



## Εργασία 1 (υποχρεωτική) – Διοχέτευση

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2021 – 2022

(ΕΚΦΩΝΗΣΗ) ΠΕΜΠΤΗ 25 ΝΟΕΜΒΡΙΟΥ 2021

(ΠΑΡΑΔΟΣΗ ΣΤΟ ECLASS ΜΕΧΡ) ΠΑΡΑΣΚΕΥΗ 17 ΔΕΚΕΜΒΡΙΟΥ 2021

Επώνυμο	Όνομα	Αριθμός Μητρώου	Email
Γιαννοπούλου	Κατερίνα	1115201600034	sdi1600034@di.uoa.gr
Φώτης	Γιάννης	1115201700182	sdi1700182@di.uoa.gr

### Πληροφορίες για τις Υποχρεωτικές Εργασίες του μαθήματος

- Οι υποχρεωτικές εργασίες του μαθήματος είναι **δύο**. Σκοπός τους είναι η κατανόηση των εννοιών του μαθήματος με χρήση αρχιτεκτονικών προσομοιωτών. Η πρώτη υποχρεωτική εργασία (αυτή) αφορά τη διοχέτευση (pipelining) και η δεύτερη θα αφορά τις κρυφές μνήμες (cache memories).
- Οι δύο εργασίες είναι υποχρεωτικές και η βαθμολογία του μαθήματος θα προκύπτει από το γραπτό (60%), την εργασία της διοχέτευσης (20%), και την εργασία των κρυφών μνημών (20%).
- Κάθε ομάδα μπορεί να αποτελείται **από 1 έως και 3 φοιτητές**. Συμπληρώστε τα στοιχεία όλων των μελών της ομάδας στον παραπάνω πίνακα. Όλα τα μέλη της ομάδας πρέπει να έχουν ισότιμη συμμετοχή και να γνωρίζουν τις λεπτομέρειες της υλοποίησης της ομάδας.
- Για την εξεταστική Σεπτεμβρίου δε θα δοθούν άλλες εργασίες. Το Σεπτέμβριο εξετάζεται μόνο το γραπτό.
- Σε περίπτωση αντιγραφής θα μηδενίζονται όλες οι ομάδες που μετέχουν σε αυτή.
- Η παράδοση της **Εργασίας Διοχέτευσης** πρέπει να γίνει μέχρι τα **μεσάνυχτα της προθεσμίας ηλεκτρονικά** και μόνο στο eclass (να ανεβάσετε ένα μόνο αρχείο zip ή rar με την τεκμηρίωσή σας σε PDF και τον κώδικά σας). **Μην περιμένετε μέχρι την τελευταία στιγμή. Η εκκώφηση της εργασίας 2 των κρυφών μνημών θα ανατεθεί αμέσως μετά.**

### Ζητούμενο

Το ζητούμενο της εργασίας είναι να σχεδιάσετε έναν μικροεπεξεργαστή MIPS με διοχέτευση (pipeline) που να εκτελεί ένα συγκεκριμένο πρόγραμμα **στον μικρότερο δυνατό χρόνο ανά μονάδα κόστους** (cost effectiveness). Η εκτέλεση του προγράμματος και η αξιολόγηση των σχεδιάσεών σας θα γίνει στον προσομοιωτή QtMips που χρησιμοποιούμε στο εργαστήριο.

Το πρόγραμμα πρέπει να επεξεργάζεται έναν πίνακα 198 θετικών ακεραίων αριθμών  $[k_1, \dots, k_{198}]$  που έχει τυχαίο και μη ταξινομημένο περιεχόμενο και κάθε αριθμός είναι μικρότερος του 100. Για κάθε τριάδα διαδοχικών αριθμών, το πρόγραμμα πρέπει να υπολογίζει τον μέγιστο κοινό διαιρέτη (gcd) και το ελάχιστο κοινό πολλαπλάσιο (lcm) αυτών. Τα αποτελέσματα πρέπει να γράφονται σε δύο ξεχωριστούς πίνακες ακεραίων μεγέθους 66 θέσεων ο καθένας (ένα στοιχείο για κάθε τριάδα). Ο αρχικός πίνακας θα έχει ήδη περιεχόμενα στο τμήμα .data του προγράμματος και δεν θα δίδονται αριθμοί από τον χρήστη ούτε θα εκτυπώνονται αποτελέσματα στην κονσόλα. Οι σωστές τιμές των πινάκων gcd και lcm πρέπει να βρίσκονται απλά στη μνήμη<sup>1</sup>.

Μπορείτε να λάβετε υπ' όψιν σας **διαφορετικές διαμορφώσεις** της διοχέτευσης του μικροεπεξεργαστή MIPS **ανάλογα με το κόστος** και να επιλέξετε αυτή που δίνει για το πρόγραμμά σας την καλύτερη απόδοση ανά μονάδα κόστους. Σε αυτή την εργασία θεωρούμε ότι η μνήμη είναι ιδανική και συνεπώς σε όλες τις διαμορφώσεις οι κρυφές μνήμες πρέπει να είναι απενεργοποιημένες και η κύρια μνήμη να έχει χρόνο προσπέλασης 1 κύκλο ρολογιού. Οι επιλογές σας για την σχεδίαση του μικροεπεξεργαστή συνοψίζονται στον ακόλουθο πίνακα. Το πρόσθετο κόστος και η επιβάρυνση στο ρολόι είναι **αθροιστικά στο βασικό κόστος Κ και στον βασικό κύκλο ρολογιού C**.

Παράμετρος	Πρόσθετο κόστος	Επιβάρυνση στο ρολόι
Αγνόηση κινδύνων δεδομένων	0	0
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	+ 2 %	0
Ανίχνευση κινδύνων δεδομένων και προώθηση	+ 5 %	+ 3 %
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	0	0
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	+ 2 %	0
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	+ 7 %	+ 1 %
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	+ 8 %	+ 2 %
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	+ 1 %	0

<sup>1</sup> Φυσικά τα παραπάνω ισχύουν για τις τελικές μετρήσεις απόδοσης και κόστους του προγράμματός σας. Κατά την ανάπτυξη του κώδικά σας και για τις δοκιμές σας μπορείτε να εκτυπώνετε ή να δίνετε εισόδους όπως το επιθυμείτε.  
[ Γιαννοπούλου – Φώτης ]

Εκτός από τα προγράμματά σας σε συμβολική γλώσσα, που πρέπει να παραδώσετε ξεχωριστά, να συμπληρώσετε τους ακόλουθους πίνακες με τις επιλογές και τα αποτελέσματά σας. Τα αναφέρετε και τα περιεχόμενα του αρχικού πίνακα με τα οποία πήρατε τις μετρήσεις που θα γράψετε στον πίνακα. Αν πάρετε μετρήσεις με περισσότερα από ένα περιεχόμενα του πίνακα να δώσετε όλα τα δεδομένα.

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	
Ανίχνευση κινδύνων δεδομένων και προώθηση	
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	

Dataset	Τελικό κόστος (ως συνάρτηση του βασικού Κ)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[\text{Χρόνος} * \text{Κόστος}]$ )

## Τεκμηρίωση

[ Σύντομη τεκμηρίωση της λύσης σας μέχρι **10 σελίδες ξεκινώντας από την επόμενη σελίδα** – μην αλλάζετε τη μορφοποίηση του κειμένου (**και παραδώστε την τεκμηρίωση σε αρχείο PDF**). Η τεκμηρίωσή σας πρέπει να περιλαμβάνει παραδείγματα ορθής εκτέλεσης του προγράμματος και σχολιασμό για την επίλυση του προβλήματος και την επίτευξη του ζητούμενου. Μπορείτε να χρησιμοποιήσετε εικόνες, διαγράμματα και ό,τι άλλο μπορεί να βοηθήσει στην εξήγηση της δουλειάς σας. ]

## Εισαγωγή:

Χρησιμοποιήσαμε 3 datasets τυχαίων ακεραίων για να διαπιστώσουμε πως επηρεάζουν τα διαφορετικά δεδομένα εισόδου την απόδοση του επεξεργαστή στις διάφορες υλοποιήσεις που επιλέξαμε. Ο βασικός κώδικάς μας βρίσκεται στο αρχείο “ca-II-handouts.s”. Μπορείτε να βρείτε τα datasets σε μορφή txt στον υποφάκελο “Datasets”, το περιεχόμενο των οποίων τοποθετείται κάθε φορά στον πίνακα “arr1”. Επιπλέον, για τις περιπτώσεις που χρησιμοποιούμε Delay slot, πρέπει να αφαιρέσουμε το σχολιασμό από τις εντολές nop μετά από κάθε εντολή τύπου jump και την πρώτη απ’ τις δύο nop μετά από κάθε εντολή branch όταν η επίλυση των branch γίνεται στο ID στάδιο. Αν η επίλυση γίνεται στο στάδιο EX, αφαιρούμε το σχολιασμό από όλες τις εντολές nop. Εναλλακτικά, για λόγους πρακτικότητας, έχουμε ετοιμάσει και τα κατάλληλα αρχεία κώδικα για τις παραπάνω δύο περιπτώσεις, τα οποία μπορείτε να βρείτε στον υποφάκελο “Other Implementations”.

Δεν θεωρήσαμε σκόπιμο να καταγράψουμε δεδομένα για την περίπτωση αγνόησης κινδύνων δεδομένων, δηλαδή αποφασίσαμε να έχουμε το data hazard unit ενεργό σε όλες τις περιπτώσεις, καθώς το κόστος του είναι μόλις 2% στην αμέσως φτηνότερη περίπτωση όπου κάνουμε stall on data hazard. Επιπλέον, μία άλλη παραδοχή που κάναμε, είναι ότι θα προσθέσουμε σε όλες τις υλοποιήσεις - εκτός από μία - το hardware που απαιτείται για να ολοκληρώνονται οι branch εντολές στο στάδιο ID αντί του EX. Ο λόγος και πάλι είναι το πολύ χαμηλό κόστος σε σχέση με το overhead που θα γλιτώσουμε αν το προσθέσουμε. Τέλος, άλλος ένας λόγος, είναι για να μειώσουμε το συνολικό αριθμό συνδυασμών που θα εξετάσουμε σε αυτή την εργασία χάριν έκτασης του παραδοτέου κειμένου. Ωστόσο, για να αποδείξουμε την ορθότητα του παραπάνω συλλογισμού, οι δύο πρώτες εκδοχές που θα παρουσιάσουμε, διαφέρουν μόνο στο στάδιο του Branch resolution.

Παρ’ όλα αυτά, έχουμε ετοιμάσει ένα συμπληρωματικό spreadsheet αρχείο (Implementation Statistics.xlsx) με στατιστικά για πολύ πιο αναλυτική σύγκριση των διαφόρων σεναρίων. Εκεί, φαίνεται ξεκάθαρα ότι σε κάθε υλοποίηση, συμφέρει όντως να έχουμε το Branch resolution στο ID στάδιο.

Τα σενάρια μας ξεκινούν από την αρχιτεκτονική με το χαμηλότερο κόστος και προσθέτουμε σταδιακά τμήματα του hardware με τη σειρά που αυτά προσφέρονται στον simulator QtMips.

### Case 0: Stall on Branch & Branch resolution @ EX & Data hazard detection without forwarding:

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	Όχι
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	Ναι
Ανίχνευση κινδύνων δεδομένων και προώθηση	Όχι
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	Όχι
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	Ναι
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	Όχι
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	Όχι
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	Όχι

Dataset	Τελικό κόστος (ως συνάρτηση του βασικού K)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[\text{Χρόνος} * \text{Κόστος}]$ )
0	1,04 K	1,00 C	19.011	13.119	1,44912	19.011,00 C	$5,058 * 10^{-5} * 1/(C * K)$
1	1,04 K	1,00 C	19.811	13.679	1,44828	19.811,00 C	$4,854 * 10^{-5} * 1/(C * K)$
2	1,04 K	1,00 C	19.431	13.413	1,44867	19.431,00 C	$4,948 * 10^{-5} * 1/(C * K)$
Average	1,04 K	1,00 C	19.417,67	13.403,67	1,44869	19.417,67 C	$4,953 * 10^{-5} * 1/(C * K)$

**Case 1:** Stall on Branch & Branch resolution @ ID & Data hazard detection without forwarding:

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	Όχι
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	Ναι
Ανίχνευση κινδύνων δεδομένων και προώθηση	Όχι
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	Όχι
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	Ναι
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	Όχι
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	Όχι
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	Ναι

Dataset	Τελικό κόστος (ως συνάρτηση του βασικού Κ)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[\text{Χρόνος} * \text{Κόστος}]$ )
0	1,05 K	1,00 C	18.107	13.119	1,38021	18.107,00 C	$5,260 * 10^{-5} * 1/(C * K)$
1	1,05 K	1,00 C	18.867	13.679	1,37927	18.867,00 C	$5,048 * 10^{-5} * 1/(C * K)$
2	1,05 K	1,00 C	18.506	13.413	1,37971	18.506,00 C	$5,146 * 10^{-5} * 1/(C * K)$
Average	1,05 K	1,00 C	18.493,33	13.403,67	1,37973	18.493,33 C	$5,151 * 10^{-5} * 1/(C * K)$

Όπως διαπιστώνουμε, όταν προσθέσουμε το hardware για να γίνει το Branch resolution στο στάδιο ID, βελτιώνονται τόσο οι συνολικοί κύκλοι που εκτελείται το πρόγραμμα – άρα και το CPI – όσο και ο χρόνος εκτέλεσης, με αποτέλεσμα να έχουμε καλύτερο λόγο απόδοσης προς κόστος. Συγκεκριμένα, παρατηρούμε βελτίωση στο CPI κατά 6,9% έχοντας πληρώσει μόνο 1% του κόστους επιπλέον. Φυσικά, περιμένουμε όσο προστίθεται hardware, το CPI να πλησιάζει πολύ το ιδανικό, επομένως η επίδραση της παραμέτρου αυτής θα έχει μικρότερο αντίκτυπο, αφού ο επεξεργαστής θα είναι πιο γρήγορος συνολικά. Ωστόσο, σε κάθε υλοποίηση που δοκιμάσαμε, η διαφορά μεταξύ επιλογής ή μη αυτής της παραμέτρου βελτιώνει το τελικό CPI, κατ' ελάχιστον, κατά 1,5% (TODO link se 2 bit). Φυσικά δεν λάβαμε υπόψιν τα CPI των περιπτώσεων του Delay slot, μιας και το CPI του δεν συνυπολογίζει τους χαμένους κύκλους από τις nop που έχουμε προσθέσει ως προγραμματιστές, αλλά μόνο αυτές του hardware.

Στη συνέχεια θα συγκρίνουμε ποιο απ' τα υπόλοιπα κομμάτια του hardware αξίζει περισσότερο να προστεθεί ανάμεσα στην ανίχνευση κινδύνων δεδομένων με προώθηση, την αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης και την πρόβλεψη διακλάδωσης με 1 ή 2 bit. Κάθε φορά θα προσθέτουμε μόνο ένα από αυτά στην βασική περίπτωση 1 (Case 1), την οποία θα θεωρούμε και ως μέτρο σύγκρισης μεταξύ τους. Με αυτό τον τρόπο, στοχεύουμε στην ιεράρχηση των διαφορετικών επιλογών που μας δίνονται με βάση την απόλυτη επίδραση της κάθε μίας στον αρχικό μας επεξεργαστή. Έτσι θα οδηγηθούμε σταδιακά στην πιο συμφέρουσα λύση χωρίς να χρειαστεί να εξετάσουμε όλους τους πιθανούς συνδυασμούς υλοποιήσεων.

**Case 2:** Stall on Branch & Branch resolution @ ID & Data hazard detection with forwarding:

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	Όχι
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	Όχι
Ανίχνευση κινδύνων δεδομένων και προώθηση	Ναι
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	Όχι
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	Ναι
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	Όχι
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	Όχι
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	Ναι

Dataset	Τελικό κόστος (ως συνάρτηση του βασικού Κ)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[\text{Χρόνος} * \text{Κόστος}]$ )
0	1,08 K	1,03 C	15.961	13.119	1,21663	16.439,83 C	$5,632 * 10^{-5} * 1/(C * K)$
1	1,08 K	1,03 C	16.641	13.679	1,21654	17.140,23 C	$5,402 * 10^{-5} * 1/(C * K)$
2	1,08 K	1,03 C	16.318	13.413	1,21658	16.807,54 C	$5,509 * 10^{-5} * 1/(C * K)$
Average	1,08 K	1,03 C	16.306,67	13.403,67	1,21658	16.795,87 C	$5,514 * 10^{-5} * 1/(C * K)$

**Case 3:** Delay slot & Branch resolution @ ID & Data hazard detection without forwarding:

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	Όχι
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	Ναι
Ανίχνευση κινδύνων δεδομένων και προώθηση	Όχι
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	Ναι
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	Όχι
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	Όχι
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	Όχι
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	Ναι

Dataset	Τελικό κόστος (ως συνάρτηση του βασικού Κ)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[\text{Χρόνος} * \text{Κόστος}]$ )
0	1,03 K	1,00 C	18.107	15.961	1,13445	18.107,00 C	$5,362 * 10^{-5} * 1/(C * K)$
1	1,03 K	1,00 C	18.867	16.641	1,13377	18.867,00 C	$5,146 * 10^{-5} * 1/(C * K)$
2	1,03 K	1,00 C	18.506	16.318	1,13409	18.506,00 C	$5,246 * 10^{-5} * 1/(C * K)$
Average	1,03 K	1,00 C	18.493,33	16.306,67	1,13410	18.493,33 C	$5,251 * 10^{-5} * 1/(C * K)$

**Case 4:** 1-bit Branch prediction & Branch resolution @ ID & Data hazard detection without forwarding:

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	Όχι
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	Ναι
Ανίχνευση κινδύνων δεδομένων και προώθηση	Όχι
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	Όχι
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	Όχι
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	Ναι
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	Όχι
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	Ναι



Dataset	Τελικό κόστος (ως συνάρτηση του βασικού K)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[\text{Χρόνος} * \text{Κόστος}]$ )
0	1,10 K	1,01 C	16.504	13.119	1,25802	16.669,04 C	$5,454 * 10^{-5} * 1/(C * K)$
1	1,10 K	1,01 C	17.184	13.679	1,25623	17.355,84 C	$5,238 * 10^{-5} * 1/(C * K)$
2	1,10 K	1,01 C	16.861	13.413	1,25706	17.029,61 C	$5,338 * 10^{-5} * 1/(C * K)$
Average	1,10 K	1,01 C	16.849,67	13.403,67	1,25711	17.018,16 C	$5,343 * 10^{-5} * 1/(C * K)$

**Case 5:** 2-bit Branch prediction & Branch resolution @ ID & Data hazard detection without forwarding:

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	Όχι
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	Ναι
Ανίχνευση κινδύνων δεδομένων και προώθηση	Όχι
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	Όχι
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	Όχι
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	Όχι
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	Ναι
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	Ναι

Dataset	Τελικό κόστος (ως συνάρτηση του βασικού K)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[\text{Χρόνος} * \text{Κόστος}]$ )
0	1,11 K	1,02 C	16.307	13.119	1,24301	16.633,14 C	$5,416 * 10^{-5} * 1/(C * K)$
1	1,11 K	1,02 C	16.987	13.679	1,24183	17.326,74 C	$5,199 * 10^{-5} * 1/(C * K)$
2	1,11 K	1,02 C	16.664	13.413	1,24238	16.997,28 C	$5,300 * 10^{-5} * 1/(C * K)$
Average	1,11 K	1,02 C	16.652,67	13.403,67	1,24240	16.985,72 C	$5,305 * 10^{-5} * 1/(C * K)$

Παρατηρούμε ότι η μεγαλύτερη βελτίωση επιτεύχθηκε όταν προσθέσαμε το data forwarding για τα data hazards. Αξίζει να σημειωθεί ότι κάτι τέτοιο δεν είναι αυτονόητο, καθώς η συγκεκριμένη επιλογή κοστίζει πολύ, ενώ παράλληλα επιβαρύνει και το ρολόι τη στιγμή που τα control hazards είναι περισσότερα απ' τα data hazards σε όλες τις περιπτώσεις. Αυτό συμβαίνει επειδή η ύπαρξη του forwarding εξαλείφει πλήρως τους χαμένους κύκλους λόγω κινδύνων δεδομένων, διότι δεν υπάρχουν εξαρτήσεις δεδομένων μεταξύ διαδοχικών εντολών (load-use hazards, R-to-R use hazards κοκ.). Αντίθετα, οι κίνδυνοι ελέγχων δεν είναι δυνατόν να μηδενιστούν ακόμη και στην περίπτωση που χρησιμοποιήσουμε 2-bit Branch predictor. Συγκεκριμένα, για τα 3 datasets που δοκιμάσαμε, τα αρχικά stalls λόγω κινδύνων δεδομένων ήταν κατά μέσο όρο 2.186,67 και αυτά των κινδύνων δεδομένων ήταν 2.903. Όμως, παρά τη χρήση 2-bit Branch predictor, τα control stalls ήταν κατά μέσο όρο 601 χαμένοι κύκλοι, άρα καταλαβαίνουμε ότι με την προσθήκη της προώθησης δεδομένων κερδίζουμε 2186,67 κύκλους και με τη χρήση πρόβλεψης διακλάδωσης κερδίζουμε 2302 κύκλους. Ως εκ τούτου, η χαμηλότερη επιβάρυνση που προσθέτει η προώθηση δεδομένων έναντι της πρόβλεψης διακλάδωσης, την καθιστά πιο συμφέρουσα επιλογή. Όσον αφορά το Delay slot - αν και η φτηνότερη λύση από τις παραπάνω - δεν είναι η πιο αποδοτική, καθώς ο χρόνος εκτέλεσης αυξάνεται σημαντικά, μειώνοντας τον λόγο απόδοσης ανά κόστος. Τέλος, η χρήση πρόβλεψης διακλάδωσης με 1 bit, δείχνει να είναι πιο συμφέρουσα από αυτή των 2 bit, αλλά δεν ξεπερνά ούτε αυτή το κέρδος που έχουμε από την προσθήκη προώθησης δεδομένων λόγω του υψηλού της κόστους.

Συμπεραίνουμε ότι στις επόμενες υλοποιήσεις πρέπει να συμπεριλάβουμε εκτός της επίλυσης διακλαδώσεων στο στάδιο ID και την προώθηση κατά τον εντοπισμό κινδύνων δεδομένων.

**Case 6:** Delay slot & Branch resolution @ ID & Data hazard detection with forwarding:

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	Όχι
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	Όχι
Ανίχνευση κινδύνων δεδομένων και προώθηση	Ναι
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	Όχι
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	Ναι
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	Όχι
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	Όχι
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	Ναι

Dataset	Τελικό κόστος (ως συνάρτηση του βασικού K)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[Χρόνος * Κόστος]$ )
0	1,06 K	1,03 C	15.961	15.961	1,00000	16.439,83C	$5,738 * 10^{-5} * 1/(C * K)$
1	1,06 K	1,03 C	16.641	16.641	1,00000	17.140,23C	$5,504 * 10^{-5} * 1/(C * K)$
2	1,06 K	1,03 C	16.318	16.318	1,00000	16.807,54C	$5,613 * 10^{-5} * 1/(C * K)$
Average	1,06 K	1,03 C	16.306,67	16.306,67	1,00000	16.795,87C	$5,618 * 10^{-5} * 1/(C * K)$

**Case 7:** 1-bit Branch prediction & Branch resolution @ ID & Data hazard detection with forwarding:

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	Όχι
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	Όχι
Ανίχνευση κινδύνων δεδομένων και προώθηση	Ναι
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	Όχι
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	Όχι
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	Ναι
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	Όχι
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	Ναι

Dataset	Τελικό κόστος (ως συνάρτηση του βασικού K)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[Χρόνος * Κόστος]$ )
0	1,13 K	1,04 C	13.917	13.119	1,06083	14.473,68C	$6,114 * 10^{-5} * 1/(C * K)$
1	1,13 K	1,04 C	14.477	13.679	1,05834	15.056,08C	$5,878 * 10^{-5} * 1/(C * K)$
2	1,13 K	1,04 C	14.211	13.413	1,05949	14.779,44C	$5,988 * 10^{-5} * 1/(C * K)$
Average	1,13 K	1,04 C	14.201,67	13.403,67	1,05955	14.769,73C	$5,993 * 10^{-5} * 1/(C * K)$



**Case 8:** 2-bit Branch prediction & Branch resolution @ ID & Data hazard detection with forwarding:

Επιλογή παραμέτρων	Ναι/Όχι
Αγνόηση κινδύνων δεδομένων	Όχι
Ανίχνευση κινδύνων δεδομένων και καθυστέρηση	Όχι
Ανίχνευση κινδύνων δεδομένων και προώθηση	Ναι
Αγνόηση κινδύνων ελέγχου και χρήση καθυστερημένης διακλάδωσης	Όχι
Ανίχνευση κινδύνων ελέγχου και καθυστέρηση	Όχι
Πρόβλεψη διακλάδωσης του 1-bit και με χρήση 6-bit για τον BHT	Όχι
Πρόβλεψη διακλάδωσης του 2-bit και με χρήση 6-bit για τον BHT	Ναι
Επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX	Ναι

Dataset	Τελικό κόστος (ως συνάρτηση του βασικού K)	Τελικός χρόνος κύκλου ρολογιού (ως συνάρτηση του βασικού C)	Πλήθος κύκλων προγράμματος	Πλήθος εντολών προγράμματος	CPI	Χρόνος εκτέλεσης (Κύκλοι * Χρόνος κύκλου)	Απόδοση προς κόστος (δηλαδή $1/[Χρόνος * Κόστος]$ )
0	1,14 K	1,05 C	13.720	13.119	1,04581	14.406,00C	$6,089 * 10^{-5} * 1/(C * K)$
1	1,14 K	1,05 C	14.280	13.679	1,04394	14.994,00C	$5,850 * 10^{-5} * 1/(C * K)$
2	1,14 K	1,05 C	14.014	13.413	1,04481	14.714,70C	$5,961 * 10^{-5} * 1/(C * K)$
Average	1,14 K	1,05 C	14.004,67	13.403,67	1,04485	14.704,90C	$5,967 * 10^{-5} * 1/(C * K)$

Στις περιπτώσεις 6, 7 και 8 συγκρίναμε ποιο από τα εναπομείναντα κομμάτια του hardware είναι προτιμότερο να προσθέσουμε στην περίπτωση 2 (Case 2), ώστε να βελτιστοποιήσουμε την απόδοση του επεξεργαστή μας. Και οι 3 επιλογές οδήγησαν σε αύξηση της απόδοσης, που σημαίνει ότι η περίπτωση 2 έχει περιθώρια βελτίωσης. Και πάλι διαπιστώνουμε ότι το Delay slot, δεν επιφέρει τη μέγιστη επιτάχυνση, επομένως, παρά το χαμηλό κόστος της, δεν μπορεί να επιλεγεί. Ανάμεσα στις τελευταίες 2 υλοποιήσεις ωστόσο, παρατηρούμε ότι καλύτερο λόγο απόδοσης προς κόστος έχει αυτή με τον Branch predictor του 1 bit αντί των 2 bit. Η διαφορά μεταξύ τους είναι πολύ μικρή και έγκειται στο γεγονός ότι η δεύτερη είναι αρκετά πιο ακριβή για την επιτάχυνση που προσφέρει.

### Συμπεράσματα:

Με βάση τις παραπάνω δοκιμές καταλήγουμε στο συμπέρασμα ότι η υλοποίηση που έχει την καλύτερη απόδοση ανά μονάδα κόστους για το πρόγραμμά μας, είναι αυτή της περίπτωσης 7 (Case 7). Δηλαδή θα έχουμε έναν επεξεργαστή με ανίχνευση κινδύνων δεδομένων και προώθηση, πρόβλεψη διακλάδωσης του 1-bit με χρήση 6-bit για τον BHT και επίλυση των διακλαδώσεων στο στάδιο ID αντί του EX.

Η αρχιτεκτονική αυτή παρουσίασε λόγο απόδοσης προς κόστος  $5,993 * 10^{-5} * 1/(C * K)$  έναντι της αμέσως καλύτερης (Case 8) με αντίστοιχο λόγο  $5,967 * 10^{-5} * 1/(C * K)$ . Αυτό σημαίνει ότι η επιλογή όλου του διαθέσιμου hardware δεν αποτελεί πάντα την καλύτερη απόφαση. Παρ' όλα αυτά, το καλύτερο δυνατό CPI και ο μικρότερος χρόνος εκτέλεσης, επιτυγχάνονται αναμενόμενα στην τελευταία εκδοχή.

Σχόλιο: Δεδομένης της μικρής αυτής διαφοράς, θα μπορούσε κανείς να παραβλέψει το παραπάνω αποτέλεσμα αν το ζητούμενο είναι οι επιδόσεις του συστήματος, συνεπώς αφήνεται στην κρίση του κάθε σχεδιαστή ποια από τις δύο υλοποιήσεις θα διαλέξει.

Ευχαριστούμε για το χρόνο σας.