|  |  |
| --- | --- |
|  | **Πανεπιστήμιο Δυτικής Αττικής**  **Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών** |

****

**Σχεδίαση Ψηφιακών Συστημάτων**

**Ακαδημαϊκό έτος 2022-2023**

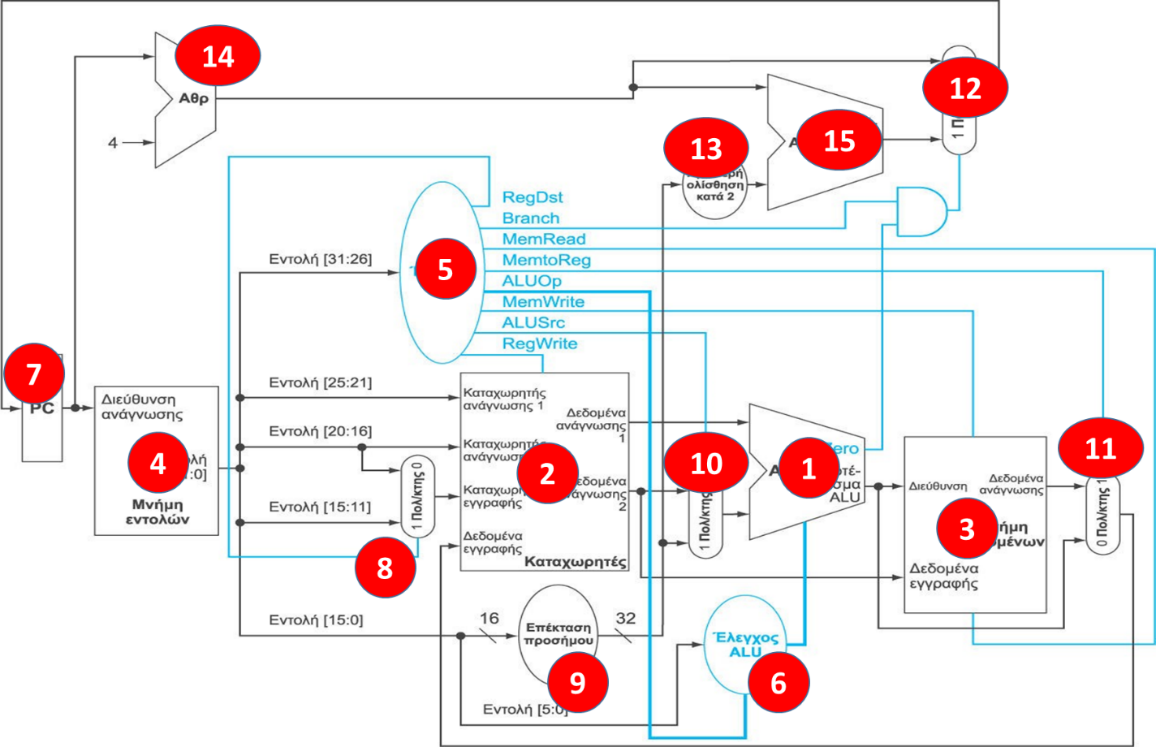
**Εξαμηνιαία Άσκηση Μαθήματος**

**Ημερομηνία παράδοσης:** 16/6/2023

Σκοπός της εργασίας είναι η σχεδίαση και υλοποίηση του επεξεργαστή MIPS απλού κύκλου. Ο επεξεργαστής έχει ως είσοδο τα σήματα reset και clock, ενώ δεν έχει έξοδο. Το σήμα reset οδηγεί τη μονάδα PC στην τιμή 0. Επίσης οδηγεί το Register file και μηδενίζει όλους τους καταχωρητές.

Ο επεξεργαστής περιλαμβάνει τις ακόλουθες μονάδες (η αρίθμηση φαίνεται στην Εικόνα 1)

|  |  |
| --- | --- |
| 1 | ALU |
| 2 | Register file |
| 3 | Μνήμη δεδομένων |
| 4 | Μνήμη εντολών (θα υλοποιηθεί σαν ROM) |
| 5 | Μονάδα ελέγχου |
| 6 | Μονάδα ελέγχου ALU |
| 7 | PC |
| 8 | 5-πλό πολυπλέκτη 2-σε-1 |
| 9 | Μονάδα επέκτασης προσήμου 16-σε-32 |
| 10, 11, 12 | 32-πλό πολυπλέκτη 2-σε-1 |
| 13 | Μονάδα ολίσθησης αριστερά κατά 2 (32-bit) |
| 14 ,15 | Αθροιστή 32 bits |
|  | Πύλη AND 2 εισόδων |



**Εικόνα 1**: Υλοποίηση Επεξεργαστή

Ο επεξεργαστής θα εκτελεί τις εντολές: add, sub, addi, lw, sw, **bne** (προσοχή, μπορεί να χρειαστεί να τροποποιηθεί ελαφρά η δομή του επεξεργαστή της Εικόνας 1)

Για ευκολία στην υλοποίηση θεωρήστε ότι:

* Η μνήμη εντολών έχει 16 θέσεις (των 32 bits)
* Η μνήμη δεδομένων έχει 16 θέσεις (των 32 bits)
* Το αρχείο καταχωρητών έχει 16 θέσεις (των 32 bits)
* Μετά την εκτέλεση κάθε εντολής το περιεχόμενο του PC αυξάνει κατά 1

Υλοποιήστε τα ακόλουθα:

**1. Entity MIPS**

Δημιουργήστε ένα entity MIPS.

**2.** Τοποθετήστε στη μνήμη εντολών τις ακόλουθες εντολές, ξεκινώντας από τη θέση 0.

addi $0, $0, 0 # δε χρειάζεται (οι καταχωρητές έχουν μηδενιστεί)

addi $2, $2, 0 # δε χρειάζεται (οι καταχωρητές έχουν μηδενιστεί)

addi $2, $4, 0 # δε χρειάζεται (οι καταχωρητές έχουν μηδενιστεί)

addi $3, $0, 1

addi $5, $0, 3

L1: add $6, $3, $0

sw $6, 0($4)

addi $3, $3, 1

addi $4, $4, 1

addi $5, $5, -1

bne $5,$0,L1

**Προσοχή:** Η τοποθέτηση των εντολών πρέπει να γίνει πριν οι μονάδες γίνουν Compile και Load.

**3. Ανάπτυξη architecture του MIPS**

Δημιουργήστε αρχιτεκτονική, η οποία θα περιλαμβάνει ως components όλα τα προηγούμενα στοιχεία κατάλληλα συνδεδεμένα.

**Σημείωση:** για ευκολία, μπορείτε να υλοποιήσετε τη μνήμη εντολών και δεδομένων ως πίνακα θέσεων 32-bits. Στην περίπτωση αυτή:

* Ο αθροιστής (14) θα προσθέτει στα περιεχόμενα του PC τον αριθμό 1 και όχι τον αριθμό 4.
* Η μονάδα αριστερής ολίσθησης κατά 2 (13) μπορεί να παραληφθεί.

**4. Έλεγχος λειτουργιών**

**4.1** Ενεργοποιήστε το σήμα reset για (κάτι λιγότερο από) ένα κύκλο.

**4.2** Εκτελέστε το πρόγραμμα. Στην προσομοίωση πρέπει να φαίνονται οι τιμές των καταχωρητών και των θέσεων της μνήμης.

**Οδηγίες Υποβολής**

**Τρόπος παράδοσης:** Υποβολή zip αρχείου στο φάκελο «Εργασίες», Εξαμηνιαία Άσκηση 2022-2023

Στα παραδοτέα της εργασίας θα πρέπει να περιλαμβάνονται τα ακόλουθα αρχεία.

|  |  |  |
| --- | --- | --- |
| ΑΑ | Περιγραφή | Ονομασία αρχείου |
| 1 | Έγγραφο doc(x) που θα περιέχει   * εισαγωγική σελίδα (στοιχεία μαθήματος & φοιτητή). * περιγραφή του κυκλώματος * κώδικα του κυκλώματος και testbench, * screenshot από την εκτέλεση όπου θα φαίνονται οι τιμές των καταχωρητών και τα περιεχόμενα των θέσεων μνήμης που αλλάζουν τιμή. | ΑΜ\_ΕΠΙΘΕΤΟ\_01\_DESCRIPTION.DOC(X) |
| 2 | Ο κώδικας του κυκλώματος σε ξεχωριστό αρχείο | ΑΜ\_ΕΠΙΘΕΤΟ\_02\_MIPS.vhd |
| 3 | Ο κώδικας του testbench σε ξεχωριστό αρχείο | ΑΜ\_ΕΠΙΘΕΤΟ\_03\_testbench.vhd |

Τα τρία αρχεία θα βρίσκονται συμπιεσμένα σε ένα αρχείο με ονομασία AM\_EPITHETO.zip

Το επίθετο θα είναι γραμμένο με λατινικούς χαρακτήρες.

**Προσοχή:** δεν θα βαθμολογηθεί και θα μηδενίζεται:

* Κώδικας ο οποίος δεν περνάει τη φάση του compile.
* Εργασία που δεν θα συνοδεύεται από την απαραίτητη τεκμηρίωση
* Εργασία που δεν θα περιλαμβάνει testbench