

Αναφορά Εργαστηριακής άσκησης 4

Ομάδα LAB31239665

Καλογεράκης Στέφανος AM:2015030064

Πίσκοπος Διονύσης AM:2015030115

Σκοπός εργαστηριακής άσκησης

Σκοπός της τέταρτης εργαστηριακής άσκησης ήταν ο σχεδιασμός της μονάδας ελέγχου του Control του Datapath, σε συνέχεια με το προηγούμενο εργαστήριο, με μια FSM αυτή την φορά για τις κατάλληλες καταστάσεις και ανάλογα με την εντολή που εκτελείται. Επιπρόσθετα, ζητείται η υλοποίηση εξαιρέσεων, για δύο διαφορετικές περιπτώσεις.

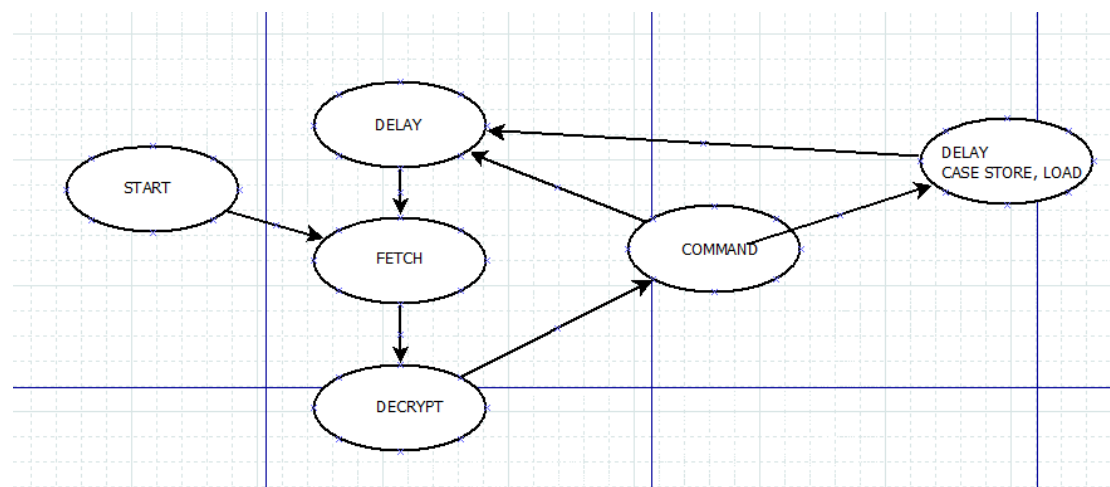
Προεργασία

Σαν παραδοτέα/προεργασία του εργαστηρίου ζητήθηκε τόσο ο πηγαίος κώδικας όλων των καινούργιων components που υλοποιήθηκαν όσο και η υλοποίηση ενός σχηματικού διαγράμματος του ολοκληρωμένου datapath με όλες τις συνδέσεις των καινούργιων components. Ακόμα μας ζητήθηκε η προσομοίωση συγκεκριμένων σημάτων που βασίστηκε σε ορισμένα αρχεία εισόδου.

Περιγραφή

Control FSM

Οι πιθανές καταστάσεις της FSM μας είναι αυτές που φαίνονται στην εικόνα



Ακολουθεί η αναλυτική περιγραφή των καταστάσεων

START: Αποτελεί την αρχική κατάσταση κατά την οποία εκκινεί το πρόγραμμα. Σε μια φυσιολογική ροή κώδικα μας η κατάσταση αυτή θα συναντάται μονάχα μια φορά κατά την εκκίνηση. Επόμενη της κατάσταση είναι η FETCH.

DELAY: Το Delay αποτελεί μια αρχική κατάσταση αναμονής, καθώς σε αυτή την κατάσταση ενεργοποιούμε το Enable του καταχωρητή IF_ID και περιμένουμε έναν κύκλο μέχρις ότου επισκεφτούμε την κατάσταση Fetch και ξεκινήσει κανονικά η ροή της κάθε εντολής.

FETCH: Από την κατάσταση Fetch περνάνε όλες οι καταστάσεις υποχρεωτικά καθώς γίνεται αύξηση του PC με σκοπό να έρθει επιτυχώς η επόμενη εντολή.

DECRYPT: Στο στάδιο decrypt πραγματοποιείται ουσιαστικά η «αποκρυπτογράφηση» της εκάστοτε εντολής που έχει έρθει από την μνήμη. Από αυτή την κατάσταση επίσης περνάνε αναγκαστικά όλες οι εντολές.

COMMAND: Το συγκεκριμένο στάδιο αποτελεί generic τύπο, το οποίο περιλαμβάνει την κάθε εντολή ξεχωριστά. Σε αυτό το στάδιο είναι και που υπάρχουν οι μεγαλύτερες διαφοροποιήσεις ανά περίπτωση, αφού καθορίζονται οι περισσότερες διεργασίες που πραγματοποιούνται.

DELAY CASE STORE, LOAD : Η συγκεκριμένη κατάσταση αποτελεί ειδική περίπτωση για τις εντολές lw, lb, sw, sb ο οποίες πρέπει να έχουν πρόσβαση στην μνήμη. Συγκεκριμένα, έχει πραγματοποιηθεί από το προηγούμενο στάδιο, η ενεργοποίηση του συγκεκριμένου καταχωρητή που είναι υπεύθυνος για την μνήμη και σε αυτό το στάδιο πραγματοποιείται η πρόσβαση στην μνήμη.

Παρατήρηση: Η αρχική μας προσέγγιση στο εργαστήριο πραγματοποιήθηκε χωρίς την χρήση καταχωρητών. Μετά από σχετική ερώτηση στον διδάσκοντα, μας επισημάνθηκε η δυνατότητα μη χρησιμοποίησης καταχωρητών για το συγκεκριμένο εργαστήριο. Παρόλα αυτά, διαφοροποιήσαμε κατάλληλα την υλοποίηση μας και τις καταστάσεις μας και προσθέσαμε τους καταχωρητές: IF_ID ο οποίος βρίσκεται μεταξύ IFSTAGE και DECTAGE, ο ID_EX ο οποίος βρίσκεται μεταξύ DECTAGE και EXSTAGE και ο MEMWB ο οποίος βρίσκεται μεταξύ EXSTAGE και Memory. Σε κάθε περίπτωση το σήμα Enable που έχουμε και είναι υπεύθυνο για τον κάθε καταχωρητή ενεργοποιείται πάντα ένα στάδιο πριν από την χρήση του καθώς οι καταχωρητές περιμένουν την θετική ακμή του επόμενου κύκλου.

Exceptions

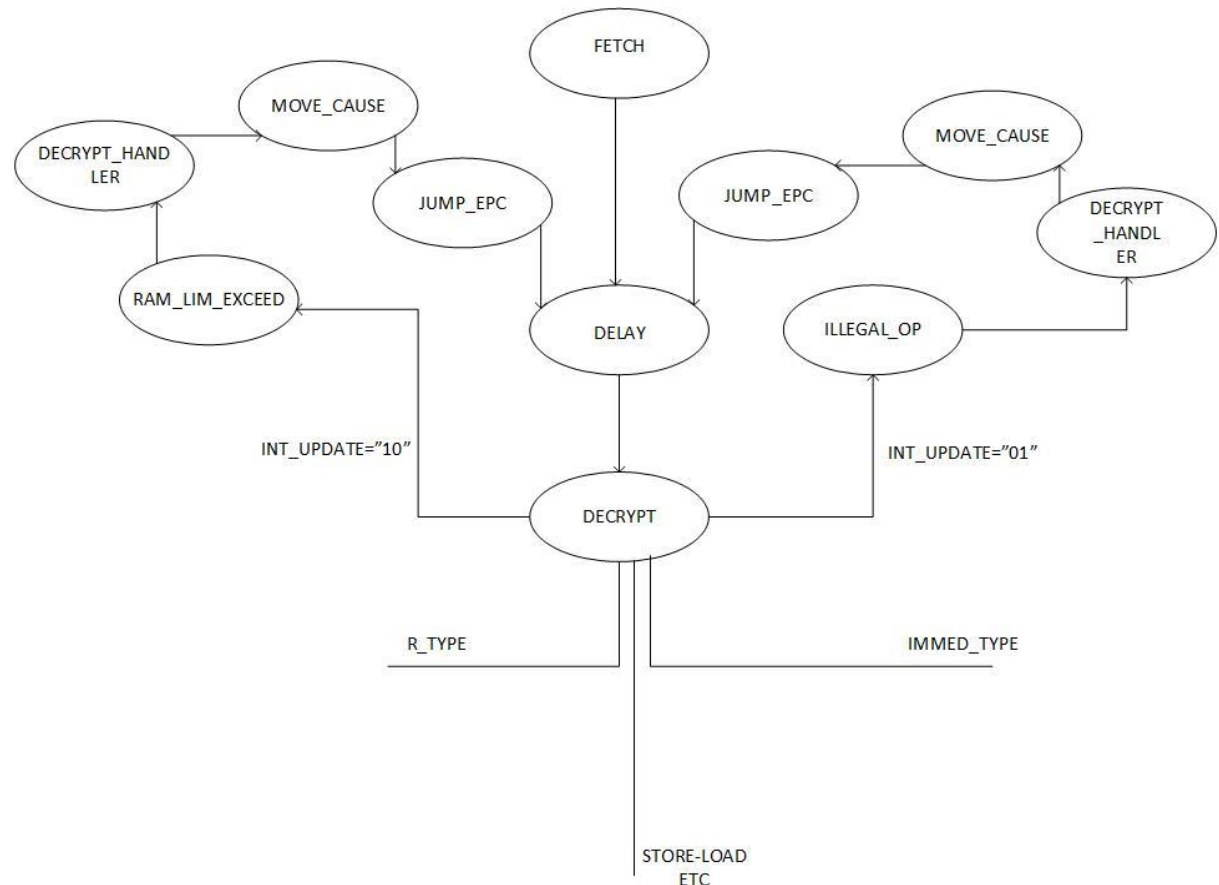
Στο κομμάτι των exceptions, χρειάστηκε να δημιουργηθούν αρκετά επιπλέον κυκλώματα ελέγχου έτσι ώστε να συμβαίνουν τα εξής στην περίπτωση των interrupts:

- Να ανιχνεύεται το σφάλμα, το οποίο μπορεί να είναι είτε λανθασμένο opcode είτε υπέρβαση του ορίου της ram και να καταγράφεται στο control.

- Στην συνέχεια, το control να ενεργοποιεί συγκεκριμένα επιπλέον σήματα, τα οποία θα αλλάζουν την διεύθυνση της ram και θα την κάνουν να μεταβαίνει στα κομμάτια κώδικα, ανάλογα την περίπτωση interrupt.

- Τέλος, να αποθηκεύονται το σφάλμα στον καταχωρητή cause, το οποίο θα τοποθετηθεί στην συνέχεια στον καταχωρητή r5, και επιπλέον ο καταχωρητής epc να κρατάει την διεύθυνση του σφάλματος, η οποία χρειάζεται για να θυμόμαστε την διεύθυνση της ram στην οποία θα μεταβούμε μόλις εκτελεστεί ο κώδικας του interrupt.

Με σκοπό να ανιχνεύσουμε το interrupt, αλλά και να εκτελέσουμε τον κατάλληλο κώδικα ανά περίπτωση, δημιουργήσαμε δύο συγκεκριμένες διαδρομές στην fsm controller, ανάλογα το σφάλμα που συμβαίνει. Οι δύο προηγούμενες, θα ελέγχονται αν συνέβησαν κατά το state DECRYPT, το οποίο και αναγνωρίζει όλες τις εντολές του συστήματος και δίνει ένα συγκεκριμένο path ανά περίπτωση εντολής. Στην συνέχεια, παρατίθεται το διάγραμμα των 2 περιπτώσεων interrupts, καθώς και το μονοπάτι που ακολουθεί μέχρι να επιστρέψει η fsm στην κανονική της λειτουργία.

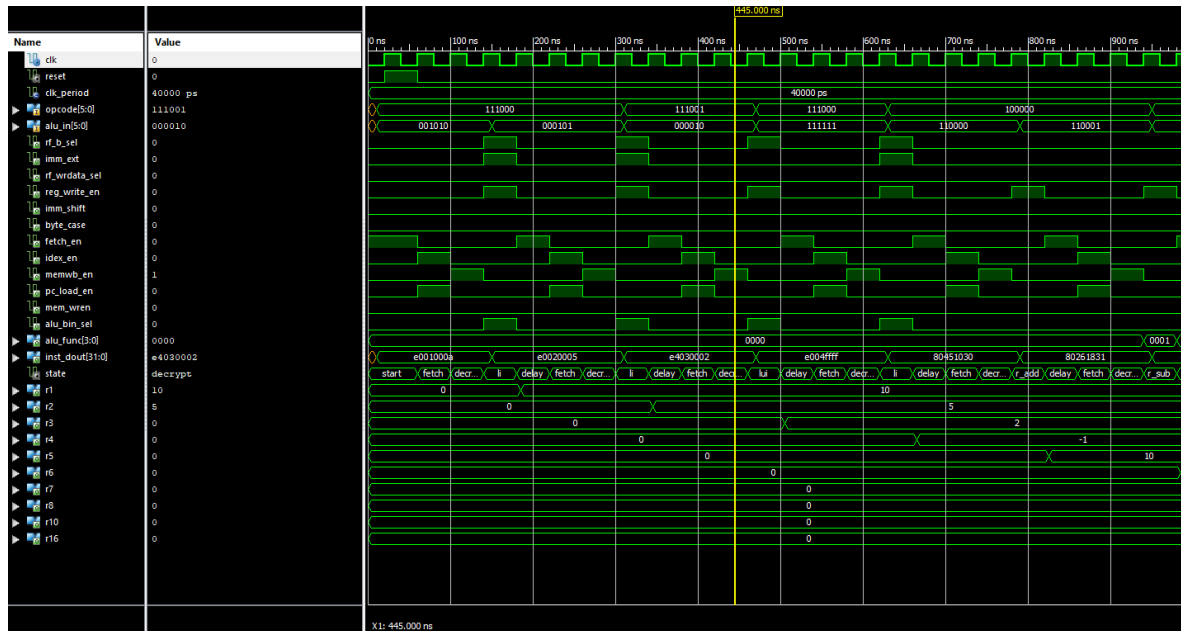


Σημειώνεται πως η επιστροφή του κάθε state στο fetch state μετα την εκτέλεσή του παραλείπεται για λόγους ευκρίνειας του κομματιού των interrupts. Η πλήρης διαδρομή της fsm δύνανται στο 1ο κομμάτι της αναφοράς.

Κυματομορφές-Προσομοίωση

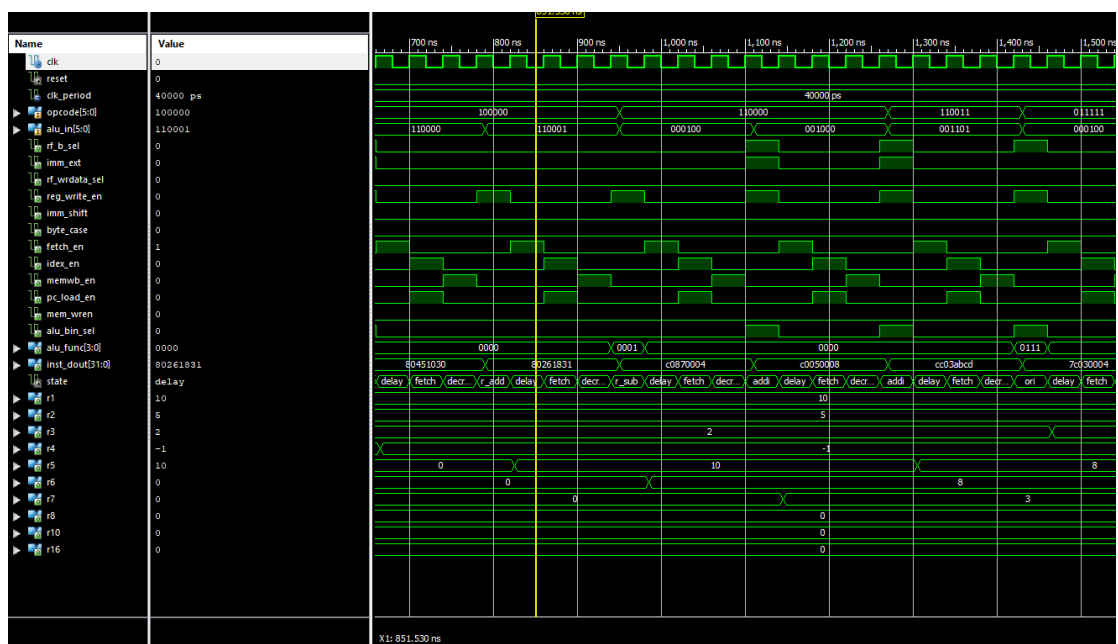
Control FSM

Παρακάτω παρουσιάζουμε ενδεικτική λειτουργία του Control μετά από δημιουργία δικού μας ram.data αρχείου



Οι εντολές που εκτελούνται στο συγκεκριμένο παράθυρο κατά σειρά είναι

- li r1, 10
- li r2, 5
- lui r3, 2
- li r4, -1

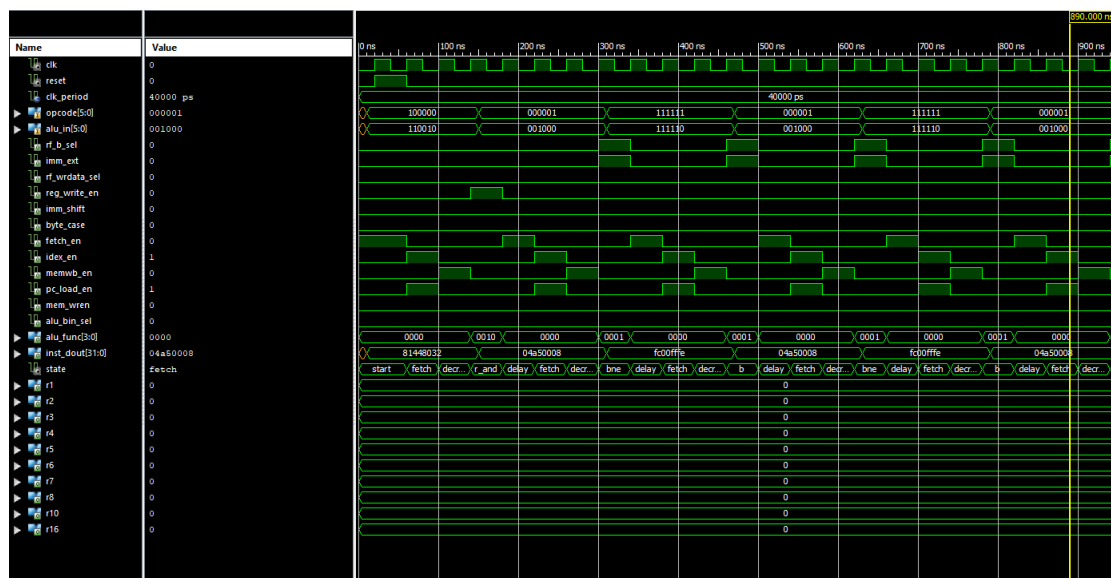


Οι εντολές που εκτελούνται στο συγκεκριμένο παράθυρο κατά σειρά είναι

- add r5, r2, r2
- sub r6, r1, r3
- addi r7, r4, 1

Μετά την εκτέλεση των συγκεκριμένων εντολών, πραγματοποιείται η εκτέλεση των προτύπων ram του προηγούμενου εργαστηρίου. Στην ήδη υπάρχουσα ram πραγματοποιείται η εκτέλεση του προγράμματος αναφοράς 1 ενώ σε ξεχωριστό αρχείο πραγματοποιείται η εκτέλεση του προγράμματος αναφοράς 2. Και στις δύο περιπτώσεις βέβαια συναντάμε τα αποτελέσματα του προηγούμενου εργαστηρίου, επαληθεύοντας την ορθή υλοποίησή μας.

Παρακάτω, παρατηρούμε χαρακτηριστικό παράδειγμα υλοποίησης του προηγούμενου εργαστηρίου, αφού απεικονίζεται ο ατέρμων βρόγχος του προγράμματος αναφοράς 2 όπως είχα συναντήσει.



Exceptions

Τέλος, ακολουθούν οι κυματομορφές που δείχνουν εκτέλεση των interrupts από την fsm control καθώς και τις επιδράσεις της σε σήματα του datapath.



Όπως γίνεται αντιληπτό από το παραπάνω διάγραμμα, έχουμε πλήρη εξυπηρέτηση των 2 interrupts, το ένα μετά το άλλο (λόγω των εντολών της ram που ζητήθηκαν στην εκφώνηση), έχουμε μεταβίβαση του καταχωρητή move_cause στον r5 και αλλά και αποθήκευση της διεύθυνσης του σφάλματος στον EPC.

Παράρτημα

Επισυνάπτεται παρακάτω το σχηματικό διάγραμμα συνδεσμολογίας του ολοκληρωμένου datapath όπως υλοποιήθηκε

