

דוקומנטציה פרויקט מבנה המחשב:

ASSEMBLER

מבנים :

label מכליל שדה של שם הלייבל ושדה שמכיל את מספר השורה בה מופיעה ההוראה הראשונה של label זה בקובץ asm.

word מתאר מבנה של הוראה על פי השדות שנדרשו בפרויקט, כלומר שדות של שמות ה-opcode, שמות הרגיסטרים ב rd,rs,rt וכן את ה imm_num ומספר imm_num.

MEM-מכליל שדה יחיד של תוכן של פקודת word. מבנה זה ימצא במערך ומספרו במערך יהיה מספרו בזיכרון על פי כתובת ה address שבפקודת ה word.

פונקציות :

is_instruction מקבלת שורה מקובץ הקלט asm מחזירה 1 אם מדובר בשורת פקודה (בין אם כוללת התחלה של label ובין אם מסתיימת ב comment-אחרת) (אם מדובר ב word, או label שורה ריקה, שורת comment או כל שילוב אחר שלהם) תחזיר 0.

find_instruction מקבלת פוינטר למיקום בקובץ asm וכן את מספר השורה בה נמצא המיקום ומחזירה את מספר השורה בה מופיעה הראשונה הוראה לאחר הופעת הלייבל.

get_labels ("קריאה ראשונה") מבצעת סריקה של קובץ ה asm וברגע שמזהה לייבל מכניסה את שמו לשדה המתאים במבנה label אשר נמצא במערך של מבנים מסוג label. לאחר מכן מעדכנת את מספר השורה שבה מופיעה ההוראה הראשונה שלו. כמו כן מחזירה את מספר הלייבלים שקיימים בקובץ הקלט.

is_word מקבלת שורה מקובץ הקלט asm מחזירה 1 אם הפקודה מסוג word. גם אם כוללת בתוכה שורת comment או 0 אם מדובר בכל מקרה אחר (למשל comment, שורה ריקה, טאבים, תווים ריקים, הוראה או כל שילוב ביניהם).

instruction_to_wd_arr (עבור "קריאה שניה") מקבלת שורת הוראה ומפצלת אותה לשדות המתאימים במבנה word. שומרת את הפקודות (עלפי סדר הופעתן בקובץ asm.) במערך של מבנים מסוג word. ברגע שמגיעה להוראה עם imm מסוג label, מחפשת את שמו של ה label ושומרת את כתובתו (מספר שורתו בקובץ ה -asm) ב hex המתאים. אחרת לא מדובר ב label שומרת את התכולה ב imm.

change_labels ("קריאה שניה") פונקציה זו עוברת על כל פקודה (כלומר על כל שורה בקובץ הקלט למעט שורות של לייבלים בלבד, או comments בלבד או שילוב של שניהם, או שורות ריקות), ומעביר כל שורה מקובץ הקלט באמצעות *instruction_to_wd_arr* למערך של מבנים מסוג word שנקרא *wd_arr* (כלומר מפצל כל פקודה לשדות המתאימים במבנה word). במידה וב immediate יש לייבל, ממירה את הלייבל לכתובת שבה

נמצא לייבל זה. במידה ויש פקודה מסוג word. שומרת את המידע במבנה מסוג MEM המוכל במערך של מבנים דומים במקום אותו מכתובה פקודת ה word. (כלומר המיקום נקבע על פי ה address) .

asm. go_down מוריד שורה בקריאת קובץ

fromWordsTotxt פונקציה זו מקבלת מערך של מבנים מסוג word וכותבת מחדש את הפקודה ל*memin* באמצעות קידוד ההוראות והרגיסטרים השונים. כמו כן במידה ובשורה מסוימת אמור להיות מידע מפקודת word. הוא יועבר לשורה זו.

main הפונקציה הראשית מבצעת את הקריאה הראשונה *get_labels* ולאחר מכן פונקצית *change_labels* מבצעת את הקריאה השניה, לבסוף פונקצית *fromWordsTotxt* לוקחת את מערך המבנים *words* וממירה רצף ההוראות מקובץ הקלט *asm* לקובץ טקסט *memin*.

Simulator

מבנים:

- **simulator** : מבנה שמכיל בתוכו כל המשתנים שהסימולטור צריך במהלך ריצתו, כדי להעביר את המשתנים לפונקציות ביותר פשטות ע"י העברת מצביע ל simulator . תוכן המבנה:
 - memory : מצביע שיצביע למערך הזיכרון
 - disk : מצביע שיצביע למערך דיסק הקשיח
 - Instruction : משתנה מסוג int שיכיל את הפקודה הנוכחית
 - PC : ערך ה PC הנוכחי
 - op : ערך ה opcode של הפקודה הנוכחית
 - rd : מספר רגיסטר rd נוכחי
 - rs : מספר רגיסטר rs נוכחי
 - rt : מספר רגיסטר rt נוכחי
 - Registers[] : מערך בגודל 16 שמכיל בתוכו את תוכן הרגיסטרים הנוכחי
 - IORegisters[] : מערך בגודל 18 שמכיל בתוכו את תוכן רגיסטרי החומרה הנוכחי
 - Irq : זהו הסיגנל
$$irq = (irq0enable \& irq0status) | (irq1enable \& irq1status) | (irq2enable \& irq2status)$$
 - next_irq2 : מחזור השעון הבא בו תתרחש פסיקה 2 אם יש כזה
 - DMA_starting_cycle : מחזור השעון בו התחילה פעולת DMA
 - timer_starting_cycle : מחזור השעון בו התחיל הטיימר
 - currently_handling_interruption : שווה 1 אם כרגע הסימולטור נמצא בשגרת הטיפול בפסיקה, אחרת שווה 0
 - files[] : מערך בגודל 11 שמכיל מצביעים לקבצי הקלט של בסימולטור
 - PC_updated : שווה 1 אם PC עודכן במהלך ביצוע הפקודה
(jal, reti, beq,..., or PC was changed because of interruption)
אחרת שווה 0

פונקציות:

- write_to_file_from_array : מקבלת מצביע לקובץ טקסט, מצביע למערך ואת גודל המערך וכותבת את תוכן המערך אל קובץ הטקסט כאשר כל איבר בשורה ובהקסאדצימלי
- read_file_to_array : מקבלת מצביע לקובץ טקסט שמכיל מספר בהקסאדצימלי בכל שורה, מצביע למערך ואת גודל המערך ומעתיקה את תוכן הקובץ למערך
- write_to_trace_file : מקבלת מצביע למבנה simulator וכותב את ה PC, הפקודה ותוכן הרגיסטרים אל תוך קובץ trace.txt
- write_to_hwregtrace_file : מקבלת מצביע למבנה simulator וכותב לתוך hwregtrace.txt את:
CYCLE READ/WRITE NAME DATA
- write_line_of_cycle_data_to_file : מקבלת מצביע למבנה simulator וכתלות בפקודה כותב בתוך leds.txt או display שורה של מחזור השעון ותוכן רגיסטר החומרה leds או display

- instruction_fetch : מקבלת מצביע למבנה simulator ומביא את הפקודה הבאה מתוך הזיכרון ושם אותה בשדה instruction של הסימולטור, מחזיר 0 אם הצליח, 1 אם לא
- instruction_partition : מקבלת מצביע ל simulator ומפצלת את ההוראה ל : opcode, rd, rs, rt, imm ועושה sign extension ל imm , ושמה את כל אחת בשדה המתאים במבנה הסימולטור ואת ה imm שמה ברגיסטר \$imm
- execute_instruction : מקבלת מצביע למבנה simulator ומבצעת את ההוראה הנוכחית
- write_to_disk_sector : מקבלת מצביע למבנה simulator ומעתיקה את תוכן diskbuffer אל הדיסק סקטור שנמצא ברגיסר חומרה disksector
- read_from_disk_sector : מקבלת מצביע למבנה simulator ומעתיקה את תוכן הדיסק סקטור שנמצא ברגיסר חומרה disksector אל diskbuffer בזיכרון
- get_next_irq2 : מקבלת מצביע למבנה simulator ומביאה מתוך irq2in.txt את מחזור השעון הבא בו תתרחש פסיקה 2 ושמה את ערכו בתוך שדה next_irq2 של מבנה הסימולטור
- check_and_handle_irq : מקבלת מצביע למבנה simulator ובודקת אם התנאי לפסיקה מתקיים ומעדכנת את רגיסטרי החומרה שתלויים בפסיקות בהתאם לכך
- handle_IO : פונקציה שמטפלת בפקודות in out
- handle_timer : פונקציה שמטפלת בטיימר ומעדכנת אותו אם יש צורך
- handle_diskcmd : פונקציה שמטפלת בהוראות כתיבה או קריאה מהדיסק הקשיח
- initialize_simulator : מקבלת מצביע למבנה simulator את מערך argv ואת argc של command line ומערך file_modes[] שמכיל את מחרוזות סוגי הפתיחה עבור כל קובץ. הפונקציה פותחת את הקבצים ומאתחלת את השדות של מבנה ה simulator.
- run_simulation : מקבלת מצביע למבנה simulator ומסמלצת את מחזורי השעון
- end_simulation : מקבלת מצביע למבנה simulator מסיימת את ריצת הסימולטור, מעתיקה את תוכן הזיכרון והדיסק הקשיח אל memout.tx -i diskout.txt וסוגרת את הקבצים ומשחררת זיכרון שהוקצה דינאמית
- check_disk : פונקציה שבודקת אם פעולת הקריאה או הכתיבה מהדיסק הקשיח הסתיימה ומעדכנת את irq1status, diskstatus, diskcmd בהתאם
- update_PC : מעדכנת את PC