תיעוד פרויקט מבנה המחשב

תאריך: 28/02/2024

שמות המגישים:

- אור שאול

הפרויקט מורכב משלושה חלקים:

- אסמבלר
- סימולטור
- 4 תוכניות בדיקה

אסמבלר

מטרת האסמבלר הינה לתרגם קוד שכתוב בשפת אסמבלי לשפת מכונה, בהתאם לפונקציות אשר ניתנו בהוראות. הוראות האסמבלי ישמרו בתחילת מערך הזיכרון להוראות. בנוסף, כאשר האסמבלר מגיע לפסאודו הוראה (פקודת word.) המידע ישמר בכתובת המתאימה בזיכרון. קבצי הפלט של האסמבלר הם: imemin.txt , dmemin.txt בפורמט הקסדצימלי. קבצים אלו הם למעשה גם קבצי קלט של הסימולטור.

פירוט אופן פעולת האסמבלר:

- פתיחת כלל הקבצים אשר נעשה בהם שימוש בתוכנית לקריאה ולכתיבה, תוך וידוא כי כלל הקבצים נפתחים באופן תקין.
- תחילה נעבור בפעם הראשונה על הקובץ על מנת לשמור את כל ה-labels במערך ייעודי אשר הוגדר עבורם. כל label יישמר בעזרת ה struct ייחודי.
- נסגור את הקבצים ונפתח מחדש לצורך הריצה השנייה. בריצה השנייה נשמור את כל ההוראות במערך inst_array כאשר כל הוראה תשמר בעזרת struct שהוגדר עבורה. בנוסף, את הפקודות שהתקבלו בהוראות ה word. נשמור במערך Dmem_array כאשר כל מילה תשמר בעזרת struct מתאים שהגדרנו.

- נעבור על מערך ההוראות שבנינו, ונעדכן את השדות imm1, imm2 של ההוראות המתאימות להיות כתובת ה-label שנשמרה במערך ה- labels .
- נכתוב את תוכן מערך ההוראות לתוך הקובץ imemin.txt .
- נכתוב את תוכן מערך המילים Dmem_array לתוך הקובץ dmemin.txt .

<u>מקרי קצה:</u>

- במקרה ולא כתבנו לרגיסטר קיים (לפי הרגיסטרים שהגדרנו באסמבלר), ברירת המחדל היא רגיסטר האפס.

סימולטור

הסימולטור מקבל כקלט את קבצי האסמבלר: imemin.txt , dmemin.txt בפורמט הקסדצימלי. כמו כן, מקבל גם irq2in.txt (קובץ פסיקות חיצוני) ו-diskin.txt (תוכן הדיסק, כרכיב IO). מטרת הסימולטור הינה לפענח בכל איטרציה בלולאת ה-while את ההוראות הנתונות בקובץ imemin.txt, ולאחר מכן לבצע את ההוראה. לכל פונקציה אשר נתמכת ע״י המעבד יש אופקוד מסוים שנקבע לפי הוראות הפרויקט. ערך PC ההתחלתי הינו 0 ובסיום כל הוראה מעדכנים את ערך PC להיות PC+1 , אלא אם כן ביצענו הוראת קפיצה או שקיבלנו פסיקה. סיום הריצה ויציאה מהסימולטור תתבצע בעת הגעה להוראה HALT. למעשה קובץ ה-dmemout יכיל dmemin בתחילת התוכנית ולפני הלולאה. נדאג לכך באמצעות מערך ייעודי Data_memory_array ולאחר מכן במהלך ריצת הקוד עבור הוראות sw/lw נעדכן או נקרא ממערך זה. בסוף ריצת הלולאה נדפיס את איברי המערך לקובץ dmemout.

<u>פירוט אופן פעולת הסימולטור:</u>

- פתיחת כלל הקבצים אשר נעשה בהם שימוש בתוכנית לקריאה ולכתיבה, תוך וידוא כי כלל הקבצים נפתחים באופן תקין.
- נשמור את קובץ הקלט imemin במערך ייעודי וכך נעשה גם עבור dmemin.
- הסימולטור רץ בלולאת while כך שנבצע עבור כל פקודה שמתקבלת מ-imemin את שלושת השלבים הבאים:
  - Fetch - נייבא את ההוראה הבאה ממערך inst_array לפי ה- PC.
  - Decode - נפענח את ההוראה ונמלא את שדות הרגיסטרים בהתאם.
  - Execute - נבצע את ההוראה לפי אופקוד מתאים.
- בסוף הלולאה נדפיס את עיקר הקבצים. יש לציין כי חלק מהקבצים יודפסו כבר במהלך ריצת הלולאה: leds.txt , display7seg, hwregtrace, monitor, diskout .

<u>מקרי קצה:</u>

- במקרה של קריאה, כתיבה או קפיצה למקום החורג מגבולות מערך הזיכרון, נבצע מודולו על הכתובת המתקבלת עם גודל מערך הזיכרון, כלומר הזיכרון ציקלי.
- במקרה שבאחת משורות ה imemin -קיים opcode לא תקין (לא בטווח שבין 0-21) הסימולטור לא יבצע את ההוראה ויתעלם ממנה.

קבצי הפלט המתקבלים מהסימולטור:

- dmemout – מכיל את תוכן הזיכרון הראשי בסיום הריצה.
- Regout – מכיל תוכן הרגיסטרים בסיום הריצה.
- Trace – משמש למעקב אחר תוכן הרגיסטרים. כל הוראה, טרם בוצעה, מתועדת בקובץ זה, ומכילה את כתובת ההוראה (PC), את ההוראה עצמה ואת תוכן כל הרגיסטרים.
- Cycles – מכיל את מספר מחזורי השעון שרצה התוכנית.
- hwregtrace – קובץ טקסט זה מציין האם כתבנו או קראנו אל/מ רגיסטר חומרה, מהו רגיסטר החומרה, ומחזור השעון בו התבצעה הפעולה.
- leds – קובץ טקסט זה מכיל בכל שורה את מחזור השעון בו עדכנו את הלדים ומהו מערך הלדים המעודכן.
- display7seg –קובץ טקסט זה מכיל בכל שורה את מחזור השעון בו עדכנו את ה- 7seg ומהו ה-7seg המעודכן.
- diskout – קובץ המציג את תוכן הדיסק בסוף ריצת הסימולטור.
- Monitor – קובץ המציג את ערכי הפיקסלים שבמוניטור.
- Monitor.yuv – כמו הקובץ Monitor אך בהצגה בינארית. באמצעות התוכנה (שקישור אליה צורף במסמך הפרויקט) נציג את תוכן המוניטור, ובמקרה של ה-circle.asm נקבל עיגול לבן מלא.

קבצי בדיקה

Mulmat

התוכנית מקבלת שתי מטריצות ומבצעת מכפלה ביניהן. נרוץ ב3 לולאות כאשר הראשונה רצה על השורות, השנייה רצה על העמודות והשלישית דואגת לסכימה של המכפלות הפנימיות של השורה הנוכחית בעמודה הנוכחית. את המכפלה המצטברת נשמור ברגיסטר $v0.

Binom

קוד אסמבלי זה מממש את תוכנית הבדיקה binom בשפת המעבד הנתונה. הקוד מחשב את הבינום של ניוטון בצורה רקורסיבית. תנאי העצירה מתקיים כאשר n==k או כאשר k==0. תנאי העצירה מומש בקוד האסמבלי בעזרת פקודות beq לאחר מכן נפתחות שתי קריאות רקורסיביות הממומשות בקוד בעזרת פקודות jal לליבל binom עם הפרמטרים המתאימים ברגיסטרים $a0 ו$a1 החדשים שעודכנו.

Disktest

אתחלנו את רגיסטר $t0 לערך 7 ובכל סוף איטרציה נוריד אותו ב -1 וכך נוכל לוודא שסיימנו להזיז את כל שמונת הסקטורים. בכל איטרציה נעתיק סקטור מהסוף ונדביק לסקטור אחד קדימה ונלך אחורה מהסוף להתחלה. כלומר, נקרא את סקטור 7 לכתובת כלשהיא בזיכרון זמנית (diskbuffer) לאחר מכן נקדם ונעתיק לסקטור 8. לאחר מכן נחזור על התהליך ונעתיק מסקטור 6 לסקטור 7 באותו האופן. וכך חוזר חלילה עד שנסיים להעתיק את כל הסקטורים.

Circle

קבענו בכתובת בזיכרון בעזרת פקודת word. ערך רדיוס שרירותי 100. שמרנו אותו בכתובת שנתונה בהוראות ב0X100 PDF. נאתחל את המשתנים x ו y להיות

x-128, y-128, על מנת שנוכל לקבל את העיגול במרכז. לאחר מכן נקבע את ערכו של ה monitordatta להיות 255 (לבן) בתחילת התוכנית. נבדוק בכל איטרציה שהפיקסל אכן הינו בתוך העיגול לפי משוואת העיגול $x^2 + y^2 \leq R^2$. במידה ומיקומו של הפיקסל הנוכחי שבמטריצת ה buffer כלומר במוניטור, נמצא בתוך העיגול, נשמור את כתובת הפיקסל לתוך ה-monitoradress. ונוכל להיווכח שאכן קיבלנו עיגול מלא בצבע לבן במרכז המסך.