



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών
Ψηφιακά συστήματα HW-1

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2021-2022

Project - Μέρος 1

ΣΧΕΔΙΑΣΗ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ ΜΙΑΣ ΜΟΝΑΔΑΣ
ΑΡΙΘΜΗΤΙΚΩΝ ΚΑΙ ΛΟΓΙΚΩΝ ΠΡΑΞΕΩΝ (ALU) ΚΑΙ ΕΝΟΣ
ΑΡΧΕΙΟΥ ΚΑΤΑΧΩΡΗΤΩΝ (REGISTER FILE)

Σκοπός αυτού του μέρους:

Είναι η σχεδίαση σε Γλώσσα Περιγραφής Υλικού (Hardware Description Language) Verilog μιας μονάδας αριθμητικών και λογικών πράξεων, ενός αρχείου καταχωρητών, καθώς και η προσομοίωση τους με το εργαλείο ISE της Xilinx.

Προαπαιτούμενα

Καλή κατανόηση της Verilog στη συμπεριφορική μορφή της (behavioral) και την δομική μορφή της (structural) καθώς και του περιβάλλοντος που προσφέρει το εργαλείο ISE της Xilinx.

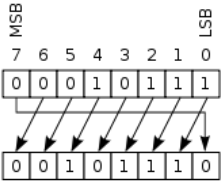
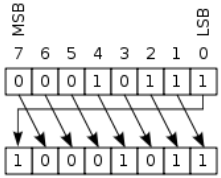
Διεξαγωγή

A) Σχεδίαση της μονάδας αριθμητικών και λογικών πράξεων

Η μονάδα έχει τις εξής εισόδους και εξόδους:

Σήμα	Είδος -Πλάτος	Λειτουργία
A	είσοδος (32 bits)	Πρώτος τελεσταίος
B	είσοδος (32 bits)	Δεύτερος τελεσταίος
Op	είσοδος (4 bits)	Κωδικός πράξης
Out	έξοδος (32 bits)	Αποτέλεσμα
Zero	έξοδος (1 bit)	Ενεργοποιημένη αν το αποτέλεσμα είναι μηδέν

Η συμπεριφορά της ALU είναι η εξής:

Κωδικός	Πράξη	Αποτέλεσμα
Op = 0000	Πρόσθεση	Out = A + B
Op = 0001	Αφαίρεση	Out = A - B
Op = 0010	Λογικό “ΚΑΙ”	Out = A & B
Op = 0011	Λογικό “Η”	Out = A B
Op = 0100	Αντιστροφή του A	Out = ! A
Op = 1000	Αριθμητική ολίσθηση δεξιά κατά 1 θέση (MSB \leftarrow (παλιό MSB))	Out= (int) A >> 1 Αποτέλεσμα = {A[31], A[31], ... A[1]}
Op = 1010	Λογική ολίσθηση δεξιά κατά 1 θέση (MSB \leftarrow ‘0’)	Out= (unsigned int) A >> 1 Αποτέλεσμα = {0, A[31], ... A[1]}
Op = 1001	Λογική ολίσθηση αριστερά κατά 1 θέση (LSB \leftarrow 0)	Out= A << 1 Αποτέλεσμα = {A[30], A[29],... A[0],0}
Op = 1100	Κυκλική ολίσθηση (rotate) αριστερά το A κατά 1 θέση	
Op = 1101	Κυκλική ολίσθηση (rotate) δεξιά το A κατά 1 θέση	

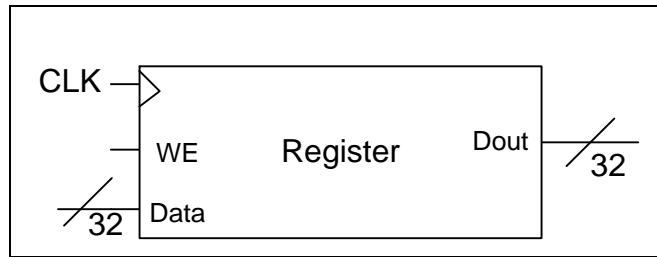
- 1) Γράψτε τον κώδικα που υλοποιεί την ALU σε Verilog. Δώστε προσοχή στην έξοδο Zero. Ποια είναι η συνθήκη που ορίζει αυτή την έξοδο.
- 2) Μεταγλωττίστε και προσομοιώστε την ALU στο περιβάλλον Xilinx.
- 3) Δώστε αρκετές διαφορετικές εισόδους στην προσομοίωση ώστε να ελέγξετε όλες τις (ενδιαφέρουσες) περιπτώσεις των σημάτων εξόδου.

B) Σχεδίαση του Αρχείου Καταχωρητών

Το αρχείο καταχωρητών είναι ένας συνδυασμός από καταχωρητές και συνδυαστική λογική.

B1. Παραγωγή Register

Αρχικά υλοποιήστε έναν καταχωρητή 32 bits σε Verilog. Ο καταχωρητής θα πρέπει να έχει τα εξής σήματα: **ρολόι (CLK - 1 bit)**, **Δεδομένα εισόδου (Data - 32 bits)**, **Δεδομένα εξόδου (Dout - 32 bits)** και ένα σήμα για **Write Enable (WE - 1 bit)**. Η διεπαφή για τον register φαίνεται στην Εικόνα 1.



Εικόνα 1: Καταχωρητής 32 bits

B2. Παραγωγή Register File

Για την παραγωγή ενός αρχείου καταχωρητών (Register File-RF) θα χρησιμοποιήσετε 32 registers, όπως αυτούς που υλοποιήσατε στο βήμα B1. Στην συνέχεια κάνοντας την συνδεσμολογία που παρουσιάζεται στην Εικόνα 2 θα υλοποιήσετε ένα αρχείο καταχωρητών με 32 καταχωρητές με τρεις θύρες(δύο ανάγνωσης και μία εγγραφής). Η διεπαφή του αρχείου καταχωρητών έχει τις εξής εισόδους και εξόδους:

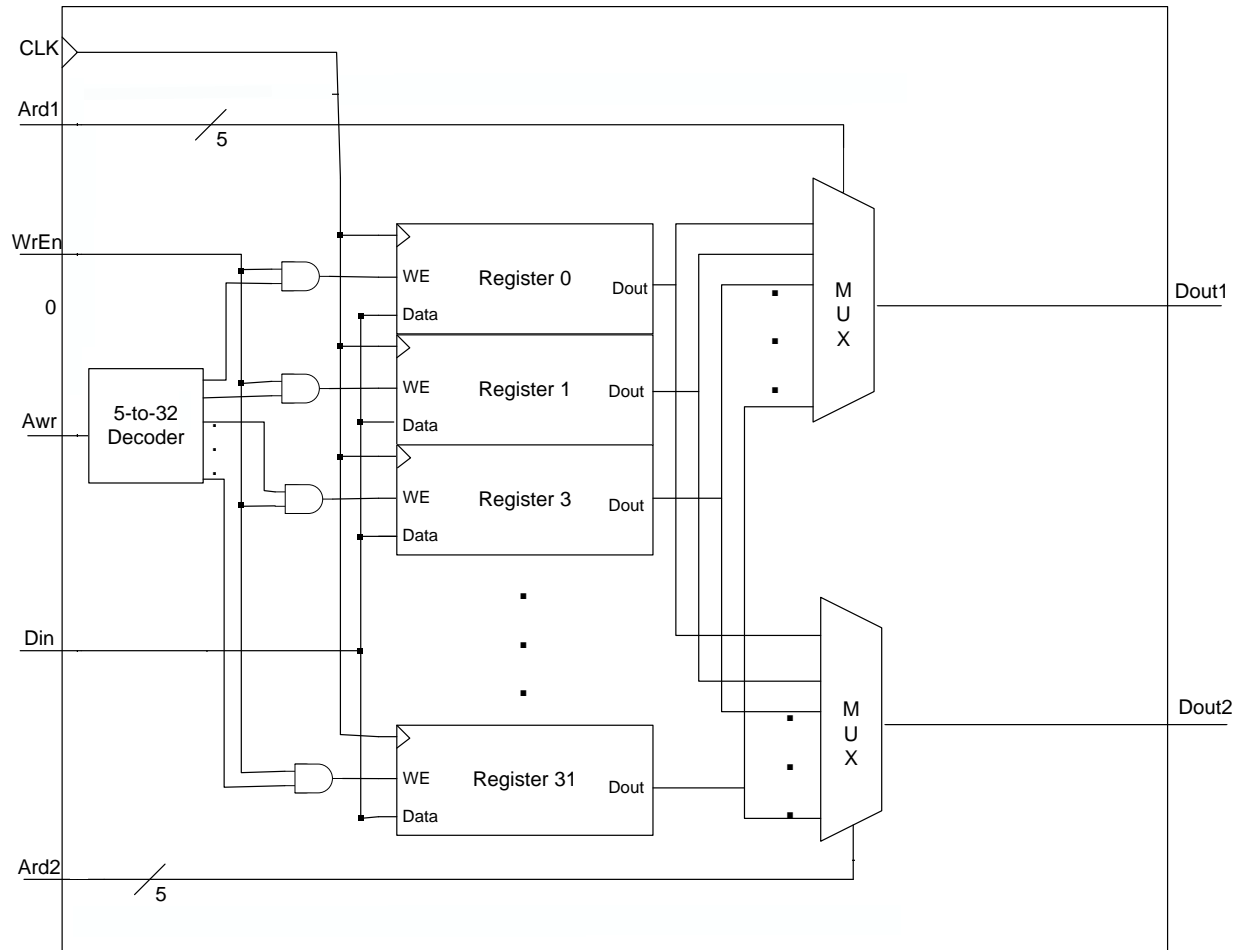
Σήμα	Είδος -Πλάτος	Λειτουργία
Ard1	Είσοδος (5 bits)	Διεύθυνση πρώτου καταχωρητή για ανάγνωση
Ard2	Είσοδος (5 bits)	Διεύθυνση δεύτερου καταχωρητή για ανάγνωση
Awr	Είσοδος (5 bits)	Διεύθυνση καταχωρητή για εγγραφή
Dout1	Έξοδος (32 bits)	Δεδομένα πρώτου καταχωρητή
Dout2	Έξοδος (32 bits)	Δεδομένα δεύτερου καταχωρητή
Din	Είσοδος (32 bits)	Δεδομένα για εγγραφή
WrEn	Είσοδος (1 bit)	Ενεργοποίηση Εγγραφής καταχωρητή
Clk	Είσοδος (1 bit)	Ρολόι

Παρατηρήσεις:

- 1) Στην διεπαφή δεν υπάρχει είσοδος ενεργοποίησης ανάγνωσης, το οποίο σημαίνει ότι το αρχείο καταχωρητών διαβάζει πάντα και από τις δύο θέσεις που υποδεικνύουν οι διευθύνσεις ανάγνωσης.
- 2) Όλα τα modules που απεικονίζονται στην Εικόνα 2 είναι ασύγχρονα (εκτός προφανώς τους registers).
- 3) Τι χρειάζεται να αλλάξετε στην συνδεσμολογία της Εικόνας 2 έτσι ώστε να μην αλλάζει ποτέ η τιμή του R0; (Ως γνωστό η τιμή του R0 παραμένει πάντα σταθερά 0).

Υλοποίηση:

- 1) Γράψτε τον κώδικα Verilog που υλοποιεί το αρχείο καταχωρητών χρησιμοποιώντας τον καταχωρητή που υλοποιήσατε στο **B1**.
- 2) Μεταγλωττίστε και προσομοιώστε την RF στο περιβάλλον Xilinx.
- 3) Δώστε αρκετές διαφορετικές εισόδους στην προσομοίωση ώστε να ελέγξετε όλες τις (ενδιαφέρουσες) περιπτώσεις των σημάτων εισόδου-εξόδου.



Εικόνα 2: Αρχείο καταχωρητών