



Ψηφιακά ΗW I

ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ

Στην παρούσα εργαστηριακή αναφορά παρουσιάζεται η 1^η εργαστηριακή άσκηση, καθώς δεν πρόλαβα να υλοποιήσω τις υπόλοιπες. Η 1^η άσκηση πραγματεύεται τη δημιουργία μιας ALU (Arithmetic Logic Unit), η οποία πραγματοποιεί μια σειρά από αριθμητικές και λογικές πράξεις ανάλογα με τις εισόδους που δέχεται. Στην ουσία παράγει μια έξοδο, δηλαδή ένα συγκεκριμένο αποτέλεσμα. Οι παράμετροι που δέχεται ως είσοδο είναι τρεις. Πιο συγκεκριμένα, δύο αριθμοί των 16-bit με ονόματα **op1**, **op2** και μια είσοδος των 4-bit με όνομα **alu_op**. Οι είσοδοι **op1** & **op2** αποτελούν τους αριθμούς πάνω στους οποίους συντελείται η αριθμητική ή λογική πράξη, ενώ η είσοδος **alu_op** προσδιορίζει την πράξη (ή αλλιώς τον τελεστή) ανάμεσα στις εισόδους αυτές. Το αποτέλεσμα των πράξεων παράγεται στη μεταβλητή εξόδου ενός αριθμού 32-bit, με όνομα **result**, ενώ υπάρχει και μια έξοδος αριθμού μεγέθους 1-bit με όνομα **zero**, η οποία τίθεται σε τιμή 1 οποτεδήποτε η έξοδος **result** ισούται με το 0. Στόχος της άσκησης είναι η δημιουργία ενός συνδυαστικού κυκλώματος Αριθμητικής και Λογικής Επεξεργασίας, το οποίο να δίνει τις απαραίτητες εξόδους σύμφωνα με τις πράξεις που υλοποιούνται στις εισόδους.

Αρχίζοντας, κωδικοποιήσαμε όλες τις δυνατές αριθμητικές και λογικές πράξεις ως εξής :

```
//encoding the operation
parameter [3:0] ALUOP_AND = 4'b0000,
                ALUOP_OR  = 4'b0001,
                ALUOP_ADD = 4'b0010,
                ALUOP_SUB = 4'b0110,
                ALUOP_LET = 4'b0111,
                ALUOP_LSR = 4'b1000,
                ALUOP_LSL = 4'b1001,
                ALUOP_ASR = 4'b1010,
                ALUOP_XOR = 4'b1101;
```

|

Στον εργαστηριακό οδηγό, όπως και από τις επισημάνσεις που δόθηκαν στην επίλυση αποριών των ασκήσεων του εργαστηρίου, αναφέρεται ρητά πως λαμβάνουμε τις εισόδους **op1** & **op2** σε δυαδικούς αριθμούς συμπληρώματος ως προς 2 και πρέπει να

λάβουμε την έξοδο **result** επίσης σε συμπλήρωμα ως προς 2.
Επομένως σε περίπτωση που η έξοδος **result** είναι 1 σε δεκαδικό σύστημα, τότε σε δυαδικό συμπληρώματος ως προς δύο είναι

111 ... 111 (σύνολο 32 bit)

ενώ το μηδενικό αποτέλεσμα σε συμπλήρωμα ως προς δύο είναι

000 ... 000 (σύνολο 32 bit)

Για περισσότερες πληροφορίες σχετικά με τα δυαδικά συμπληρώματα ως προς 2 μπορούμε να ανατρέξουμε σε συγκεντρωτικούς πίνακες ή σε ηλεκτρονικούς μετατροπείς αριθμών. Λαμβάνοντας υπόψη αυτά και τις οδηγίες του εργαστηριακού οδηγού, πηγάζει και ο επισυναπτόμενος στα αρχεία κώδικας σε Verilog που υλοποίησα. Είναι εμπλουτισμένος με σχόλια, όπου το θεώρησα κρίσιμο. Παραθέτω επίσης, για διευκόλυνση, την υλοποίηση ενός δικού μου testbench για την επαλήθευση του κυκλώματος. Επίσης το σχηματικό διάγραμμα του συνδυαστικού κυκλώματος που προκύπτει, σύμφωνα με τον κώδικα, αποτυπώνεται παρακάτω :



