



Εργασία #2: Δημιουργία επεξεργαστή πολλαπλών κύκλων και μετατροπή του σε pipeline

Η εργασία χωρίζεται σε 2 φάσεις

4^η φάση: «Δημιουργία επεξεργαστή πολλαπλών κύκλων»

– Μελετήστε πρώτα καλά την εκφώνηση –

Σκοπός

Ο σκοπός της 4^{ης} φάσης είναι να μετατρέψετε τον επεξεργαστή ενός κύκλου που σχεδιάσατε στην Εργασία#1 σε επεξεργαστή πολλαπλών κύκλων.

Προαπαιτούμενα

Λειτουργική σχεδίαση επεξεργαστή ενός κύκλου

Διεξαγωγή

A) Μετατροπή του datapath του επεξεργαστή ενός κύκλου σε datapath πολλαπλών κύκλων

Μελετήστε το datapath που κατασκευάσατε στην Εργασία#1 και προσθέστε καταχωρητές μεταξύ των βαθμίδων όπου χρειάζεται ώστε να κρατάτε τις τιμές για κάποια σήματα που παράγονται από μια βαθμίδα και πρέπει να χρησιμοποιηθούν σε επόμενη βαθμίδα. Μπορείτε να κάνετε όσες επιπλέον αλλαγές κρίνετε απαραίτητες, είτε εσωτερικά σε επίπεδο σχεδίασης του datapath είτε σε επίπεδο διεπαφής. Ονομάστε το αρχείο σας **DATAPATH_MC.vhd**.

B) Σχεδιασμός και υλοποίηση της μονάδας ελέγχου (control)

Σχεδιάστε τη μονάδα ελέγχου, η οποία θα είναι μια Μηχανή Πεπερασμένων Καταστάσεων (FSM) που θα ελέγχει τη ροή εκτέλεσης της κάθε εντολής γεννώντας τα απαιτούμενα σήματα ελέγχου σε κάθε κύκλο ρολογιού. Η FSM έχει σαν είσοδο την εντολή (opcode), πιθανώς flags όπως το Zero, κ.λ.π., και σαν εξόδους όλα τα σήματα ελέγχου του Datapath. Ονομάστε το αρχείο σας **CONTROL_MC.vhd**.

Γ) Ολοκλήρωση και έλεγχος ορθότητας

Συνδέστε το datapath με το control ώστε να υλοποιήσετε την πλήρη λειτουργία ενός επεξεργαστή πολλαπλών κύκλων. Ονομάστε το αρχείο σας **PROCESSOR_MC.vhd**. Η κύρια μνήμη θα πρέπει να βρίσκεται εκτός αυτού του module.

Δείξτε την ορθότητα της σχεδίασης σας με τα προγράμματα αναφοράς της Εργασίας#1 και επίσης δημιουργήστε τουλάχιστον ένα ακόμη δικό σας πρόγραμμα αναφοράς το οποίο θα περιλαμβάνει όλες τις εντολές του ISA τουλάχιστον από μία φορά.

5^η φάση: «Δημιουργία pipeline επεξεργαστή»

– Μελετήστε πρώτα καλά την εκφώνηση –

Σκοπός

Ο σκοπός της 5^{ης} φάσης είναι να μετατρέψετε τον επεξεργαστή πολλαπλών κύκλων σε pipeline επεξεργαστή.

Προαπαιτούμενα

Λειτουργική σχεδίαση επεξεργαστή πολλαπλών κύκλων

Διεξαγωγή

A) Αλλαγές στο datapath του επεξεργαστή πολλαπλών κύκλων

Προσθέστε του κατάλληλους καταχωρητές pipeline όπου κρίνετε ότι χρειάζεται ώστε για όλες τις βαθμίδες του datapath που υλοποιήσατε στην 4^η φάση η έξοδος της κάθε βαθμίδας να γίνεται είσοδος στην επόμενη. Λάβετε επιπρόσθετα υπόψη ότι χρειάζεται προώθηση (forwarding) ή/και stalls για την αντιμετώπιση των κινδύνων δεδομένων (data hazards). Βάση αυτού υλοποιήστε τις απαιτούμενες επεκτάσεις στη σχεδίαση σας. Ονομάστε το αρχείο σας **DATAPATH_PIPELINE.vhd**.

B) Σχεδιασμός και υλοποίηση της μονάδας ελέγχου (control)

Σχεδιάστε τη μονάδα ελέγχου ώστε να δημιουργεί τα απαιτούμενα σήματα ελέγχου για κάθε εντολή. Μπορείτε να χρησιμοποιήσετε είτε τη μονάδα ελέγχου του επεξεργαστή ενός κύκλου (Εργασία#1) είτε τη μονάδα ελέγχου του επεξεργαστή πολλαπλών κύκλων (4^η φάση) ως σημείο εκκίνησης. Ανάλογα με τη σχεδίαση της μονάδας ελέγχου που θα χρησιμοποιήσετε ως σημείο εκκίνησης, θα χρειαστεί να κάνετε και τις απαιτούμενες αλλαγές. Θα πρέπει να ακολουθήσετε τη θεωρία και να σχεδιάσετε το control για τον pipeline επεξεργαστή ανάλογα με τα παραδείγματα που παρουσιάστηκαν στις διαλέξεις. Λάβετε υπόψη ότι υπάρχουν και data hazards μεταξύ εντολών που μπορούν να αντιμετωπιστούν με forwarding ή/και με stalls και πιθανό να χρειαστούν επιπλέον επεκτάσεις στο control (ανάλογα με τη συνολική σχεδίαση του επεξεργαστή σας). Αγνοήστε τα control hazards. Ονομάστε το αρχείο σας **CONTROL_PIPELINE.vhd**.

Προτείνεται να βασιστείτε στην μονάδα ελέγχου του επεξεργαστή ενός κύκλου ως σημείο εκκίνησης.

Γ) Ολοκλήρωση και έλεγχος ορθότητας

Συνδέστε το datapath με το control ώστε να υλοποιήσετε την πλήρη λειτουργία ενός pipeline επεξεργαστή. Ονομάστε το αρχείο σας **PROCESSOR_PIPELINE.vhd**.

Δείξτε την ορθότητα της σχεδίασης σας με ένα δικό σας πρόγραμμα αναφοράς τα οποίο θα περιλαμβάνει μόνο τις εντολές **li**, **lw**, **sw** και **add** πολλές φορές ώστε να φανεί η λειτουργία του pipeline και η αντιμετώπιση των κινδύνων δεδομένων.

Αναλυτική περιγραφή παραδοτέων Εργασίας #2

Ο τελικός φάκελος που θα υποβάλλετε θα φέρει το όνομα της ομάδας. Πχ. LAB12345678. Εσωτερικά θα περιέχει 3εις υποφακέλους: REPORT, SOURCES, WAVEFORMS.

1. Αναφορά (μέχρι 6 σελίδες, PDF)

Θα βρίσκεται στον υποφάκελο REPORT. Η αναφορά δεν θα περιλαμβάνει κώδικα εκτός αν είναι code snippet μέχρι 10 γραμμές και ακολουθείται από τον αντίστοιχο σχολιασμό. Η αναφορά θα πρέπει να περιλαμβάνει block diagrams για τους δύο επεξεργαστές που έχετε σχεδιάσει.

2. Κώδικας VHDL

Όλα τα αρχεία vhd που έχετε δημιουργήσει στον φάκελο SOURCES, οργανωμένα σε υποφακέλους όπως περιγράφεται παρακάτω.

Περιεχόμενα φακέλου SOURCES

1. Φάκελος **MC**: όλα τα αρχεία vhd για τον επεξεργαστή πολλαπλών κύκλων
2. Φάκελος **PIPELINE**: όλα τα αρχεία vhd για τον pipeline επεξεργαστή

Διευκρίνιση

Δεν χρειάζεται να υποβάλλετε τα vhd αρχεία που υποβάλλατε στην Εργασία#1 εκτός αν τα έχετε τροποποιήσει. Σε αυτή την περίπτωση δημιουργήστε ένα επιπλέον φάκελο μέσα στον φάκελο SOURCES με όνομα **SUBMODULES** και προσθέστε τα εκεί. Κάθε vhd αρχείο στο φάκελο SUBMODULES θα πρέπει να περιέχει σε σχόλια μια σύντομη περιγραφή και αιτιολόγηση των αλλαγών που έγιναν.

3. Κυματομορφές Προσομοίωσης και Προγράμματα Αναφοράς

Όλες οι κυματομορφές προσομοίωσης θα βρίσκονται στον φάκελο WAVEFORMS. Θα σας βοηθήσει η χρήση του Waveform Configuration File (.wcfg) που προσφέρει η Xilinx, για ν' αποθηκεύετε τα σήματα σε αρχείο. Το αρχείο αυτό θα το κάνετε Load μετά από κάθε compile, για να εμφανίζονται αυτόματα τα σήματα που θέλετε να παρακολουθείτε. Στον ίδιο φάκελο θα βρίσκονται και τα προγράμματα αναφοράς που δημιουργήσατε, συνοδευόμενο το καθένα από ένα text αρχείο που θα περιλαμβάνει τις assembly εντολές του προγράμματος αναφοράς.

ΗΜΕΡΟΜΗΝΙΑ ΥΠΟΒΟΛΗΣ

18 Μαΐου