MIC-1 Simulator

Ο εξομοιωτής, βασισμένος στην JAVA, ονομάζεται mic1 και υλοποιεί την αρχιτεκτονική Mic-1 που περιγράφεται στο κεφάλαιο 4 του βιβλίου του A. S. Tanenbaum, Structured Computer Organization, 4η έκδοση. Το διαθέσιμο αυτό λογισμικό είναι σχεδιασμένο να υποστηρίξει τους εκπαιδευτές και τους σπουδαστές για να χρησιμοποιήσουν τη θεωρία.

Το mic1 είναι γραμμένο σε Java και απαιτεί το Java Development Kit (JDK) 1.0 ή νεότερο για να τρέξει. Η τεχνολογία Java διατίθεται από την SUN ή από τους τεχνικούς συνεργάτες της. Το site http://www.java.sun.com περιέχει τα απαραίτητα.

Ο microassembler του mic αξιοποιεί την γεννήτρια αναλύσεων για Java της CUP. Όλα τα άλλα προγράμματα και module είναι γραμμένα κατευθείαν σε Java και είναι συμβατά με το JDK 10 ή νεότερο.

Το mic1 διανέμεται με τους όρους της GNU General Public License. Ο πηγαίος κώδικας περιέχεται στην έκδοση.

Το πακέτο της έκδοσης του mic1 περιλαμβάνει τα ακόλουθα:

- Εγχειρίδιο χρήσης του mic1
- Σημειώσεις για την έκδοση
- Συχνές Ερωτήσεις
- Προδιαγραφές για IJVM Assembly Language
- Προδιαγραφές για Micro-Assembly Language (MAL)
- Σελίδα οδηγιών για το mic1sim
- Σελίδα οδηγιών για το mic1asm
- Σελίδα οδηγιών για το ijvmasm

Το λογισμικό mic1 περιλαμβάνει:

- Έναν απλό Mic-1 simulator (mic1sim) που επιτρέπει στον χρήστη την εκτέλεση ενός Mic-1 μικροπρογράμματος.
- Έναν απλό Mic-1 assembler (mic1asm) που μεταφράζει ένα αρχείο κειμένου που περιέχει γλώσσα Mic-1 microassembly (MAL) σε δυαδικό αρχείο (binary) Mic-1 μικροοδηγιών (microinstructions), έτοιμο να φορτώνεται στον mic1 simulator control store.
- Ένα δείγμα Mic-1 μικροπρογράμματος που ερμηνεύει ένα τροποποιημένο (μόνο ακέραιο) υποσύνολο της αρχιτεκτονικής συνόλου οδηγιών (ISA) της Java Virtual Machine (JVM) της SUN (το οποίο αναφέρεται ως IJVM).
- Ένα δείγμα IJVM assembler (ijvmasm), το οποίο μεταφράζει ένα αρχείο κειμένου που περιέχει IJVM σε δυαδικό (binary) αρχείο IJVM οδηγιών, έτοιμο να φορτωθεί στο κύριο πρόγραμμα του εξομοιωτή mic1.
- Ένα δείγμα IJVM προγράμματος, το οποίο μπορεί να χρησιμοποιηθεί για να επιδείξει τη λειτουργία του μικροπρογραμματισμού mic1, με τη βοήθεια του εξομοιωτή mic1.

Μερικά πιθανά project (με αύξουσα σειρά πολυπλοκότητας)

- Γράψτε ένα πρόγραμμα σε IJVM. Αποδείξτε ότι το πρόγραμμά σας λειτουργεί σωστά χρησιμοποιώντας τον IJVM assembler και τον εξομοιωτή Mic-1.
- Προσθέστε ένα νέο χαρακτηριστικό στον IJVM assembler.
- Προσθέστε μια νέα οδηγία στο IJVM και εφαρμόστε την ως προέκταση στο Mic-1 IJVM μικροπρόγραμμα. Τροποποιήστε τον IJVM assembler ώστε να αναγνωρίζει και να παράγει σωστά κώδικα για τη νέα σας οδηγία. Αποδείξτε ότι το νέο σας Mic-1 μικροπρόγραμμα ερμηνεύει σωστά ένα κατάλληλο πρόγραμμα ελέγχου που περιέχει τη νέα σας οδηγία.
- Γράψτε ένα μικροπρόγραμμα σε Mic-1 microassembly γλώσσα.
- Προσθέστε νέο χαρακτηριστικό στον Mic-1 microassembler.
- Προσθέστε μια νέα καταχώρηση στην αρχιτεκτονική Mic-1.
- Υλοποιήστε έναν εξομοιωτή για την αρχιτεκτονική Mic-2
- Υλοποιήστε έναν εξομοιωτή για την αρχιτεκτονική Mic-3
- Υλοποιήστε έναν εξομοιωτή για την αρχιτεκτονική Mic-4

Ενώ αρκετά από τα παραπάνω project υποστηρίζουν το υλικό που καλύπτει το κεφάλαιο 4, Το Επίπεδο Μικροαρχιτεκτονικής, υπάρχουν αρκετά που είναι εφαρμόσιμα και σε άλλα τμήματα του βιβλίου. Για παράδειγμα, υπάρχει ένας αριθμός από αυτά που δημιουργούν σκέψεις που παρουσιάζονται στο κεφάλαιο 7, Το Επίπεδο Γλώσσας Assembly.

Εγκατάσταση του mic1

- Αρχικά δημιουργούμε έναν κατάλογο mic1 στον σκληρό δίσκο.
- Κατεβάζουμε το αρχείο **mic1win.exe** και το τοποθετούμε στον κατάλογο mic1. Το αρχείο αυτό θα το βρούμε με link από το site http://www.ontko.com/mic1/.
- Εκτελούμε το αρχείο mic1win.exe και αυτομάτως αποσυμπιέζει και εγκαθιστά όλα τα αρχεία (προγράμματα, πηγαίο κώδικα, τεκμηρίωση) του εξομοιωτή mic1 μέσα στο συγκεκριμένο κατάλογο.

Πριν εκτελέσουμε οποιοδήποτε από τα προγράμματα του mic1, πρέπει να τροποποιήσουμε το αρχείο env.bat, ως εξής:

Α) Προσθέτουμε REM στις εξής γραμμές

```
echo   NOTE: YOU NEED TO EDIT THE FILE ENV.BAT BEFORE YOUR mic1
echo   SOFTWARE will WORK CORRECTLY.
pause
goto end
```

ώστε να γίνουν έτσι:

```
REM echo   NOTE: YOU NEED TO EDIT THE FILE ENV.BAT BEFORE YOUR mic1
REM echo   SOFTWARE will WORK CORRECTLY.
REM pause
REM goto end
```

Β) Μεταβάλλουμε την αναφορά του path, κάτω στο βήμα 2, ώστε ο κατάλογος που αναφέρεται εκεί να δείχνει στον κατάλογο bin του Java Development Kit (σημείωση: αφήνουμε το τμήμα ;%path% στο τέλος της αναφοράς του path). Δηλαδή, μέσα στο αρχείο env.bat η γραμμή

path C:\jdk1.2\bin;%path%

γίνεται τώρα

path C:\java\bin;%path%

αν το JDK έχει εγκατασταθεί στον κατάλογο c:\java
(Σημείωση: Για την εγκατάσταση του JDK ανατρέξτε στο τέλος του εγγράφου)

Γ) Μεταβάλλουμε την αναφορά CLASSPATH στο βήμα 3 ώστε να δείχνει το αρχείο classes.zip στον κατάλογο που είναι εγκατεστημένο το mic1. Δηλαδή, μέσα στο αρχείο env.bat η γραμμή

set CLASSPATH=C:\mic1\classes.zip

παραμένει όπως είναι

Έλεγχος της εγκατάστασης

Μπορούμε να ελέγξουμε αν όλα τα τμήματα της εγκατάστασης λειτουργούν σωστά με το να συναρμολογήσουμε (assembling) και να τρέξουμε το πρόγραμμα ijvmtest. Αυτό γίνεται ως εξής:

Συναρμολόγηση του προγράμματος ijvmtest

- Εκτελούμε το αρχείο ijvmasm.bat με διπλό κλικ.
- Ανοίγει ένα παράθυρο όπου πρέπει να εισάγουμε δύο ονόματα αρχείων. Δίνουμε ijvmtest.jas ως το αρχείο εισαγωγής, και το ijvmtest.ijvm ως το αρχείο εξαγωγής.
- Πατάμε το πλήκτρο compile.

Εκτέλεση του εξομοιωτή Mic-1, mic1sim

- Εκτελούμε το αρχείο mic1sim.bat με διπλό κλικ.
- Από το μενού File, επιλέγουμε ”Load Microprogram” και δίνουμε το αρχείο mic1ijvm.mic1 ως το μικροπρόγραμμα που θα φορτωθεί.
- Πάλι από το μενού File, επιλέγουμε ”Load Macroprogram” και δίνουμε το αρχείο ijvmtest.ijvm ως το μακροπρόγραμμα που θα φορτωθεί.
- Κάνουμε κλικ στο πλήκτρο RUN για να ξεκινήσει η μετάφραση του μακροπρογράμματος από το μικροπρόγραμμα. Μετά από μια σύντομη περίοδο, όσο ο εξομοιωτής τρέχει, θα πρέπει να εμφανιστούν οι ακόλουθες λέξεις στην περιοχή “Standard out”:

OK
End of run.

Βήμα προς βήμα μετάφραση ενός προγράμματος σε γλώσσα Assembly από ένα μικροπρόγραμμα.

Το πρόγραμμα mic1sim είναι ένα java πρόγραμμα με γραφικό περιβάλλον αλληλεπίδρασης με το χρήστη, που επιτρέπει να παρατηρούμε την μετάφραση ενός Προγράμματος Επιπέδου Αρχιτεκτονικής Συνόλου Οδηγιών (Instruction Set Architecture (ISA) – Level Program), από ένα πρόγραμμα Επιπέδου Μικροαρχιτεκτονικής.

Στο παράδειγμα που ακολουθεί, χρησιμοποιούμε τον ijvmasm assembler για να παράγουμε ένα πρόγραμμα επιπέδου ISA από πηγαίο κώδικα, και τον mic1asm microassembler για να παράγουμε ένα πρόγραμμα επιπέδου μικροαρχιτεκτονικής από πηγαίο κώδικα.

1] Συναρμολογούμε (assemble) ένα δείγμα IJVM πρόγραμμα

- Στην προτροπή του συστήματος γράφουμε ijvmasm.bat
- Η ενοτλή αυτή κάνει τον IJVM Assembler (ijvmasm) να διαβάσει το αρχείο κειμένου echo.jas (που περιέχει τον πηγαίο κώδικα μας σε IJVM assembly γλώσσα) και να παράγει ένα δυαδικό (binary) αρχείο echo.ijvm (που περιέχει την εκτελέσιμη έκδοση του ISA-Level κώδικά μας).

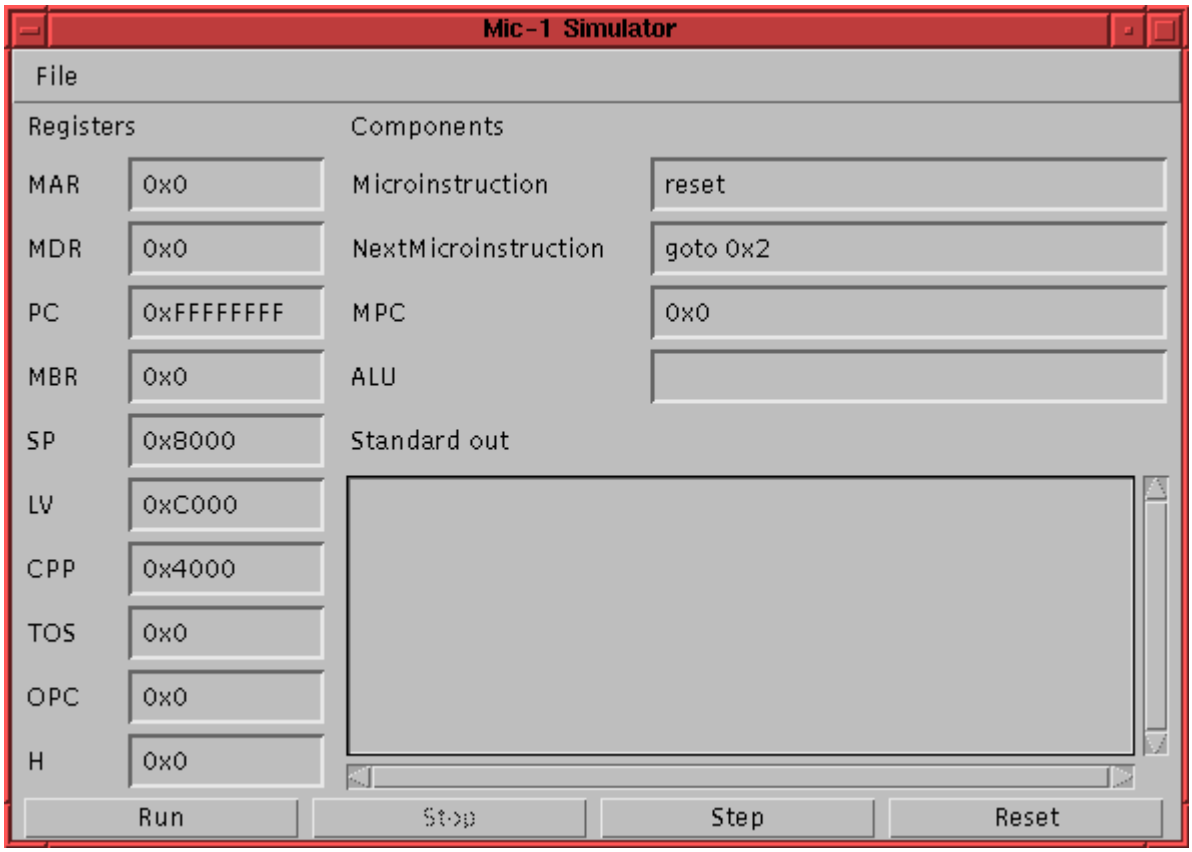
2] Συναρμολογούμε ένα μικροπρόγραμμα

- Στην προτροπή του συστήματος γράφουμε mic1asm.bat
- Η εντολή αυτή κάνει τον Microassembler του Mic-1 (mic1asm) να διαβάσει το αρχείο κειμένου mic1ijvm.mal (που περιέχει τον πηγαίο κώδικά μας σε γλώσσα Micro Assembly για έναν απλοποιημένο διερμηνέα Java Virtual Machine ώστε να εκτελείτε στην αρχιτεκτονική Mic1) και να παράγει ένα δυαδικό (binary) αρχείο mic1ijvm.mic1 (που περιέχει τα πραγματικά δεδομένα, π.χ. μικροκώδικας, που πρόκειται να φορτωθεί στο control store μιας μηχανής που υλοποιεί την αρχιτεκτονική Mic1).

3] Τρέχουμε τον εξομοιωτή

- Στην προτροπή του συστήματος γράφουμε mic1sim.bat

Η εντολή αυτή θα εμφανίσει το ακόλουθο παράθυρο:



- Εδώ θα πρέπει να φορτώσουμε το μικροπρόγραμμα και το μακροπρόγραμμα, επιλέγοντας από το μενού File “Load Microprogram” και “Load Macroprogram”.

- Στη συνέχεια μπορούμε να πατήσουμε το πλήκτρο STEP για να εκτελέσουμε το μικροπρόγραμμα μικροεντολή προς μικροεντολή, ή μπορούμε να πατήσουμε τα πλήκτρα RUN και STOP για να ξεκινάμε και να αναστέλλουμε την εκτέλεση του μικροπρογράμματος (το οποίο φυσικά διερμηνεύει το ISA – Level πρόγραμμά μας). Το πλήκτρο RESET, επαναφέρει τους καταχωρητές του Mic1 στις αρχικές τους συνθήκες (αλλά όχι τη μνήμη).

4] Πατάμε το πλήκτρο RUN

- Όσο διερμηνεύουμε το πρόγραμμα echo.ijvm, θα παρατηρήσουμε ότι αν πατήσουμε το πλήκτρο RUN, οτιδήποτε που πληκτρολογούμε θα εμφανίζεται στο πεδίο κειμένου με τίτλο “standard output”.

5] Πατάμε το πλήκτρο STOP και μετά το πλήκτρο RESET.

6] Πατάμε το πλήκτρο STEP αρκετές φορές, αργά, και παρακολουθούμε προσεκτικά τι συμβαίνει. Προσέχουμε πως εμφανίζονται οι αναφορές του microassembly προγράμματος στο πεδίο μικροοδηγιών, και πως αλλάζουν οι τιμές των καταχωρητών κάθε φορά που πατάμε το πλήκτρο STEP. Μετά από προσεκτική εξέταση των οδηγιών και των καταχωρητών, θα πρέπει να είμαστε ικανοί να προβλέπουμε τι αλλαγές θα γίνονται σε κάθε καταχωρητή, βασιζόμενοι στη γνώση μας για την μικροεντολή που πρόκειται να εκτελεστεί.

Γράφοντας και δοκιμάζοντας ένα πρόγραμμα γλώσσας Assembly σε IJVM.

Μπορούμε να χρησιμοποιήσουμε τον IJVM Assembler, ijvmasm, για να εξασκηθούμε στο γράψιμο προγραμμάτων σε γλώσσα assembly, τα οποία στη συνέχεια μπορούμε να τα ελέγξουμε στον εξομοιωτή Mic1 (mic1sim). Επεξηγήσεις για τη δομή ενός προγράμματος IJVM υπάρχουν στο τμήμα IJVM Assembly Language Specifications, που υπάρχει στην εγκατάσταση.

1] Σε ένα οποιοδήποτε επεξεργαστή κειμένου, δημιουργούμε ένα αρχείο κειμένου που θα περιέχει το δικό μας IJVM πρόγραμμα και το αποθηκεύουμε με την κατάληξη **.jas**, που δηλώνει ότι πρόκειται για ένα αρχείο σε γλώσσα Java Assembly. Μπορούμε να πάρουμε ως βάση ένα αντίγραφο των αρχείων echo.jas ή add.jas.

2] Τρέχουμε τον IJVM assembler:

java ijvmasm input-filename [output-filename]

π.χ.

java ijvmasm my_echo.jas my_echo.ijvm

3] Ελέγχουμε το πρόγραμμά μας με τον εξομοιωτή Mic1:

java mic1sim mic1ijvm.mic1 macroprogram

π.χ.

java mic1sim mic1ijvm.mic1 my_echo.ijvm

επαναλαμβάνουμε την παραπάνω διαδικασία όσο ελέγχουμε και αποσφαλματώνουμε το IJVM πρόγραμμά μας.

Γράφοντας και Ελέγχοντας Μικροπρογράμματα για την Αρχιτεκτονική Mic1

Μπορούμε να χρησιμοποιήσουμε τον Mic1 microassembler, mic1asm, για να γράψουμε τα δικά μας μικροπρογράμματα για την αρχιτεκτονική Mic1.

1] Με οποιοδήποτε επεξεργαστή κειμένου, δημιουργούμε ένα αρχείο κειμένου που περιέχει το κείμενο σε Micro Assembly γλώσσα (MAL) για το μικροπρόγραμμά μας. Αποθηκεύουμε το αρχείο με κατάληξη **.mal** που δηλώνει ότι πρόκειται για αρχείο γλώσσας Micro Assembly. Μπορούμε να χρησιμοποιήσουμε ένα αντίγραφο του αρχείου mic1ijvm.mal ως παράδειγμα.

2] Τρέχουμε τον Mic1 Micro-Assembly Language assembler:

java mic1asm input-filename output-filename

π.χ.

java mic1asm my_ijvm.mal my_ijvm.mic1

3] Ελέγχουμε το μικροπρόγραμμά μας με τον εξομοιωτή Mic1:

java mic1sim microprogram macroprogram

π.χ.

java mic1sim my_ijvm.mic1 echo.ijvm

Επαναλαμβάνουμε τα παραπάνω βήματα καθώς τροποποιούμε, ελέγχουμε και αποσφάλματώνουμε το IJVM πρόγραμμά μας.

Σημείωση: Αν προσθέσουμε νέες οδηγίες στο IJVM, ή εφαρμόσουμε μια μικροπρογραμματισμένη διερμηνεία για μια τελείως διαφορετική γλώσσα ISA-Level, θα πρέπει να τροποποιήσουμε ή να δημιουργήσουμε έναν νέο assembler για τη γλώσσα αυτή. Αυτό περιγράφεται σύντομα στην επόμενη ενότητα.

Τροποποιώντας και Ελέγχοντας έναν Assembler

Ίσως κάποτε θελήσουμε να τροποποιήσουμε τη λειτουργικότητα ενός IJVM assembler, ή πιθανώς να δημιουργήσουμε ένα ISA-Level Language Assembler από μόνοι μας. Προσθέτοντας νέες οδηγίες που είναι αντίστοιχες με αυτές που ήδη υπάρχουν, είναι στην πραγματικότητα αρκετά εύκολο. Θα πρέπει να είμαστε σε θέση απλά να τροποποιήσουμε το αρχείο ijvm.conf (φυσικά θα χρειαστούμε να υλοποιήσουμε κάθε νέα οδηγία στο μικροπρόγραμμα, αλλά η ουσία εδώ είναι πως η προσθήκη νέας οδηγίας, γενικά, δεν απαιτεί την αλλαγή στον IJVM Assembler). Αν θέλουμε να προσθέσουμε ένα νέο τύπο οδηγιών, ή να προσθέσουμε μια ψευδο-οδηγία, ή αλλιώς να αλλάξουμε τη συμπεριφορά του assembler, θα πρέπει να τον τροποποιήσουμε. Μπορούμε να δούμε τον πηγαίο κώδικα του ijvmasm στον κατάλογο source που δημιουργήθηκε όταν εγκαταστήσαμε το πακέτο του εξομοιωτή.

1] Με οποιοδήποτε επεξεργαστή κειμένου, τροποποιούμε το αρχείο ijvmasm.java ή τα σχετικά class αρχεία του κατάλληλα.

2] Μεταγλωττίζουμε το τροποποιημένο ijvmasm πρόγραμμα και τα σχετικά του Java classes αρχεία:

```
java –depend ijvmasm
```

3] Ελέγχουμε το μεταγλωτισμένο assembler χρησιμοποιώντας ένα βολικό αρχείο ελέγχου:

```
java ijvmasm my_text.jas my_test.ijvm
```

4] Ελέγχουμε το αρχείο εξόδου από τον δικό μας assembler.
Προφανώς, μπορούμε να ελέγξουμε αυτό το αρχείο χρησιμοποιώντας τον εξομοιωτή mic1sim και τον διερμηνέα mic1ijvm. Μπορούμε ακόμα να βρούμε χρήσιμο να επιθεωρήσουμε οπτικά την έξοδο από τον assembler χρησιμοποιώντας το δικό μας Java-based πρόγραμμα dump. Για παράδειγμα:

```
Java dump my_test.ijvm
```

Αυτό παράγει μια λίστα τριών στηλών, κάθε γραμμή της οποίας είναι για κάθε byte του αρχείου. Η πρώτη στήλη είναι το δεκαδικό offset (π.χ. η διεύθυνση) του byte, η δεύτερη στήλη είναι το δεκαεξαδική τιμή της τοποθεσίας, και η τρίτη είναι η δεκαδική τιμή της τοποθεσίας.

Το λειτουργικό μας σύστημα μπορεί επίσης να παρέχει μια πιο ισχυρή εφαρμογή για αυτό τον σκοπό.

Επαναλαμβάνουμε τα παραπάνω βήματα καθώς τροποποιούμε, ελέγχουμε και αποσφαλματώνουμε τον assembler μας.

Τροποποιώντας και Ελέγχοντας ένας Microassembler

Ίσως κάποτε θελήσουμε να παρέχουμε πρόσθετη λειτουργικότητα στον microassembler, mic1asm. Για παράδειγμα, ίσως θέλουμε να έχουμε πρόσθετες ψευδο-οδηγίες, ή να αλλάξουμε τη σύνταξη του κώδικα Java στις μικροοδηγίες.

Το mic1asm είναι γραμμένο σε Java, και αρκετή από τη σημασιολογία του περιέχεται στο πηγαίο αρχείο mic1asm.java. Ωστόσο, ο αναλυτής της Micro Assembly Language (MAL) είναι γραμμένος με πρόγραμμα τύπου yacc που λέγεται "CUP Parser Generator for Java ™". Αυτό παράγει έναν Java LALR αναλυτή με εμπεδωμένες δράσεις βασιζόμενες σε λεπτομερείς παρουσιάσεις που δίνονται στο απλό αρχείο κειμένου Mic1Parser.cup. Αν αλλάξουμε τη σύνταξη ή κάποιες σημασιολογίες του mic1asm, θα πρέπει να τροποποιήσουμε αυτό το αρχείο.

1] Κατεβάζουμε και εγκαθιστούμε στον CUP Parser Generator for Java™.
Το λογισμικό είναι διαθέσιμο στη διεύθυνση:
http://www.cs.princeton.edu/~appel/modern/java/CUP/
Σιγουρευόμαστε να προσθέσουμε τον κατάλογο του CUP (αυτόν που περιέχει τον java_cup κατάλογο) στο CLASSPATH.
setenv CLASSPATH=.:.\cup

2] Με οποιονδήποτε επεξεργαστή κειμένου, τροποποιούμε τον κατάλληλο πηγαιό κώδικα java (.java αρχεία), ή το αρχείο ορισμών του αναλυτή (parser) Mic1Parser.cup.

3] Δημιουργούμε το parser class:

java java_cup.Main –parser Mic1Parser –symbol Mic1Symbol < Mic1Parser.cup

Αυτό υλοποιεί το class Main στον κατάλογο source/java_cup, παράγει έναν αναλυτή (parser) που λέγεται Mic1Parser, χρησιμοποιώντας ένας λιξιλογικό σαρωτή που παράγει σύμβολα από την κλάση Mic1Symbol, και διαβάζει τις εισροές από το Mic1Parser.cup

4] Μεταγλωττίζουμε τον Mic1 microassembler mic1asm:

javac –depend mic1asm.java

5] Ελέγχουμε τον μεταγλωττισμένο microassembler χρησιμοποιώντας κατάλληλο αρχείο ελέγχου:

java mic1asm my_test.mal my_test.mic1

6] Ελέγχουμε το αρχείο εξόδου του microassembler.
Προφανώς, μπορούμε να το ελέγξουμε αυτό χρησιμοποιώντας τον εξομοιωτή mic1sim και κάποιο βολικό πρόγραμμα ελέγχου. Επίσης, είναι αρκετά χρήσιμο να κάνουμε οπτική επιθεώρηση της εξόδου του assembler με το Java-based πρόγραμμα dump.

Άλλο ένα εργαλείο που περιέχεται στο mic1 και το οποίο ίσως φανεί χρήσιμο είναι ο java-based mic1 disassembler, mic1dasm. Για παράδειγμα:

Java mic1dasm my_test.mic1

Αυτό διαβάζει ένα δυαδικό (binary) αρχείο που περιέχεται στο control store (που παράγεται πιθανώς από το mic1asm) και παράγει μια λίστα που δείχνει την αποκωδικοποιημένη οδηγία σε κάθε τοποθεσία του 512 λέξεων control store.

Επαναλαμβάνουμε τα σχετικά βήματα της παραπάνω διαδικασίας καθώς τροποποιούμε, ελέγχουμε και αποσφαλματώνουμε τον δικό μας microassembler.

Τροποποιώντας και Ελέγχοντας έναν Εξομοιωτή Μικροαρχιτεκτονικής

Το να προσθέσουμε χαρακτηριστικά στον εξομοιωτή Mic1 είναι κυρίως μιας εξάσκηση στον προγραμματισμό σε Java, με μια ελαφριά έμφαση στην κατανόηση της αλληλεπίδρασης μεταξύ της διασύνδεσης του χρήστη και των θεμελιωδών συστατικών της αρχιτεκτονικής του εξομοιωτή mic1sim.

Διαδικασία:

1] Με οποιονδήποτε επεξεργαστή κειμένου, τροποποιούμε το mic1sim.java ή άλλα σχετικά με αυτό java classes.
2] Μεταγλωττίζουμε το mic1sim πρόγραμμα:
java –depend mic1sim.java
3] Ελέγχουμε τον τροποποιημένο εξομοιωτή Mic1.

Επαναλαμβάνουμε τα παραπάνω βήματα της τροποποίησης, ελέγχου και αποσφαλμάτωσης όπως πρέπει.

Υλοποιώντας ένας εξομοιωτή για μια τροποποιημένη Μικροαρχιτεκτονική

Κυρίως, αυτό περιλαμβάνει την τροποποίηση της διαμόρφωσης των καθοριστικών συστατικών που χρησιμοποιεί ο mic1sim εξομοιωτής και πως αυτά χειρίζονται στον κύριο βρόχο επεξεργασίας του mic1sim εξομοιωτή.

Διαδικασία:

1] Με οποιονδήποτε επεξεργαστή κειμένου, τροποποιούμε το αρχείο mic1sim.java ή άλλα σχετικά με αυτό java classes.
2] Μεταγλωττίζουμε το πρόγραμμα mic1sim:
java –depend mic1sim.java
3] Ελέγχουμε τον τροποποιημένο Mic1 εξομοιωτή.

Επαναλαμβάνουμε τα παραπάνω βήματα της τροποποίησης, ελέγχου και αποσφαλμάτωσης, όπως πρέπει.

ΠΑΡΑΡΤΗΜΑ

Αρχεία

.bat	Δέσμη εντολών για εκτέλεση από το λειτουργικό σύστημα
.conf	Αρχείο κειμένου με πληροφορίες διαμόρφωσης ενός προγράμματος
.ijvm	Δυαδικό αρχείο γλώσσας Integer Java Virtual Machine, αντικειμενικός κώδικας IJVM
.jas	Αρχείο κειμένου γλώσσας Java Assembly, πηγαίος IJVM κώδικας
.mal	Αρχείο κειμένου γλώσσας Micro Assembly, πηγαίος κώδικας για μικρο-πρόγραμμα
.mic1	Δυαδικό αρχείο μικροπρογράμματος αρχιτεκτονικής Mic1, πηγαίος κώδικας

Δείγματα Προγραμμάτων

- **ljvmtest.jas**, ένα πρόγραμμα που ελέγχει όλα τα χαρακτηριστικά της αρχιτεκτονικής mic1.
- **Echo.jas**, ένα πρόγραμμα που δέχεται είσοδο από το πληκτρολόγιο και τυπώνει κάθε πάτημα πλήκτρου σε μια στάνταρ περιοχή κειμένου.
- **Add.jas,** ένα πρόγραμμα που δέχεται δύο δεκαεξαδικούς αριθμούς από το πληκτρολόγιο και απεικονίζει το σύνολό τους.

Πηγαίος Κώδικας

Ο πηγαίος κώδικας (σε java) για όλα τα προγράμματα java βρίσκεται στον κατάλογο source που δημιουργήθηκε μετά την εγκατάσταση του mic1.

Εργασίες για τους φοιτητές

Οι φοιτητές μπορούν να χρησιμοποιήσουν αυτά τα προγράμματα, τον δικό τους πηγαίο κώδικα, ή άλλα δείγματα προγραμμάτων, ώστε να μπορούν να:

- Παρατηρούν την βήμα προς βήμα μετάφραση ενός προγράμματος σε γλώσσα assembly από ένα μικροπρόγραμμα.
- Γράφουν και ελέγχουν προγράμματα γλώσσας assembly σε IJVM.
- Γράφουν και ελέγχουν μικροπρογράμματα για την αρχιτεκτονική Mic1.
- Τροποποιούν και ελέγχουν έναν assembler.
- Τροποποιούν και ελέγχουν έναν microassembler.
- Τροποποιούν και ελέγχουν έναν εξομοιωτή μικροαρχιτεκτονικής.
- Υλοποιούν έναν εξομοιωτή για τροποποιημένη μικροαρχιτεκτονική.

ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ JAVA DEVELOPER'S KIT (JDK) release 1.0.2

- Από τη διεύθυνση ftp://ftp.javasoft.com/pub/jdk1.0.2/JDK-1_0_2-win32-x86.exe κατεβάζουμε το αρχείο JDK-1_0_2-win32-x86.exe και το αποθηκεύουμε στον σκληρό δίσκο στη θέση c:\
- Στον υπολογιστή μας δεν θα πρέπει να υπάρχει καμιά άλλη προηγούμενη έκδοση του JDK. Αν υπάρχει πρέπει να την διαγράψουμε.
- Εκτελούμε το αρχείο που κατεβάσαμε, ώστε να αποσυμπιεστούν τα δεδομένα που περιέχει. Θα δημιουργηθεί κατάλογος c:\java, όπου εκεί θα μπουν όλα τα απαραίτητα αρχεία.

Μετά την αποσυμπίεση θα πρέπει να κάνουμε τις παρακάτω ρυθμίσεις περιβάλλοντος:

- Η αναφορά Path πρέπει να δείχνει στον κατάλογο c:\java\bin. Αυτό μπορεί εύκολα να γίνει αν τροποποιήσουμε το αρχείο Autoexec.bat του υπολογιστή μας.
- Αν έχουμε ρυθμίσει τη μεταβλητή περιβάλλοντος CLASSPATH, τότε θα πρέπει να την ανανεώσουμε, αντικαθιστώντας τις καταχωρήσεις με τη διαδρομή c:\java/lib/classes.zip.

Μετά από τα παραπάνω, ο υπολογιστής μας είναι έτοιμος να χρησιμοποιήσει το Java Developer's Kit.

Ξεκινάμε τον Applet Viewer ως εξής:

cd java/demo/TicTacToe
appletviewer example1.html

Περισσότερα στο site
http://www.java.sun.com/products/jdk/1.0.2/installation-win32-x86.html