



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

Εργαστηριακή Άσκηση 2

Αριθμός Ομάδας: 56

Ονοματεπώνυμο 1: Χαϊντούτη Μαρία

Ονοματεπώνυμο 2: Χειλαδάκης Νικόλαος

Σκοπός της άσκησης: Ο σκοπός της άσκησης αυτής είναι η μετατροπή του επεξεργαστή ενός κύκλου που σχεδιάστηκε στην φάση 1 σε επεξεργαστή πολλαπλών κύκλων.

Περιγραφή της Σχεδίασης: Για την υλοποίηση της φάσης αυτής έγιναν αλλαγές στο “Datapath”, η μονάδα ελέγχου “Control” και το συνολικό σύστημα “System”.

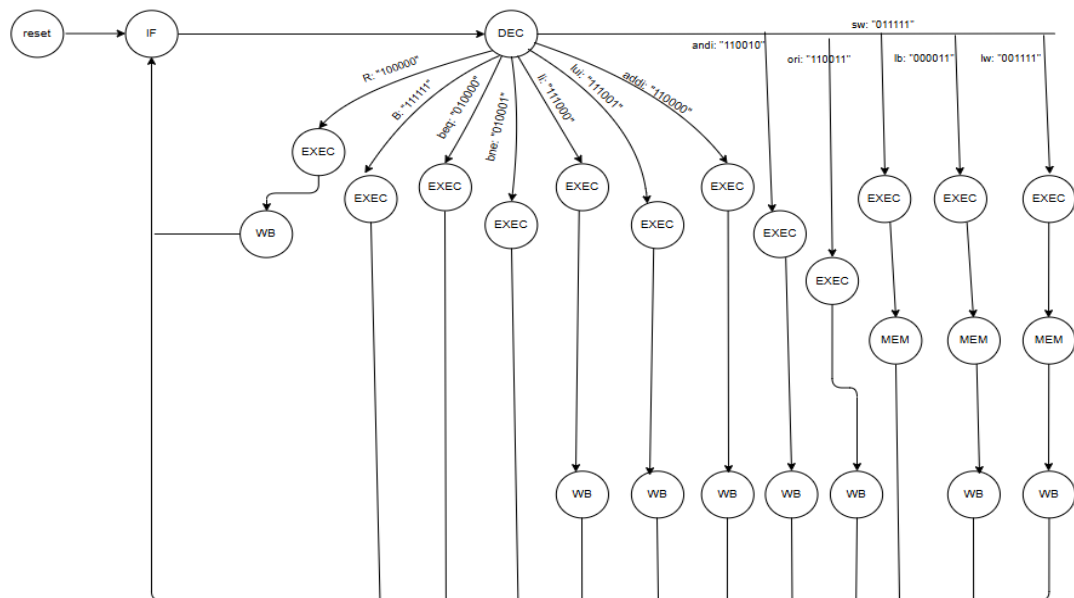
A) Στο Datapath του επεξεργαστή πολλών κύκλων προστέθηκαν ενδιάμεσοι καταχωρητές μεταξύ των σταδίων IF, DEC, EXEC και MEM. Αυτοί οι καταχωρητές είναι οι ακόλουθοι:

- If_Dec_reg
- Dec_Exec_reg_RF_A
- Dec_Exec_reg_RF_B
- Dec_Exec_reg_RF_Immed
- Exec_Mem_reg
- Mem_Wb_reg

Συνολικά προστέθηκαν έξι καταχωρητές, ένας καταχωρητής δηλαδή για κάθε σήμα εξόδου των παραπάνω σταδίων, προκειμένου να διατηρούνται τα δεδομένα σωστά και να είναι διαθέσιμα όταν χρειάζεται στους επόμενους κύκλους του ρολογιού.

B) Η μονάδα ελέγχου πλέον είναι μια Μηχανή Πεπερασμένων Καταστάσεων (FSM) που ελέγχει τη ροή εκτέλεσης της κάθε εντολής. Σε κάθε κατάσταση, τα σήματα ελέγχου ρυθμίζονται σύμφωνα με τον τύπο της εντολής και τις ανάγκες της επόμενης ενέργειας. Το σήμα “next_state” καθορίζει την επόμενη κατάσταση στον επόμενο κύκλο ρολογιού και το “cur_state” την κατάσταση όπου βρισκόμαστε.

Παρακάτω ακολουθεί το state diagram της FSM και πινακάκια με τις αλλαγές του κάθε control σήματος στο αντίστοιχο stage, προκειμένου να φανεί ακριβέστερα ο τρόπος λειτουργίας της μονάδας ελέγχου αυτής της άσκησης:



Πινακάκια:

- Fetch:**

Control Signals//Opcode	111111	010000	0100001	111000	0111111
ALU_func	1111	1111	1111	1111	1111
PC_sel	0	0	0	0	0
PC_LdEn	0	0	0	0	0
RF_WrEn	0	0	0	0	0
RF_WrData_sel	0	0	0	0	0
RF_B_sel	0	1	1	0	0
ALU_Bin_sel	0	0	0	0	0
MEM_WrEn	0	0	0	0	0
byteOp	0	0	0	0	0
Immed_sel	00	01	01	00	00

- Decode:**

Control Signals//Opcode	100000	111111	010000	010001	011111	000011	001111
ALU_func	1111	1111	0001	0001	1111	1111	1111
PC_sel	0	0	1(zero=1)	1(zero=0)	0	0	0
PC_LdEn	0	1	1	1	0	0	0
RF_WrEn	0	0	0	0	0	0	0
RF_WrData_sel	0	0	0	0	0	0	0
RF_B_sel	0	1	0	0	0	1	1
ALU_Bin_sel	0	0	0	0	0	0	0
MEM_WrEn	0	0	0	0	0	0	0
byteOp	0	0	0	0	0	0	0
Immed_sel	00	01	00	00	00	00	00

- **Decode συνέχεια:**

Control Signals//Opcode	111000	111001	110000	110010	110011
ALU_func	1111	1111	1111	1111	1111
PC_sel	0	0	0	0	0
PC_LdEn	0	0	0	0	0
RF_WrEn	0	0	0	0	0
RF_WrData_sel	0	0	0	0	0
RF_B_sel	1	1	1	1	1
ALU_Bin_sel	0	0	0	0	0
MEM_WrEn	0	0	0	0	0
byteOp	0	0	0	0	0
Immed_sel	00	11	00	10	10

- **Execute:**

Control Signals//Opcode	100000	111111	010000	010001	011111	000011	001111
ALU_func	Instr(3 downto 0)	1111	1111	1111	0000	0000	0000
PC_sel	0	1	0	0	0	0	0
PC_LdEn	1	0	0	0	1	0	0
RF_WrEn	0	0	0	0	0	0	0
RF_WrData_sel	0	0	0	0	0	0	0
RF_B_sel	0	0	0	0	1	0	0
ALU_Bin_sel	0	0	0	0	1	1	1
MEM_WrEn	0	0	0	0	0	0	0
byteOp	0	0	0	0	0	0	0
Immed_sel	00	00	00	00	00	00	00

- **Execute συνέχεια:**

Control Signals//Opcode	111000	111001	110000	110010	110011
ALU_func	0000	0000	0000	0010	0011
PC_sel	0	0	0	0	0
PC_LdEn	1	1	1	1	1
RF_WrEn	0	0	0	0	0
RF_WrData_sel	0	0	0	0	0
RF_B_sel	0	0	0	0	0
ALU_Bin_sel	1	1	1	1	1
MEM_WrEn	0	0	0	0	0
byteOp	0	0	0	0	0
Immed_sel	00	00	00	00	00

- **Memory:**

Control Signals//Opcode	011111	000011	001111
ALU_func	1111	1111	1111
PC_sel	0	0	0
PC_LdEn	0	1	1
RF_WrEn	0	0	0
RF_WrData_sel	0	0	0
RF_B_sel	0	0	0
ALU_Bin_sel	0	0	0
MEM_WrEn	1	0	0
byteOp	0	1	0
Immed_sel	00	00	00

- **WriteBack:**

Control Signals//Opcode	100000	111000	111001	110000	110010	110011	000011	001111
ALU_func	1111	1111	1111	1111	1111	1111	1111	1111
PC_sel	0	0	0	0	0	0	0	0
PC_LdEn	0	0	0	0	0	0	0	0
RF_WrEn	1	1	1	1	1	1	1	1
RF_WrData_sel	0	0	0	0	0	0	1	1
RF_B_sel	0	0	0	0	0	0	0	0
ALU_Bin_sel	0	0	0	0	0	0	0	0
MEM_WrEn	0	0	0	0	0	0	0	0
byteOp	0	0	0	0	0	0	0	0
Immed_sel	00	00	00	00	00	00	00	00

Στο παρακάτω σχηματικό διάγραμμα φαίνεται το συνολικό σύστημα με την προσθήκη των νέων καταχωρητών. Χάριν απλούστευσης του διαγράμματος στην μονάδα ελέγχου φαίνεται μόνο το “instr” ως είσοδος σε αυτή, καθώς οι υπόλοιπες εισόδους (clk, rst, alu_zero_out) και οι έξοδοι παραλείπονται.

