



Πολυτεχνείο Κρήτης

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΔΙΑΤΑΣΣΟΜΕΝΑ ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΗΡΥ591

Αναφορά 2ου Milestone

ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΡΟΗΣ ΔΕΔΟΜΕΝΩΝ

Ον/μο	A.M
Ζήσкас Χρήστος	2014030191
Χρυσομάλλης Ιάσων	2014030078

ΖΗΤΟΥΜΕΝΑ

Η τελική σχεδίαση απαιτεί την κατασκευή 3 αντικειμένων:

- Κατασκευή καταχωρητών στους οποίους αποθηκεύονται οι τιμές των κανόνων.
- Κατασκευή καταχωρητών στους οποίους αποθηκεύονται ο αριθμός των φορών που έχουν ενεργοποιηθεί οι κανόνες.
- Ενημέρωση των υπόλοιπων στοιχείων του συστήματος καθώς και της απαιτούμενης επικοινωνίας μεταξύ τους.

ΥΛΟΠΟΙΗΣΗ

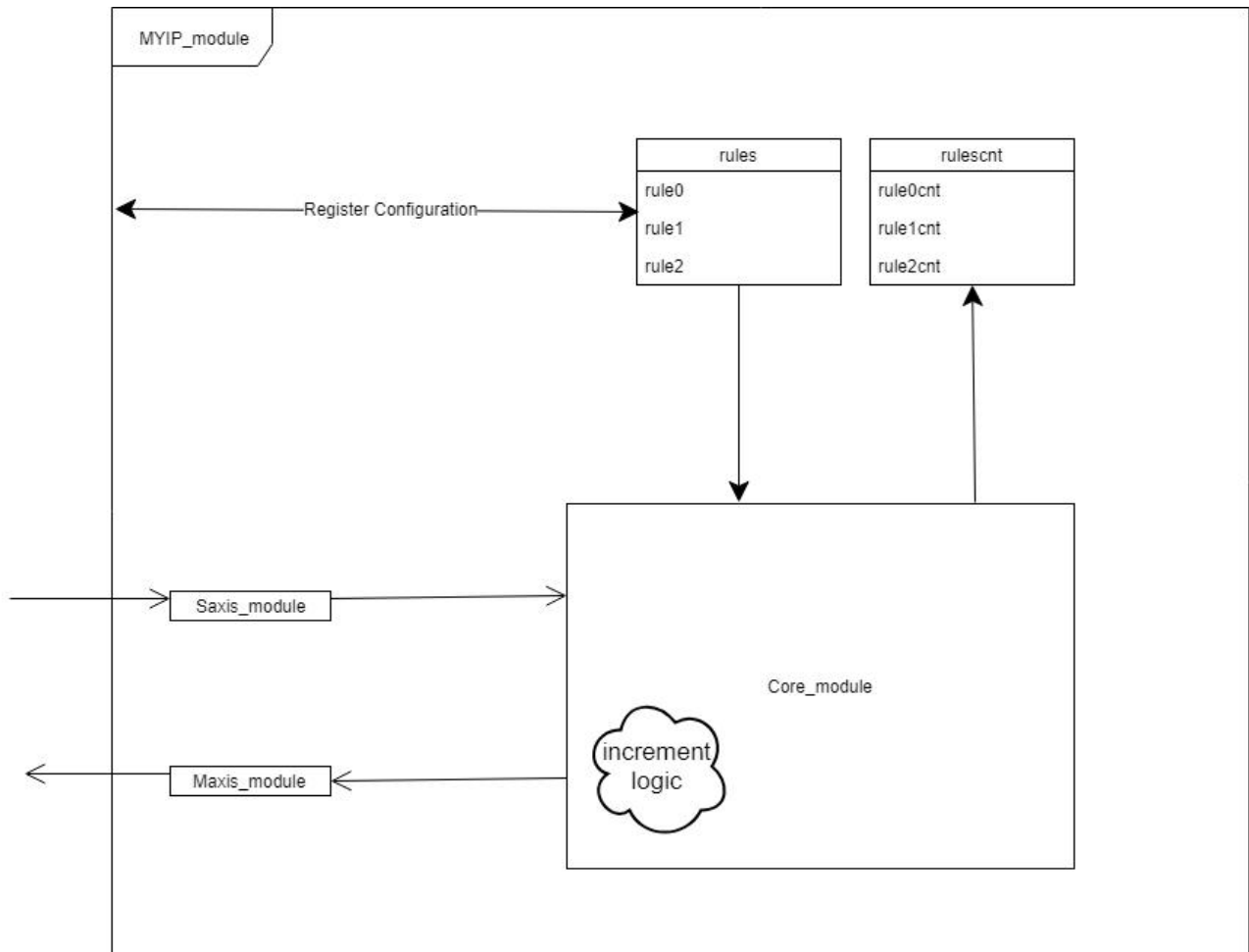
Η σχηματική αναπαράσταση των παράλληλων modules έχει προσαρμοσθεί ώστε να ακολουθεί την ζητούμενη υλοποίηση . Η κατασκευή της τελικής σχεδίασης απαιτεί την προσθήκη module που ανταποκρίνεται στις απαιτήσεις των καταχωρητών . Επιπλέον υπάρχει τροποποίηση του module του core, ώστε τα δεδομένα προς επεξεργασία να υφίστανται την επιθυμητή διαχείριση, ανάλογα την συμπεριφορά των κανόνων. Οι τιμές των μετρητών για κάθε κανόνα τοποθετούνται ως εξεγόμενα του πυρήνα. Η FSM του core εμπλουτίζεται με την προσθήκη διαχείρισης των δεδομένων που έχει αναφερθεί . Τα states στις FSM των slave και master heap έχουν διατηρηθεί.

Τροποποιήσεις/Προσθήκες

- 1) Εισαγωγή task που αφορά την κατασκευή συνάρτησης για εκχώρηση τιμών σε κάθε καταχωρητή.
- 2) Εισαγωγή task που αυξάνει τον counter που αφορά τον εκάστοτε κανόνα στη περίπτωση που τα δεδομένα και ο κανόνας συμπίπτουν.
- 3) Ενημέρωση των directives του myip module για την τοποθέτηση του επιθυμητού πλήθους καταχωρητών.
- 4) Εκτύπωση των μετρητών των κανόνων σε κάθε έλεγχο δεδομένων.

Σχηματική αναπαράσταση της υλοποίησης

Η σχεδίαση των διεπαφών για την επικοινωνία των component του συστήματος αποτυπώνεται στη σχεδίαση.



MODULE HEAPS

Δεν έχει υποστεί τροποποίηση σε σχέση με την αρχική σχεδίαση. Η απεικόνιση των fsm θεωρήθηκε μη ουσιαστική καθώς διαθέτει μόνο ένα state και μια είσοδο που οδηγεί στην ίδια κατάσταση. Αν η ουρά περιλαμβάνει δεδομένα, τότε διαβάζονται και τα εφοδιάζει σε ένα `stream<axi.word>`. Η saxis heap δέχεται δεδομένα από το χρήστη και τα αποστέλνει στο core ενώ το maxis heap δέχεται δεδομένα από το core και τα αποστέλνει στην έξοδο του συστήματος.

MODULE CORE

Προστίθεται λειτουργικότητα ως προς τον έλεγχο των δεδομένων. Με την νέα προσθήκη τα δεδομένα φιλτράρονται πριν την έξοδο. Αρχικά ελέγχεται η `saxis heap` να διαθέτει δεδομένα προς ανάγνωση. Εφόσον τα δεδομένα ικανοποιούν κάποιον από τους κανόνες, τότε υπάρχει μοναδιαία αύξηση του αντίστοιχου counter που εκπροσωπεί τον κανόνα και τα δεδομένα παραλείπονται. Σε αντίθετη περίπτωση τα δεδομένα εξάγονται προς εγγραφή στο `maxis_heap`. Οι τιμές των μετρητών εξάγονται από τον πυρήνα ώστε σε επόμενο στάδιο να ενημερωθούν οι καταχωρητές του `myip_block`. Η fsm διατηρεί τα states σύμφωνα με την αρχική σχεδίαση. Η απεικόνιση της fsm θεωρήθηκε μη ουσιαστική καθώς διαθέτει μόνο ένα state και μια είσοδο που οδηγεί στην ίδια κατάσταση.

MODULE RULES

Το module εμπλουτίζεται με συναρτήσεις που περιγράφουν τα χαρακτηριστικά των καταχωρητών. Υπάρχουν τρεις συναρτήσεις:

okanonas(uint32 rule,uint32 &exodos): Συνάρτηση εκχώρησης τιμής εισόδου σε μεταβλητή εξόδου ώστε να αποθηκεύεται η τιμή του κανόνα στον καταχωρητή.

okanonas2(uint32 rule,uint32 &exodos2): Συνάρτηση πρόσθεσης τιμής εισόδου στην μεταβλητή εξόδου ώστε να υπάρχει ενημέρωση των counter των κανόνων.

print(uint32 num1, uint32 num2, uint32 num3): Συνάρτηση εκτύπωσης αποτελεσμάτων. Εκτυπώνονται οι counter των rules μετά από κάθε data που υπάρχει στο stream

MODULE MYIP

Αρχικά έχουν προστεθεί τα κατάλληλα directives ώστε το σύστημα να εμπλουτιστεί με τον κατάλληλο αριθμό καταχωρητών. Συνολικά έξι καταχωρητές. Τρεις από τους καταχωρητές λαμβάνουν την τιμή που θα εκπροσωπεί τον εκάστοτε κανόνα μέσω της συνάρτησης **okanonas** . Έπειτα από την επεξεργασία των δεδομένων που έχει υλοποιηθεί στον πυρήνα, ενημερώνονται οι μεταβλητές των counter των κανόνων. Σε κάθε ταυτοποίηση ο counter παίρνει την τιμή μονάδας . Αλλιώς λαμβάνει μηδενική τιμή. Ο **okanonas2** ενεργοποιείται και προσθέτει την τιμή της ταυτοποίησης στην τιμή του καταχωρητή και έτσι ολοκληρώνεται η ενημέρωση. Σε κάθε έλεγχο δεδομένων τα counts εκτυπώνονται για τον έλεγχο ορθής υλοποίησης.

SIMULATION

Η επιτυχής λειτουργικότητα της σχεδίασης αποτυπώνεται στο επίπεδο προσομοίωσης. Έχει χρησιμοποιηθεί simulation που απεικονίζει το functionality του myip_module. Έχουν προκύψει δυο σενάρια λειτουργίας ώστε να επαληθευθεί η συνολική λειτουργικότητα του συστήματος

Το πρώτο σενάριο σχετίζεται με την επαναληπτική ροή ενός stream δεδομένων που αντιστοιχούν στις τιμές στο διάστημα [0~100]. Η επαναληπτική ροή πραγματοποιείται 3 φορές. Το πρώτο σενάριο επιλέχθηκε ώστε να διαπιστωθεί η ανταπόκριση του κυκλώματος σε μια απλή ροή δεδομένων

Το δεύτερο σενάριο αφορά την επανελειμμένη φραγή των δεδομένων για δεδομένα που ταυτίζονται με τους κανόνες. Το δεύτερο σενάριο επιλέχθηκε ώστε να διαπιστωθεί μια corner case περίπτωση στην οποία ανταποκρίνεται το σύστημα . Σε αυτή τη περίπτωση, όλα τα δεδομένα ταυτοποιούνται με τους κανόνες ,οπότε οι μετρητές αυξάνονται συνεχώς .

Στόχος του σεναρίου είναι η απόδειξη της ορθής λειτουργικότητα του συστήματος απαλλασόμενο πλέον από τον παράγοντα τύχαιας εκτέλεσης.

ANALYSIS SCENARIO#1

Οι counters είναι αρχικοποιημένοι σε μηδενικές τιμές. Σε αυτό το scenario, ζητούμενο είναι η επαλήθευση της βασικής λειτουργίας ροή δεδομένων, αναγνώρισης κανόνων και ενημέρωσης των counter. Το σενάριο ξεκινάει με τα παρακάτω αποτελέσματα.

```
Data stream starts
>-----<
Register 1 counter=0 after calculations
Register 2 counter=0 after calculations
Register 3 counter=0 after calculations
Time:0: read data: 1

Register 1 counter=0 after calculations
Register 2 counter=0 after calculations
Register 3 counter=0 after calculations
Time:1: read data: 2

Register 1 counter=0 after calculations
Register 2 counter=0 after calculations
Register 3 counter=0 after calculations
Time:2: read data: 3

Register 1 counter=0 after calculations
Register 2 counter=0 after calculations
Register 3 counter=0 after calculations
Time:3: read data: 4

Register 1 counter=0 after calculations
Register 2 counter=0 after calculations
Register 3 counter=0 after calculations
Time:4: read data: 5
```

Αρχικά, η ροή δεδομένων είναι κενή . Τα δεδομένα προς εγγραφή δημιουργούνται μέσω δυο loop

το ένα εμφωλευμένο στο άλλο. Το εξωτερικό loop έχει δημιουργηθεί ως μεσο για την επανάληψη του stream προς εγγραφή. Το εμφωλευμένο loop αναφέρεται στα περιεχόμενα του stream. Το περιεχόμενο του data είναι ο δείκτης του εμφωλευμένου loop αυξημένο κατά μια μονάδα. Εκτελείται η my_IP με παραμέτρους τους σωρούς , τα rules και τα rule_counters. Αν η maxis_hear είναι άδεια δηλώνει ότι τα δεδομένα απορρίφθηκαν λόγω των κανόνων. Επομένως οι counter αυξάνονται. Σε αντίθετη περίπτωση τα δεδομένα παρουσιάζονται στην έξοδο.

Τα περιεχόμενα του data αποκτούν τιμές στο διάστημα [1~100]. Στο σενάριο οι τιμές των κανόνων είναι [10 30 50] ενώ το stream συγκεντρώνει τις τιμές [1,2,3,4,5...100].

Αναμένεται ότι ο αριθμός των μετρητών αυξάνεται όταν το περιεχόμενο του stream είναι [10,30,50]. Για 3 επαναλήψεις οι counters τελικά θα έχουν λάβει τις τιμές [3 3 3].

```
Register 1 counter=3 after calculations
Register 2 counter=2 after calculations
Register 3 counter=2 after calculations
Time:9: no data available!
```

```
Register 1 counter=3 after calculations
Register 2 counter=3 after calculations
Register 3 counter=2 after calculations
Time:29: no data available!
```

```
Register 1 counter=3 after calculations
Register 2 counter=3 after calculations
Register 3 counter=3 after calculations
Time:49: no data available!
```

Το στιγμιότυπο επαληθεύει την ορθή λειτουργία της ροής των δεδομένων καθώς έχει ολοκληρωθεί η μεταφορά δεδομένων.

```
Register 1 counter=3 after calculations
Register 2 counter=3 after calculations
Register 3 counter=3 after calculations
Time:98: read data: 99
```

```
Register 1 counter=3 after calculations
Register 2 counter=3 after calculations
Register 3 counter=3 after calculations
Time:99: read data: 100
```

ANALYSE SCENARIO#2

Για τους σκοπούς του scenario#2 , οι μετρητές δεν έχουν αρχικοποιηθεί στο μηδέν και συνεχίζουν να λαμβάνουν τις τελικές τιμές του προηγούμενου σεναρίου . Όπως και να έχει η επαλήθευση της λειτουργικότητας είναι εφικτή.

```
Data :10
Register 1 counter=4 after calculations
Register 2 counter=3 after calculations
Register 3 counter=3 after calculations
Time:0: no data available!
```

```
Data :10
Register 1 counter=5 after calculations
Register 2 counter=3 after calculations
Register 3 counter=3 after calculations
Time:1: no data available!
```

```
Data :10
Register 1 counter=6 after calculations
Register 2 counter=3 after calculations
Register 3 counter=3 after calculations
Time:2: no data available!
```

```
Data :10
Register 1 counter=7 after calculations
Register 2 counter=3 after calculations
Register 3 counter=3 after calculations
Time:3: no data available!
```

Σε αυτό το scenario ζητούμενο είναι η επαλήθευση της εξειδικευμένης λειτουργίας ανάγνωσης και εγγραφής καθώς και η ορθή λειτουργία της ενημέρωσης των counter μέσω ταυτοποίησης δεδομένων και κανόνων . Το stream αποκτά δεδομένα συγκεκριμένων τιμών:

- ✧ Όταν ο αριθμός του Iteration είναι μικρότερος του 10, τότε το data γίνεται 10.
- ✧ Όταν ο αριθμός του Iteration είναι μικρότερος του 30, τότε το data γίνεται 30.
- ✧ Σε οποιαδήποτε άλλη περίπτωση το data γίνεται 50.


```
Data :10
Register 1 counter=4 after calculations
Register 2 counter=3 after calculations
Register 3 counter=3 after calculations
Time:0: no data available!

Data :10
Register 1 counter=5 after calculations
Register 2 counter=3 after calculations
Register 3 counter=3 after calculations
Time:1: no data available!

Data :30
Register 1 counter=13 after calculations
Register 2 counter=4 after calculations
Register 3 counter=3 after calculations
Time:10: no data available!

Data :30
Register 1 counter=13 after calculations
Register 2 counter=5 after calculations
Register 3 counter=3 after calculations
Time:11: no data available!

Data :50
Register 1 counter=13 after calculations
Register 2 counter=23 after calculations
Register 3 counter=72 after calculations
Time:98: no data available!

Data :50
Register 1 counter=13 after calculations
Register 2 counter=23 after calculations
Register 3 counter=73 after calculations
Time:99: no data available!
```

Τα rules διατηρούνται και σε αυτό το σενάριο ως [10 30 50].

Αναμένεται διαφοροποίηση των μετρητών σε κάθε iteration καθώς τα δεδομένα για κάθε επανάληψη συνάδουν με κάποιον απο τους κανόνες.

Τα τελικά αποτελέσματα είναι τα αναμενόμενα και αποτυπώνονται παρακάτω.

```
Data :50
Register 1 counter=13 after calculations
Register 2 counter=23 after calculations
Register 3 counter=72 after calculations
Time:98: no data available!

Data :50
Register 1 counter=13 after calculations
Register 2 counter=23 after calculations
Register 3 counter=73 after calculations
Time:99: no data available!
```