

פרויקט סקורבורד

בפרויקט זה נממש סימולטור של מעבד FLOATING POINT המשתמש באלגוריתם scoreboard.

המעבד כולל:

- 16 floating point registers, שנשמנים F0-F15, כל אחד ברוחב 32 סיביות ושומר מספר נקודה צפה בפורמט single precision (ביט אחד עבור הסימן, 8 עבור האקספוננט ו-23 עבור המנטיסה). הניחו כי אין הגבלה על מספר הקריאות והכתיבות למערך הרגיסטרים שניתן לבצע במקביל כל מחזור שעון. בתחילת הריצה כל רגיסטר מכיל מספר השווה לאינדקס שלו: F0 מכיל 0, F1 מכיל 1.0, וכך הלאה.
- יחידת Fetch שקוראת עד הוראה אחת מזיכרון ההוראות בכל מחזור שעון אל תוך ה- Instruction Queue, שהינו בגודל 16 הוראות, כל עוד יש מקום. במחזור שעון מספר 0 מתבצע Fetch של ההוראה הראשונה וכתיבה אל תור ההוראות.
- המעבד מבצע Issue של עד הוראה אחת בכל מחזור שעון. הפענוח מתבצע בז'ומנית עם ה- Issue, ולא לוקח מחזור שעון נוסף. למשל במחזור שעון 1 כבר מבצעים issue של ההוראות הראשונה שנקראה במחזור 0, וכמובן שבמקביל מבצעים כבר Fetch של ההוראה הבאה.
- מספר יחידות פונקציונאליות לנקודה צפה עבור חיבור, חיסור, כפל, חילוק, טעינה מזיכרון, וכתיבה לזיכרון. היחידות לא מצונרות, ובעלות השהייה ניתנת לקנפוג. מחזור השעון שבו נקראים האופרנדים לאחר שהם מוכנים בשלב ה- Read Operands הוא גם המחזור השעון הראשון של שלב הבצוע ביחידה הפונקציונאלית. למשל אם השהיית היחידה הפונקציונאלית היא 2, אז שלב ה- Exec יסתיים מחזור שעון אחד לאחר Read Operands. שלב ה- Write Result מתחיל מחזור שעון אחד לאחר שלב סיום הביצוע.
- הזיכרון הראשי בגודל 4096 מילים בנות 32 סיביות כל אחת. כל יחידות הטעינה והכתיבה לזיכרון יכולות לגשת במקביל לזיכרון, אולם יש לדאוג כי לא יהיו גישות כתיבה וקריאה, או מספר גישות כתיבה, הפעילות במקביל לאותה הכתובת. קריאות במקביל מאותה הכתובת מותרות אם אין כתיבה במקביל לכתובת זו. ניתן להניח כי לא יהיו בתוכנית הוראות LD או ST אשר ניגשות לכתובות אשר מכילות הוראות להרצה.

כל הוראה מקודדת ב- 32 סיביות, בפורמט אחיד:

bits	31-28	27-24	23-20	19-16	15-12	11-0
	0	OPCODE	DST	SRC0	SRC1	IMM

ה- OPCODE מתאר את ההוראה שאותה יש לבצע. שדה ה- DST הוא רגיסטר היעד, והשדות SRC0, SRC1 הם שני רגיסטרי המקור. השדה IMM הוא קבוע בן 12 סיביות.

כאשר סט ההוראות מכיל:

opcode name	number	description
LD	0	$F[DST] = MEM[IMM]$
ST	1	$MEM[IMM] = F[SRC1]$
ADD	2	$F[DST] = F[SRC0] + F[SRC1]$
SUB	3	$F[DST] = F[SRC0] - F[SRC1]$
MULT	4	$F[DST] = F[SRC0] * F[SRC1]$
DIV	5	$F[DST] = F[SRC0] / F[SRC1]$
HALT	6	exit simulator

בכל הגישות לזיכרון, הכתובת לזיכרון היא של מילים בנות 32 סיביות כל אחת (לא של בתים).

1 סביבות תכנות:

הפרויקט ימומש בשפת C בסביבת Windows, ב- Visual Studio 2022. יש להגיש את כל ספריית ה- Solution כך שנוכל לקמפל ע"י build solution.

2 הרצה וקבצים:

הפרויקט יבנה אל תוך command line application שנקרא sim.exe, ויורץ עם רשימת קבצי טקסט בתור פרמטרים:

sim.exe cfg.txt memin.txt memout.txt regout.txt traceinst.txt traceunit.txt

כאשר cfg.txt ו־ memin.txt הינם קבצי קלט, ושאר הקבצים הינם קבצי פלט.

קובץ הקונפיגורציה cfg.txt מכיל שורות מהצורה parameter = value, כאשר הפרמטרים הינם:

- add_nr_units = x : מספר יחידות החיבור.
- sub_nr_units = x : מספר יחידות החיסור.
- mul_nr_units = x : מספר יחידות הכפל.
- div_nr_units = x : מספר יחידות החילוק.
- ld_nr_units = x : מספר יחידות הטעינה מהזיכרון.
- st_nr_units = x : מספר יחידות הכתיבה לזיכרון.
- add_delay = x : השהיית יחידות החיבור במחזורי שעון.
- sub_delay = x : השהיית יחידות החיסור במחזורי שעון.
- mul_delay = x : השהיית יחידות הכפל במחזורי שעון.
- div_delay = x : השהיית יחידות החילוק במחזורי שעון.
- ld_delay = x : השהיית יחידות הטעינה מהזיכרון במחזורי שעון.
- st_delay = x : השהיית יחידות הכתיבה לזיכרון במחזורי שעון.
- trace_unit = name : שם היחידה שעבורה נכתוב את קובץ ה־ traceunit.

קובץ תמונת הזיכרון memin.txt מכיל 4096 שורות של תמונת הזיכרון הראשי כאשר כל שורה מכילה 32 סיביות ב־ 8 ספרות הקסאדצימליות. התוכנית מתחילה לרוץ מ־ PC=0, כאשר ההוראה שם מקודדת בשורה הראשונה בקובץ.

אם החל מכתובת מסויימת ועד הסוף תוכן הזיכרון מכיל רק אפסים, מותר לא לרשום שורות אלו, בהבנה שהזיכרון יכיל אפסים. זה נעשה רק כדי לחסוך בגודל הקבצים ואינו חובה, כלומר אפשר גם בכל מקרה לכתוב את כל 4096 השורות.

הקובץ memout.txt הינו באותו הפורמט כמו memin.txt, ומכיל את תמונת הזיכרון בסיום הרצת התוכנית.

הקובץ regout.txt מכיל את פלט רגיסטרי ה־ floating point בסיום ריצת התוכנית. יהיו שם 16 שורות, כאשר כל שורה i הינה מספר עשרוני עבור תוכן הרגיסטר $F[i]$.

הקובץ traceinst.txt מכיל שורות בפורמט הבא:

instruction pc unit cycle.issued cycle.read.operands cycle.execute_end cycle.write.result

כאשר יש שורה עבור כל הוראה (מלבד הוראת HALT) לפי סדר ה־ ISSUE (לא לפי סדר ה־ COMPLETION).

- שדה ה־ instruction הוא קידוד ההוראה בשמונה ספרות הקסא כפי שנקראו מהזיכרון.
- שדה ה־ pc הינו ה־ pc של ההוראה (0 עבור ההוראה הראשונה בזיכרון, 1 עבור השנייה וכך הלאה).

- שדה ה־ unit מכיל את שם היחידה הפונקציונאלית שמקבלת את ההוראה. לדוגמא שמות אפשריים הינם ADD0, MUL2, DIV1, LD3. השמות ממוספרים החל מאפס.
- שדה ה־ cycle_issued הוא מחזור השעון שבו ההוראה נכנסה לאחת היחידות.
- שדה ה־ cycle_read_operands הוא מחזור השעון שבו נקראו האופרנדים וההוראה התחילה להתבצע ביחידה הפונקציונאלית.
- שדה ה־ cycle_execute_end הינו מחזור השעון האחרון שבו ההוראה עדיין מתבצעת ביחידה הפונקציונאלית.
- שדה ה־ cycle_write_result הינו מחזור השעון שבו התוצאה נכתבה לרגיסטר. לשם קונסיסטנטיות, גם עבור הוראות ST יהיה מחזור שעון יעודי עבור write_result למרות שאין רגיסטר יעד.

הקובץ traceunit.txt מכיל שורות בפורמט הבא:

```
cycle unit Fi Fj Fk Qj Qk Rj Rk
```

כאשר יש שורה עבור כל מחזור שעון שבה היחידה trace_unit כפי שהוגדרה בקובץ הקונפיגורציה הייתה במצב Busy.

- שדה ה־ cycle הוא מחזור השעון.
- שדה ה־ unit מכיל את שם היחידה הפונקציונאלית או ה־ LD/ST buffer שמקבלת את ההוראה. לדוגמא שמות אפשריים הינם ADD0, MUL2, DIV1, LD3. השמות ממוספרים החל מאפס.
- שדות Fi, Fj, Fk הינם שמות הרגיסטרים כפי שהוגדרו באלגוריתם הסקורבורד.
- שדות Qj, Qk מכילים את שם היחידה כפי שהוגדר באלגוריתם בכיתה, או הסימן מינוס אם לא קיים.
- שדות Rj, Rk מכילים Yes או No כפי שהוגדר באלגוריתם בכיתה.

3 דוקומנטציה:

הקפידו שהקוד יהיה קריא, ומכיל comments לגבי מבני הנתונים והפונקציות. כמו כן יש להגיש דוקומנטציה חיצונית המתארת באופן כללי את הפרויקט.

4 בדיקות:

הפרויקט שלכם יבדק בן השאר ע"י תוכניות בדיקה שלא תקבלו מראש. לכן חשוב מאוד לבדוק נכונות ע"י בנייה של קטעי קוד שונים, וכמו כן בדיקה עם פרמטרים שונים בקובץ הקונפיגורציה. יש להגיש 3 ספריות בדיקה, כאשר בכל ספרייה יהיו קבצי הקלט והפלט עבור הבדיקה. יש לתאר בדוקומנטציה את הבדיקות שבוצעו.