

Ερωτήσεις αυτοαξιολόγησης 1^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 1ου μαθήματος (“Εισαγωγή στην Αρχιτεκτονική Η/Υ”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Ποια η σχέση της αρχιτεκτονικής των υπολογιστών και της τεχνολογίας;
2. Ποια η σημασία των διεπαφών (interfaces) και της ιεραρχικής σχεδίασης;
3. Ποιος ο ρόλος της διεπαφής ISA (Instruction Set Architecture); Μεταξύ ποιών επιπέδων σε έναν υπολογιστή βρίσκεται;
4. Γιατί το λογισμικό είναι μέρος της αρχιτεκτονικής;
5. Ποιοι οι στόχοι της αρχιτεκτονικής Η/Υ ανάλογα με την κάθε κατηγορία υπολογιστικών συστημάτων;
6. Ποιες οι βασικές μονάδες ενός υπολογιστή και ποια η λειτουργικότητα τους;
7. Τι σημαίνει ότι η αρχιτεκτονική ενός υπολογιστή είναι 32- ή 64-bits;
8. Γιατί είναι ζωτικής σημασίας η συνέχιση της σμίκρυνσης των διαστάσεων των τρανζίστορ; Ποια τα οφέλη από τη σμίκρυνση αυτή;
9. Γιατί τα τρανζίστορ και η θεωρητική δυαδική λογική μπορούν να συνδυαστούν με επιτυχία;
10. Βεβαιωθείτε ότι γνωρίζετε τους βασικούς κανόνες της άλγεβρας Boole.

Θέματα 1^{ου} εργαστηρίου

1. Σχεδιάστε έναν αντιστροφέα (πύλη NOT) χρησιμοποιώντας ένα τρανζίστορ PMOS και ένα τρανζίστορ NMOS. Θυμηθείτε ότι το τρανζίστορ PMOS άγει με 0 στην πύλη του και “περνάει” ισχυρό 1, ενώ το NMOS άγει με 1 στην πύλη του και “περνάει” ισχυρό 0.
2. Επεκτείνοντας την προηγούμενη ερώτηση, σχεδιάστε μία πύλη NAND με τον ίδιο ακριβώς τρόπο. Θυμηθείτε ότι ο συνδυασμός των PMOS (παράλληλα ή σε σειρά) είναι ο συμπληρωματικός του συνδυασμού των NMOS.
3. Σχεδιάστε με τον ίδιο τρόπο μία πύλη NAND, Παρατηρήστε ότι με τη μέθοδο στατικής σχεδίασης με συμπληρωματικές διατάξεις PMOS και NMOS μπορούμε να κατασκευάσουμε μόνο αναστρέφουσες λογικές συναρτήσεις. Πώς μπορούμε να φτιάξουμε πύλες AND ή OR;
4. Πώς μπορούμε να φτιάξουμε μια πύλη XOR; Αποδείξτε ότι $A \text{ XOR } B = AB' + A'B$. Πόσα τρανζίστορ θα χρειάζονταν για την κατασκευή μιας πύλης XOR με τη μέθοδο που χρησιμοποιήσατε προηγουμένως; Συνήθως οι πύλες XOR κατασκευάζονται με διαφορετική μέθοδο για μείωση του αριθμού των τρανζίστορ.
5. Ποια συνάρτηση υλοποιεί η πύλη XNOR;
6. Θέλουμε να υλοποιήσουμε τη συνάρτηση της πλειοψηφίας μεταξύ τριών εισόδων: αν δύο ή περισσότερες από τις εισόδους είναι 1, τότε η έξοδος θα είναι 1. Σε κάθε άλλη περίπτωση η έξοδος θα είναι 0. Σχεδιάστε τον πίνακα αλήθειας και προσπαθήστε εμπειρικά να βρείτε τον συνδυασμό πυλών που υλοποιεί τη συνάρτηση αυτή.
7. Στην προσημασμένη δυαδική πρόσθεση, έχουμε υπερχειλίση όταν οι δύο προστιθέμενοι αριθμοί έχουν ίδιο πρόσημο αλλά το αποτέλεσμα έχει διαφορετικό. Ποια συνθήκη (σε γλώσσα προγραμματισμού) περιγράφει την κατάσταση υπερχειλίσης; Σχεδιάστε το κύκλωμα που ως έξοδο έχει την ένδειξη της υπερχειλίσης.

Ερωτήσεις αυτοαξιολόγησης 2^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 2ου μαθήματος (“Ψηφιακή Λογική και Σχεδίαση – Συνδυαστικά Κυκλώματα”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Βεβαιωθείτε ότι γνωρίζετε τους βασικούς κανόνες της άλγεβρας Boole.
2. Ποιο είναι το γενικό σχήμα υλοποίησης λογικών συναρτήσεων με την στατική τεχνολογία CMOS;
3. Πού χρησιμοποιούνται τα τρανζίστορ NMOS και PMOS;
4. Πώς υλοποιούνται οι πύλες NOT, NAND και NOR;
5. Ποιο είναι το βασικό χαρακτηριστικό ενός μπλοκ συνδυαστικής λογικής σε σχέση με τις τιμές των εξόδων του;
6. Τι ονομάζεται καθυστέρηση διάδοσης;
7. Πώς λειτουργεί ο αποκωδικοποιητής από N-σε-2^N; Πώς μπορεί να κατασκευαστεί με λογικές πύλες;
8. Ποια η λειτουργία του πολυπλέκτη 2^N γραμμών σε 1 γραμμή; Πώς υλοποιείται με λογικές πύλες;
9. Ποια η λειτουργία ενός ημιαθροιστή; Γιατί δεν αρκεί για την πρόσθεση δυαδικών αριθμών;
10. Πώς διαδίδεται το κρατούμενο όταν χρησιμοποιούνται πολλαπλοί πλήρεις αθροιστές;

Θέματα 2^{ου} εργαστηρίου

1. Χρήση του προγράμματος σχεδίασης ψηφιακών κυκλωμάτων Electric. Σχηματική σχεδίαση με πύλες. Δημιουργία cells. Ιεραρχική σχεδίαση με cells. Εξομοίωση κυκλώματος και έλεγχος λειτουργίας.
(<http://ftp.gnu.org/pub/gnu/electric/>)
2. Σχεδιάστε το κύκλωμα του ημιαθροιστή (half adder) στο πρόγραμμα Electric, και ελέγξτε τη λειτουργία του (Tool-> Simulation (Built-In)->ALS).
3. Σχεδιάστε στο πρόγραμμα Electric το κύκλωμα του πλήρους αθροιστή (full adder), χρησιμοποιώντας 2 ημιαθροιστές και μια πύλη OR. Ελέγξτε τη λειτουργία του κυκλώματος για τους 8 συνδυασμούς εισόδων.

Ερωτήσεις αυτοαξιολόγησης 3^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 3ου μαθήματος (“Ψηφιακή Λογική και Σχεδίαση – Στοιχεία μνήμης και μέθοδοι χρονισμού”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Από τι εξαρτάται η τιμή των εξόδων ενός ακολουθιακού κυκλώματος;
2. Σε τι χρησιμεύει το σήμα ρολογιού και πότε αποθηκεύεται η κατάσταση στα στοιχεία μνήμης;
3. Γιατί επιτρέπεται η ανάδραση όταν χρησιμοποιούμε στοιχεία μνήμης;
4. Σε τι διαφέρουν τα στοιχεία μνήμης που χρησιμοποιούνται για καταχωρητές από εκείνα που χρησιμοποιούνται στην κύρια μνήμη;
5. Περιγράψτε τη λειτουργία του “μανδαλωτή” (latch).
6. Περιγράψτε τη λειτουργία ενός D flip-flop. Σε τι διαφέρει από το latch;
7. Πώς υλοποιείται ένας καταχωρητής από στοιχεία flip-flop; Πόσα bits αποθηκεύονται σε έναν καταχωρητή;
8. Τι είναι το register file (ομάδα καταχωρητών) σε μια ΚΜΕ; Πόσες λειτουργίες μπορούν να εκτελούνται ταυτόχρονα;
9. Πώς διενεργείται η ανάγνωση και η εγγραφή σε ένα register file;
10. Με ποιο κύκλωμα μπορώ να διαβάσω ταυτόχρονα 2 καταχωρητές από το register file;

Θέματα 3^{ου} εργαστηρίου

1. Σχεδιάστε το κύκλωμα ενός πολυπλέκτη 4-σε-1 στο πρόγραμμα Electric, χρησιμοποιώντας 4 πύλες AND και 1 πύλη OR. Ελέγξτε τη λειτουργία του κυκλώματος για τους 4 συνδυασμούς των 2 σημάτων ελέγχου.
2. Σχεδιάστε μια αριθμητική-λογική μονάδα (ALU) του ενός bit, η οποία εκτελεί τις πράξεις AND, OR, XOR και ΠΡΟΣΘΕΣΗ μεταξύ 2 bits εισόδου. Χρησιμοποιήστε τον πολυπλέκτη του προηγούμενου ερωτήματος και τον πλήρη αθροιστή από το προηγούμενο εργαστήριο. Το κύκλωμά σας θα πρέπει επίσης να έχει ως είσοδο το carry-in του αθροιστή και ως έξοδο το carry-out.

Ερωτήσεις αυτοαξιολόγησης 4^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 4ου μαθήματος (“Αρχιτεκτονικές Συνόλου Εντολών”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Τι είναι ακριβώς ο (μικρο)επεξεργαστής; Πόσες κατηγορίες επεξεργαστών υπάρχουν;
2. Τι περιέχεται σήμερα μέσα στο τυπικό ολοκληρωμένο κύκλωμα του επεξεργαστή;
3. Ποιο υπολογιστικό σύστημα περιγράφεται με το μοντέλο von Neumann;
4. Περιγράψτε τα βασικά στάδια του κύκλου μηχανής (εκτέλεσης εντολών).
5. Περιγράψτε τις λειτουργίες που μπορούν να εκτελεστούν με τις εντολές μηχανής.
6. Ποιος ο ρόλος του μετρητή προγράμματος (program counter);
7. Πώς γίνεται η εκκίνηση του υπολογιστικού συστήματος; Πώς παίρνει αρχική τιμή ο μετρητής προγράμματος; Πότε τερματίζεται η εκτέλεση;
8. Τι περιγράφει η αρχιτεκτονική συνόλου εντολών; Ως διεπαφή (interface) η αρχιτεκτονική συνόλου εντολών μεταξύ ποιων μερών βρίσκεται;
9. Ποια η γενική μορφή κωδικοποίησης των εντολών μηχανής; Ποια τα μέρη μιας τυπικής εντολής;
10. Το μήκος των εντολών σε έναν επεξεργαστή είναι σταθερό ή μεταβλητό; Ποια τα πλεονεκτήματα και μειονεκτήματα κάθε προσέγγισης;

Θέματα 4^{ου} εργαστηρίου

1. Θα πρέπει να έχετε στη σχεδιαστική σας βιβλιοθήκη τα εξής τμήματα: α) πλήρη αθροιστή β) πολυπλέκτη 4-σε-1.
2. Σχεδιάστε μια αριθμητική-λογική μονάδα (ALU) του ενός bit, η οποία εκτελεί τις πράξεις AND, OR, XOR και ΠΡΟΣΘΕΣΗ μεταξύ 2 bits εισόδου. Χρησιμοποιήστε τον πολυπλέκτη του προηγούμενου ερωτήματος και τον πλήρη αθροιστή από το προηγούμενο εργαστήριο. Το κύκλωμά σας θα πρέπει επίσης να έχει ως είσοδο το carry-in του αθροιστή και ως έξοδο το carry-out.
3. Προσθέστε μια πύλη XOR πριν τον πλήρη αθροιστή για να μπορείτε να συμπληρώνετε κατά βούληση την είσοδο B, πριν αυτή φτάσει στον αθροιστή. Μπορείτε να σκεφτείτε έναν τρόπο για να υλοποιήσετε την πράξη της διαίρεσης με το κύκλωμα αυτό;

Ερωτήσεις αυτοαξιολόγησης 5^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 5ου μαθήματος (“Αρχιτεκτονικές Συνόλου Εντολών II”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Τι περιγράφει μια αρχιτεκτονική συνόλου εντολών;
2. Πώς κωδικοποιούνται οι εντολές μηχανής;
3. Ποια τα πλεονεκτήματα και μειονεκτήματα των εντολών σταθερού και μεταβλητού μήκους;
4. Τι είναι η γλώσσα assembly και πώς σχετίζεται με τις εντολές μηχανής;
5. Ποια η πιθανή προέλευση των δεδομένων σε μια εντολή;
6. Περιγράψτε 2 παραλλαγές προέλευσης δεδομένων, στις αρχιτεκτονικές των σύγχρονων υπολογιστών.
7. Ποια η διαφορά των αρχιτεκτονικών CISC και RISC;
8. Όταν η εντολή μηχανής περιλαμβάνει λειτουργία από/προς τη μνήμη, πώς περιγράφεται η διεύθυνση της μνήμης αυτής; Πόσες μεθόδους σχηματισμού διεύθυνσης μνήμης μπορεί να χρησιμοποιήσει ένας μεταγλωττιστής και για ποιον σκοπό την καθεμία;
9. Περιγράψτε τη μορφή των 3 βασικών κατηγοριών εντολών μηχανής: αριθμητικών/λογικών πράξεων, μεταφοράς δεδομένων από/προς τη μνήμη και διακλάδωσης.
10. Γιατί οι εντολές που υποστηρίζουν την κλήση συναρτήσεων κατατάσσονται στις εντολές διακλάδωσης; Σε τι διαφέρουν από μια απλή μετάβαση σε άλλο σημείο του κώδικα;

Θέματα 5^{ου} εργαστηρίου

1. Θα πρέπει να έχετε στη σχεδιαστική σας βιβλιοθήκη το εξής τμήμα: alu1, το οποίο εκτελεί τις πράξεις AND, OR, XOR και ΠΡΟΣΘΕΣΗ (ή ΑΦΑΙΡΕΣΗ) μεταξύ δύο bits εισόδου.
2. Χρησιμοποιώντας 8 τμήματα alu1 σχεδιάστε ένα νέο τμήμα alu8, το οποίο θα εκτελεί τις προηγούμενες πράξεις σε 2 bytes εισόδου και θα παράγει αποτέλεσμα ομοίως του ενός byte.
3. Προσθέστε τα εξής σήματα κατάστασης:
 - z (zero): 1 όταν το αποτέλεσμα είναι μηδενικό, σε κάθε άλλη κατάσταση 0.
 - v (overflow): 1 όταν το αποτέλεσμα της προσημασμένης πρόσθεσης παράγει υπερχειλίση, αλλιώς 0.

Ερωτήσεις αυτοαξιολόγησης 6^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του βου μαθήματος (“Κεντρική Μονάδα Επεξεργασίας - Σχεδιασμός και λειτουργία μιας απλής ΚΜΕ”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Ποια η διαφορά μεταξύ των όρων “μικροεπεξεργαστής” και “κεντρική μονάδα επεξεργασίας”;
2. Τι είδη εντολών περιγράφει μια αρχιτεκτονική “load-store”;
3. Ποιος ο ρόλος της αριθμητικής-λογικής μονάδας (ΑΛΜ);
4. Ποιες λειτουργίες μπορούν να εκτελεστούν σε μια συστοιχία καταχωρητών (register file);
5. Ποια τμήματα απαιτούνται για τη συγκρότηση ενός απλού “μονοπατιού δεδομένων” (datapath);
6. Ποιος ο ρόλος της μονάδας ελέγχου;
7. Πώς σχηματίζεται η διεύθυνση της επόμενης εντολής;
8. Ποια τα γενικά στάδια εκτέλεσης μιας εντολής;
9. Ποια τα βήματα για την εκτέλεση αριθμητικών εντολών, εντολών διακλάδωσης και εντολών load-store;
10. Τι περιγράφουν τα μεγέθη clocks per instruction (CPI) και clock cycle (CC);

Εργαστήριο #6

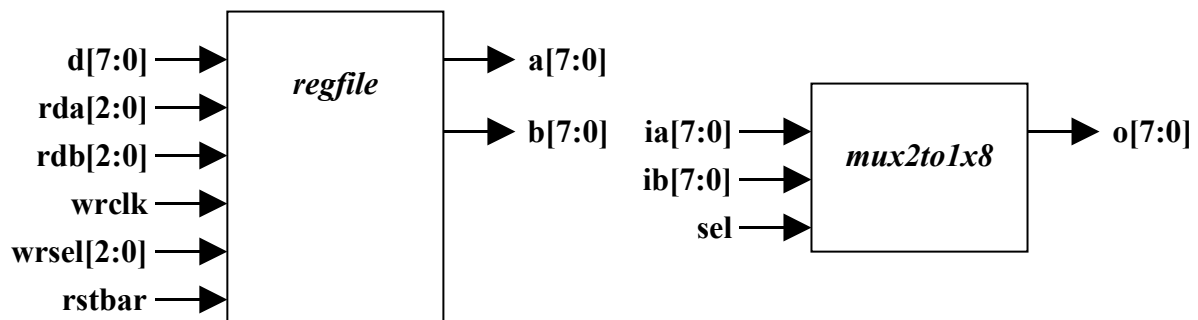
1. Τι θα πρέπει να έχετε ολοκληρώσει πριν συνεχίσετε...

Από τα προηγούμενα εργαστήρια, θα πρέπει να έχετε σχεδιάσει και ελέγξει:

- Ένα νέο τμήμα **alu8**, το οποίο θα εκτελεί τις πράξεις AND, OR, XOR και ΠΡΟΣΘΕΣΗ (ή ΑΦΑΙΡΕΣΗ) σε 2 bytes εισόδου και θα παράγει αποτέλεσμα ομοίως του ενός byte.
- Το τμήμα **alu8** θα παράγει τα εξής σήματα κατάστασης: α) z (zero): 1 όταν το αποτέλεσμα είναι μηδενικό, σε κάθε άλλη κατάσταση 0 και β) v (overflow): 1 όταν το αποτέλεσμα της **προσημασμένης** πρόσθεσης παράγει υπερχείλιση, αλλιώς 0.

2. Μονοπάτι δεδομένων (datapath) ενός κύκλου.

Στο παρόν εργαστήριο θα συνδυάσετε την ΑΛΜ των 8 bits που έχετε ήδη φτιάξει και ελέγξει, μαζί με δύο νέα τμήματα, τα οποία θα βρείτε έτοιμα στη βιβλιοθήκη **lablib.jelib** στο site του εργαστηρίου. Τα δύο αυτά τμήματα είναι:



α) **regfile**, ομάδα 8 καταχωρητών (R0..R7) για ανάγνωση (έξοδοι A και B) και εγγραφή. Ο R0 έχει ειδική λειτουργία: ό,τι γράφεται σε αυτόν δεν αποθηκεύεται, ενώ όταν διαβαστεί το περιεχόμενό του είναι πάντα 0. Τα σήματα εισόδου και εξόδου είναι τα εξής

Όνομασία	Κατεύθυνση	Λειτουργία
d0 ... d7	είσοδοι	εισαγωγή δεδομένων προς εγγραφή
wrsel0 ... wrsel2	είσοδοι	επιλογή καταχωρητή για εγγραφή
wrclk	είσοδος	τα δεδομένα εγγράφονται στην ανερχόμενη ακμή του σήματος αυτού <i>*τα δεδομένα πρέπει να είναι σταθερά τουλάχιστον 5ns πριν και μετά την ανερχόμενη ακμή του wrclk, ενώ η διάρκεια του παλμού (υψηλού ή χαμηλού) του wrclk πρέπει να είναι τουλάχιστον 5ns</i>
rstbar	είσοδος	όσο το σήμα αυτό είναι 0 (Low), το περιεχόμενο όλων των καταχωρητών παραμένει 0
rda0 ... rda2	είσοδος	επιλογή καταχωρητή ανάγνωσης για την έξοδο A
rdb0 ... rdb2	είσοδος	επιλογή καταχωρητή ανάγνωσης για την έξοδο B
a0 ... a7	έξοδοι	έξοδος ανάγνωσης A
b0 ... b7	έξοδοι	έξοδος ανάγνωσης B

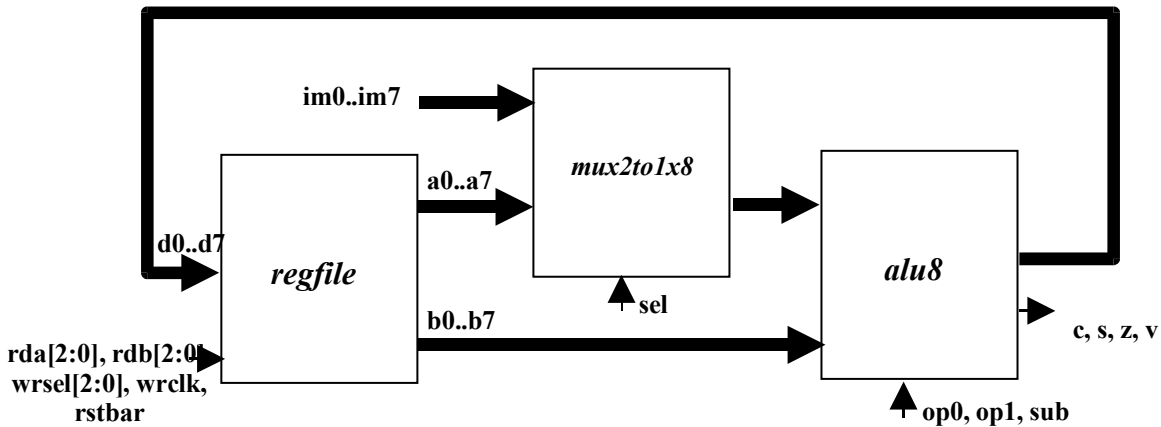
β) **mux2to1x8**, 8πλός πολυπλέκτης 2 σε 1. Από δύο ομάδες εισόδων των 8 bits η κάθε μία, επιλέγει ποια ομάδα θα περάσει στην έξοδο (επίσης των 8 bits) με βάση ένα σήμα επιλογής. Οι εισοδοί και έξοδοί έχουν ως εξής:

Όνομασία	Κατεύθυνση	Λειτουργία
ia0 ... ia7	είσοδοι	πρώτη ομάδα εισόδου (8 bits)
ib0 ... ib7	είσοδοι	δεύτερη ομάδα εισόδου (8 bits)
o0 ... o7	έξοδοι	έξοδος (8 bits)
sel	είσοδος	επιλογή ομάδας 8 bits που θα εμφανιστεί στην έξοδο. Με 0 επιλέγεται η ia, ενώ με 1 η ib.

3. Σχεδιασμός μονοπατιού δεδομένων.

Το μονοπάτι δεδομένων που θα σχεδιάσετε θα εκτελεί σε έναν κύκλο ρολογιού μια πράξη μεταξύ δύο πηγών δεδομένων και θα αποθηκεύει το αποτέλεσμα στο regfile. Η αποθήκευση θα ξεκινά στην ανερχόμενη ακμή του ρολογιού και θα ολοκληρώνεται μέσα στον επόμενο κύκλο. Το σχήμα του μονοπατιού δεδομένων θα επιτρέπει την εκτέλεση πράξεων:

- α) μεταξύ δεδομένων από 2 καταχωρητές (έξοδοι A και B του regfile)
- β) μεταξύ δεδομένων από 1 καταχωρητή (έξοδος B) και μέσω της άμεσης εισόδου im0..im7



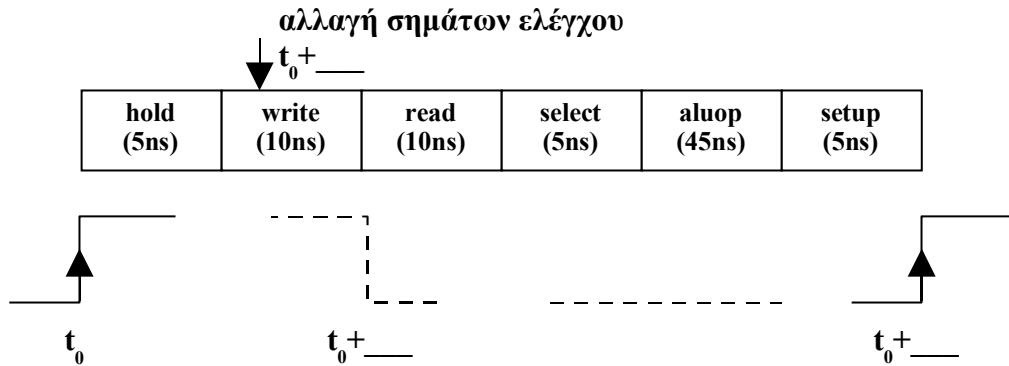
4. Περίοδος ρολογιού (CC).

Για να υπολογίσετε την απαιτούμενη περίοδο ρολογιού, λάβετε υπόψη σας τις εξής λειτουργίες που εκτελούνται σε έναν κύκλο:

- α) **hold**, εγγραφή δεδομένων από προηγούμενη εντολή στο regfile, κανένα σήμα ελέγχου δεν πρέπει να αλλάξει στο χρονικό αυτό διάστημα (5ns).
- β) **write**, ολοκλήρωση εγγραφής δεδομένων προηγούμενης εντολής (10ns).
- γ) **read**, επιλογή καταχωρητών δεδομένων για επόμενη πράξη (10ns).
- δ) **select**, καθυστέρηση επιλογής μεταξύ άμεσης εισόδου και regfile (5ns).
- ε) **aluop**, λειτουργία ΑΛΜ (45ns).
- στ) **setup**, χρόνος που τα δεδομένα παραμένουν σταθερά πριν εγγραφή (5ns).

Τα σήματα ελέγχου πρέπει να αλλάζουν μέσα στο διάστημα write, ενώ το ρολόι πρέπει να παραμείνει σε υψηλή στάθμη τουλάχιστον για το διάστημα hold.

A. Οι χρονισμοί του κύκλου ρολογιού. Συμπληρώστε τους χρόνους οδήγησης των σημάτων ελέγχου και του clock:



5. Δοκιμαστική λειτουργία.

α) Πώς θα θέσετε μια συγκεκριμένη τιμή σε έναν καταχωρητή μέσα σε έναν κύκλο ρολογιού;

β) Δοκιμάστε να εκτελέσετε σε 3 κύκλους τα εξής:

$R1 \leftarrow 10$

$R2 \leftarrow 20$

$R3 \leftarrow R1 + R2$

Ερωτήσεις αυτοαξιολόγησης 7^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 7ου μαθήματος (“Απόδοση ΚΜΕ - Μέτρηση και τεχνικές βελτίωσης απόδοσης”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Ποια η διαφορά μεταξύ των όρων “απόδοση ΚΜΕ” και “απόδοση υπολογιστικού συστήματος”;
2. Ποιο είναι το μέγεθος μέτρησης της απόδοσης; Ποια η σχέση του με την απόδοση;
3. Πώς συγκρίνουμε την απόδοση δύο υπολογιστών;
4. Αναλύστε τα μέρη του χρόνου εκτέλεσης ενός προγράμματος.
5. Ποια είναι τα τρία μεγέθη, στα οποία αναλύεται ο χρόνος εκτέλεσης; Τι αντιπροσωπεύει το καθένα και ποια η σχέση του με το λογισμικό;
6. Ποια πρέπει να είναι τα χαρακτηριστικά των μετροπρογραμμάτων;
7. Περιγράψτε τον νόμο του Amdahl και τις επιπτώσεις του στις προσπάθειες αρχιτεκτονικών βελτιώσεων.
8. Σε μια ΚΜΕ πολλαπλών κύκλων ανά εντολή, με τι ισούται ο κύκλος ρολογιού;
9. Τι είναι το μικροπρόγραμμα;
10. Ποια τα πλεονεκτήματα και μειονεκτήματα μιας ΚΜΕ πολλαπλών κύκλων ανά εντολή;

Ερωτήσεις αυτοαξιολόγησης 8^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 8ου μαθήματος (“Παραλληλισμός σε επίπεδο εντολών”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

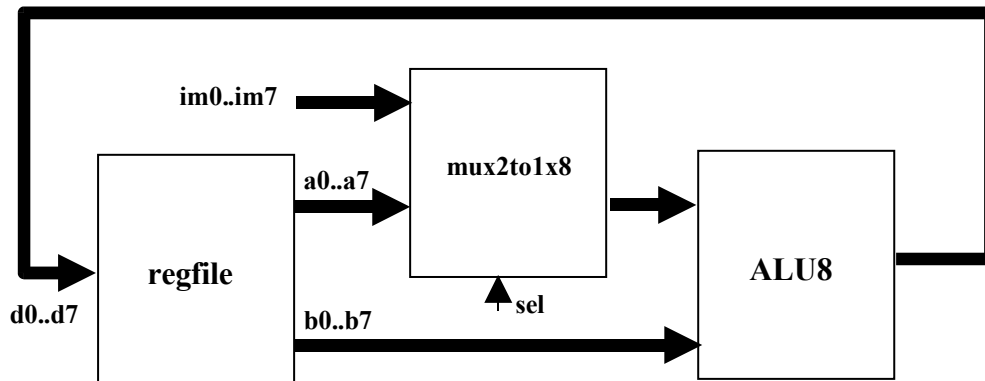
1. Για ποιον λόγο μια ΚΜΕ πολλαπλών κύκλων ανά εντολή εκμεταλλεύεται καλύτερα τον χρόνο εκτέλεσης από μια απλή ΚΜΕ ενός κύκλου ανά εντολή (single-cycle);
2. Ποια παρατήρηση στη χρήση του υλικού στην ΚΜΕ πολλαπλών κύκλων ανοίγει τον δρόμο για περαιτέρω βελτιστοποίηση της απόδοσης;
3. Ποια είναι η βασική λειτουργία ενός pipeline;
4. Γιατί στο pipeline είναι διαφορετικός ο χρόνος ολοκλήρωσης μιας εντολής από τον ρυθμό ολοκλήρωσης εντολών; Ποιο μέγεθος είναι σημαντικότερο;
5. Ποιες οι δυσκολίες στην εφαρμογή του ιδανικού pipeline; Περιγράψτε τι αντιπροσωπεύουν τα structural, data και control hazards.
6. Γιατί οι εντολές διακλάδωσης προβάλλουν εμπόδια στην εφαρμογή του σχήματος του pipeline; Τι συμβαίνει με τις εντολές που ακολουθούν την εντολή διακλάδωσης, όταν μια διακλάδωση εκτελείται;
7. Τι συμβαίνει στο pipeline κατά την εμφάνιση μιας διακοπής η ενός σφάλματος;
8. Πώς μπορεί να επιτευχθεί $CPI < 1$ σε έναν επεξεργαστή; Τι έχει ως συνέπεια αυτό στην πολυπλοκότητα του υλικού;
9. Τι είναι η εικαζόμενη εκτέλεση (speculative execution);
10. Ποια η διαφορά μεταξύ superscalar και VLIW επεξεργαστών;

Εργαστήριο #8

1. Μονοπάτι δεδομένων (datapath) ενός κύκλου.

Το μονοπάτι δεδομένων που έχετε σχεδιάσει εκτελεί σε έναν κύκλο ρολογιού μια πράξη μεταξύ δύο πηγών δεδομένων και αποθηκεύει το αποτέλεσμα στο regfile. Η αποθήκευση ξεκινά στην ανερχόμενη ακμή του ρολογιού και ολοκληρώνεται μέσα στον επόμενο κύκλο. Το σχήμα του μονοπατιού δεδομένων επιτρέπει την εκτέλεση πράξεων:

- α) μεταξύ δεδομένων από 2 καταχωρητές (έξοδοι A και B του regfile)
- β) μεταξύ δεδομένων από 1 καταχωρητή (έξοδος B) και μέσω της άμεσης εισόδου im0..im7



2. Υλοποίηση λειτουργιών (πράξεων) στο μονοπάτι δεδομένων.

Στους παρακάτω πίνακες συμπληρώστε τις τιμές των σημάτων ελέγχου για την υλοποίηση των αντίστοιχων πράξεων. Για κάθε μία πράξη ελέγξτε την ορθότητα των επιλογών σας μέσω εξομοίωσης. Σήματα που δεν επηρεάζουν την κάθε πράξη πρέπει να σημειωθούν ως 'x' (don't care).

(Τα σήματα *rstbar* και *wrclk* δεν αναφέρονται στους πίνακες)

α) $R_i \leftarrow$ άμεση τιμή (8 bits)

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

β) $R_i \leftarrow R_j + R_k$

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

γ) $R_i \leftarrow R_j +$ άμεση τιμή (8 bits)

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

δ) $R_i \leftarrow 2 * R_j$

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

ε) $R_i \leftarrow R_j - R_k$

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

στ) $R_i \leftarrow R_j - 1$

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

ζ) $R_i \leftarrow R_j$ λογ-πραξ R_k (λογ-πραξ = AND/OR/XOR)

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

η) $R_i \leftarrow \text{NOT } R_j$

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

θ) $R_i \leftarrow 0$

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

ι) $R_i - R_j$ (για σύγκριση, το αποτέλεσμα δεν αποθηκεύεται)

rda[2:0]	rdb[2:0]	im[7:0]	sel	op0	op1	sub	wrsel[2:0]

3.Μια μικρή ακολουθία πράξεων.

Υλοποιήστε μια ακολουθία πράξεων για τον υπολογισμό του $1+2+3+4+\dots+n$, **περνώντας από τα im[7:0] μόνο τον αριθμό 1**. Επειδή δεν έχετε τη δυνατότητα επανάληψης (δεν έχετε μονάδα ελέγχου στον σχεδιασμό σας), υλοποιήστε τις πράξεις ακολουθιακά μέχρι το $n=5$.

Ερωτήσεις αυτοαξιολόγησης 9^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 9ου μαθήματος (“Τεχνολογίες Κύριας Μνήμης”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Συγκρίνετε ιστορικά την ανάπτυξη των κυκλωμάτων επεξεργασίας και των κυκλωμάτων αποθήκευσης.
2. Ποιο το μοντέλο της Μνήμης Τυχαίας Προσπέλασης;
3. Ποια είναι η χωρητικότητα μνήμης με εύρος λέξης M bits και εύρος διεύθυνσης N bits;
4. Ποια η διαφορά μεταξύ byte και word addressing;
5. Τι περιγράφουν τα μεγέθη access time και cycle time;
6. Ποιες οι διαφορές μεταξύ στατικής και δυναμικής μνήμης RAM σε σχέση με την ταχύτητα, τη χωρητικότητα και το κόστος;
7. Ποια θα ήταν η “ιδανική μνήμη” σε ένα υπολογιστικό σύστημα και ποια η πραγματική εικόνα;
8. Τι περιγράφει η αρχή της τοπικότητας;
9. Ποιος ο σκοπός της ιεραρχίας μνήμης;
10. Περιγράψτε την αποθήκευση των δεδομένων στην ιεραρχία μνήμης.

Εργαστήριο #9

1. Σύνοψη προηγούμενων εργαστηρίων.

Στα προηγούμενα εργαστήρια ολοκληρώσατε τη σχεδίαση ενός **μονοπατιού δεδομένων (datapath) των 8 bits**, το οποίο εκτελεί πράξεις: α) μεταξύ δεδομένων από 2 καταχωρητές και β) μεταξύ δεδομένων από 1 καταχωρητή και δεδομένων από την άμεση είσοδο $im[7:0]$. Η λειτουργία του μονοπατιού δεδομένων ελέγχεται από 23 σήματα ελέγχου/δεδομένων (**είσοδοι στο μονοπάτι δεδομένων**):

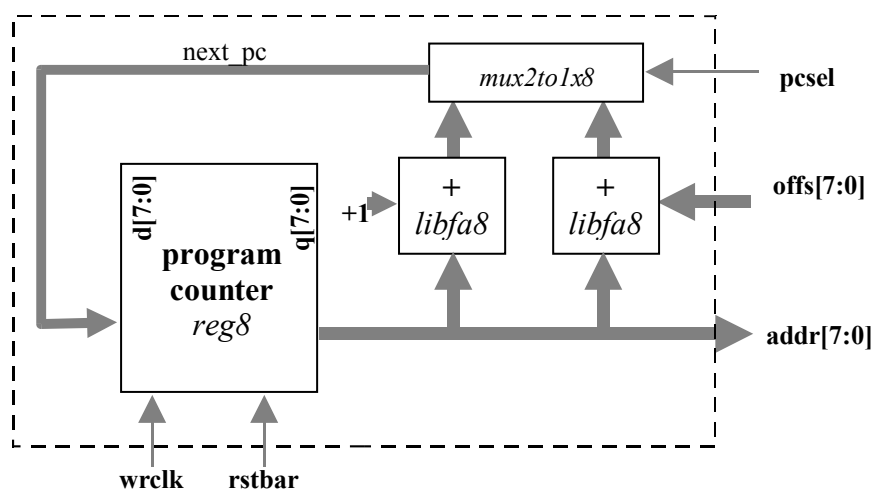
Σήματα	Λειτουργία
sub, op1, op0	επιλογή πράξης ΑΛΜ
sel	επιλογή εισόδου στην ΑΛΜ (1=από register file-έξοδος A, 0=από άμεση είσοδο $im[7:0]$)
rdb[2:0]	επιλογή καταχωρητή για την έξοδο B του register-file
rda[2:0]	επιλογή καταχωρητή για την έξοδο A του register-file
wrsel[2:0]	επιλογή καταχωρητή για εγγραφή
$im[7:0]$	άμεση είσοδος (8-bit)
wrclk	Εγγραφή δεδομένων στην ανερχόμενη ακμή του wrclk
rstbar	Μηδενισμός περιεχομένων καταχωρητών εάν $rstbar=0$

Επίσης, το μονοπάτι δεδομένων παράγει ως **εξόδους τα σήματα κατάστασης (flags) C/Z/V**, τα οποία θα χρησιμοποιήσετε αργότερα για τον έλεγχο των εντολών διακλάδωσης.

2. Μονάδα μετρητή εντολών (program counter - PC).

Στη συνέχεια θα αρχίσετε να υλοποιείτε τη **μονάδα ελέγχου**, η οποία θα συνδυαστεί με το **μονοπάτι δεδομένων** για την κατασκευή μιας **πλήρους ΚΜΕ**. Στο παρόν εργαστήριο θα υλοποιήσετε τη μονάδα του μετρητή εντολών, η οποία:

- Διαθέτει έναν 8-bit καταχωρητή (PC) για τη διεύθυνση της επόμενης εντολής.
- Αυξάνει την τιμή του PC κατά 1 μετά την εκτέλεση μιας εντολής.
- Παρέχει τη δυνατότητα αλλαγής τιμής στον PC, για την υλοποίηση των εντολών διακλάδωσης. Η αλλαγή υλοποιείται μέσω της πρόσθεσης ενός 8-bit offset στην τρέχουσα τιμή του PC.



3. Σχεδίαση της μονάδας μετρητή εντολών.

Δημιουργήστε ένα νέο cell στη βιβλιοθήκη σας και ονομάστε το **pclogic**. Το σχηματικό διάγραμμα της μονάδας φαίνεται στο προηγούμενο σχήμα.

Ο μετρητής εντολών (PC) φυλάσσεται σε έναν **καταχωρητή των 8 bits (reg8**, θα τον βρείτε στη βιβλιοθήκη **lablib.jelib**). Η έξοδος του καταχωρητή $q[7:0]$ παρέχει τη διεύθυνση $addr[7:0]$ για τη μνήμη εντολών.

Το περιεχόμενο του PC περνά μέσα από 2 **αθροιστές (libfa8**, θα τους βρείτε στη βιβλιοθήκη **imemlib.jelib** στο site του εργαστηρίου). Κάθε αθροιστής προσθέτει 2 8-bit εισόδους $a[7:0]$ και $b[7:0]$ και εμφανίζει το αποτέλεσμα στην έξοδο $s[7:0]$. Ο πρώτος αθροιστής παράγει το (PC+1) και ο δεύτερος το (PC+offset).

Η επιλογή μεταξύ του (PC+1) και του (PC+offset) γίνεται σε **πολυπλέκτη (mux2to1x8**, βιβλιοθήκη **lablib.jelib**) μέσω του σήματος **pcsel** (με 0 περνά το PC+1, με 1 το PC+offset).

Η τιμή που περνά από τον πολυπλέκτη θα αποτελέσει τη νέα τιμή του PC, η οποία θα αποθηκευτεί στον καταχωρητή στην επόμενη ανερχόμενη ακμή του σήματος ρολογιού **wrclk**. Ο καταχωρητής διαθέτει είσοδο άμεσου μηδενισμού (**rstbar**) για την αρχικοποίηση του PC σε 0.

4. Διαδικασία ελέγχου.

Εξομοιώστε τη λειτουργία της νέας μονάδας. Θέστε τα **rstbar**, **wrclk** και **pcsel** σε 0 (low). Στη συνέχεια το **rstbar** πρέπει να γίνει 1 (high) για κανονική λειτουργία. Αμέσως μετά:

α) Μετακινήστε το σημείο χρόνου 20ns δεξιότερα και επιλέξτε το σήμα **wrclk**.

γ) Κάνετε zoom-out στο παράθυρο της εξομοίωσης, πιέζοντας 4 φορές Ctrl-0.

β) Από το μενού **Tools, Simulation (Built-in)** επιλέξτε **Set clock on selected signal...** και ορίστε ως περίοδο τα 0.000000150s (150ns).

Παρατηρήστε τις γραμμές διεύθυνσης ($addr[7:0]$). Θα πρέπει να αυξάνονται κατά 1 σε κάθε κύκλο ρολογιού.

Θέστε τα σήματα $offs[7:0]$ σε διάφορες τιμές και το **pcsel** σε 1. Τώρα οι διευθύνσεις θα πρέπει να αυξάνονται κατά **offset** σε κάθε κύκλο ρολογιού.

5. Τμήμα ελέγχου διακλάδωσης.

Ως τελευταίο βήμα θα σχεδιάσετε τη μονάδα ελέγχου εκτέλεσης των εντολών διακλάδωσης. Μια εντολή διακλάδωσης χρησιμοποιεί 3 bits ($cc[2:0]$) για να δηλώσει τη **συνθήκη διακλάδωσης**:

$cc[1:0]$	Συνθήκη διακλάδωσης
000	bn (branch never): όχι διακλάδωση
001	bz (branch on zero): διακλάδωση εάν το αποτέλεσμα της εντολής θέτει το Z flag (είναι μηδενικό)
010	bc (branch on carry): διακλάδωση εάν το αποτέλεσμα της εντολής θέτει το C flag (παράγει κρατούμενο)
010	bv (branch on overflow): διακλάδωση εάν το αποτέλεσμα της εντολής θέτει το V flag (προκλήθηκε υπερχειλίση)
111	ba (branch always): διακλάδωση πάντοτε (χωρίς συνθήκη)

Εάν η απαιτούμενη συνθήκη διακλάδωσης είναι αληθής, τότε το σήμα **pcsel** γίνεται 1. Το ζητούμενο λοιπόν είναι η σχεδίαση ενός cell (ονομάστε το **brlogic**), το οποίο θα δέχεται ως είσοδο τα $cc[2:0]$ και τα **z**, **c**, **v** από την ΑΛΜ και θα παράγει το κατάλληλο σήμα **pcsel** (0 όταν δεν θα εκτελεστεί η διακλάδωση και 1 όταν θα εκτελεστεί).

Υπόδειξη: Θα πρέπει να χρησιμοποιήσετε έναν αποκωδικοποιητή από 3 σε 8 (dec3to8, θα τον βρείτε στη βιβλιοθήκη **lablib.jelib**), καθώς και διάφορες λογικές πύλες. Ο αποκωδικοποιητής χρησιμοποιείται για να παράγει '1' σε 8 διαφορετικές γραμμές ανάλογα με τον συνδυασμό των $cc[2:0]$. Οι γραμμές αυτές (δείτε και τον πίνακα των συνθηκών διακλάδωσης) πρέπει να συνδυαστούν με τα Z, C, S, V (με πύλες AND) για να οδηγήσουν τελικά (μέσω πύλης OR) το σήμα **pcsel** που καθορίζει αν θα εκτελεστεί η διακλάδωση ή όχι.

Ερωτήσεις αυτοαξιολόγησης 10^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 10ου μαθήματος (“Κρυφές Μνήμες”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Ποια τα χαρακτηριστικά της “ιδανικής μνήμης” που προσπαθεί να προσεγγίσει η ιεραρχία μνήμης;
2. Περιγράψτε την χωρική και χρονική τοπικότητα, καθώς και εφαρμογές των δύο αυτών μορφών τοπικότητας.
3. Ποιος διαχειρίζεται την κρυφή μνήμη; Ο κώδικας κάθε εκτελούμενης εφαρμογής γνωρίζει τι συμβαίνει στην κρυφή μνήμη;
4. Περιγράψτε την αποθήκευση δεδομένων στην ιεραρχία μνήμης.
5. Με ποιον τρόπο μεταφέρονται δεδομένα μεταξύ κρυφής και κύριας μνήμης; Γιατί;
6. Περιγράψτε την προσπέλαση δεδομένων από την ΚΜΕ στις περιπτώσεις cache hit και cache miss.
7. Πώς οργανώνεται μια κρυφή μνήμη; Τι αποθηκεύεται σε κάθε θέση της κρυφής μνήμης;
8. Σε ποια θέση της κρυφής μνήμης τοποθετείται ένα μπλοκ δεδομένων από την κύρια μνήμη; Ποιος αποφασίζει για τη θέση του; Περιγράψτε την τεχνική άμεσης απεικόνισης για την τοποθέτηση των μπλοκ στην κρυφή μνήμη.
9. Ποια τα χαρακτηριστικά μεγέθη για την απόδοση της κρυφής μνήμης; Πώς υπολογίζονται οι χαμένοι κύκλοι ρολογιού κατά την προσπέλαση μνήμης;
10. Ποιο το κόστος των cache misses; Ποια μεγέθη μπορούν να βελτιωθούν ως προς την απόδοση;

Εργαστήριο #10

1. Κωδικοποίηση εντολών.

Σε κάθε ΚΜΕ τα bits μιας εντολής χρησιμοποιούνται για την παραγωγή των σημάτων ελέγχου του μονοπατιού δεδομένων (datapath). Τα σήματα ελέγχου παράγονται από την αποκωδικοποίηση των bits της εντολής, κάθε σήμα ελέγχου εξάγεται δηλαδή από μια λογική συνάρτηση των bits της εντολής. Στις πραγματικές ΚΜΕ, όπου κάθε εντολή ολοκληρώνεται σε πολλά στάδια, τα σήματα ελέγχου εξαρτώνται και από τη χρονική φάση της εκτέλεσης της εντολής.

Για την απλοποίηση της εργαστηριακής άσκησης θα θεωρήσετε ότι ισχύουν οι παρακάτω απλουστεύσεις:

- Η ΚΜΕ εκτελεί **μία εντολή ανά κύκλο** ρολογιού.
- Τα bits της εντολής αντιστοιχούν **ένα-προς-ένα** με τα σήματα ελέγχου (**δεν χρειάζεται δηλαδή αποκωδικοποίηση της εντολής**).
- Η μνήμη εντολών αποτελείται από λέξεις, με **εύρος κάθε λέξης όσα ακριβώς τα bits της εντολής**.

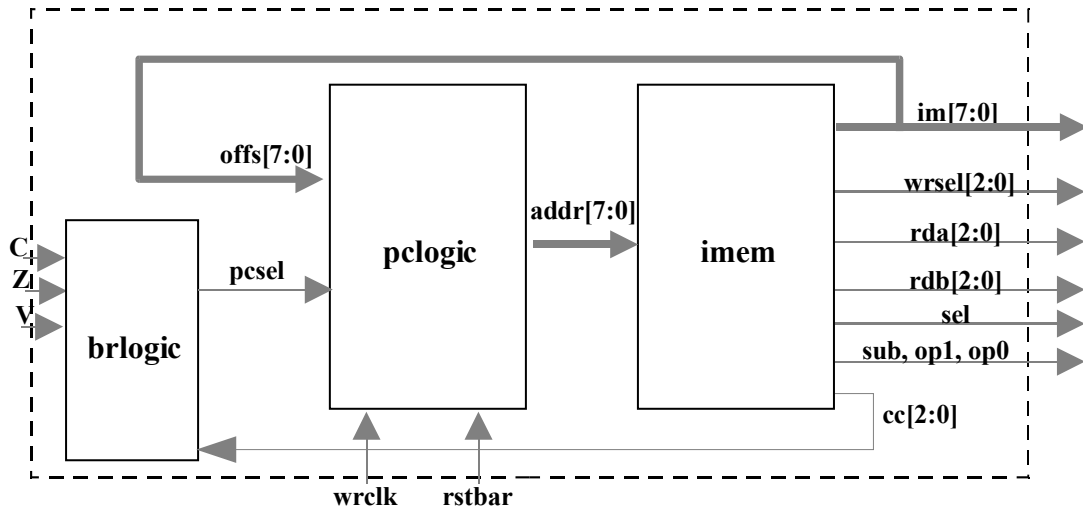
Οι εντολές που θα χρησιμοποιήσετε έχουν μήκος 24 bits (23..0), και η αντιστοιχία τους με τα σήματα ελέγχου του μονοπατιού δεδομένων θα είναι η ακόλουθη:

bit 23							bit 0
cc[2:0]	sub	op[1:0]	sel	rdb[2:0]	rda[2:0]	wrsel[2:0]	im[7:0]

Bits εντολής	Σήματα ελέγχου	Λειτουργία
23, 22, 21	cc[2:0]	έλεγχος διακλάδωσης
20, 19, 18	sub, op1, op0	επιλογή πράξης ΑΛΜ
17	sel	επιλογή εισόδου στην ΑΛΜ (1=από register file-έξοδος A, 0=από άμεση είσοδο im[7:0])
16, 15, 14	rdb[2:0]	επιλογή καταχωρητή για την έξοδο B του register-file
13, 12, 11	rda[2:0]	επιλογή καταχωρητή για την έξοδο A του register-file
10, 9, 8	wrsel[2:0]	επιλογή καταχωρητή για εγγραφή
7, 6, 5, 4, 3, 2, 1, 0	im[7:0]	άμεση είσοδος (8-bit)

2. Μονάδα ελέγχου (control unit) της ΚΜΕ.

Για τη μονάδα ελέγχου της ΚΜΕ, δημιουργήστε ένα καινούργιο cell στη βιβλιοθήκη σας και ονομάστε το **instretl**. Εκεί μέσα συνδυάστε τα cells **pclogic** και **brlogic** που έχετε φτιάξει στο προηγούμενο εργαστήριο, με το cell **imem**, το οποίο θα βρείτε στη βιβλιοθήκη **imemlib.jelib**. Το cell imem είναι ένα απλό μοντέλο μνήμης εντολών με χωρητικότητα έως 256 λέξεις των 24 bits η καθεμία.



3. Εξομοίωση της μονάδας ελέγχου.

Ξεκινήστε την εξομοίωση της μονάδας ελέγχου, ακριβώς όπως και στα προηγούμενα εργαστήρια. Η μόνη διαφορά είναι ότι η μνήμη εντολών (imem) δεν έχει περιεχόμενο. Για να το ορίσετε, ακολουθήστε τα πιο κάτω βήματα:

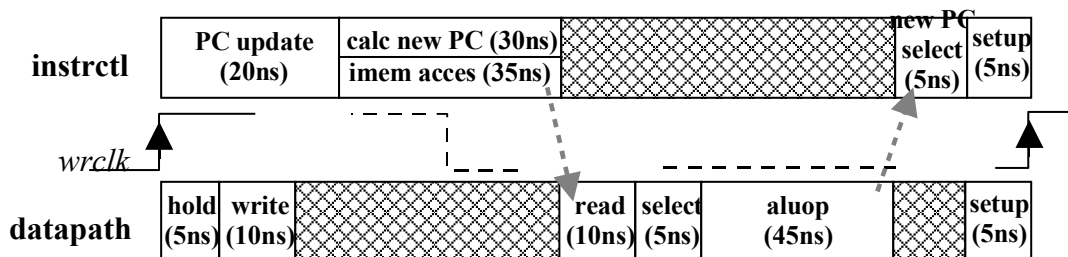
α) Με τη βοήθεια του προγράμματος **δημιουργίας περιεχομένου μνήμης** (θα το βρείτε στο site του εργαστηρίου), κατασκευάστε το μοντέλο της μνήμης σε μορφή αρχείου κειμένου.

β) Στο electric, ξεκινήστε την εξομοίωση, αλλά αφού τακτοποιήσετε τα σήματα που θέλετε να δείτε, κλείστε το παράθυρο της εξομοίωσης. Ανοίξτε το **instrectl{net.als}**. Πηγαίνατε στην αρχή του κειμένου και αντικαταστήστε (copy-paste) το μοντέλο της μνήμης (model imem..) με το κείμενο που δημιουργήσατε.

γ) Ενώ έχετε ανοιχτό το **instrectl{net.als}** (κι όχι το σχηματικό), ξεκινήστε την εξομοίωση. Οδηγήστε το rstbar και το wrclk όπως στο προηγούμενο εργαστήριο, με περίοδο 150ns. Θα πρέπει τώρα να βλέπετε τα σήματα ελέγχου (έξοδοι από τη μνήμη εντολών) να αλλάζουν σύμφωνα με το ρολόι wrclk και τις εισόδους c, z, v.

3. Ολοκλήρωση της ΚΜΕ.

Σχεδιάστε το τελικό τμήμα **cpu**, συνδυάζοντας τα τμήματα **datapath** και **instrectl**.



Παρατηρήστε ότι στον τελικό σχεδιασμό, η επιλογή του νέου PC εξαρτάται από το αποτέλεσμα της ΑΛΜ: για τον λόγο αυτόν η επιλογή του PC μπορεί να γίνει μόνο μετά την εκτέλεση της πράξης στην ΑΛΜ.

4. Δοκιμή της ολοκληρωμένης ΚΜΕ.

Υλοποιήστε μια επαναληπτική ακολουθία εντολών για τον υπολογισμό του $1+2+3+4+...+n$.

Ερωτήσεις αυτοαξιολόγησης 11^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 11ου μαθήματος (“Εικονική Μνήμη”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Τι είναι η “δευτερεύουσα μνήμη”; Ποια η θέση της στην ιεραρχία μνήμης;
2. Γιατί η τεχνική των overlays δεν έλυσε το πρόβλημα της περιορισμένης φυσικής μνήμης;
3. Ποιος είναι ο χώρος διευθύνσεων ενός προγράμματος;
4. Γιατί η αποσύνδεση λογικών και φυσικών διευθύνσεων επιτρέπει την τοποθέτηση ενός προγράμματος οπουδήποτε στη μνήμη;
5. Ποια μέρη του υπολογιστικού συστήματος πρέπει να συνεργαστούν για τη λειτουργία της Εικονικής Μνήμης;
6. Περιγράψτε δύο λόγους για την εφαρμογή της Εικονικής Μνήμης. Ποιος είναι ο σημαντικότερος σήμερα;
7. Περιγράψτε τη σχέση σελίδων και πλαισίων κύριας μνήμης-δίσκου.
8. Ποια είναι τα μέρη της εικονικής (λογικής) και φυσικής διεύθυνσης; Πώς γίνεται η μετάφραση από εικονικές σε φυσικές διευθύνσεις;
9. Πού βρίσκονται οι πίνακες σελίδων; Τι είναι το TLB και για ποιον λόγο είναι αναγκαίο;
10. Γιατί κάθε ξεχωριστή διεργασία πρέπει να έχει διαφορετικούς πίνακες σελίδων; Πώς το υπολογιστικό σύστημα εγγυάται ότι ένα πρόγραμμα δεν μπορεί να προσπελάσει δεδομένα ενός άλλου προγράμματος;

Ερωτήσεις αυτοαξιολόγησης 12^ο μαθήματος

Έχοντας κατανοήσει την ύλη του 12ου μαθήματος (“Διασύνδεση Εισόδου-Εξόδου (I) - δίαυλοι και μέθοδοι επικοινωνίας”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Ποιος ο ρόλος ενός διασυνδεδετικού διαύλου στο υπολογιστικό σύστημα; Τι μπορεί να διασυνδέει ένας δίαυλος;
2. Γιατί υπάρχει τόσο μεγάλη ποικιλία στα χαρακτηριστικά των διαφόρων διαύλων;
3. Πώς διαμορφώνεται η απόδοση ανάλογα με τη θέση του διαύλου στο υπολογιστικό σύστημα; Τι συμβαίνει με άλλα μεγέθη, όπως η τυποποίηση, η ευχρηστία και η αντοχή;
4. Πώς λειτουργεί ένας παράλληλος και ένας σειριακός δίαυλος; Ποια τα πλεονεκτήματα και τα μειονεκτήματα του καθενός; Σε ποια θέση στο υπολογιστικό σύστημα χρησιμοποιούνται οι δύο αυτοί τύποι διαύλων;
5. Πώς συγχρονίζεται η αποστολή και λήψη δεδομένων;
6. Περιγράψτε δύο τοπολογίες διαύλων. Ποια χρησιμοποιείται περισσότερο σήμερα;
7. Ποιος ελέγχει τη μεταφορά δεδομένων σε μια επικοινωνία master-slave; Πώς ορίζεται κατά σύμβαση η ανάγνωση και η εγγραφή;
8. Τι επιλέγει η διεύθυνση ανάγνωσης ή εγγραφής;
9. Ποιες είναι οι τυπικές φάσεις κατά τη μεταφορά δεδομένων σε έναν δίαυλο;
10. Πώς μεταφέρονται οι διευθύνσεις, οι εντολές και τα δεδομένα σε παράλληλους και σειριακούς διαύλους;

Ερωτήσεις αυτοαξιολόγησης 13^{ου} μαθήματος

Έχοντας κατανοήσει την ύλη του 13ου μαθήματος (“Διασύνδεση Εισόδου-Εξόδου (II) - συσκευές και διάλογοι E/E”) θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Ποια μέρη ενός υπολογιστικού συστήματος χαρακτηρίζονται ως συσκευές E/E;
2. Γιατί οι συσκευές και οι διάλογοι E/E πρέπει να υπακούν σε πρότυπα;
3. Περιγράψτε τρεις τρόπους αναγνώρισης-διασύνδεσης συσκευών E/E σε ένα υπολογιστικό σύστημα.
4. Ποιος ο ρόλος ενός ελεγκτή E/E; Γιατί είναι αναγκαίος;
5. Τι περιγράφει το “μοντέλο καταχωρητών” στον τρόπο διασύνδεσης ενός ελεγκτή E/E με την ΚΜΕ;
6. Περιγράψτε τις διαφορές μεταξύ απεικονισμένης στη μνήμη και απομονωμένης E/E.
7. Περιγράψτε τη διαδικασία polling.
8. Πώς λειτουργεί ο μηχανισμός των ασύγχρονων διακοπών (interrupts);
9. Με ποιους τρόπους μπορούν να μετακινηθούν τα δεδομένα μεταξύ ελεγκτή E/E και κύριας μνήμης; Περιγράψτε τον μηχανισμό DMA.
10. Τι είναι ένας περιφερειακός διάλογος και από ποια τμήματα συγκροτείται;