דו"ח מכין מעבדה 4 :

1. **הסבר מה היא פסיקה והסבר את הצורך בה:**

פסיקה היא אות חשמלי המתקבל ב-CPU ואפשר לשנות את סדר ביצוע הפקודות בתוכנית שכתבנו ללא בעזרת לולאות.
הצורך בפסיקה היא יצירת קשר בין האירועים הקורים מחוץ ל- CPU לצורך ביצוע פעולות רבות.
זאת אומרת, הפסיקה שולחת אותנו לרצף פקודות חדש הרלוונטי לפסיקה שנשלחה.

2. **הסבר את היתרון של שימוש בפסיקה (interrupt), לעומת תשאול (polling) מתי וכיצד נוכל לשלב בין השניים?**

כאשר משתמשים ב – polling, אנו בעצם מתשאלים אם התבצעה לחיצה, פעולה המשאירה את ה-CPU בלולאה עד שהמשימה התבצעה, דבר שגורר בזבוז ושימוש לא נכון ב-CPU.
לעומת זאת, כאשר משתמשים בפסיקה, אין צורך לתשאל אם התבצעה לחיצה וגם אין צורך להשאיר את ה-CPU בלולאה, אלא, ה- CPU עובד כרגיל ורק כאשר מתבצעת לחיצה, יעצור את משימתו העכשווית ויפנה לטיפול אליה הלחצן הפנה אותו.
ניתן לשלב ביניהם כך: בדומה למימוש של המשימה, הפסיקה עובדת מאחורי הקלעים ומשנה ערך כלשהו ואנו מבצעים את ה-תשאול בזמננו הפנוי ולא משתמשים ב-תשאול אינטנסיבי שמשפיע על ביצועי התוכנית.

3. **הסבר את שלוש סוגי הפסיקות ומה הצורך בכל סוג:**

❁ פסיקה חיצונית – פסיקה הנגרמת על ידי רכיב חומרה, ללא תלות במצב הנוכחי של הריצה בתוכנית.
הצורך בה הוא שתהיה שליטה על הבקר או על הצג על ידי רכיבים חומרתיים.
לדוגמה: שימוש במתגים שלנו כדי לשנות את הרוטינה במהלך ביצוע התוכנית.
❁ פסיקה פנימית – פסיקה שהגדרנו מראש שתשנה את רצף הפקודות המבוצע בתוכנית.
הצורך בה הוא שנגדיר סט פקודות ידוע מראש, נקפוץ לרצף פקודות מסוים כך שללא התערבות חיצונית נבצע את אותו סט פקודות.
לדוגמה: בחלק המעשי של דוח מכין שלנו אנו צריכים לייצר שפעולה תשתנה באופן אוטומטי לאחר 10שניות.
❁פסיקת תוכנה – פסיקה הנגרמת בעקבות הדלקת דגל כתוצאה מביצוע הוראה מיוחדת.
הדגל הנ"ל הינו דבר הקיים לכל מודול חומרתי והוא בעצם קורא ל-CPU לבצע רצף פקודות ספציפי בהתאם המודול שהעלה את הדגל. לכל מודול יש מספר המגדיר רמת העדיפות שלו, לדוגמה: RESET הוא בעל רמת העדיפות הגבוהה ביותר וכך ה- CPU יודע להרים את הדגל בעל רמת העדיפות הגבוהה ביותר.
הצורך בו הוא שכאשר ישנה הוראה שהוגדרה מראש כהוראה שגוררת דגל, ה- CPU יבצע סט פקודות שהוגדר מראש להוראה הנ"ל, לדוגמה: כאשר נלחץ על RESET, יקפוץ הדגל של ה- RESET ואז ה-CPU יעבור לתחילת התוכנית.

4. **הסבר את מושג אופני העבודה של הבקר, הסבר כל אופן בנפרד ומתי תבחר להשתמש בו:**

כאשר מסתיימת התוכנית, בשביל למנוע בזבוז אנרגיה, נרצה להגדיר לבקר רמות שונות של פעולה.
זאת אומרת, נוכל להגדיר לבקר איך הוא התנהל בזמן שהוא ממתין להוראה הבאה.
בין אופני העבודה של הבקר ישנם יש מספר הבדלים, ההבדלים הללו נעוצים במספר רכיבים מרכזיים, כך שלפני שנראה את ההבדלים, נסביר את הרכיבים ומה שהם עושים:
❁ CPU – יחידת הבקרה שמבצעת את הפקודות ומריצה את התוכניות.
❁ MCLK – שעון מרכזי שסופר את זמן ריצת התוכנית
❁ SMCLK – שעון בשביל רכיבים חיצוניים לCPU, כמו טיימרים. עובד על תדר גבוה.
❁ ACLK – שעון בשביל רכיבים חיצוניים לCPU, אך בניגוד ל-SMCLK, עובד על תדרים נמוכים.
❁ DCO – סוג של שעון שמייצר אות בתדירות שנקבעת על ידי רכיב חיצוני הנקרא DC Generator.
ישנן שישה אופנים שונים של עבודה של הבקר:

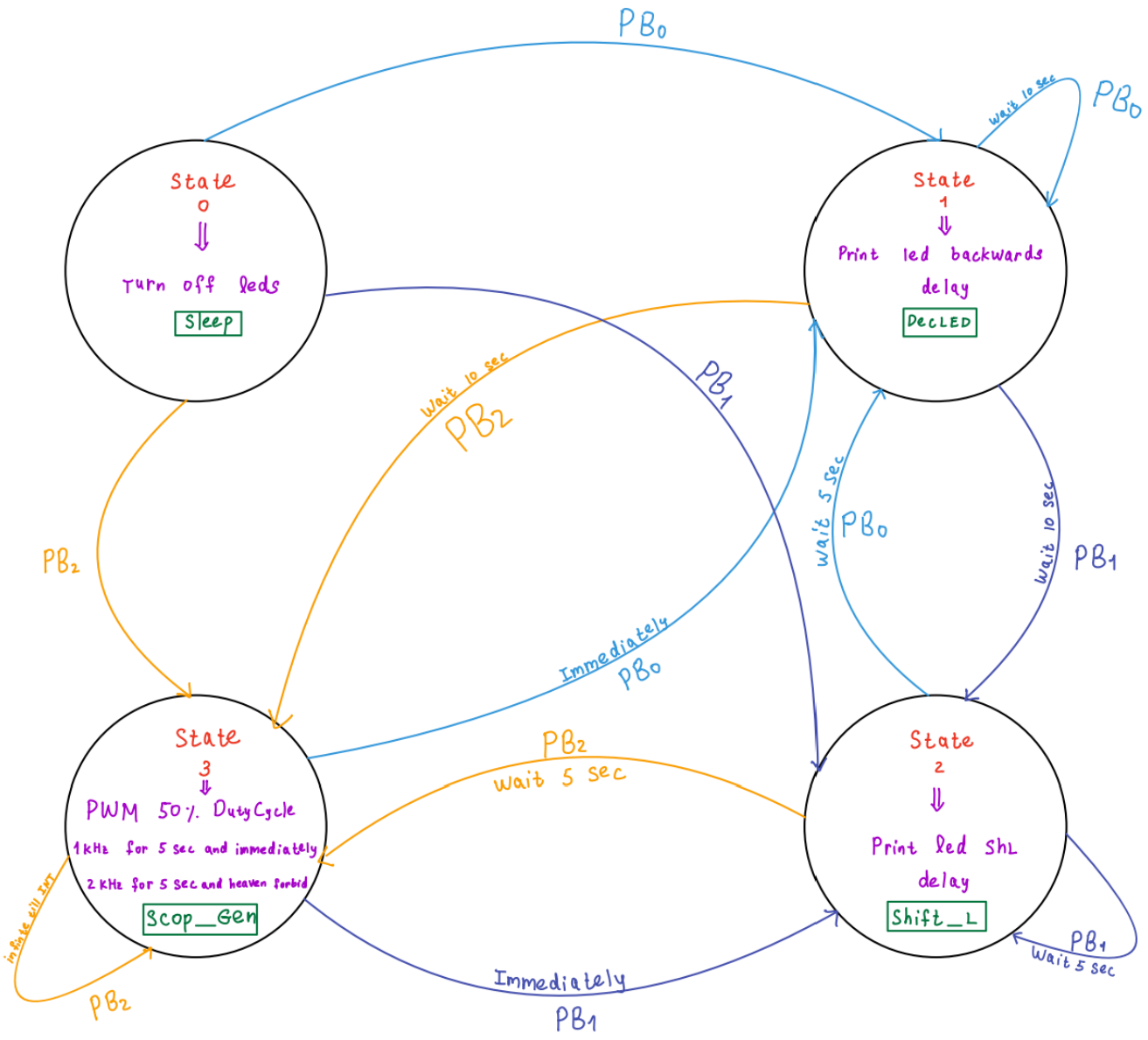
א. מצב פעיל (AM) – כאשר הבקר מריץ תוכנית פעילה. במצב זה כל הרכיבים פועלים
ב. מצב LPM0 – ה-CPU והשעון MCLK כבויים, לעומת זאת, שאר הרכיבים פעילים.
ג. מצב LPM1 – ה-CPU, MCLK וה-DCO כבויים, אך ACLK וSMCLK פעילים
ד. מצב LPM2 – רק ה-ACLK פעיל, כל השאר כבויים
ה. מצב LPM3 – רק ה- ACLKפעיל, כל השאר כבויים
ו. מצב LPM4 – כל הרכיבים כבויים.

המצבים הללו מתארים באופן יורד צריכת אנרגיה, כאשר המצב הראשון הפעיל צורך הכי הרבה אנרגיה, לעומת זאת, המצב האחרון צורך הכי מעט אנרגיה, עד כדי אפסית.

5. **רשום את השלבים כדי לקנפג את רגל P2 0.כך שבירידת מתח מ-'1 'ל- '0 'תתבצע בקשת פסיקה:**

```
BIC.B #1, &P2SEL
BIC.B #1, &P2DIR
BIS.B #1, &P2IES
BIS.B #1, &P2IE
BIC.B #0x01, &P2IFG
```

שרטוט דיאגרמת FSM:



הערה: בסוף מצב 1 המעבר חוזר למצב שינה.

מגישים :

יאיר טיירי- 207973017

עומר גראוברט- 322480971