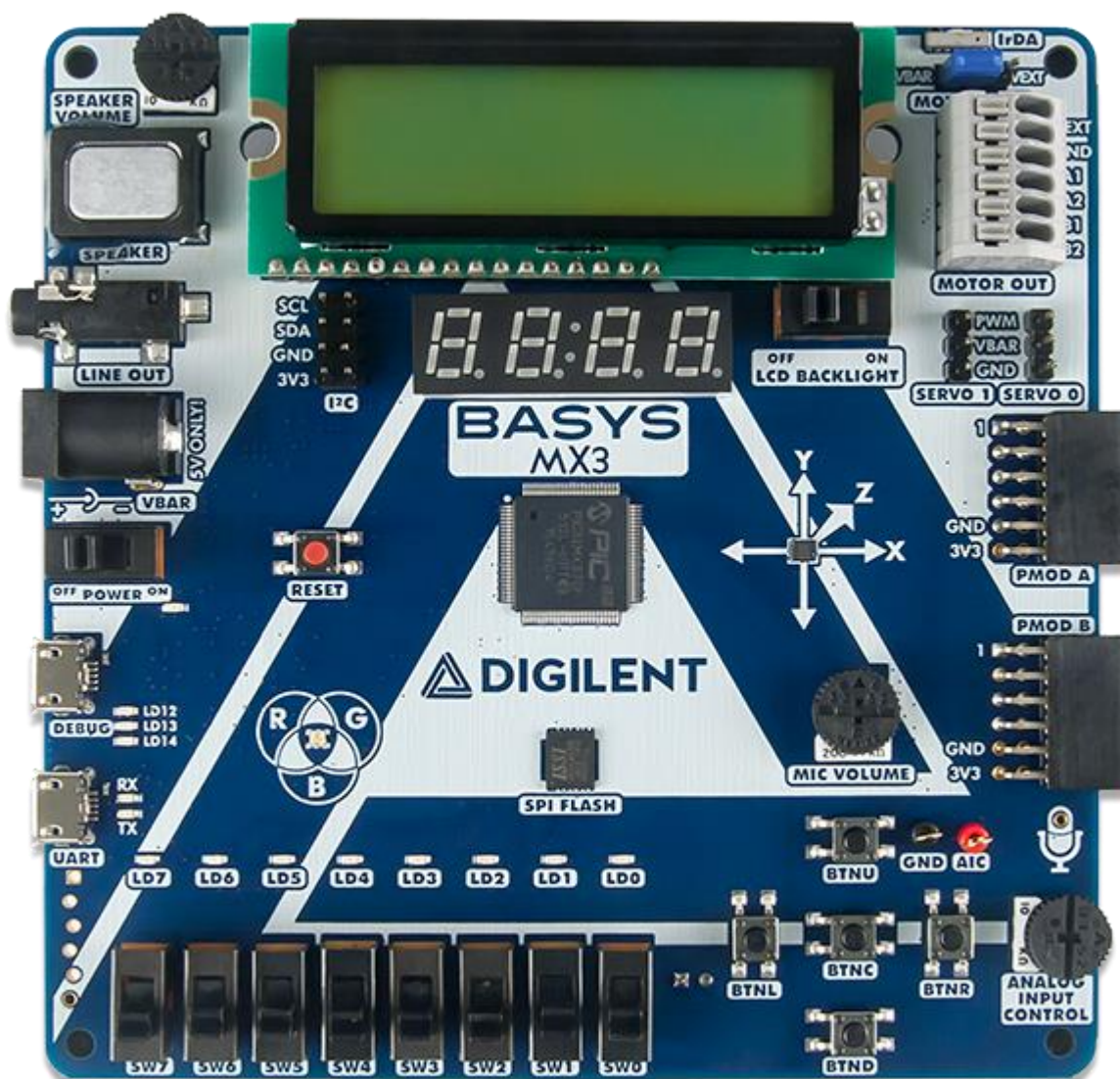


אוניברסיטת תל-אביב, הפקולטה להנדסה

פרויקט BASYS בקורס: מבנה המחשב 0512.4400

שנת הלימודים תשפ"ג, סמסטר ב'

בפרויקט זה נממש סימולטור למעבד SIMP, שהוגדר בפרויקט הראשון, ע"י תכנות כרטיס מחשב BASYS MX3 של חברת DIGILENT בשפת סי:



כרטיס זה כולל מיקרוקונטרולר PIC32MX370 מחברת Microchip, המכיל מעבד MIPS, זיכרונות, טיימרים, ותמיכה בפסיקות ו-DMA. מעבד זה מחובר על גבי הכרטיס להתקני קלט/פלט שונים כמתואר בשרטוט מעלה: תצוגת LCD, תצוגת 7-segment, מפסקים וכפתורים, מיקרופון, רמקול, ועוד. הכרטיס מתחבר למחשב האישי ע"י כבל MicroUSB.

פעולת הסימולטור

בזמן היציאה מ-Reset, תוכן הזיכרון הראשי יטען באחת משתי תוכניות אסמבלי, כתלות במפסק SW7. לאחר טעינת התוכנית לזיכרון, PC יאותחל לאפס והתוכנית תתחיל לרוץ עד להגעה להוראת HALT. כל הוראה רצה ב- 976.5625 מיקרושניות, כלומר קצב ריצה ההוראות הוא 1024 הוראות לשנייה, וכל 976.5625 מיקרושניות ה-PC משתנה.

מצבי PAUSE ו-SINGLE STEP: לחיצה על BTNL מבצעת PAUSE בריצת התוכנית. כלומר בלחיצה ראשונה על BTNL, הסימולטור עוצר וה-PC מצביע על ההוראה שאנו נמצאים בה. במצב PAUSE, לחיצה על BTNR מבצעת SINGLE STEP כלומר הסימולטור מריץ הוראה אחת נוספת, ועוצר שוב. לחיצה נוספת על BTNL משחררת את מצב ה-PAUSE, וממשיכים לרוץ כרגיל, עד לכניסה הבאה ל-PAUSE או עד לסיום התוכנית.

תצוגת LCD:

בשורה השנייה יוצגו PC, RSP, TIME בפורמט הבא:

XXX YYY TTTTTTTT

כאשר XXX, YYY הינם בהקסאדצימאלי, והזמן TTTTTTTT הינו הזמן שעבר בשניות מאז היציאה מ-RESET.

המפסקים SW1, SW0 שולטים על תצוגת השורה הראשונה ב-LCD:

- כאשר SW1=OFF, SW0=OFF, ההוראה INST שאותה מריצים כעת תוצג בשורה הראשונה באותו פורמט כמו ה-trace בפרויקט הראשון: 8 ספרות הקסאדצימליות.

- כאשר SW1=OFF, SW0=ON, תוכן הרגיסטרים יוצג בשורה הראשונה בפורמט RXX = YYYYYYYY כאשר XX הינו מספר הרגיסטר שמוצג כרגע בספרות דצימאליות, החל מרגיסטר אפס, ו- YYYYYYYY הינו תוכנו ב- 8 ספרות הקסאדצימליות. לחיצה על BTNU תקדם את XX באחד כך שניתן יהיה להציג את כל הרגיסטרים. קידום מספר הרגיסטר מעל 15 חוזר חזרה לרגיסטר אפס.

- כאשר SW1=ON, SW0=OFF, תוכן הזיכרון יוצג בשורה הראשונה, בפורמט MAAA = DDDDDDDD כאשר AAA הינו הכתובת בזיכרון ו- DDDDDDDD הינו הדאטא, שניהם בספרות הקסאדצימליות.

-
לחיצה על BTNU תקדם את הכתובת באחד. חריגה מחוץ לגבולות הזיכרון (IFF) מבצעת "סיבוב" חזרה לאפס.

- כאשר $SW1=ON, SW0=ON$, יוצג בשורה הראשונה מספר ההוראות שבוצעו עד עתה, בהקסאדצימאלי.

שימו לב שאין קשר בין המפסקים המשפיעים על התצוגה במסך ה-LCD לבין מצב ה-PAUSE ו-SINGLE STEP שנקבעים ע"י הכפתורים BTNL ו-BTNR. מותר למשל לשנות את התצוגה ע"י המפסקים בזמן שתוכנית האסמבלי רצה ואין לעצור את התוכנית. כמו כן מותר לשנות את המפסקים גם בזמן שהתוכנית במצב PAUSE.

כמו כן אין צורך לעדכן את תצוגת ה-LCD מהר יותר מכפי שהעין מסוגלת לעקוב. פעם בשנייה או שתי שניות מספיק.

תוכנית האסמבלי שאותה מסמלצים נקבעת ע"י מצב SW7 לאחר היציאה מ-RESET. בזמן העבודה אין צורך לתמוך בשינוי מצב SW7 עד הפעם הבאה שמתבצע RESET.

תוכנית אסמבלי 1:

כאשר $SW7=OFF$, תורץ הדוגמא של סדרת פיבונאצ'י מפרויקט ה-ISA.

תוכנית אסמבלי 2:

כאשר $SW7=ON$, תורץ תוכנית אסמבלי חדשה, סטופר, המציגה את הזמן שעבר מאז RESET באופן MM.SS בתצוגת ה-7-segment display, כאשר SS הינו מספר השניות (0 עד 59) ו-MM מספר הדקות (0 עד 59). לאחר שעה הסטופר מסתובב חזרה ל-00:00 וממשיך לספור.

לחיצה על BTNC מבצעת PAUSE בסטופר. לחיצה נוספת על BTNC משחררת את ה-PAUSE.

לחיצה על BTND מאפסת את הסטופר חזרה ל-00:00.

יש להקפיד שהסטופר יהיה מדויק ולא יצבור טעות של יותר משנייה עבור כל דקה ספירה במצב שהוא "רץ חופשי" ללא בצוע PAUSE בתוכנית האסמבלי או בסימולטור.

תמיכה בקלט/פלט

לצורך מימוש הסטופר באסמבלי, נוסף למעבד תמיכה בקלט ופלט ע"י 3 הוראות חדשות, "רגיסטרי חומרה", ופסיקות. לגישה לרגיסטרי החומרה נשתמש באופקודים 17 ו- 18, שהיו reserved בפרויקט הראשון, תוך שימוש בקבוע, וכמו כן נוסף הוראת חזרה מפסיקה באופקוד

: 16

| Opcode | Name | Meaning |
|--------|------|-------------------------------------|
| 16 | reti | PC = IORegister[7] |
| 17 | in | $R[rd] = IORegister[R[rs] + R[rt]]$ |
| 18 | out | $IORegister[R[rs] + R[rt]] = R[rd]$ |

IORegister[14] הינו מערך של 14 "רגיסטרי חומרה", כפי שמתכנת האסמבלי חושב שהם, שמאותחלים לאפס ויסומלצו ע"י הסימולטור :

| IORegister Number | Name | number bits | Meaning |
|-------------------|-------------|-------------|--|
| 0 | irq0enable | 1 | IRQ 0 enabled if set to 1, otherwise disabled. |
| 1 | irq1enable | 1 | IRQ 1 enabled if set to 1, otherwise disabled. |
| 2 | irq2enable | 1 | IRQ 2 enabled if set to 1, otherwise disabled. |
| 3 | irq0status | 1 | IRQ 0 status. Set to 1 when irq 0 is triggered. |
| 4 | irq1status | 1 | IRQ 1 status. Set to 1 when irq 1 is triggered. |
| 5 | irq2status | 1 | IRQ 2 status. Set to 1 when irq 2 is triggered. |
| 6 | irqhandler | 9 | PC of interrupt handler |
| 7 | irqreturn | 9 | PC of interrupt return address |
| 8 | clks | 32 | cyclic clock counter. Starts from 0 and increments every clock. After reaching 0xffffffff, the counter rolls back to 0. |
| 9 | reserved | 32 | Reserved for future use. |
| 10 | display7seg | 16 | Connected to 7-segment display of 4 letters. Each 4 bits displays one digit from 0 – F, where bits 3:0 control the rightmost digit, and bits 15:12 the leftmost digit. |

| | | | |
|----|--------------|----|---------------------------------------|
| 11 | timerenable | 1 | 1: timer enabled 0: timer disabled |
| 12 | timercurrent | 32 | current timer counter |
| 13 | timermax | 32 | max timer value |

פסיקות

מעבד SIMP תומך ב- 3 פסיקות: $irq0$, $irq1$, $irq2$. פסיקה 0 משויכת לטיימר, וקוד האסמבלי יכול לתכנת כל כמה זמן הפסיקה תתרחש.

פסיקה 1 היא BTNC. היא מתרחשת כאשר זוהתה לחיצה על BTNC.

פסיקה 2 היא BTND. היא מתרחשת כאשר זוהתה לחיצה על BTND.

במחזור השעון בו הפסיקה מתקבלת, הסימולטור מדליק את אחד הרגיסטרים $irq0status$, $irq1status$, $irq2status$ בהתאמה. אם מספר פסיקות מתקבלות באותו מחזור שעון, ידלקו בהתאמה מספר רגיסטרי סטטוס.

כל מחזור שעון, המעבד בודק את הסיגנל:

$$irq = (irq0enable \& irq0status) | (irq1enable \& irq1status) | (irq2enable \& irq2status)$$

במידה ו- $irq == 1$, והמעבד לא נמצא כרגע בתוך שגרת הטיפול בפסיקה, המעבד קופץ לשגרת הטיפול בפסיקה שכתובתה בזיכרון נתונה ברגיסטר חומרה $irqhandler$. כלומר במחזור שעון זה מתבצעת ההוראה ב- $PC = irqhandler$ במקום ב- PC המקורי. באותו מחזור שעון ה- PC המקורי נשמר לתוך רגיסטר חומרה $irqreturn$.

לעומת זאת במידה ו- $irq == 1$ והמעבד עדיין נמצא בתוך שגרת הטיפול בפסיקה קודמת (כלומר עדיין לא הריץ את הוראת ה- $reti$), המעבד יתעלם, לא יקפוץ וימשיך להריץ את הקוד כרגיל בתוך שגרת הפסיקה (כאשר המעבד יחזור מהפסיקה, הוא יבדוק שוב את irq ואם יהיה צורך יקפוץ שוב לשגרת הפסיקה).

קוד האסמבלי של שגרת הפסיקה יבדוק את הביטים של $irqstatus$, ולאחר טיפול מתאים בפסיקה יכבה את הביטים.

חזרה משגרת הפסיקה מתבצעת באמצעות הוראת $reti$, שתציב $PC = irqreturn$.

טיימר

מעבד SIMP תומך בטיימר של 32 ביטים, המחובר לפסיקה $irq0$. הוא מאופשר כאשר $timerenable = 1$.

ערך מונה הטיימר הנוכחי שמור ברגיסטר חומרה $timercurrent$. בכל מחזור שעון שבו הטיימר מאופשר, רגיסטר $timercurrent$ מקודם באחד.

במחזור השעון שבו $timercurrent = timermax$, מדליקים את $irqstatus0$. במחזור שעון זה במקום לקדם את $timercurrent$, מאפסים אותו חזרה לאפס.

IORegister[10] קובע את התצוגה של ה- 7-segment display. כל 4 ביטים קובעים ספרה בודדת שתוצג, מימין לשמאל. כלומר ביטים 3:0 קובעים את הספרה הימנית, ביטים 7:4 את הספרה השנייה מימין, ביטים 11:8 את הספרה השנייה משמאל, וביטים 15:12 את הספרה השמאלית. כל ספרה יכולה להכיל 0 עד F, אבל בתוכנית האסמבלי יש לכתוב בספרת האחדות ערכים מאפס עד 9 ובספרת העשרות ערכים מאפס עד 5.

עבור כתובות החורגות ממערך רגיסטרי החומרה, יש להתעלם מהכתיבה ולהחזיר אפס בקריאה.

ניהול נורות ה-LED:

בכל פעם שהמעבד מריץ את הוראת `reti`, יש להדליק נורית אחרת שתישאר דלוקה עד הרצת הוראת `reti` בפעם הבאה, באופן סיבובי שמאלה החל מ LD0. לדוגמא:

בפעם הראשונה שהמעבד נתקל בהוראת `reti`, מדליקים את LD0 ומכבים את LD7 (שהיה במקרה כבר מכובה כי זו הפעם הראשונה).

בפעם השנייה שמריצים `reti`, מדליקים את LD1 ומכבים את LD0.

...

בפעם השמינית שמריצים `reti`, מדליקים את LD7 ומכבים את LD6.

וחוזר חלילה, בפעם התשיעית נדליק את LD0 ונכבה את LD7, וכך הלאה.

הדלקת/כיבוי הנורות מנוהל ע"י קוד הסימולטור בסי, לא ע"י קוד האסמבלי.

דרישות הגשה

1. יש להגיש קובץ דוקומנטציה של הפרויקט, חיצוני לקוד, בפורמט pdf.
2. הפרויקט יכתב בשפת התכנות סי בסביבת MPLAB IDE. יש להגיש את כל ספרית הפרויקט כך שניתן יהיה לבנות אותו ע"י לחיצה על `build project`.
3. יש להגיש גם את קוד האסמבלי של תוכנית הסטופר.
4. יש להגיש את העדכונים שבצעתם לאסמבלר ולסימולטור מהפרויקט הראשון על מנת לתמוך בהוראות החדשות.
5. בנוסף להגשה במודל, יש לתאם מועד להצגת הפרויקט (DEMO) על הכרטיס.