חלק תיאורטי- דוח מכין 1

1. סביבת פיתוח(IDE- Integrated Development Environment) היא חבילת תוכנה אליה ניתן לכתוב פקודות קוד, בשפה כלשהי, לפי הסינטקס של אותה שפה. הסביבה תיקח את הפקודות שנכתבות לה כטקסט, ובאמצעות כלים הבנויים בה כמו קומפיילר או אינטרפטר תמיר אותה לשפת מכונה. בסביבת הפיתוח קיימים גם כלים נוספים כמו דיבאגר למשל, שמקלים את פיתוח התוכנה על המשתמש. בקורס אנחנו עובדים עם סביבת IAR שמשמשת אותנו לכתיבת פקודות קוד באסמבלי.

2. בבקר קיימות פקודות מובנות אותם ניתן לכתוב לסביבת הפיתוח, וכאשר מקמפלים וצורבים אותן לבקר, הבקר יודע לפעול על פי הן. לבקר 430MSP שאנחנו נשתמש בו במעבדה קיימות 27 פקודות ליבה ו24 פקודות אמולציה. ההבדל בינהן הוא שפקודות אמולציה בנויות מפקודות ליבה ונועדו להקל על המשתמש בבקר לכתוב לו פקודות. פקודות ליבה הן פקודות שיש להן קוד פעולה ייחודי שמפוענח באופן ישיר על ידי המעבד ופקודות האמולציה הן פקודות שלא קיים להן קוד פעולה ייחודי.

3. במעבד (CPU) יש קובץ רגיסטרים, המכיל 16 רגיסטרים ויחידה אריתמטית לוגית. רגיסטר הוא יחידת זכרון באורך 16 ביטים (שני bytes) בה ניתן לאחסן מידע למטרות שונות. רגיסטרים 4-16R משמשים למטרות כלליות- לשימוש המשתמש לשמור כתובות ומספרים מ0 עד 1-(2^16).
רגיסטרים 0-3R משמשים לתפקידים ייעודיים:
   - R0- Program Counter (PC) מצביע על הכתובת של ההוראה הבאה לביצוע. מאותחל בכתובת ההתחלתית של קטע הקוד שנמצא במרחב זיכרון ה – FLASH.

   - R1- Stack Pointer (SP) משמש לניהול המחסנית, המשמשת לאחסון נתונים זמניים במהלך ריצת הקוד. R1 מצביע לערך העליון הנוכחי של המחסנית (ראש המחסנית).

   - R2 –Status Register (SR) ברגיסטר זה נמצאים הדגלים V,C,N,Z המושפעים מפעולות אריתמטיות שבוצעו (הסבר מפורט בתשובה לשאלה מס' 4). ומשמש גם כמחולל קבועים.

   - R3 – Constant Register (CG) - רגיסטר זה משמש גם כמכולל קבועים, כלומר מייצר קבועים הנמצאים בשימוש בתדירות גבוהה ובכך מקצר כמות הפקודות הדרושה בעת שימוש בקבועים אלו (הגישה לזיכרון לצורך שליפת הקבוע נחסכת). בנוסף, רגיסטר זה מאפשר את התמיכה ב – 24 הוראות האמולציה (אליהן התייחסנו בתשובה לשאלה מס' 2).

4. פירוט הדגלים שנמצאים ברגיסטר R2:
   - Carry bit ערכו של ביט זה הוא 1 כאשר הפעולה האריתמטית האחרונה שבוצעה יצרה נשא, ו – 0 אם לא יצרה נשא. אם בוצעה פעולה שבעקבותיה לא יכול להיווצר נשא, ערכו של ביט זה נשמר. מיקום הbit ברגיסטר הוא 0.
     דוגמא לפקודה:
     CLRC

- Zero bit- ערכו של ביט זה הוא 1 כאשר תוצאת הפעולה האריתמטית האחרונה שבוצעה היא אפס, ואחרת ערכו הוא 0. מיקום הbit ברגיסטר הוא 1.
  דוגמא לפקודה:
  SETZ

- Negative bit- שווה אחד כאשר התוצאה של פעולה היא מספר בינארי שלילי ושווה 0 כאשר הוא חיובי (בשיטת משלים ל-2). מיקום הbit ברגיסטר הוא 2.
  דוגמא לפקודה:
  CLRN

- 8- Overflow bit- ביט זה מקבל את הערך 1 כאשר מתקבלת תוצאה עבור פעולה אריתמטית שאינה הגיונית – ומתריע לנו כי לא ניתן לייצג את התוצאה המבוקשת באמצעות מספר הסיביות הנתון, בהתאם למגבלות שיטת משלים ל-2.
  לדוגמה, אם נחבר שני מספרים חיוביים ובתוצאה תתקבל כי ספרת ה- MSB היא 1, ביט הOverflow יעבור לערך 1.זאת מכיוון שלא ייתכן כי חיבור שני מספרים חיוביים ייתן מספר שלילי. באופן דומה יתרחש המקרה עבור חיבור שני מספרים שליליים וקבלת MSB שערכה 0- כלומר קבלת מספר חיובי. מיקום הbit ברגיסטר הוא 8.
  דוגמא לפקודה:
  SXT(על פי המפרט של MSP430 לא קיימת פקודה שמשפיעה אך ורק על ה Overflow bit) .

5. טיפוסים וגדלים בMSP430:

- Char: גודל 8 :סיביות .(1 byte)

  סוג נתונים זה משמש בדרך כלל לאחסון תו יחיד, ויכול להכיל ערך מספרי בין 0 -256 1=255=2^1-8.

- int: גודל: 16 סיביות (2 .(bytes
  משמש לאחסון מספרים שלמים, טבעיים ושליליים. טווח המספרים הטבעיים החיוביים שהוא יכול לייצג- עד 2^16 = 65536. טווח של מספרים שליליים וחיוביים: -32,768 עד 32,767.
- Long: גודל 32 :סיביות .(4 bytes)
  מתאים לערכים גדולים יותר מאלה של int. יכול לאחסן ערכים עם סימן (מ- 2,147,483,648- עד 2,147,483,647) או ללא סימן (0 עד 4,294,967,295).

<u>פרטי ריצה של המשימה:</u>

<u>גודל תכנית</u>: 0x2162 – 0x2100 = 0x0062h (98 bytes)

<u>זמן ריצה:</u> time = 156*Tmclk = 156*0.954*10^(-6) = 148.824micro sec

מגישים:

כרמי פרנק 206463846

מתן יוסף טנצר 208314302