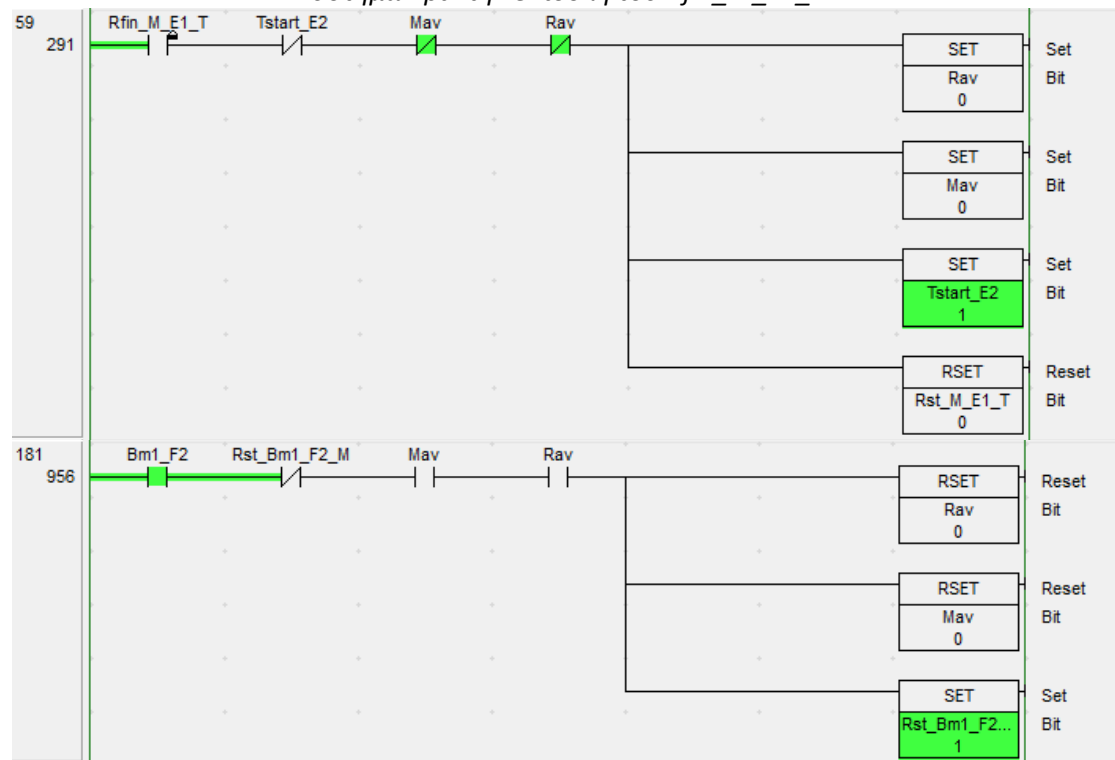
Σύστημα πριν την επιβολή του `Mfinished_E1`



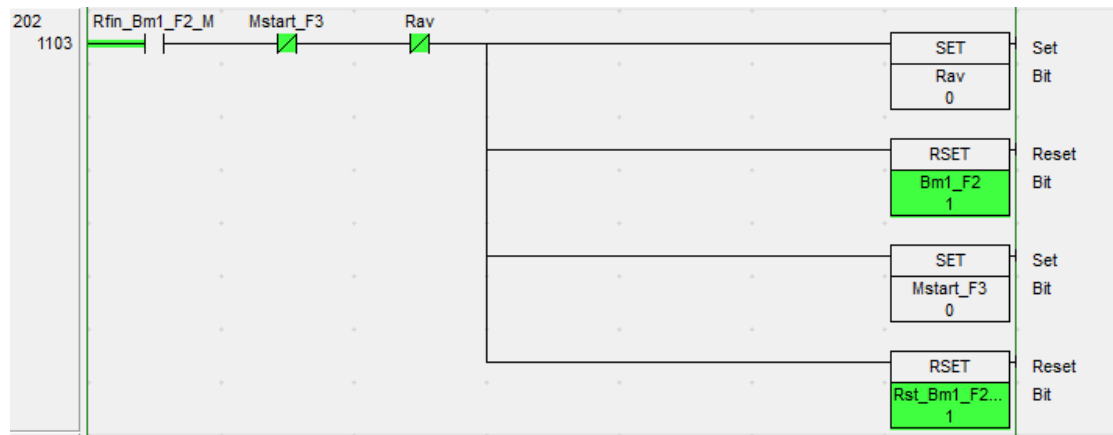
Σύστημα μετά την επιβολή του `Mfinished_E1`



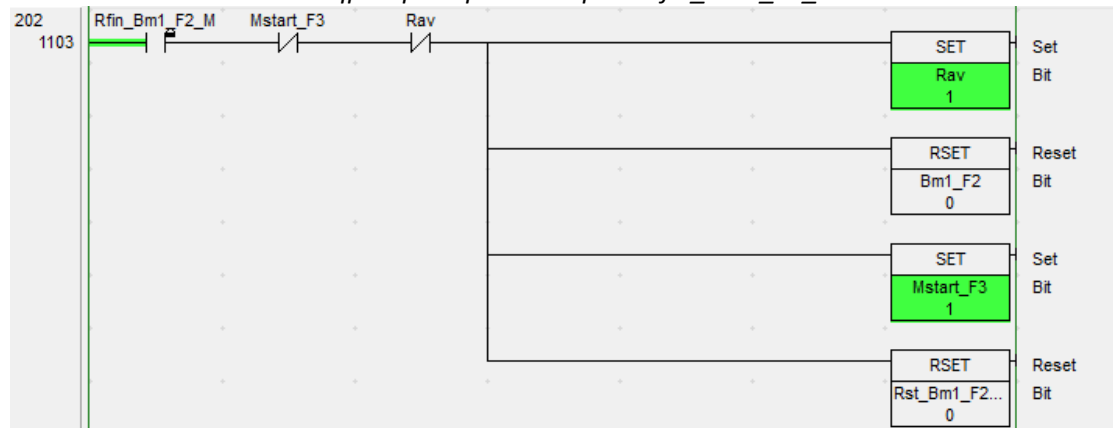
Σύστημα πριν την επιβολή του Rfin_M_E1_T



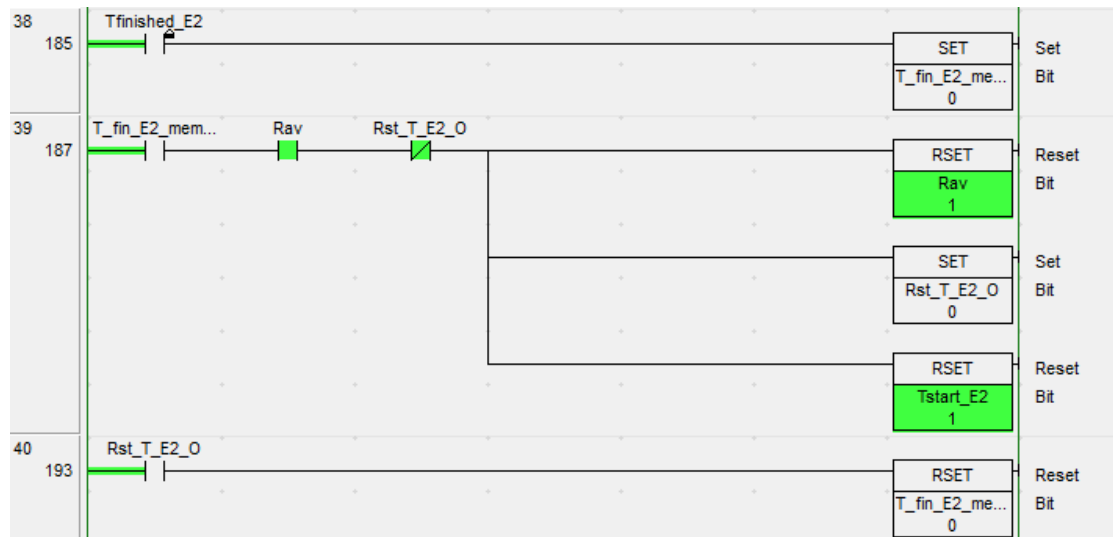
Σύστημα μετά την επιβολή του Rfin_M_E1_T



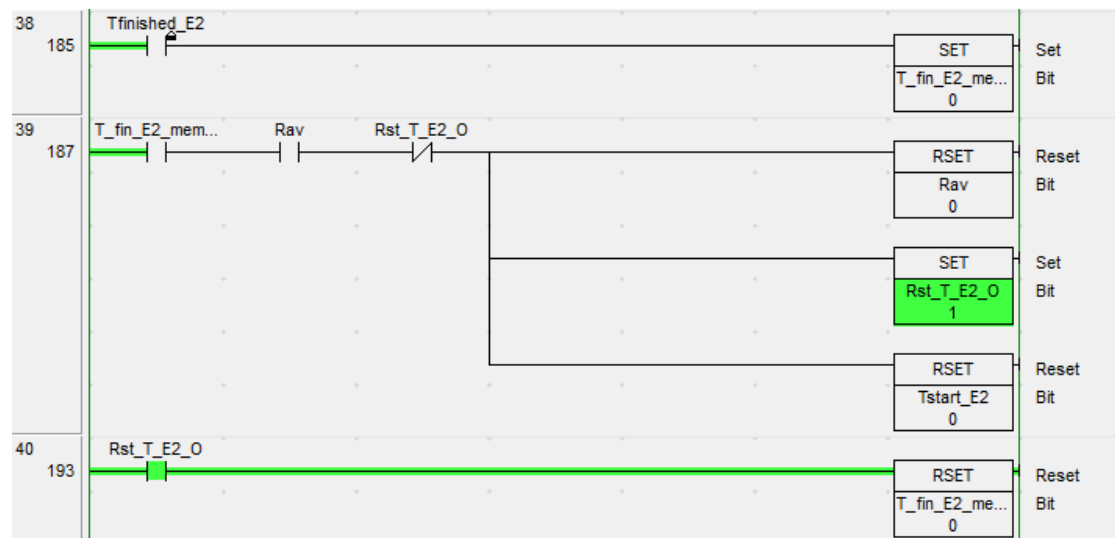
Σύστημα πριν την επιβολή του Rfin_Bm1_F2_M



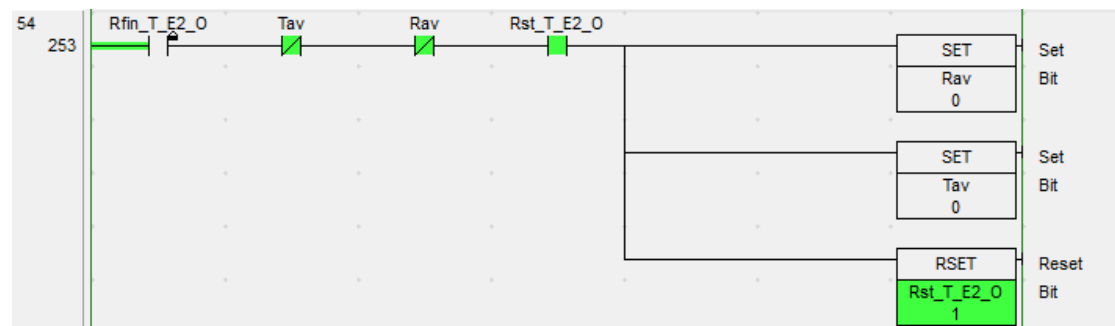
Σύστημα μετά την επιβολή του Rfin_Bm1_F2_M



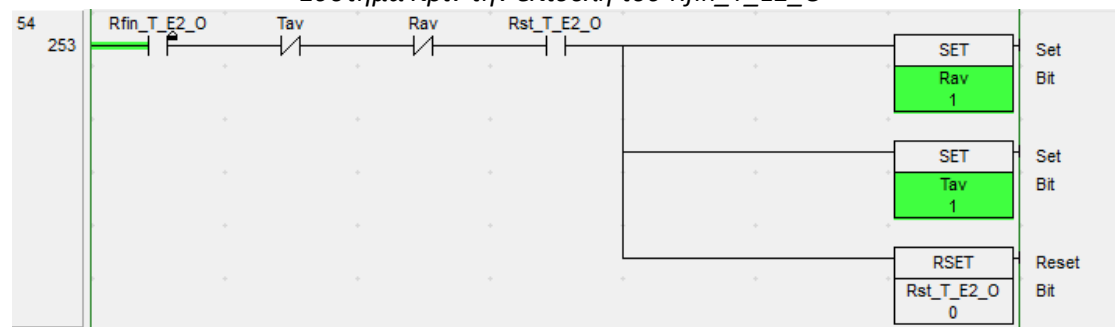
Σύστημα πριν την επιβολή του Tfinished_E2



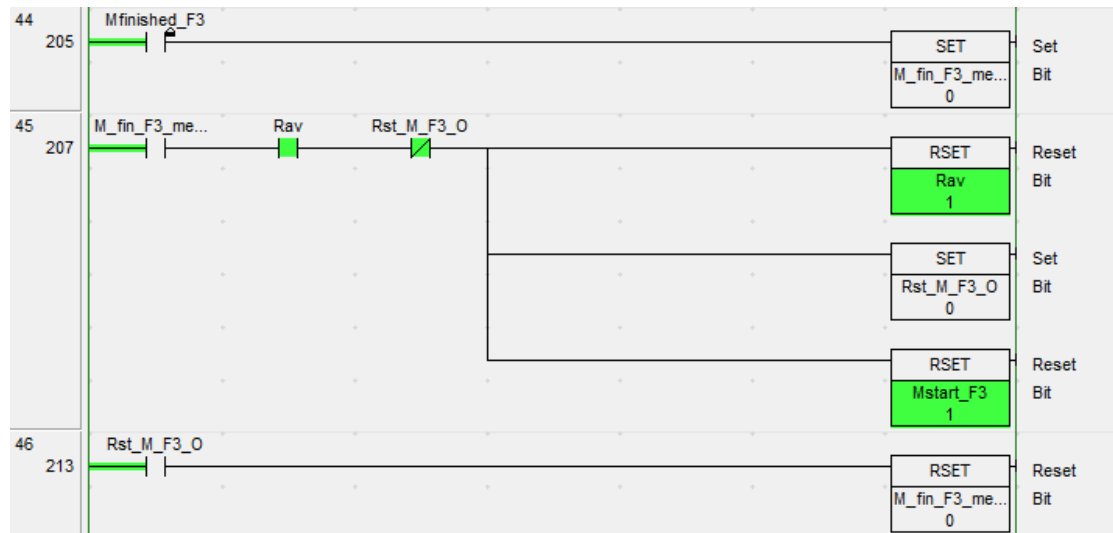
Σύστημα μετά την επιβολή του `Tfinished_E2`



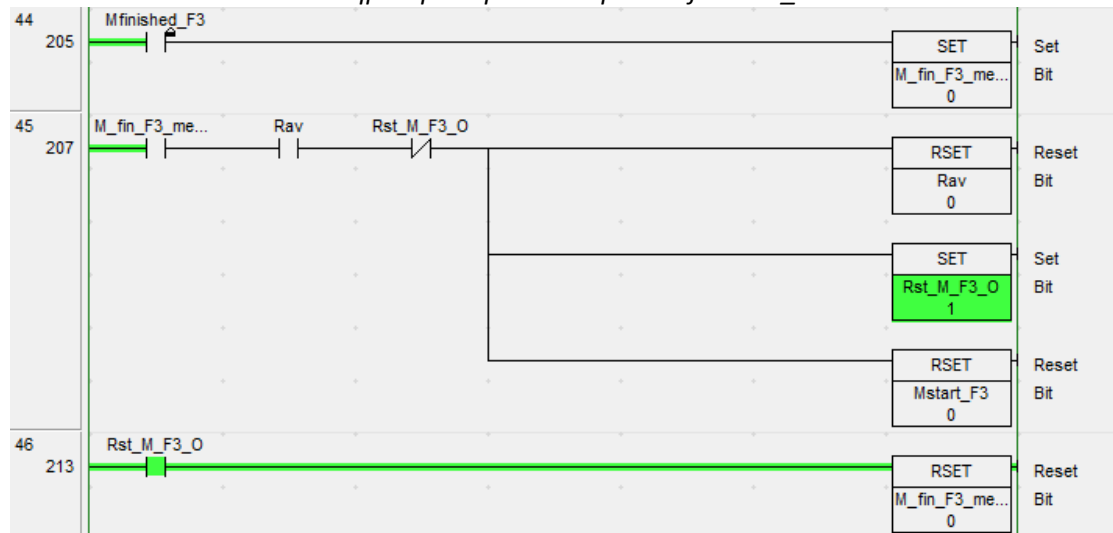
Σύστημα πριν την επιβολή του `Rfin_T_E2_O`



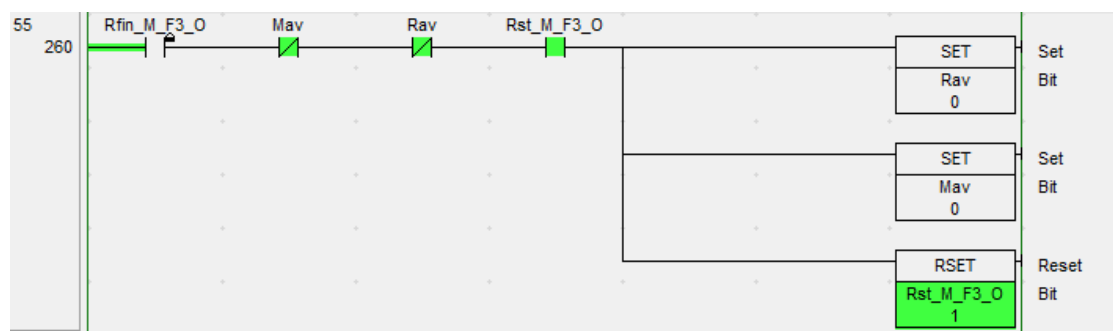
Σύστημα μετά την επιβολή του `Rfin_T_E2_O`



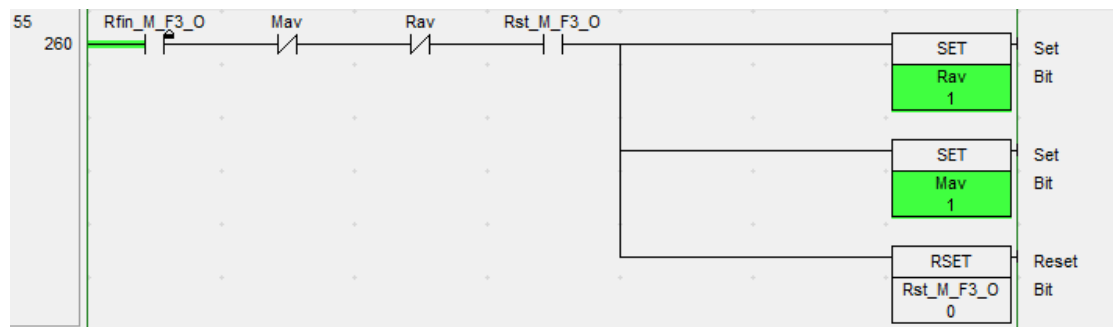
Σύστημα πριν την επιβολή του *Mfinished_F3*



Σύστημα μετά την επιβολή του *Mfinished_F3*

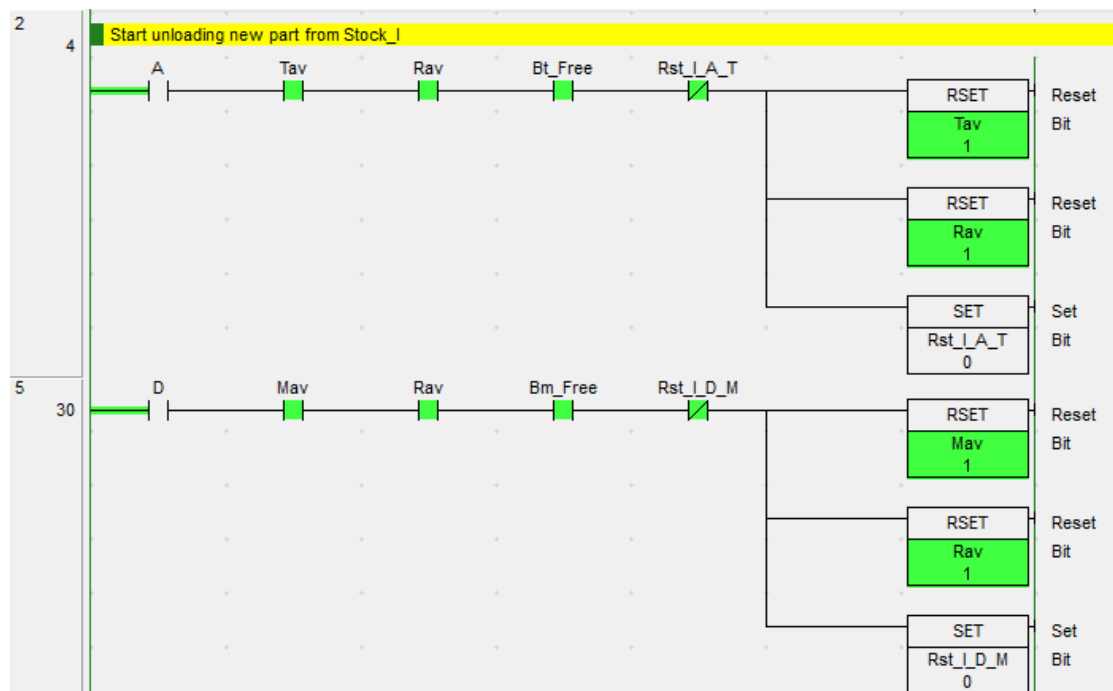


Σύστημα πριν την επιβολή του *Rfin_M_F3_O*



Σύστημα μετά την επιβολή του Rfin_M_F3_O

Όπως αποδείχτηκε όλα τα τεμάχια οδηγήθηκαν στη θέση εξόδου ολοκληρώνοντας όλες τις κατεργασίες που περιλαμβάνει το process plan τους. Στα παρακάτω rungs που αντιστοιχούν στην τελική κατάσταση του συστήματος (μετά την επιβολή του τελευταίου σήματος εισόδου) φαίνεται ότι το σύστημα επανέρχεται στην αρχική του κατάσταση: οι μηχανές κατεργασίας είναι διαθέσιμες και τα δύο buffer είναι άδεια όπως είχε επιβεβαιωθεί από την προσομοίωση του μοντέλου cph.



Τελική κατάσταση του συστήματος (το σύστημα έχει επανέλθει στην αρχική του κατάσταση)

Κεφάλαιο 6 – Συμπεράσματα

6.1 Χρησιμότητα δικτύων Petri

Όπως αποδείχτηκε στην παρούσα εργασία, η θεωρία των δικτύων Petri αποτελεί ένα πολύ χρήσιμο εργαλείο για τη μοντελοποίηση, την ανάλυση και τον έλεγχο συστημάτων διακριτών γεγονότων. Για το ευέλικτο σύστημα κατεργασιών που μελετήθηκε στη συγκεκριμένη διπλωματική, η εφαρμογή της συγκεκριμένης θεωρίας κατά το σχεδιασμό του ελέγχου αποδείχτηκε ιδιαίτερα χρήσιμη, αποτελεσματική και αξιόπιστη. Με τη βοήθεια του προγράμματος CPN tools που βασίζεται σε μαθηματικά ορισμένες σχέσεις της θεωρίας των δικτύων Petri επιβεβαιώθηκαν τα εξής χαρακτηριστικά του συστήματος:

- ✓ δεν υπάρχει υπερχείλιση των πόρων του συστήματος
- ✓ δεν υπάρχουν ανεπιθύμητες καταστάσεις του συστήματος λόγω σφαλμάτων στο σχεδιασμό που θα μπορούσαν να οδηγήσουν ενδεχομένως και σε καταστροφή των μηχανών (π.χ. το ρομπότ να μεταφέρει ένα τεμάχιο σε μηχανή που ήδη κατεργάζεται άλλο τεμάχιο)
- ✓ δεν υπάρχουν κολλήματα του συστήματος δηλαδή καταστάσεις όπου οποιοδήποτε σήμα εισόδου και να επιβληθεί στο σύστημα δε μεταβάλεται η κατάστασή του

6.2 Αξιολόγηση προγράμματος CPN tools

Το πρόγραμμα CPN tools, βασισμένο στα Coloured Petri Nets, επέκταση της θεωρίας των κλασικών PN, χρησιμοποιήθηκε για τη μοντελοποίηση και σύνθεση του σχεδιασμού ελέγχου του συστήματος της παρούσας εργασίας. Η αυξημένη δυναμική μοντελοποίησης του συγκεκριμένου προγράμματος, μέσω της χρήσης τύπων δεδομένων και μεταβλητών, έδωσε αρχικά τη δυνατότητα μιας συνοπτικής και ευανάγνωστης απεικόνισης του μοντέλου. Με αυτόν τον τρόπο, οι πολλές διαφορετικές κατεργασίες του συστήματος αλλά και αρκετοί πόροι του συστήματος συγχωνεύονται διευκολύνοντας τη διαχείριση του μοντέλου και την παρακολούθηση της προσομοίωσης. Μετά από την εμπειρία χρήσης του προγράμματος cpn tools θα μπορούσαμε να πούμε ότι οι πιο σημαντικές δυνατότητες που παρέχει είναι :

- η προσομοίωση του μοντέλου βήμα βήμα και η εκκίνηση των ενεργοποιούμενων μεταβάσεων κατ' επιλογή του χρήστη. Η λειτουργία αυτή δίνει τη δυνατότητα εποπτικής και ακριβούς παρακολούθησης της ροής των καταστάσεων του συστήματος.

- η ανάλυση του χώρου καταστάσεων που επιτρέπει όχι μόνο την επιβεβαίωση των ιδιοτήτων του συστήματος αλλά και την επιδιόρθωση σφαλμάτων κατά τον προγραμματισμό του ελέγχου. Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, με την ανάλυση state space υπογίζονται όλες οι καταστάσεις αλλά και όλες οι πιθανές «διαδρομές» από μία κατάσταση σε μία άλλη. Με αυτόν τον τρόπο, μπορεί να μελετηθεί η ροή των καταστάσεων που οδήγησε σε μία ανεπιθύμητη κατάσταση ή μια νεκρή σήμανση και να διορθωθεί το προγραμματιστικό σφάλμα που την προκάλεσε.

Ωστόσο, κατά τη μοντελοποίηση του μελετούμενου συστήματος παρατηρήθηκε ότι στο CPN tools υπήρχε περιορισμός στη μοντελοποίηση του μηδενικού ελέγχου που χρειάστηκε για να εκφραστεί η μη διαθεσιμότητα των πόρων ως συνθήκη. Το πρόγραμμα περιλαμβάνει τη δυνατότητα χρήσης inhibitor arc που χρησιμοποιούνται στα κλασικά δίκτυα Petri όχι όμως τόξων συνδεδεμένων με θέσεις που αντιστοιχούν σε κάποιο τύπο δεδομένων (colourset). Τελικά, το θέμα αυτό λήθηκε χρησιμοποιώντας «αντίστροφες θέσεις» που όμως αύξησαν τον αριθμό θέσεων και την πολυπλοκότητα του τελικού μοντέλου.

6.3 Γενικότητα μοντέλου

Το ευέλικτο σύστημα κατεργασιών που μελετήθηκε στην παρούσα εργασία περιλαμβάνει δύο διαφορετικές μηχανές κατεργασίας και φασεολόγια μιας έως τριών διαδοχικών κατεργασιών. Στο συγκεκριμένο σύστημα οποιαδήποτε αλλαγή στον τρόπο κατεργασίας των τεμαχίων αφορά τους ελεγκτές του τόννου και της φρέζας (δηλαδή τις εργασίες που είναι αποθηκευμένες ως A1,B1,B2,C1 κ.λ.π.) και δεν απαιτούν τροποποιήσεις στον κεντρικό ελεγκτή του συστήματος που αντιστοιχεί στο μοντέλο cpn.

Όμως, αν πρόκειται να γίνουν αλλαγές που σχετίζονται με τις προτεραιότητες ενεργειών του συστήματος, θα πρέπει να ανατοποθετηθούν τα τόξα και να τροποποιηθούν συνθήκες των τόξων και μεταβάσεων του μοντέλου cpn. Δεδομένου ότι η δομή και οι λειτουργίες του συστήματος δεν αλλάζουν, οι θέσεις, οι τύποι δεδομένων, οι μεταβλητές και οι μεταβάσεις του συστήματος παραμένουν ίδιες.

6.4 Επεκτασιμότητα μοντέλου

Η συγκεκριμένη μοντελοποίηση θα μπορούσε να επεκταθεί για ένα μεγαλύτερο σύστημα που περιλαμβάνει περισσότερες μηχανές επεξεργασίας π.χ λείανση, καθαρισμός, βαφή ή ακόμη και συναρμολόγηση των παραγόμενων τεμαχίων. Οι

επεκτάσεις αυτές βέβαια απαιτούν την προσθήκη νέων θέσεων, τύπων δεδομένων και μεταβλητών αλλά και νέων (και ενδεχομένως πιο πολύπλοκων) συνθηκών στο μοντέλο cpn. Ωστόσο, ειδικά η κατηγορία των Coloured Petri δικτύων ενδείκνυται για τέτοιου είδους επεκτάσεις στον έλεγχο των συστημάτων καθώς παρέχει τη δυνατότητα συγχώνευσης των πόρων και των λειτουργιών του συστήματος. Έτσι η πρόσθήκη νέων σταθμών επεξεργασίας των τεμαχίων ή περισσότερων αποθηκευτικών χώρων δεν αυξάνει υπερβολικά τον όγκο του μοντέλου (οπως θα συνέβαινε στη μοντελοποίηση με απλά δίκτυα Petri).

6.5 Ευελιξία συστήματος

Όπως επιβεβαιώθηκε από την προσομοίωση του μοντέλου στο cpn tools αλλά και από την προσομοίωση του προγράμματος ladder το μελετούμενο σύστημα χαρακτηρίζεται από ευελιξία ως προς τα εξής:

- Ευελιξία κατεργασίας. Μπορεί να υποστηρίξει το συντονισμό πολλαπλών κατεργασιών των τεμαχίων σε δύο διαφορετικές μηχανές κατεργασίας αλλά και περισσότερων.
- Ευελιξία προϊόντος. Το σύστημα μπορεί να περάσει από ένα μείγμα τύπων τεμαχίων σε ένα διαφορετικό μείγμα αλλάζοντας τις εργασίες που είναι αποθηκευμένες στους ελεγκτές των μηχανών κατεργασίας χωρίς να τροποποιηθεί ο κεντρικός έλεγχος του συστήματος.
- Ευελιξία διαδρομής. Το σύστημα ακολουθεί εναλλακτικές λύσεις όταν ένα ή περισσότερα σημεία της διαδρομής ενός ή περισσότερων τεμαχίων γίνονται προσπελάσιμα λόγω π.χ. καθυστέρησης μιας κατεργασίας προκειμένου να ελαχιστοποιήσει τους «νεκρούς χρόνους» κατά την παραγωγή.

6.6 Μεταροπή μοντέλου CPN σε γλώσσα plc

Στην παρούσα εργασία αποδείχτηκε ότι ένα coloured petri δίκτυο μπορεί να μετατραπεί σε μορφή διαγράμματος ladder ώστε να υλοποιηθεί σε εμπορικά διαθέσιμους ελεγκτές. Βέβαια, η μετάφραση του μοντέλου cpn σε διάγραμμα ladder απαιτούσε αρκετές επεκτάσεις της υπάρχουσας θεωρίας μετατροπής απλού PN σε LLD. Παράλληλα, χρειάστηκε και η αντιστοίχιση σημάτων εισόδου και εξόδου προς/από τον κεντρικό ελεγκτή σε ένα από τα τόξα εισόδου και εξόδου κάθε μετάβασης που επίσης δεν περιλαμβάνεται στην υπάρχουσα θεωρία μετατροπής.

Από την προσομοίωση του τελικού προγράμματος ladder επιβεβαιώθηκε ότι το πρόγραμμα ελέγχου δεν παρουσίαζε κολλήματα (deadlocks), ανεπιθύμητες

καταστάσεις και νεκρούς χρόνους και τηρούσε τις προτεραιότητες που είχαν τεθεί κατά το σχεδιασμό.

Συνεπώς, η θεωρία μετάφρασης CPN σε γλώσσα plc μπορεί να διερευνηθεί και να ανατυχθεί περαιτέρω παρέχοντας ένα χρήσιμο εργαλείο ελέγχου προγραμματισμού των συστημάτων διακριτών γεγονότων ή ακόμα να ενσωματωθεί και στο λογισμικό των plc.

6.7 Αντικείμενα για περαιτέρω μελέτη

- Μοντελοποίηση ευέλικτου συστήματος κατεργασιών με περισσότερους σταθμούς επεξεργασίας ή και μεταφοράς των τεμαχίων με τη βοήθεια των Coloured Petri δικτύων
- Εφαρμογή του προγράμματος που δημιουργήθηκε σε γλώσσα plc στην πραγματική διάταξη συστήματος κατεργασιών που διαθέτει το εργαστήριο

Παράρτημα

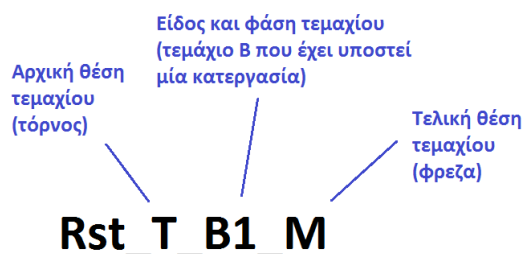
Σήματα εισόδου ,εξόδου και βοηθητικές μνήμες που έχουν χρησιμοποιηθεί στο διάγραμμα ladder:

- Διαθεσιμότητα πόρων (βοηθητικές μνήμες στο PLC)

Διαθεσιμότητα ρομπότ	Rav
Διαθεσιμότητα τórνου	Tav
Διαθεσιμότητα φρέζας	Mav

- Σήματα - Εντολές προς το ρομπότ (έξοδοι από το PLC)

Οι εντολές προς το ρομπότ έχουν την παρακάτω κωδικοποιημένη μορφή που φαίνεται και στο παρακάτω παράδειγμα :



Μεταφορά τεμαχίου A από την αποθήκη εισόδου στον τόρνο	Rst_I_A_T
Μεταφορά τεμαχίου B από την αποθήκη εισόδου στον τόρνο	Rst_I_B_T
Μεταφορά τεμαχίου C από την αποθήκη εισόδου στον τόρνο	Rst_I_C_T
Μεταφορά τεμαχίου D από την αποθήκη εισόδου στη φρέζα	Rst_I_D_T
Μεταφορά τεμαχίου E από την αποθήκη εισόδου στη φρέζα	Rst_I_E_T
Μεταφορά τεμαχίου F από την αποθήκη εισόδου στη φρέζα	Rst_I_F_T
Μεταφορά τεμαχίου B1 από τον τόρνο στη φρέζα	Rst_T_B1_M
Μεταφορά τεμαχίου B1 από τον τόρνο στο Buffer_m (θέση 1)	Rst_T_B1_Bm1
Μεταφορά τεμαχίου B1 από τον τόρνο στο Buffer_m (θέση 2)	Rst_T_B1_Bm2
Μεταφορά τεμαχίου B1 από τον τόρνο στο Buffer_m (θέση 3)	Rst_T_B1_Bm3
Μεταφορά τεμαχίου B1 από τον τόρνο στο Buffer_m (θέση 4)	Rst_T_B1_Bm4

Μεταφορά τεμαχίου C1 από τον τόρνο στη φρέζα	Rst_T_C1_M
Μεταφορά τεμαχίου C1 από τον τόρνο στο Buffer_m (θέση 1)	Rst_T_C1_Bm1
Μεταφορά τεμαχίου C1 από τον τόρνο στο Buffer_m (θέση 2)	Rst_T_C1_Bm2
Μεταφορά τεμαχίου C1 από τον τόρνο στο Buffer_m (θέση 3)	Rst_T_C1_Bm3
Μεταφορά τεμαχίου C1 από τον τόρνο στο Buffer_m (θέση 4)	Rst_T_C1_Bm4
Μεταφορά τεμαχίου F2 από τον τόρνο στη φρέζα	Rst_T_F2_M
Μεταφορά τεμαχίου F2 από τον τόρνο στο Buffer_m (θέση 1)	Rst_T_F2_Bm1
Μεταφορά τεμαχίου F2 από τον τόρνο στο Buffer_m (θέση 2)	Rst_T_F2_Bm2
Μεταφορά τεμαχίου F2 από τον τόρνο στο Buffer_m (θέση 3)	Rst_T_F2_Bm3
Μεταφορά τεμαχίου F2 από τον τόρνο στο Buffer_m (θέση 4)	Rst_T_F2_Bm4
Μεταφορά τεμαχίου E1 από τη φρέζα στον τόρνο	Rst_M_E1_T
Μεταφορά τεμαχίου E1 από τη φρέζα στο Buffer_t (θέση 1)	Rst_M_E1_Bt1
Μεταφορά τεμαχίου E1 από τη φρέζα στο Buffer_t (θέση 2)	Rst_M_E1_Bt2
Μεταφορά τεμαχίου E1 από τη φρέζα στο Buffer_t (θέση 3)	Rst_M_E1_Bt3
Μεταφορά τεμαχίου E1 από τη φρέζα στο Buffer_t (θέση 4)	Rst_M_E1_Bt4
Μεταφορά τεμαχίου F1 από τη φρέζα στον τόρνο	Rst_M_F1_T
Μεταφορά τεμαχίου F1 από τη φρέζα στο Buffer_t (θέση 1)	Rst_M_F1_Bt1
Μεταφορά τεμαχίου F1 από τη φρέζα στο Buffer_t (θέση 2)	Rst_M_F1_Bt2
Μεταφορά τεμαχίου F1 από τη φρέζα στο Buffer_t (θέση 3)	Rst_M_F1_Bt3
Μεταφορά τεμαχίου F1 από τη φρέζα στο Buffer_t (θέση 4)	Rst_M_F1_Bt4
Μεταφορά τεμαχίου C2 από τη φρέζα στον τόρνο	Rst_M_C2_T
Μεταφορά τεμαχίου C2 από τη φρέζα στο Buffer_t (θέση 1)	Rst_M_C2_Bt1
Μεταφορά τεμαχίου C2 από τη φρέζα στο Buffer_t (θέση 2)	Rst_M_C2_Bt2
Μεταφορά τεμαχίου C2 από τη φρέζα στο Buffer_t (θέση 3)	Rst_M_C2_Bt3
Μεταφορά τεμαχίου C2 από τη φρέζα στο Buffer_t (θέση 4)	Rst_M_C2_Bt4
Μεταφορά τεμαχίου B1 από τη θέση 1 του Buffer_m στη φρέζα	Rst_Bm1_B1_M
Μεταφορά τεμαχίου B1 από τη θέση 2 του Buffer_m στη φρέζα	Rst_Bm2_B1_M
Μεταφορά τεμαχίου B1 από τη θέση 3 του Buffer_m στη φρέζα	Rst_Bm3_B1_M

Buffer_m στη φρέζα	
Μεταφορά τεμαχίου B1 από τη θέση 4 του Buffer_m στη φρέζα	Rst_Bm4_B1_M
Μεταφορά τεμαχίου C1 από τη θέση 1 του Buffer_m στη φρέζα	Rst_Bm1_C1_M
Μεταφορά τεμαχίου C1 από τη θέση 2 του Buffer_m στη φρέζα	Rst_Bm2_C1_M
Μεταφορά τεμαχίου C1 από τη θέση 3 του Buffer_m στη φρέζα	Rst_Bm3_C1_M
Μεταφορά τεμαχίου C1 από τη θέση 4 του Buffer_m στη φρέζα	Rst_Bm4_C1_M
Μεταφορά τεμαχίου F2 από τη θέση 1 του Buffer_m στη φρέζα	Rst_Bm1_F2_M
Μεταφορά τεμαχίου F2 από τη θέση 2 του Buffer_m στη φρέζα	Rst_Bm2_F2_M
Μεταφορά τεμαχίου F2 από τη θέση 3 του Buffer_m στη φρέζα	Rst_Bm3_F2_M
Μεταφορά τεμαχίου F2 από τη θέση 4 του Buffer_m στη φρέζα	Rst_Bm4_F2_M
Μεταφορά τεμαχίου E1 από τη θέση 1 του Buffer_t στον τόρνο	Rst_Bt1_E1_T
Μεταφορά τεμαχίου E1 από τη θέση 2 του Buffer_t στον τόρνο	Rst_Bt2_E1_T
Μεταφορά τεμαχίου E1 από τη θέση 3 του Buffer_t στον τόρνο	Rst_Bt3_E1_T
Μεταφορά τεμαχίου E1 από τη θέση 4 του Buffer_t στον τόρνο	Rst_Bt4_E1_T
Μεταφορά τεμαχίου F1 από τη θέση 1 του Buffer_t στον τόρνο	Rst_Bt1_F1_T
Μεταφορά τεμαχίου F1 από τη θέση 2 του Buffer_t στον τόρνο	Rst_Bt2_F1_T
Μεταφορά τεμαχίου F1 από τη θέση 3 του Buffer_t στον τόρνο	Rst_Bt3_F1_T
Μεταφορά τεμαχίου F1 από τη θέση 4 του Buffer_t στον τόρνο	Rst_Bt4_F1_T
Μεταφορά τεμαχίου C2 από τη θέση 1 του Buffer_t στον τόρνο	Rst_Bt1_C2_T
Μεταφορά τεμαχίου C2 από τη θέση 2 του Buffer_t στον τόρνο	Rst_Bt2_C2_T
Μεταφορά τεμαχίου C2 από τη θέση 3 του Buffer_t στον τόρνο	Rst_Bt3_C2_T
Μεταφορά τεμαχίου C2 από τη θέση 4 του Buffer_t στον τόρνο	Rst_Bt4_C2_T
Μεταφορά τεμαχίου A1 από τον τόρνο στην αποθήκη εξόδου	Rst_T_A1_O
Μεταφορά τεμαχίου B2 από τη φρέζα στην αποθήκη εξόδου	Rst_M_B2_O
Μεταφορά τεμαχίου C3 από τον τόρνο στην αποθήκη εξόδου	Rst_T_C3_O
Μεταφορά τεμαχίου D1 από τη φρέζα στην αποθήκη εξόδου	Rst_M_D1_O
Μεταφορά τεμαχίου E2 από τον τόρνο στην αποθήκη εξόδου	Rst_T_E2_O
Μεταφορά τεμαχίου F3 από τη φρέζα στην αποθήκη εξόδου	Rst_M_F3_O

- **Σήματα από το ρομπότ (είσοδοι στο PLC)**

Ολοκλήρωση μεταφοράς τεμαχίου A από την αποθήκη εισόδου στον τόρνο	Rfin_I_A_T
Ολοκλήρωση μεταφοράς τεμαχίου B από την αποθήκη εισόδου στον τόρνο	Rfin_I_B_T
Ολοκλήρωση μεταφοράς τεμαχίου C από την αποθήκη εισόδου στον τόρνο	Rfin_I_C_T
Ολοκλήρωση μεταφοράς τεμαχίου D από την αποθήκη εισόδου στη φρέζα	Rfin_I_D_M
Ολοκλήρωση μεταφοράς τεμαχίου E από την αποθήκη εισόδου στη φρέζα	Rfin_I_E_M
Ολοκλήρωση μεταφοράς τεμαχίου F από την αποθήκη εισόδου στη φρέζα	Rfin_I_F_M
Ολοκλήρωση μεταφοράς τεμαχίου B1 από τον τόρνο στη φρέζα	Rfin_T_B1_M
Ολοκλήρωση μεταφοράς τεμαχίου B1 από τον τόρνο στο Buffer_m (θέση 1)	Rfin_T_B1_Bm1
Ολοκλήρωση μεταφοράς τεμαχίου B1 από τον τόρνο στο Buffer_m (θέση 2)	Rfin_T_B1_Bm2
Ολοκλήρωση μεταφοράς τεμαχίου B1 από τον τόρνο στο Buffer_m (θέση 3)	Rfin_T_B1_Bm3
Ολοκλήρωση μεταφοράς τεμαχίου B1 από τον τόρνο στο Buffer_m (θέση 4)	Rfin_T_B1_Bm4
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τον τόρνο στη φρέζα	Rfin_T_C1_M
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τον τόρνο στο Buffer_m (θέση 1)	Rfin_T_C1_Bm1
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τον τόρνο στο Buffer_m (θέση 2)	Rfin_T_C1_Bm2
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τον τόρνο στο Buffer_m (θέση 3)	Rfin_T_C1_Bm3
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τον τόρνο στο Buffer_m (θέση 4)	Rfin_T_C1_Bm4
Ολοκλήρωση μεταφοράς τεμαχίου F2 από τον τόρνο στη φρέζα	Rfin_T_F2_M
Ολοκλήρωση μεταφοράς τεμαχίου F2 από τον τόρνο στο Buffer_m (θέση 1)	Rfin_T_F2_Bm1
Ολοκλήρωση μεταφοράς τεμαχίου F2 από τον τόρνο στο Buffer_m (θέση 2)	Rfin_T_F2_Bm2
Ολοκλήρωση μεταφοράς τεμαχίου F2 από τον τόρνο στο Buffer_m (θέση 3)	Rfin_T_F2_Bm3
Ολοκλήρωση μεταφοράς τεμαχίου F2 από τον τόρνο στο Buffer_m (θέση 4)	Rfin_T_F2_Bm4
Ολοκλήρωση μεταφοράς τεμαχίου E1 από τη φρέζα στον τόρνο	Rfin_M_E1_T
Ολοκλήρωση μεταφοράς τεμαχίου E1 από τη φρέζα στο Buffer_t (θέση 1)	Rfin_M_E1_Bt1
Ολοκλήρωση μεταφοράς τεμαχίου E1 από τη φρέζα στο Buffer_t (θέση 2)	Rfin_M_E1_Bt2
Ολοκλήρωση μεταφοράς τεμαχίου E1 από τη φρέζα στο Buffer_t (θέση 3)	Rfin_M_E1_Bt3
Ολοκλήρωση μεταφοράς τεμαχίου E1 από τη φρέζα στο Buffer_t (θέση 4)	Rfin_M_E1_Bt4

τη φρέζα στο Buffer_t (θέση 4)	
Ολοκλήρωση μεταφοράς τεμαχίου F1 από τη φρέζα στον τόρνο	Rfin_M_F1_T
Ολοκλήρωση μεταφοράς τεμαχίου F1 από τη φρέζα στο Buffer_t (θέση 1)	Rfin_M_F1_Bt1
Ολοκλήρωση μεταφοράς τεμαχίου F1 από τη φρέζα στο Buffer_t (θέση 2)	Rfin_M_F1_Bt2
Ολοκλήρωση μεταφοράς τεμαχίου F1 από τη φρέζα στο Buffer_t (θέση 3)	Rfin_M_F1_Bt3
Ολοκλήρωση μεταφοράς τεμαχίου F1 από τη φρέζα στο Buffer_t (θέση 4)	Rfin_M_F1_Bt4
Ολοκλήρωση μεταφοράς τεμαχίου C2 από τη φρέζα στον τόρνο	Rfin_M_C2_T
Ολοκλήρωση μεταφοράς τεμαχίου C2 από τη φρέζα στο Buffer_t (θέση 1)	Rfin_M_C2_Bt1
Ολοκλήρωση μεταφοράς τεμαχίου C2 από τη φρέζα στο Buffer_t (θέση 2)	Rfin_M_C2_Bt2
Ολοκλήρωση μεταφοράς τεμαχίου C2 από τη φρέζα στο Buffer_t (θέση 3)	Rfin_M_C2_Bt3
Ολοκλήρωση μεταφοράς τεμαχίου C2 από τη φρέζα στο Buffer_t (θέση 4)	Rfin_M_C2_Bt4
Ολοκλήρωση μεταφοράς τεμαχίου B1 από τη θέση 1 του Buffer_m στη φρέζα	Rfin_Bm1_B1_M
Ολοκλήρωση μεταφοράς τεμαχίου B1 από τη θέση 2 του Buffer_m στη φρέζα	Rfin_Bm2_B1_M
Ολοκλήρωση μεταφοράς τεμαχίου B1 από τη θέση 3 του Buffer_m στη φρέζα	Rfin_Bm3_B1_M
Ολοκλήρωση μεταφοράς τεμαχίου B1 από τη θέση 4 του Buffer_m στη φρέζα	Rfin_Bm4_B1_M
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τη θέση 1 του Buffer_m στη φρέζα	Rfin_Bm1_C1_M
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τη θέση 2 του Buffer_m στη φρέζα	Rfin_Bm2_C1_M
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τη θέση 3 του Buffer_m στη φρέζα	Rfin_Bm3_C1_M
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τη θέση 4 του Buffer_m στη φρέζα	Rfin_Bm4_C1_M
Ολοκλήρωση μεταφοράς τεμαχίου F2 από τη θέση 1 του Buffer_m στη φρέζα	Rfin_Bm1_F2_M
Ολοκλήρωση μεταφοράς τεμαχίου F2 από τη θέση 2 του Buffer_m στη φρέζα	Rfin_Bm2_F2_M
Ολοκλήρωση μεταφοράς τεμαχίου F2 από τη θέση 3 του Buffer_m στη φρέζα	Rfin_Bm3_F2_M
Ολοκλήρωση μεταφοράς τεμαχίου F2 από τη θέση 4 του Buffer_m στη φρέζα	Rfin_Bm4_F2_M
Ολοκλήρωση μεταφοράς τεμαχίου E1 από τη θέση 1 του Buffer_t στον τόρνο	Rfin_Bt1_E1_T
Ολοκλήρωση μεταφοράς τεμαχίου E1 από τη θέση 2 του Buffer_t στον τόρνο	Rfin_Bt2_E1_T
Ολοκλήρωση μεταφοράς τεμαχίου E1 από τη θέση 3 του Buffer_t στον τόρνο	Rfin_Bt3_E1_T
Ολοκλήρωση μεταφοράς τεμαχίου E1 από τη θέση 4 του Buffer_t στον τόρνο	Rfin_Bt4_E1_T
Ολοκλήρωση μεταφοράς τεμαχίου F1 από τη θέση 1 του Buffer_t στον τόρνο	Rfin_Bt1_F1_T

Ολοκλήρωση μεταφοράς τεμαχίου F1 από τη θέση 2 του Buffer_t στον τόρνο	Rfin_Bt2_F1_T
Ολοκλήρωση μεταφοράς τεμαχίου F1 από τη θέση 3 του Buffer_t στον τόρνο	Rfin_Bt3_F1_T
Ολοκλήρωση μεταφοράς τεμαχίου F1 από τη θέση 4 του Buffer_t στον τόρνο	Rfin_Bt4_F1_T
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τη θέση 1 του Buffer_t στον τόρνο	Rfin_Bt1_C2_T
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τη θέση 2 του Buffer_t στον τόρνο	Rfin_Bt2_C2_T
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τη θέση 3 του Buffer_t στον τόρνο	Rfin_Bt3_C2_T
Ολοκλήρωση μεταφοράς τεμαχίου C1 από τη θέση 4 του Buffer_t στον τόρνο	Rfin_Bt4_C2_T
Ολοκλήρωση μεταφοράς τεμαχίου A1 από τον τόρνο στην τελική αποθήκη	Rfin_T_A1_O
Ολοκλήρωση μεταφοράς τεμαχίου B2 από τη φρέζα στην τελική αποθήκη	Rfin_M_B2_O
Ολοκλήρωση μεταφοράς τεμαχίου C3 από τον τόρνο στην τελική αποθήκη	Rfin_T_C3_O
Ολοκλήρωση μεταφοράς τεμαχίου D1 από τη φρέζα στην τελική αποθήκη	Rfin_M_D1_O
Ολοκλήρωση μεταφοράς τεμαχίου E2 από τον τόρνο στην τελική αποθήκη	Rfin_T_E2_O
Ολοκλήρωση μεταφοράς τεμαχίου F3 από τη φρέζα στην τελική αποθήκη	Rfin_M_F3_O

- Σήματα-εντολές προς τον τόρνο (έξοδοι από το PLC)**

Κατεργασία A1	Tstart_A1
Κατεργασία B1	Tstart_B1
Κατεργασία C1	Tstart_C1
Κατεργασία C3	Tstart_C3
Κατεργασία E2	Tstart_E2
Κατεργασία F2	Tstart_F2

- Σήματα από τον τόρνο (είσοδοι στο PLC)**

Τέλος κατεργασίας A1	Tfinished_A1
Τέλος κατεργασίας B1	Tfinished_B1
Τέλος κατεργασίας C1	Tfinished_C1
Τέλος κατεργασίας C3	Tfinished_C3
Τέλος κατεργασίας E2	Tfinished_E2
Τέλος κατεργασίας F2	Tfinished_F2

- **Σήματα-εντολές προς τη φρέζα (έξοδοι από το PLC)**

Κατεργασία D1	Mstart _D1
Κατεργασία E1	Mstart _E1
Κατεργασία F1	Mstart _F1
Κατεργασία F3	Mstart _F3
Κατεργασία B2	Mstart _B2
Κατεργασία C2	Mstart _C2

- **Σήματα από τη φρέζα (είσοδοι στο PLC)**

Τέλος κατεργασίας D1	Mfinished _D1
Τέλος κατεργασίας E1	Mfinished _E1
Τέλος κατεργασίας F1	Mfinished _F1
Τέλος κατεργασίας F3	Mfinished _F3
Τέλος κατεργασίας B2	Mfinished _B2
Τέλος κατεργασίας C2	Mfinished _C2

- **Καταστάσεις ενδιάμεσης αποθήκης (βοηθητικές μνημες στο PLC)**

Τεμάχιο B1 αποθηκευμένο στη θέση 1 του Buffer της φρέζας	Bm1_B1
Τεμάχιο B1 αποθηκευμένο στη θέση 2 του Buffer της φρέζας	Bm2_B1
Τεμάχιο B1 αποθηκευμένο στη θέση 3 του Buffer της φρέζας	Bm3_B1
Τεμάχιο B1 αποθηκευμένο στη θέση 4 του Buffer της φρέζας	Bm4_B1
Τεμάχιο C1 αποθηκευμένο στη θέση 1 Buffer της φρέζας	Bm1_C1
Τεμάχιο C1 αποθηκευμένο στη θέση 2 Buffer της φρέζας	Bm2_C1
Τεμάχιο C1 αποθηκευμένο στη θέση 3 Buffer της φρέζας	Bm3_C1
Τεμάχιο C1 αποθηκευμένο στη θέση 4 Buffer της φρέζας	Bm4_C1
Τεμάχιο F2 αποθηκευμένο στη θέση 1 του Buffer της φρέζας	Bm1_F2
Τεμάχιο F2 αποθηκευμένο στη θέση 2 του Buffer της φρέζας	Bm2_F2
Τεμάχιο F2 αποθηκευμένο στη θέση 3 του Buffer της φρέζας	Bm3_F2
Τεμάχιο F2 αποθηκευμένο στη θέση 4 του Buffer της φρέζας	Bm4_F2
Τεμάχιο E1 αποθηκευμένο στη θέση 1 του Buffer του τóρνου	Bt1_E1
Τεμάχιο E1 αποθηκευμένο στη θέση 3 του Buffer του τóρνου	Bt2_E1
Τεμάχιο E1 αποθηκευμένο στη θέση 3 του Buffer του τóρνου	Bt3_E1
Τεμάχιο E1 αποθηκευμένο στη	Bt4_E1

Θέση 4 του Buffer του τόρνου	
Τεμάχιο F1 αποθηκευμένο στη θέση 1 του Buffer του τόρνου	Bt1_F1
Τεμάχιο F1 αποθηκευμένο στη θέση 2 του Buffer του τόρνου	Bt2_F1
Τεμάχιο F1 αποθηκευμένο στη θέση 3 του Buffer του τόρνου	Bt3_F1
Τεμάχιο F1 αποθηκευμένο στη θέση 4 του Buffer του τόρνου	Bt4_F1
Τεμάχιο C2 αποθηκευμένο στη θέση 1 του Buffer του τόρνου	Bt1_C2
Τεμάχιο C2 αποθηκευμένο στη θέση 2 του Buffer του τόρνου	Bt2_C2
Τεμάχιο C2 αποθηκευμένο στη θέση 3 του Buffer του τόρνου	Bt3_C2
Τεμάχιο C2 αποθηκευμένο στη θέση 4 του Buffer του τόρνου	Bt4_C2

Το Buffer του τόρνου πλήρες	Bt_Free
Το Buffer της φρέζας πλήρες	Bm_Free
1 τεμάχιο αποθηκευμένο στο Buffer_m	Bm_1
2 τεμάχια αποθηκευμένα στο Buffer_m	Bm_2
3 τεμάχια αποθηκευμένα στο Buffer_m	Bm_3
4 τεμάχια αποθηκευμένα στο Buffer_m	Bm_4
1 τεμάχιο αποθηκευμένο στο Buffer_t	Bt_1
2 τεμάχια αποθηκευμένα στο Buffer_t	Bt_2
3 τεμάχια αποθηκευμένα στο Buffer_t	Bt_3
4 τεμάχια αποθηκευμένα στο Buffer_t	Bt_4

Βιβλιογραφία

- [1] M.C. Zhou , K. Venkatesh (1998). Modeling simulation and control of Flexible Manufacturing Systems A Petri Net Approach, Arizona :World Scientific
- [2] Γ.Βοσνιάκος 2000, Σχεδιασμός συστημάτων ελέγχου για συστήματα κατεργασιών
- [3] J. Jang, P-H. Koo και S.Y. Nof (1997). Application of design and control tools in a multirobot cell : Computers & Industrial Engineering ,32 (1) , pp 89-100
- [4] A. Zimmermann (1994). A Modeling Method for Flexible Manufacturing Systems based on Coloured Petri Nets : International Workshop on New directions of Control and Manufacturing , pp 147-154
- [5] Ξ.Β. Γωγουβίτης (2010) Εικονικά Συστήματα Κατεργασιών Βασισμένα σε Ευφυή Αντικείμενα , Διδακτορική διατριβή, ΕΜΠ, Σχολή Μηχανολόγων Μηχανικών, Τομέας Τεχνολογίας των Κατεργασιών.
- [6] S. Haddad, F. Vernardat, (2009), Analysis methods for Petri, Chapter 3 in M. Diaz (ed): Petri Nets: Fundamental Models, Verification and Applications , Wiley-ISTE.
- [7] L. Recalde , M. Silva, J Ezpeleta, E.Teruel (2004). Lectures on Concurrency and Petri Nets, Petri nets and manufacturing systems : An examples-driven tour , Berlin : Springer.
- [8] M. Dotoli, M. Pia Fantl (2005), A Coloured Petri Net Model for Automated Storage and Retrieval Systems Serviced by rail-Guided vehicles : a Control Perspective: International Journal of Computer Integrated Manufacturing 18(2-3) , pp 112-136
- [9] K. Jensen , L. M. Kristensen, L. Wells (2007) Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems : International Journal on Software Tools for Technology Transfer, 9, pp 213-254
- [10] N.Viswanadham, Y.Narahari (2014), Coloured Petri Net Models for Automated Manufacturing Systems ,. Paper presented at : Robotics and Automation Proceedings. 1987 IEEE International Conference on, Hong Kong, Volume: 4 , pp 1987-1990
- [11] K. Jensen (2009). Coloured Petri Nets Modelling and Validation of Concurrent Systems - Resource allocation (Sect. 1.2 of Vol. 1), Denmark : Springer
- [12] M. A. Barghash, O. M.Abuzeid, A. N.Al-Rabadi, A. M.Jaradat (2011). Petri Nets and Ladder Logic for Fully Automating and Programmable Logic Control of Semi-Automatic Machines and Systems : American Journal of Engineering and Applied Science, 4 ,pp 252-264
- [13] T. Perme (2009).Translation of Extended Petri Net Model into Ladder Diagram and Simulation with PLC : Journal of Mechanical Engineering , 10,pp 608-622
- [14] A. Torres Ferraz de Mello, M. Castro Barbosa, D. Jose dos Santos Filho, P. Eigi Miyagi, F. Junqueira (2012) A transcription tool from Petri Net to CLP Programming Languages : ABCM Symposium Series in Mechatronics, 5,pp 781-790

- [15] J-L Chirn and D.C. McFarlane(2000) Petri Nets Based Design of Ladder Logic diagrams: IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews) (IEEE T SYST MAN CY C), 34(4):523 - 531
- [16] G. Bum Lee, H. Zandong, J.S. Lee (2004). Automatic generation of ladder diagram with control Petri Net : Journal of Intelligent Manufacturing in the Netherlands, 15 ,pp 245-252
- [17] S.G.Tzafestas , M.G. Pantelelis, D.L. Kostis (2001). Design and Implementation of a Logic Controller using Petri Nets and Ladder Logic Diagrams : Systems Analysis Modeling and Simulation 42(1),pp 135-167
- [18] K.Feldmann, C.Schnur and W.Colombo (1996). Modularised, distributed real-time control of flexible production cells, using Petri Nets : Control Engineering practice, 4, pp 1067-1078
- [19] X. Li,. "Discrete Event System Modeling and Simulation." Proceedings of ICECS'99 3 (2000), pp. 1721-1724.
- [20] J.C. Quezada, J. Medina, E.Flores, J.C. Seck Tuoh, N.Hernandez (2014). Formal Design methodology for transforming ladder diagram to Petri nets : International Journal Advanced manufacturing technology, 73(5), pp. 821-836.
- [21] S. Klein, G. Frey, M. Minas (2003) , PLC Programming with Signal Interpreted Petri Nets. Paper presented at : ICATPN'03 Proceedings of the 24th international conference on Applications and theory of Petri nets, Eindhoven, June 2003 (pp 440-449), Springer-Verlag Berlin