

# מבנה מחשבים ספרתיים - דו"ח מכין מעבדה 1

מגישים: יעקב קוזמינסקי 206511966, אור יעקובי 206827164

## שאלות חלק תאורטי

1.

עבור פורט המשמש I/O נגדיר את הרגיסטרים הבאים: PxDIR, PxSEL, PxIN, PxOUT. כאשר כל ביט ברגיסטר משמש בהתאמה עבור פין בפורט. התפקידים:

PxSEL - משמש לקביעת צורת העבודה של הפורט. 0- מסמן קלט ופלט ו1- מסמן שהפורט במצב שימוש אחר (ייעודי) אותו ניתן לראות בפירוט מלא במפרט הרכיב.  
PxDIR - משמש לקביעת כיוונית הפורט, האם הוא במצב כתיבה לבקר או בקריאה מהבקר. כאשר כתיבה לבקר זה 0, קריאה מהבקר זה 1.  
PxIN - ברגיסטר זה נוכל לראות את הערך הנכנס לכל פין המגודר להיות אינפוט. כאשר 1 מסמן שאנחנו כותבים ו0 אומר שלא.  
PxOUT - בדומה לרגיסטר PxIN, במידה ונראה 1 נוכל לדעת שיש קריאה (הוצאת מתח) בפורט. ו אם נראה 0 אז אין הוצאת מידע.

2.

ברירת המחדל של הפורטים לאחר ביצוע אתחול הוא מצב קלט. זאת משום שהכנסת מתח לבקר שהוא לא במצב קלט עלול לפגוע ברכיביו הפנימיים.

3.

שלבי הקינפוג של פורט9 למצב I/O כאשר כניסות בעלי אינדקס זוגי יהיה output ו כניסות א"ז יהיה input יהיו:

Bic.b #0xff, &PxDIR  
Bis.b #0xaa, &PxDIR

כאשר מה שעשינו בפועל הוא איפוס הרגיסטרים בDIR כדי לקבוע עבודה במצב קלט, פלט ולאחר מכן קינפגנו אילו כניסות יהיה פלט ואילו קלט לפי ההצגה הבינארית של AA.

4.

נשתמש בנתוני השעון:

$$f_{MCLK} = 32 \cdot 32768 = 2^{20} = 1,048,576 \text{ Hz} \rightarrow T_{MCLK} = \frac{1}{2^{20}} \approx 0.954 \mu\text{sec}$$

לכן ע"מ לקבל מחזור שעון של מילי שנייה אחת, נרצה DS של 50%, נחשב:

$$N_{1_{cycles}} = \frac{\text{total '1' time}}{\text{time per cycle}} = \frac{0.5 \cdot 10^{-3}}{\frac{1}{f_{MCLK}}} = \frac{0.5 \cdot 10^{-3}}{2^{-20}} = 524 \frac{36}{125} \approx 524_{cycles}$$

5. פסיקה:

פסיקה היא קבלת התראה ( מהמשתמש או מהמערכת עצמה) שמטרתה לסעף את התוכנית ולבצע קטע קוד אחר שהוגדר מראש. לאחר סיום קטע הקוד הנ"ל, נחזור לנקודה האחרונה בה היינו בהרצת התוכנית המקורית, ונמשיך ממנו.

## 6. יתרונות הפסיקה כנגד תשאול:

בשימוש בתשאול, נדרש לבצע בדיקה אחת לזמן מסוים- דבר הדורש הגדרת תדירות נדרשת וקבועה לאורך כל התוכנית. תשאול תכוף זה יגביל את התכנית שכן כל כמה זמן נצטרף לעצור ולבדוק את מצב המערכת. לעומתה, בפסיקות נוכל לתכנת ללא התחשבות בתשאול הקבוע ועצירת התוכנית- אלא נוכל לרוץ באופן רציף ולעצור רק כשנצטרך. בנוסף, קיים אספקט של צריכת אנרגיה אשר גבוהה יותר בתשאול (יותר פעולות בדיקה). נוכל לשלב בין שני השיטות כאשר למשל לעבור לקטע קוד בו נקבל נתונים מהמשתמש באמצעות כפתורים (זו תהיה פסיקה) ובזמן עיבוד המידע- נשתמש בתשאול רציף.

## 7.נסביר את סוגי הפסיקות:

- פסיקה שלא ניתנת למיסוך- פסיקה בעדיפות עליונה, שלא ניתן להתעלם ממנה. הצורך בפסיקה זו הוא עבור פעולות קריטיות. לדוגמא: לחיצה על כפתור RESET לאתחול התוכנית. פסיקה זו תתבצע ללא תנאי הקשור לביצוע הנוכחי של התוכנית.
- פסיקה שניתנת למיסוך – פסיקות הניתנות למיסוך באופן פרטני או כללי, הצורך בפסיקות מסוג זה הוא עבור ביצוע פעולות תזמון שונות ומקורות מידע ממקורות חיצוניים.

## ישנם 3 סוגי פסיקות:

- חיצונית- פסיקה הנגרמת מרכיב חומרתי באופן שאינו תלוי בריצת התוכנית הנוכחית. תקרא גם פסיקה אסינכרונית.
- פנימית- פסיקה הנגרמת מרכיב חומרה או חלק בתוכנה אשר מתבצעת בעקבות ביצוע קוד מסוים, לכן היא תקרא סינכרונית. שכן היא מתבצעת לפי תכנון.
- תוכנה- פסיקה הנגרמת בעקבות שינוי מצב דגל מסוים בבקר, נקראת גם פסיקת מלכודת.

## 8.

### נפרט אופני עבודת הבקר:

אופני עבודת הבקר הם מצב פעיל ומצב שינה. בהפעלה, הבקר יריץ את התוכנית שורה אחר שורה. כאשר נרצה השהייה(הפסקת עבודה) עד להגעה של פסיקה- נכניס את הבקר למצב שינה בה הוא לא יתקדם בקוד אלא ישמור על מצבו הנוכחי ויחכה לפסיקה שתגיד לו איך עליו לפעול. (דרך נוספת למצב זה ללא הכנסה למצב שינה – היא להכניסו ללואה אינסופית) לבקר שלנו יש כמה מצבי שינה המוגדרים כ LPM0-LPM4, אשר נעים מהבזבזני(ערני) לחסכני ביותר (לפי מספר הרכיבים העובדים).

## 9.

השלבים לקנפוג רגל 0 של P2, כך שבירידת מתח תתבצע בקשת פסיקה:

Bic.b #0x01, &P2SEL

Bic.b #0x01, &P2DIR

Bis.b #0x01, &P2IES

Bis.b #0x01, &P2SIE

Bic.b #0x01, &P2IFG

Blocking	ביצוע משימות באופן טורי והמתנה לסיום בכל שלב, לא נתחיל פעולה אחרת עד סיום עם הפעולה הנוכחית, לדוגמא נבדוק באופן רציף האם קיים שינוי במצב מתג במקום קבלת פסיקה.
Non-Blocking	ביצוע משימות באופן טורי אך במידה ונדרש להמתין ונתקדם הלאה, לדוגמא קבלת הודעה, נמשיך בפעולת התוכנית עד סיום קבלת ההודעה ובמקום פרדיגמה יותר מודרנית שיכולה. להמתין לסיום להיות יותר חסכונית ויעילה.
Event Driven	העברת מידע באמצעות "אירועים" בין ישויות, בדרך כלל מבוצע באמצעות המתנה לאירועים באופן רציף פעולת התוכנית לפיהם. "האירועים" יעבירו את המידע בן חלקי התוכנית או בתצורת העבודה, לדוגמא שימוש בעכבר להזזת מצביע באופן רציף.
Interrupt driven	הפעלת תהליכים בעקבות "הפרעות" המתקבלות מרכיב חומרה פריפריאליים למעבד, בקשות הפסיקה יכולות להתחלק לסוגים שונים ורמות שונות. לא מתבצעת לרוב בדיקה האם התקיימה פסיקה אלא ביצוע פעולות אחרות בתוכנית, קבלת פסיקה באופן ישיר, ביצוע הפעולות הנדרשות מהפסיקה, חזרה לנקודה בה עצרנו בתוכנית והמשך עבודה.

## דיאגרמת FSM

