



**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
**ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ**  
**ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:**  
**Οργάνωση Υπολογιστών**  
**ΗΡΥ 302**  
**<http://www.mhl.tuc.gr>**  
**ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2024**

## **Αναφορά Εργαστηριακής Άσκησης 1**

Αριθμός Ομάδας:66

Λαμπράκης Μιχάλης 2020030077  
Δήμας Χρήστος 2021030183

### **1.Σκοπός της Άσκησης**

Ο σκοπός της Άσκησης αυτής είναι η σχεδίαση ενός πλήρους λειτουργικού επεξεργαστή μονο κύκλου με την χρήση της γλώσσας VHDL. Η σχεδίαση χωρίστηκε σε 3 φάσεις με σκοπό την πλήρη κατανόηση της λειτουργίας του επεξεργαστή.

Στην 1<sup>η</sup> φάση σχεδιάστηκε μια μονάδα αριθμητικών και λογικών πράξεων (ALU) καθώς και ένα αρχείο καταχωρητών(Register File).

Στην 2<sup>η</sup> σχεδιάστηκαν οι βασικές βαθμίδες του Datapath του επεξεργαστή. Αυτές είναι βαθμίδα ανάκλησης εντολών(IFSTAGE), αποκωδικοποίησης εντολών(DECSTAGE), εκτέλεσης εντολών(EXSTAGE) και πρόσβασης μνήμης (MEMSTAGE). Για την σχεδίαση χρειάστηκαν η (ALU) και ο (Register file) από την προηγούμενη φάση καθώς και με την χρήση του εργαλείου Xilinx Core Generator υλοποιήθηκαν 2 μνήμες, μια ROM 1024x32 για την αποθήκευση των βασικών εντολών και μια κύρια μνήμη RAM 1024x32.

Τέλος, στην 3<sup>η</sup> φάση συνδέθηκαν τα επιμέρους στοιχεία για να κατασκευαστεί το ενιαίο DATAPATH καθώς και υλοποιήθηκε το CONTROL του επεξεργαστή το οποίο παράγει τα σωστά σήματα ελέγχου για την κάθε εντολή.

### **2.Περιγραφή της Σχεδίασης**

#### **1<sup>η</sup> φάση**

Στην 1<sup>η</sup> φάση αρχικά σχεδιάστηκε η μονάδα αριθμητικών και λογικών πράξεων (ALU) η οποία παίρνει σαν είσοδο δύο 32bit τελεστέους και έναν 4bit κωδικό πράξης. Ανάλογα με τον κωδικό, η ALU βγάζει στην έξοδο το αποτέλεσμα της πράξης καθώς και 3 διαφορετικά flags που επισημαίνουν αν το αποτέλεσμα της πράξης είναι α) 0, β)έχει κρατούμενο εξόδου 1, γ)έχει υπερχειλίση. Όσον αφορά την υπερχειλίση, μπορεί να προκύψει μόνο στην πρόσθεση και στην αφαίρεση. Στην πρόσθεση, υπάρχει υπερχειλίση αν και οι δύο προσθετέοι είναι ομόσημοι μεταξύ τους και ετερόσημοι με το αποτέλεσμα, ενώ στην αφαίρεση έχουμε αν ο μειωτέος είναι είναι θετικός και ο αφαιρετέος αρνητικός και η διαφορά αρνητική η το ανάποδο. Για την υλοποίηση των πράξεων χρησιμοποιήθηκαν οι βιβλιοθήκες IEEE.NUMERIC\_STD.ALL και IEEE.STD\_LOGIC\_SIGNED.ALL ώστε να μην χρειαστεί να υλοποιηθούν από την αρχή τα κυκλώματα για τις πράξεις.

Στη συνέχεια σχεδιάστηκαν τα component τα οποία αποτελούν το αρχείο καταχωρητών (register file). Αρχικά σχεδιάστηκε ένας σύγχρονος καταχωρητής 32bit με Reset και WriteEnable. Στη συνέχεια ένας πολυπλέκτης ο οποίος ανάλογα με το σήμα sel (5bit) επιλέγει ποιά απο τις 32 εισόδους θα περάσει στην έξοδο και τέλος ένας αποκωδικοποιητής που ανάλογα με το σήμα εισόδου awr (5bit) επιλέγει ποιά απο τα 32bit της εξόδου θα είναι 1.

Το αρχείο καταχωρητών περιέχει 2 εισόδους ασύγχρονης ανάγνωσης που ελέγχουν 2 πολυπλέκτες και καθορίζουν τις εξόδους που θα διαβαστούν, 1 είσοδο σύγχρονης εγγραφής που μέσω του αποκωδικοποιητή καθορίζει σε ποιόν καταχωρητή θα γράψουμε, την είσοδο των δεδομένων, τα σήματα ελέγχου Reset και WriteEnable και το ρολόι. Χρειάστηκαν 32 καταχωρητές οι οποίοι δημιουργήθηκαν μέσω for-generate. Από αυτούς η τιμή του R0 είναι πάντα 0.

## 2<sup>η</sup> φάση

Η 2<sup>η</sup> φάση αποτελείται από 4 στάδια. Αρχικά το 1<sup>ο</sup> στάδιο είναι η βαθμίδα ανάκληση εντολών (IF Stage). Σε αυτό το στάδιο διαβάζεται μια εντολή από την μνήμη (χρησιμοποιούμε μια Distributed ROM) και την φορτώνουμε σε έναν καταχωρητή, τον Program Counter, ο οποίος σε κάθε κύκλο περιέχει την εντολή που εκτελείται την συγκεκριμένη χρονική στιγμή. Στον επόμενο κύκλο ο PC προχωράει στην επόμενη εντολή και παίρνει την τιμή (PC+4) εκτός αν η εντολή είναι brunch, brunch equal ή brunch not equal όπου τότε ο PC παίρνει την τιμή (PC+4+Immediate) και τότε η εκτέλεση του προγράμματος μεταφέρεται στην διεύθυνση αυτή. Η μνήμη λαμβάνει ως είσοδο τα [11:2] bits της εξόδου του καταχωρητή αφού αυτή είναι οργανωμένη σε words. Έτσι όσο αυξάνεται ο PC κατά 4 η μνήμη θα αυξάνεται κατά 1.

Το 2<sup>ο</sup> στάδιο αποτελεί η βαθμίδα αποκωδικοποίησης εντολών (DEC Stage). Σε αυτό το στάδιο η εντολή που φορτώνεται από την μνήμη αποκωδικοποιείται, με σκοπό την εγγραφή και προσπέλαση της στο αρχείο καταχωρητών. Επιπλέον, γίνεται και η κατάλληλη επεξεργασία της τιμής Immediate. Ο καταχωρητής ανάγνωσης 1 είναι πάντα ο rs ενώ ο καταχωρητής 2 είναι είτε ο rt είτε ο rd. Ένα πολυπλέκτης καθορίζει ποιός από τους 2 θα περάσει, όπου εκεί πρακτικά γίνεται και ο διαχωρισμός των R-Type από τις I-Type εντολές. Ο καταχωρητής εγγραφής είναι πάντα ο rd. Ακόμη η επεξεργασία του Immediate γίνεται στην μονάδα "Cloud" όπου ανάλογα με την τιμή του σήματος CloudControl γίνεται 1.Zero Fill, 2.Sign Extend, 3.Zero Fill & Shift και 4.Sign Extend & Shift 2. Τέλος, η εισαγωγή των δεδομένων προς εγγραφή στο Register File επιλέγεται μέσω ενός άλλου πολυπλέκτη να είναι είτε από την μνήμη RAM σε περίπτωση των εντολών load και store είτε από την έξοδο της ALU στις υπόλοιπες εντολές.

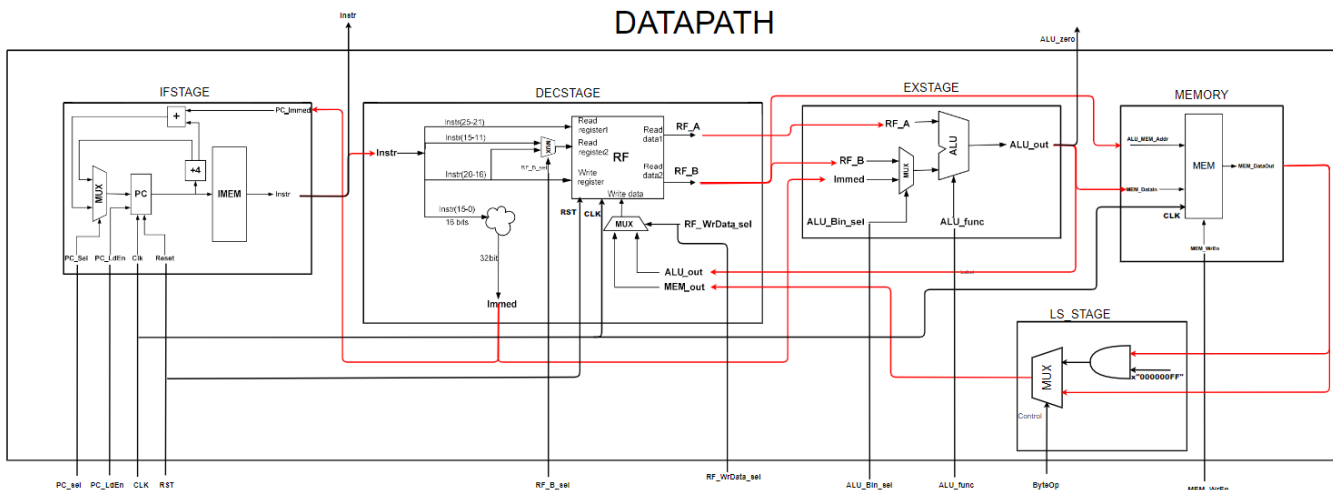
Το 3<sup>ο</sup> στάδιο είναι η βαθμίδα εκτέλεσης εντολών (EX Stage). Σε αυτό το στάδιο γίνεται η εκτέλεση της εντολής και ο υπολογισμός του αποτελέσματος της ALU. Το 1<sup>ο</sup> σήμα εισόδου της αποτελούν τα δεδομένα ανάγνωσης του καταχωρητή 1(έξοδος από Register File) και το 2<sup>ο</sup> μέσω ενός πολυπλέκτη είναι είτε τα δεδομένα ανάγνωσης του καταχωρητή 2 είτε το επεξεργασμένο Immediate.

Το 4<sup>ο</sup> και τελευταίο στάδιο αποτελεί η βαθμίδα πρόσβασης μνήμης (MEM Stage). Σε αυτό το στάδιο υλοποιήθηκε μια μνήμη RAM 1024 θέσεων των 32bit. Η μνήμη είναι Read First και έχει μια θύρα ανάγνωσης και εγγραφής.

Τέλος για την υλοποίηση της εντολής load byte έχει δημιουργηθεί ένα LS Stage το οποίο παίρνει σαν είσοδο την έξοδο της μνήμης RAM και μέσω ενός πολυπλέκτη επιλέγει εάν θα βγει αυτούσια στην έξοδο είτε αν ,στην περίπτωση της εντολής load byte, γίνει Zero Fill στα bit [31:8].

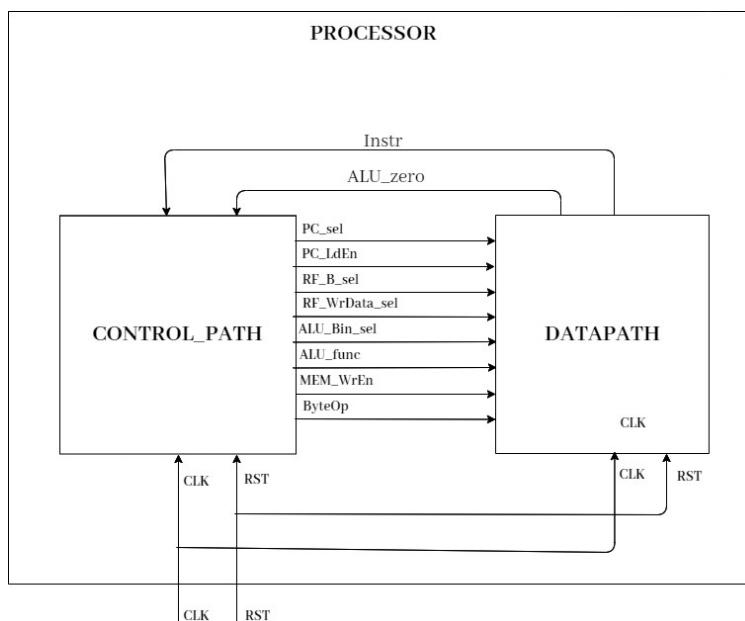
### 3<sup>η</sup> φάση

Στην 3<sup>η</sup> φάση ενώθηκαν τα 4 στάδια και υλοποιήθηκε το DATAPATH το οποίο πρακτικά είναι υπεύθυνο για να πραγματοποιήσει όλες τις λειτουργίες του επεξεργαστή, ανάλογα με τις εντολές που εισάγονται.



Στην συνέχεια δημιουργήθηκε το CONTROL το οποίο είναι υπεύθυνο να στέλνει τα κατάλληλα σήματα ελέγχου στο DATAPATH ανάλογα με την εντολή που εκτελείται. Παίρνει σαν εισόδους ένα ασύγχρονο Reset καθώς και το flag Zero της ALU το οποίο χρησιμοποιείται στις εντολές διακλαδώσεων brunch και παράγει εξόδους για όλα τα σήματα εισόδου του DATAPATH.

Το DATAPATH και το CONTROL ενώθηκαν σε ένα Top Level αρχείο PROCESSOR το οποίο τρέχει και επιβεβαιώνει την ορθή λειτουργία του επεξεργαστή. Δεν έχει εισόδους και εξόδους πέρα από το Reset και το ρολόι.



#### 4. Συμπεράσματα

Αυτή η άσκηση είναι μια πολύ καλή εξάσκηση για την κατανόηση της λειτουργίας ενός επεξεργαστή ενός κύκλου. Εντρυφήσαμε στην VHDL, την κατανοήσαμε και μάθαμε να χειριζόμαστε τις βιβλιοθήκες της για την διευκόλυνση της υλοποίησης των κυκλωμάτων. Μέσω εξαντλητικών δοκιμών (testbenches) καταφέραμε να επιβεβαιώσουμε την λειτουργία του επεξεργαστή και κατανοήσαμε πόσο σημαντική είναι αυτή η διαδικασία για την αποφυγή λαθών και την εξοικονόμηση χρόνου. Τέλος, μάθαμε πως να κάνουμε ιεραρχική σχεδίαση, να χωρίζουμε τα sub modules και να τα συνδέουμε σε ένα Top Level αρχείο.