**טיימר (זמנן- קוצב זמן) – 8254**

**PIT – Programmable Interval Timer – זמנן מרווחי זמן בר תכנות**

רכיב 8254 הוא הוא קוצב זמן מתוכנת המכיל 3 מונים בני 16 סיביות כל אחד , הסופרים כלפי מטה – Down Counters , ואוגר בקרה המאפשר לשלוט על כל מונה בנפרד.

לכל מונה יש שתי כניסות: כניסת אות שעון (Clock) וכניסת שער בקרה (Gate) ויציאה אחת (Out).

 כניסת פולסי שעון לספירה CLK

מונה

 שער GATE

 יציאה OUT

שרטוט מספר 1 : 2 כניסות ויציאה אחת מכל מונה

כניסת ה CLK היא הדק כניסת פולסי הספירה למונה.

השער GATE הוא הדק שתפקידו תלוי באיזה אופן עבודה המונה מופעל (אחד מתוך ששה אפשריים). באופן כללי ניתן לומר שתפקידו לקבוע האם המונה סופר או לא.

יציאת OUT מודיעה לנו שהמונה סיים לספור מהמספר שטוענים אותו עד ל 0 . ניתן לחבר יציאה זו להדק פסיקה של המעבד.

נציין חלק מהתכונות שלו:

1. תואם לכל המיקרו מעבדים של אינטל ורוב המיקרו מעבדים האחרים.
2. מטפל בכניסות מ DC ועד 8MHz ברכיב 8254 ועד 10MHz ברכיב 8254-2 .
3. פקודת קריאה חוזרת של הסטאטוס של הרכיב .
4. 6 אופני עבודה מתוכנתים (נקבעים על ידי פקודות תוכנה ).
5. 3 מונים בני 16 ביט כל מונה לא תלויים אחד בשני.
6. ספירה בינארית או עשרונית ((BCD.
7. עובד עם ספק כוח יחיד של 5 וולט.

הרכיב נועד לפתור בעיות משותפות בתכנון מערכות מיקרו-מחשבים כמו :

א. יצירת השהיית זמן, מבלי שהמעבד עצמו יבצע את ההשהיה. המעבד יוכל לטפל בדברים האחרים שבמערכת. המעבד יפעיל את אחד המונים לספירה ממספר כלשהו וכאשר המונה מסיים את הספירה הוא מודיע (על ידי פסיקה , לדוגמה) על סיום הספירה.

ב. ספירה של אירועים חיצוניים שונים. אם המעבד יבצע ספירה כזו הוא עסוק כל הזמן בבדיקות חוזרות של מבואות המחשב. רכיב כזה מקל בצורה רבה על עבודת המחשב.

ג. יצירת אותות זמן מחזוריים מדויקים. ניתן לקבל גל מרובע סימטרי או גל מלבני עם מנת דופק ( Duty Cycle ) משתנה .

ד. ניתן להפעיל את המונים כמערכת לא יציבה (קבלת גל מחזורי כמוזכר בסעיף הקודם ) או כמעגל חד יציב.

המספר המקסימאלי שהמונה יכול לספור הוא כאשר שולחים אליו את המספר 0 ואז הוא סופר 216 פולסי שעון עד שהוא מתאפס פעם נוספת. 65536 = 216  .

המספר המקסימאלי שהמונה יכול לספור, עבור ספירה ב BCD (ספירה עשרונית ) הוא כאשר שולחים אליו את המספר 0 ואז הוא סופר 9999+1 =10000 פולסי שעון עד שהוא מתאפס פעם נוספת.

בשרטוט מספר 2 נתון המבנה הפנימי של הרכיב ותאור ההדקים שלו.



שרטוט 2 : סכמה מלבנית ושרטוט ההדקים במבנה DIP

**הסבר השרטוט:**

השרטוט מורכב מחלק ימני שבו נראים המלבנים המרכיבים את הרכיב ומחלק שמאלי שבו רואים את הדקי הרכיב ( במבנה DIP - Dual In line Package - זיווד של 2 שורות ) .

1. בצד ימין של הסכמה המלבנית רואים את 3 המונים counter 0 , counter 1 ו counter 2 שלכל אחד מהם יש כניסת CLK כניסת GATE ויציאת OUT .
2. מצד שמאל למעלה בסכמה המלבנית רואים את חוצץ פס הכתובות - DATA BUS BUFFER - שאליו מחוברים 8 הדקים מ D0 ועד D7 . הדקים אלו מתחברים להדקי פס הנתונים של המעבד. לחוצץ יש התנגדות כניסה גבוהה ביותר והוא דואג לכך שהרכיב שלנו לא יעמיס את פס הנתונים של המעבד.
3. מתחת לחוצץ יש את מלבן "לוגיקת קריאה/כתיבה" – READ/ WRITE LOGIC . תפקידו לנהל את כיוון זרימת הנתונים מהמעבד אל אחד מהמלבנים שבתוך הרכיב ומאחד מהמלבנים החוצה אל המעבד. ההדקים המתחברים אליו הם :
4. RD - ReaD - קריאה . הדק זה מתחבר להדק הקריאה של המעבד. בעזרת הדק זה המעבד "קורא" נתון מהרכיב. ההדק פעיל בנמוך ( 0 ).
5. WR - WRite – כתיבה. ההדק מתחבר אל הדק הכתיבה של המעבד. בעזרת הדק זה המעבד כותב נתון לרכיב. ההדק פעיל בנמוך( 0 ).
6. A1 A0 – בעזרת 2 הדקים אלו המעבד מודיע למי הוא פונה בתוך הרכיב. הדקים אלו מתחברים בדרך כלל אל הדקי A1 A0 של מעבד ומכאן שהרכיב "תופס" 4 כתובות במרחב הקלט פלט של מעבד. הפנייה אל המעגלים בתוך הרכיב בעזרת 2 הדקים אלו היא לפי הטבלה הבאה :

|  |  |
| --- | --- |
|  **פניה אל :** | **A1 A0** |
| מונה 0 |  0 0 |
| מונה 1 |  0 1 |
| מונה 2 |  1 0 |
| רגיסטר מילת הבקרה(בכתיבה)  |  1 1 |

 טבלה 1 : הפנייה אל המעגלים בתוך הרכיב בעזרת A1 A0

1. CS - Chip Select - בחירת רכיב. 0 בהדק זה אומרת לרכיב "שמדברים" אליו. 1 בהדק זה אומר לרכיב שלא " מדברים " אליו.
2. CONTROL WORD REGISTER - רגיסטר מילת הבקרה. לרגיסטר זה שולחים מילת בקרה שבה מתכנתים כיצד יפעל כל מונה. באופן מעשי שולחים BYTE ולא מילה למרות שקוראים לו "מילת בקרה". שולחים ביית עבור כל מונה, כלומר אם רוצים ל,תכנת את שלושת המונים שולחים 3 בתים. רואים שרגיסטר זה מתחבר לשלושת המונים ולחוצץ. הרגיסטר קובע לכל מונה, אחד מ 6 אופני עבודה (Mode) , האם הספירה תהיה בינארית או עשרונית ואת צורת הפנייה לכל מונה (ניתן לפנות רק לחלק הנמוך (8 ביטים נמוכים) או רק לחלק הגבוה ( 8 ביטים גבוהים) או קודם לנמוך ואח"כ לגבוה (יש לרכיב 8 ביטים מ D0 ועד D7 , ולכל מונה יש 16 ביטים). כמו כן הוא יכול לקבל מכל מונה את הסטטוס שלו (האם הסתיימה ספירה ועוד (יוסבר בהמשך ).

**מבנה פנימי של מונה**

****

שרטוט 3 : מבנה פנימי של מונה

בשרטוט 3 רואים כיצד בנוי מונה. האות n מבטאת את אחד המספרים 0 עד 2 ומתאימה למונה 0 , 1 או 2 . נעבור על כל המלבנים והקשר ביניהם.

1. CONTROL WORD REGISTER – רגיסטר מילת הבקרה. מלבן זה מקבל את מילת הבקרה שהמתכנת שלח לכל מונה. הרגיסטר שולט על המלבן שמתחתיו שנקרא CONTROL LOGIC ואומר לו כיצד להפעיל את כל המעגלים שבמונה. כמו כן הוא מתחבר אל רגיסטר הסטטוס – STATUS REGISTER ואומר לו מתי לבדוק את הסטטוס של המונה.
2. CONTROL LOGIC - לוגיקת בקרה. למלבן זה מתחברים ההדקים של 2 הכניסות CLK ו GATE והיציאה OUT . מלבן זה נשלט על ידי רגיסטר מילת הבקרה – CONTROL WORD REGISTER כמו שהוסבר בסעיף א. זהו המלבן השולט על המונה בהתאם למילת הבקרה שנשלחה אל המונה.
3. STATUS REGISTER – רגיסטר המצב. זהו רגיסטר שניתן לקרא את התוכן שלו ולדעת באיזה מצב נמצא המונה. הרגיסטר מקבל את מצב ההדק OUT ואז ניתן לדעת האם בהדק יש 0 או 1 (לדוגמה - האם המונה סיים לספור או לא), כמו כן ניתן לדעת פרטים נוספים על מצב המונה ויתוארו בהמשך.
4. STATUS LATCH - נועל הסטטוס. רגיסטר הסטטוס מקבל את מצבו של המונה. כאשר נשלח פקודה לקבלת מצב רגיסטר הסטטוס, יינעל המצב של הרגיסטר על ידי מעגל נועל הסטטוס ויועבר אל פס הנתונים הפנימי – INTERNAL BUS וממנו אל חוצץ פס הנתונים ואל המעבד.
5. CE – Counting Element – האלמנט הסופר – אלו מעגלי הספירה . זה המונה בן 16 הביטים.
6. CRm Count Register Least byte - CRl - Count Register Most byte. אלו 2 אוגרי המספר, כשלכל אחד 8 ביטים. ה l הוא Byte Least significant (8 הביטים הנמוכים) , ה m הוא Most significant byte ( 8 הביטים הגבוהים) . כאשר שולחים למונה מספר שממנו רוצים שיספור , המספר לא מגיע ישירות למונה אלא ל 2 רגיסטרים אלו. המספר יעבור מ 2 הרגיסטרים אל המונה רק ברגע מסוים שתלוי באופן העבודה שבו עובד המונה.

**כששולחים BYTE לתכנת את אחד המונים אז באופן אוטומטי ה CRm ו CRl מתאפסים** !

1. OLl Output Latch Least byte ו OLm - Output Latch Most byte . נועל היציאה של 8 הביטים הנמוכים ונועל יציאה של 8 הביטים הגבוהים. כאשר רוצים לדעת לאיזה מספר הגיע המונה ניתן לשלוח פקודת נעילה שנועלת את המספר שבמונה ב 2 הנועלים האלה.

**מבנה רגיסטר מילת הבקרה**



שרטוט 4 : תבנית מילת הבקרה

נזכיר שכדי לשלוח את מילת הבקרה אל רגיסטר מילת הבקרה, יש לדאוג לכך שההדקים A1=A0=1 . הדקים אלו הם כניסות חומרה ובעזרתם המעבד קובע למי הוא פונה בתוך הרכיב.

נסביר את הסיביות במילת הבקרה:

1. SC1 SC0 – Select Counter – בחירת מונה. 2 ביטים אלו קובעים איזה מונה מתכנתים. הטבלה שמתחת 2 הביטים מראה מה מצב הביטים ואיזה מונה מתכנתים. במצב של 1 בשני ביטים אלו מבצעים Read Back Command – פקודת קריאה חוזרת (יוסבר בהמשך ).
2. RW1 RW0 – Read Write – קריאה כתיבה. בעזרת 2 ביטים אלו קובעים את אופן הפנייה אל הרכיב. בטבלה שמצד שמאל של השרטוט למטה רואים שאם 2 הביטים הם 01 בהתאמה פונים רק אל הBYTE הנמוך של המונה ( 8 ביטים נמוכים), במצב 10 פונים רק לBYTE הגבוה 8( הביטים הגבוהים) , במצב 11 פונים אל ה BYTE הנמוך ולאחריו אל הגבוה. במצב 00 מבצעים נעילה של המונים (יוסבר בהמשך ).
3. M2 M1 M0 - MODE . בעזרת 3 ביטים אלו קובעים באיזה אופן מתוך 6 מצבי עבודה נפעיל את המונה. בטבלה הימנית בשרטוט מופיע מצב הביטים ואופן העבודה המתאים לו. במקום שרשום X ניתן לשים 0 או 1 , אם כי רשום בהערה למטה בשרטוט שלמען יישומים עתידיים יש לרשום 0 ולא 1 .
4. BCD – בעזרת ביט זה קובעים האם הספירה תהיה בינארית או BCD , עשרוני מקודד בינארית, כלומר כל 4 ספרות בינאריות של המונה הן ספרה עשרונית אחת עם ערכים בין 0 ל 9 ולא מ 0 עד F . במילים אחרות – ספירה עשרונית. דוגמה : אם במונה יש את המספר 1000H ומגיע פולס ספירה אז במונה עשרוני נרד ל 0999H ובמונה בינארי נרד ל 0FFFH .

**תרגיל**

רכיב 8254 מחובר בתחום הכתובות 308h – 30bh . תכנן אותו שיפעל בצורה הבאה :

1. מונה 0 יעבוד באופן 5 , הפנייה רק לנמוך . יש לטעון אותו למספר 10H .הספירה עשרונית.
2. מונה 1 יעבוד באופן 3 , הפנייה גם "לנמוך" וגם "לגבוה". יש לטעון אליו את המספר 1234h , הספירה תהיה בינארית.
3. מונה 2 יעבוד באופן 1, פנייה גם לנמוך" וגם "לגבוה". יש לטעון אליו את המספר 10000 . ספירה בינארית.

**פתרון**

מתחום הכתובות בו נמצא הרכיב , לפי טבלה 1 - ניתן לראות באיזה כתובת נמצא כל מונה:

308H - counter0 (A1=0 A0 = 0)

309H - counter1 (A1=0 A0=1 )

30AH – counter2 (A1=1 A0=0)

30BH – control word (A1=1 A0=1)

1. מילת הבקרה למונה 0 נראית כך :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BCD | M0 |  M1 |  M2 | RW0 | RW1 | SC0 | SC1 |

0 0 0 1 1 0 1 1

נשלח את מילת הבקרה אל רגיסטר מילת הבקרה בעזרת אסמבלי 86 :

Mov al,1bh

Mov dx,30bh

Out dx,al

נשלח למונה 0 את המספר אליו רוצים שייטען :

Mov al,10h

Mov dx,308h

Out dx,al

אותן פקודות בשפת C בהנחה שנכללו קבצי הכותר וההצהרות המתאימות:

Out32(0x30b,0x1b); // control word to program counter0

Out32(0x308,0x10); // to counter0

1. מילת הבקרה למונה 1 :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BCD | M0 |  M1 |  M2 | RW0 | RW1 | SC0 | SC1 |

 0 1 1 1 0 1 1 0

התוכנית באסמבלי:

Mov al,76h

Mov dx,30bh

Out dx,al

נשלח למונה 1 את המספר אליו רוצים שייטען :

Mov al,34h

Mov dx,309h

Out dx,al ; loading least byte

Mov al,34h

Out dx,al ; loading most byte

אותן פקודות בשפת C בהנחה שנכללו קבצי הכותר המתאימים:

Out32(0x30b,0x76); // control word to program counter1

Out32(0x309,0x34); // to counter1, least byte

Out32(0x309,0x12); // to counter1, most byte

1. מילת הבקרה למונה 2 :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BCD | M0 |  M1 |  M2 | RW0 | RW1 | SC0 | SC1 |

 1 0 1 1 0 0 1 0

התוכנית באסמבלי:

Mov al,0b2h

Mov dx,30bh

Out dx,al

נשלח למונה 2 את המספר אליו רוצים שייטען : 10000=2710H

Mov al,10h

Mov dx,30ah

Out dx,al ; loading least byte

Mov al,27h

Out dx,al ; loading most byte

אותן פקודות בשפת C בהנחה שנכללו קבצי הכותר המתאימים. בשפת C היתן לבקש מהקומפיילר שיעשה את פעולת החלוקה והשארית:

Out32(0x30b,0xb2); // control word to program counter1

Out32(0x30a,10000%256); // to counter1, least byte

Out32(0x30a,10000/256); // to counter1, most byte

**אופני הפעולה**

כדי להבין את אופני העבודה נגדיר מספר מושגים :

1. CLK PULSE – פולס שעון – ההגדרה של פולס שעון אומרת : עליה ל 1 שאחריה יש ירידה ל 0 בהדק ה CLK של מונה.

 ב. TRIGGER – התנעה – עליה מ 0 ל 1 בהדק ה GATE של המונה.

1. COUNTER LOADING - טעינת מונה. מעבר של המספר מה CRm וה CRl אל ה CE , כלומר מרגיסטרי המונה אל המונה עצמו.

**האופנים השונים**

קיימים 6 אופני עבודה שונים. לכל אופן עבודה משורטטים 3 קבוצות של צורות גלים. בכל קבוצה יש את הדק ה WR של הרכיב, הדק ה CLK של המונה, הדק ה GATE של המונה , יציאת ה OUT של המונה ומתחת הדק זה מסומן מה המספר הנמצא במונה. הקבוצה הראשונה מתארת את אופן הפעולה הרגילה ואילו ב 2 הקבוצות האחרות נותנים דוגמה מה קורה אם נשנה את ה GATE או נשלח מספר חדש לספירה.

נעבור על אופני העבודה אחד אחרי השני. נסביר בהרחבה את צורות הגל של אופן 0 ואז נוכל להבין את צורות הגל באופנים האחרים.

1. **אופן 0:** פסיקה בסיום ספירה (Interrupt on Terminal Count).

 אופן 0 משמש בדרך כלל לספירת אירועים. כותבים למונה הרצוי מילת בקרה שיעבוד באופן 0 ושולחים אליו מספר שממנו רוצים שיספור. כאשר כותבים לו את מילת הבקרה היציאה OUT יורדת ל 0 . כאשר המונה מסיים את הספירה היציאה OUT עולה ל 1 . ניתן לחבר את היציאה למבוא פסיקה של מעבד ולקבל פסיקה בסיום הספירה של המונה. נתאר את הפעולה באופן זה בעזרת צורות הגלים.



שרטוט מספר 5 : אופן 0 של מונה.

בשרטוט 5 מתואר אופן עבודה 0 . זוהי קבוצה ראשונה מתוך 3 הקבוצות עבור כל אופן. נסביר את השרטוט:

צורת הגל העליונה מתארת את הדק הכתיבה המתחבר מהמעבד אל הרכיב. רואים שהמעבד ביצע 2 מחזורי כתיבה. במחזור הראשון הוא שלח את מילת הבקרה (מיוצג על ידי CW=10 ) ובמחזור השני הוא שלח את המספר שאליו המונה ייטען וממנו הוא יתחיל לספור ( LSB=4 ). נבדוק מה אומרת מילת הבקרה CW=10H . מילת הבקרה נשלחת אל רגיסטר מילת הבקרה בכתובת שבה הדקי הרכיב A1=A0=1 .

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  0 |  0  |  0 |  0 |  1  |  0 |  0 |  0 |

 ספירה בינארית אופן 0 פנייה רק אל ה BYTEהנמוך תכנות מונה 0

רואים שמתכנתים את מונה 0 לעבודה באופן 0 , ספירה בינארית , הפנייה רק לבית הנמוך של המונה.

במחזור כתיבה הבא נשלח המספר 4 אל מונה 0 עצמו : LSB=4 . ההדקים A1=A0=0

 מתוך צורות הגל רואים שאחרי כתיבת מילת הבקרה היציאה OUT יורדת לנמוך (0) ותישאר כך עד שהספירה במונה תגיע ל 0. הירידה ל 0 מתוארת בעזרת הקו האנכי השמאלי המקווקו. צורת הגל בנקודה זו אומרת שאם היציאה הייתה ב 1 היא יורדת ל 0 ואם היציאה הייתה ב 0 היא ממשיכה להיות 0 .

 הקו המקווקו השני מראה את מעבר המספר מה CR אל ה CE . יש לשים לב שטעינת המספר למונים בוצעה רק אחרי דופק השעון הבא. לפני הקו המקווקו היה במונה את המספר N (מספר כלשהו). ברגע הטעינה המספר במונה הוא 04 כאשר 0 בחלק הגבוה של המונה ו 4 בחלק הנמוך שלו. פולס זה איננו מפחית את המספר, כך שעבור ספירה התחלתית ממספר N היציאה OUT לא יורדת ל 0 אלא לאחר N+1 פולסי ספירה.

 מרגע זה המונה מתחיל לספור ( בהדק GATE יש 1 למרות שבשרטוט זה לא כל כך ברור.. ). רואים שאחרי כל פולס שעון המספר במונה קטן. כשהספירה מגיעה ל 0 היציאה עולה לגבוה (1 ) ותישאר בגבוה עד שתתחיל ספירה חדשה או עד שתיכתב מילת בקרה חדשה לעבודה באופן 0. מהשרטוט ניתן לראות שהמונה עצמו ממשיך לספור אבל היציאה OUT תישאר בגבוה.\

בקבוצת צורות הגלים השנייה לאופן 0 – שרטוט מספר 6 - ניתן לראות איך משפיע ה GATE על הספירה של המונה.



שרטוט מספר 6 – אופן 0 עם הפסקת הספירה.

משרטוט 6 רואים ששלחנו את אותה מילת בקרה ולמונה עצמו שלחנו את המספר 3. המונה התחיל לספור וכאשר ירד ל 2 הורדנו את רגל ה GATEל 0 . המונה עוצר את הספירה שלו על המספר 2 . רק ההחזרה של ה GATE ל 1 ממשיכה את הספירה של המונה.

קבוצת הגלים השלישית של אופן 0 מתוארת בשרטוט 7 .



שרטוט מספר 7 – אופן 0 עם טעינת מספר חדש למונה לפני שסיים ספירה קודמת.

 מילת הבקרה הנשלחת היא כמו בדוגמאות הקודמות, והמספר שבו טוענים את המונה הוא 3 . אם נכתב מספר חדש למונה הוא ייטען רק לאחר פולס השעון הבא והספירה תתחיל מהמספר החדש. בשרטוט רואים שהמונה החל בספירה שלו והגיע עד ל 1 . בינתיים נשלח מספר חדש – 2 למונה. בפולס השעון הבא המונה נטען למספר החדש ומתחיל לספור ממנו.

 אם צורת הפניה היא על ידי 2 בתים (קודם הנמוך ואח"כ הגבוה), יקרה התהליך הבא : 1. כתיבת ה BYTE הראשון תעצור את הספירה היציאה OUT יורדת מיד ל 0 (לא נדרש פולס שעון). 2. כתיבת ה BYTE השני מאפשרת למספר החדש להיטען בפולס השעון הבא.

2. **אופן 1:** חד-יציב בר התנעה חוזרת בחומרה (Hardware Retriggerable One-Shot).

למעגל חד יציב יש 2 מצבים :

א. מצב יציב שבו יציאת המונה – OUT - נמצאת ב 1 .

ב. מצב לא יציב שבו היציאה OUT נמצאת ב 0 . למצב הלא יציב מגיעים על ידי התערבות חיצונית. לאחר זמן מסוים המערכת חוזרת למצבה היציב.

המעבר למצב הלא יציב מתבצע כאשר ניתן טריגר בשער המונה. . לאחר שהמונה יספור (מהמספר שיש בו ) ויגיע ל 0 הוא יחזור למצבו היציב והיציאה תחזור ל 1 . אם נטען למונה מספר גדול יותר אז הזמן הלא יציב יהיה גדול יותר ולהפך.

שרטוט מספר 8 מתאר את 3 הקבוצות המתאימות לאופן 1 .

**הקבוצה 1** : מילת הבקרה הנשלחת 12H אומרת שמתכנתים את מונה 0 לעבודה באופן 1 , הפנייה רק לחלק הנמוך של המונה והספירה בינארית.

אחרי כתיבת מילת הבקרה לרגיסטר הבקרה , היציאה OUT עולה ל 1 . לאחר מכן שולחים לרגיסטרי המונה את המספר 3. המספר עובר למונה רק כאשר יש עלייה בשער ולאחריה יש פולס שעון. אז המספר נטען למונה, היציאה יורדת ל 0 (עוברים למצב הלא יציב) והספירה מתחילה. כאשר המונה יספור 3 פולסים ויגיע ל 0 היציאה OUT חוזרת ל 1 וחוזרים למצב היציב. פולס התנעה חדש ב GATE גורם אחרי פולס שעון נוסף למעבר למצב הלא יציב, ירידת OUT ל 0 ושוב ספירה עד ה 0 וחזרה למצב היציב( החזרה למצב היציב בפעם הזו לא נראית בשרטוט) .

**הקבוצה 2**: צורות הגלים בקבוצה השנייה דומים לאלו שבקבוצה הראשונה. מילת הבקרה והמספר זהים לצורת הגלים הראשונה. ההבדל הוא שבזמן הספירה של המונה מ 3 ל 2 ובמעבר מ 2 ל 1 הכנסנו טריגר נוסף בשער. הדבר גורם לטעינת מונה חדשה למספר 3 כאשר מגיע פולס השעון הבא והמונה שוב מתחיל לספור מהמספר הראשוני – 3 כלפי מטה.

**הקבוצה 3**: ההבדל בין קבוצה זו ל 2 הקבוצות הקודמות הוא שכאן בזמן שהמונה סופר שלחנו מספר חדש . המונה יסיים את הספירה שלו ורק בטריגר הבא שניתן בשער הוא ייטען למספר החדש – 4 .



קבוצה 3

קבוצה 2

קבוצה 1

שרטוט מספר 8 – אופן 1

**3. אופן 2:** מחולל קצב (RATE GENERATOR).

 באופן זה מתפקד המונה כמחלק תדר ב N . הוא משמש בדרך כלל ליצירת פסיקה בשעון זמן אמת. לאחר שליחת מילת הבקרה היציאה OUT תהיה בגבוה. המונה ייטען למספר שנשלח אליו בפולס השעון הבא ויתחיל לספור. כאשר המונה יגיע למספר 1 היציאה OUT יורדת ל 0 למשך פולס שעון אחד וחוזרת ל 1 . המונה שוב נטען למספר הראשוני שלו ושוב ממשיך לספור. התהליך הוא מחזורי ונקבל גל מלבני שבו היציאה OUT נמצאת ב 1 למשך זמן של N-1 פולסי שעון וזמן ל 0 למשך זמן של פולס שעון אחד.

GATE=1 מאפשר ספירה. GATE=0 מפסיק את הספירה. כאשר נחזיר את ה GATE ל 1 אז בפולס השעון הבא ייטען המונה למספר שנשלח אליו ותתחיל ספירה חדשה.

שרטוט מספר 9 מתאר את צורות הגל של אופן 2 .



קבוצה 1

קבוצה 2

קבוצה 3

שרטוט מספר 9 - אופן 2

 גם בשרטוט זה 3 קבוצות של צורות גלים. **בקבוצה 1** רואים ששלחנו את מילת הבקרה 14H האומרת שמתכנתים את מונה 0 לאופן 1 , הפנייה רק "לנמוך" והספירה בינארית. היציאה OUT עוברת לגבוה. לאחר מכן שולחים למונה את המספר 3 . המונה נטען למספר בפולס השעון הבא. 2 פולסי שעון המונה יהיה ב 1 , כאשר המונה מגיע ל 1 היציאה יורדת ל 0 למשך פולס שעון אחד. היציאה עולה ל 1 , המונה נטען שוב למספר 3 והתהליך חוזר על עצמו באופן מחזורי. קיבלנו כאן גל מלבני עם מנת דופק – DUTY CYCLE של 2/3 כלומר כ 67% . אם היינו שולחים למונה את המספר 10 היינו מקבלים 9 פולסי שעון שבהם היציאה OUT נמצא ב 1 ופולס שעון אחד שבו היציאה נמצאת ב 0 , כלומר מנת דופק של 90% .

 **הקבוצה 2** דומה לראשונה, ההבדל הוא בהדק ה GATE שאותו מורידים ל 0 באמצע הספירה. ניתן לראות שהספירה נעצרה על המספר 2 , עד שה GATE עולה ל 1 , המונה נטען שוב למספר 3 והמונה מתחיל הספירה ממספר זה.

 **בקבוצה 3** של צורות הגל שולחים את המספר 4 למונה. כאשר הוא סופר שלחנו אליו את המספר 5 . המונה ממשיך את הספירה שלו ורק בסיום הספירה הוא ייטען למספר החדש – 5 , ויספור ממנו.

 **דוגמה**: רכיב 8254 מחובר בתחום הכתובות 468CH - 468FH . לכניסת CLK2 נכנס גל מרובע בתדר של 1KHz . השער נמצא ב 1 . רשום קטע תכנית להוצאת גל מרובע בתדר 50Hz ביציאת OUT2 . מה מנת הדופק – Duty Cycle של הגל ?

 **פתרון** :

 נתאר את מילת הבקרה :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  0 |  0  |  1 |  0 |  1  |  0 |  0 |  1 |

 ספירה בינארית אופן 2 פנייה רק אל ה BYTEהנמוך תכנות מונה 2

קטע התוכנית באסמבלי: ; address of control word mov dx,468fh

 Mov al,94h

 Out dx,al

Mov al, 20 ; to divide : 1KHz / 20 = 50Hz

Mov dx,468eh ; address of counter 2

Out dx,al

התכנית בשפת C , בהנחה שהוגדרו קבצי הכותר המתאימים וההצהרות הנכונות:

Out32(0x468f,0x94); // control word

Out32(0x468e,20); // to counter 2 to divide 1KHz/20=50Hz

הגל המלבני שנקבל יהיה עם 19 פולסי שעון שבהם OUT=1 ופולס אחד של 0= OUT. מכאן שמנת הדופק היא : 19/20 כלומר 95%.

**הערה** : ניתן היה לשלוח מילת בקרה שבה הספירה תהיה ב BCD ולטעון לחלק הנמוך של המונה את המספר 20H .

**4. אופן 3:** גל מרובע (SQUARE WAVE).

 משמש ליצירת גל מרובע. אופן3 דומה לאופן 2 חוץ ממנת הדופק – Duty Cycle של היציאה OUT. אחרי שליחת מילת הבקרה היציאה OUT תהיה בגבוה. המונה ייטען למספר ששולחים אליו בפולס השעון הבא. כאשר חצי מהספירה מסתיימת היציאה OUT עוברת ל 0 למשך חצי הספירה הבאה היציאה OUT עולה שוב ל 1 והמונה נטען למספר ששלחנו אליו. אופן 3 הוא מחזורי ונקבל ביציאה OUT גל מרובע בתדר הכניסה לחלק במספר N ששלחנו למונה. שרטוט 10 מתאר את 3 הקבוצות באופן זה.



קבוצה 2

קבוצה 1

קבוצה 3

שרטוט 10 - אופן 3

 **בקבוצה 1** רואים ששלחנו את מילת הבקרה 16H האומרת שמתכנתים את מונה 0 לאופן 3 , הפנייה רק לנמוך והספירה בינארית. לאחר מילת הבקרה היציאה OUT עוברת לגבוה. לאחר מכן שלחנו למונה את המספר 4 . בפולס השעון הבא המספר נטען למונה והוא מתחיל לספור. הספירה מתבצעת מהמספר 4 , בפולס שעון הבא עוברת ל 2 ובפולס שעון הבא היציאה מתחלפת ל 0 והמונה שוב נטען ל 4 . גם כאן הספירה יורדת ב 2 כל פולס שעון וחוזר חלילה.

 **בקבוצה 2** שלחנו מספר אי זוגי. המונה נטען למספר 5 וסופר מ 4 ל 2 ול 0 , מחליף את מצב היציאה ל 0 ונטען למספר 4 , אבל עכשיו הוא סופר ומגיע ל 2 ובפולס שעון הבא מחליף מצב ל 1 . במקרה של מספר אי זוגי נקבל גל מלבני שבו מצב ה 1 ארוך בפולס שעון אחד ממצב ה 0 .

 **קבוצה 3** שונה בהדק ה GATE שאותו מורידים באמצע הספירה ל 0 . הספירה עוצרת ורק כאשר נעלה את ה GATE בחזרה ל 1 אז בפולס השעון הבא המספר 4 ייטען למונה והגל המרובע יתחיל להיווצר מחדש.

 **דוגמה:**

 רכיב 8254 מחובר בתחום הכתובות 20H - 23H . למונה 1 שלו מחובר גל מרובע בתדר של 1MHz . ל GATE2 חיברו מתח של 5 וולט ('1' לוגי ). רשום קטע תכנית לקבלת גל מרובע בתדר של 1KHz ביציאה OUT2 .

 **פתרון** : כדי לקבל תדר של 1KHz יש לחלק את תדר ה 1MHz ב 1000 עשרוני ( 3E8H).

 ניתן להגיד למונה לספור עשרוני צ 1000H , או בינארית מ 3E8h . ניתן לעשות זאת בשתי דרכים. בהתחלה נתאר ספירה עשרונית. אם נשלח לחלק העליון של המונה 10H אז ה BYTE הנמוך מתאפס והא יספור כלפי מטה כאשר מ 1000H הוא יעבור ל 0999H ( ולא ל 0FFFH ) וכך הלאה.

 נתאר את מילת הבקרה :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  1 |  1  |  1 |  0 |  0  |  1 |  1 |  0 |

 ספירה עשרונית אופן 3 פנייה רק אל ה BYTEהגבוה תכנות מונה 1

אסמבלי: ; address of control word register mov dx,23h

 Mov al,67h

 Out dx,al

 Mov al, 10h ; to divide : 1MHz / 1000 = 1KHz

 Mov dx,21h ; address of counter 1

 Out dx,al

 התכנית בשפת C , בהנחה שהוגדרו קבצי הכותר המתאימים וההצהרות הנכונות:

 Out32(0x23,0x67); // control word

 Out32(0x21,0x10); // to counter 1 to divide 1MHz/1000=1KHz

 **דרך נוספת** : יכולים לשלוח מילת בקרה שבה פונים לחלק הנמוך ואח"כ לגבוה, לבקש ספירה בינארית והיות ו 3E8H= 1000 לשלוח לחלק הנמוך של המונה את המספר E8H ולחלק הגבוה את המספר 3 .

נתאר את מילת הבקרה :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  0 |  1  |  1 |  0 |  1  |  1 |  1 |  0 |

 ספירה בינארית אופן 3 פנייה גם אל ה BYTEהנמוך וגם לגבוה תכנות מונה 1

אסמבלי: mov al,76h

 Out 23h,al

 Mov al, 0E8h ; המספר לביית הנמוך

 Out 21h,al ; הוצאת המספר לחלק הנמוך של המונה

 Mov al,3

 Out 21h,al ; שליחת המספר לחלק הגבוה של המונה

התכנית בשפת C , בהנחה שהוגדרו קבצי הכותר המתאימים וההצהרות הנכונות:

 Out32(0x23,0x76); // control word

 Out32(0x21,0xE8); // out32(0x21,1000%256); יכולנו גם

 Out32(0x21,3); // out32(0x21,1000/256); יכולנו גם

**5. אופן 4:** פעימת התנעה בתוכנה (SOFTWARE TRIGGERED STROBE).

 היציאה OUT תעלה ל1 אחרי קבלת מילת הבקרה. המספר ייטען בפולס השעון הבא והמונה יתחיל לספור. כאשר הוא מגיע ל 0 אז היציאה OUT יורדת ל 0 למשך פולס שעון אחד ועולה חזרה ל 1 . הספירה איננה מחזורית ורק כאשר נשלח מספר אל המונה הספירה תתחיל פעם נוספת. GATE=1 מאפשר ספירה ו GATE=0 עוצר את הספירה. יש לזכור שעבור מספר N ששלחנו אל המונה נקבל 0 בהדק OUT רק אחרי N+1 פולסי שעון.

שרטוט מספר 11 מתאר את אופן העבודה 4 .



קבוצה 3

קבוצה 2

קבוצה 1

שרטוט מספר 11 - אופן 4

 גם כאן יש 3 קבוצות של צורות גלים.

**בקבוצה 1** רואים שלאחר שליחת מילת הבקרה היציאה עוברת ל 1 . מילת הבקרה 18H אומרת שמתכנתים את מונה 0 לאופן עבודה 4 , הפנייה רק לנמוך והספירה בינארית. לאחר שליחת המספר 3 למונה הוא נטען בפולס השעון הבא ומתחיל לספור ממנו. כאשר הוא מגיע ל 0 היציאה יורדת ל 0 לפרק זמן של פולס שעון אחד וחוזרת ל 1 .

**קבוצה 2** מתארת את השפעת ה GATE על הספירה. רואים שכאשר הוא ב 0 אין ספירה. המספר 3 נטען למונה אחרי פולס השעון המגיע לאחר ששלחנו את המספר למונה אבל המונה נשאר ב 3 כי ב GATE יש 0 . רק העלאת ה GATE ל 1 תתחיל את הספירה.

**קבוצה 3** מתארת מה קורה אם שולחים מספר חדש אל המונה לפני שהספירה הקודמת מסתיימת. ניתן לראות ששלחנו את המספר 3 אל המונה והוא נטען למספר זה בפולס השעון הבא. הספירה מתחילה ולפני סיומה שלחנו את המספר 2 . המונה ייטען למספר החדש 2 ויתחיל את הספירה ממנו מבלי לסיים את הספירה הקודמת.

1. **אופן 5:** פעימת התנעה בחומרה ברת התנעה חוזרת

 (HARDWARE TRIGGERED STROBE RETRIGGERABLE).

 היציאה OUT תעלה ל 1 אחרי קבלת מילת הבקרה. המספר ייטען בפולס השעון הבא אחרי שיש עלייה מ 0 ל 1 בשער והמונה יתחיל לספור. כאשר המונה מגיע ל 0 אז היציאה OUT יורדת ל 0 למשך פולס שעון אחד ועולה חזרה ל 1 . הספירה איננה מחזורית ורק כאשר ניתן עלייה נוספת בשער הספירה תתחיל פעם נוספת. יש לזכור שעבור מספר N ששלחנו אל המונה נקבל 0 בהדק OUT רק אחרי N+1 פולסי שעון.

שרטוט מספר 12 מתאר את אופן העבודה 5 .



קבוצה 3

קבוצה 2

קבוצה 1

שרטוט מספר 12 - אופן 5

 גם כאן יש 3 קבוצות של צורות גלים.

**בקבוצה 1** רואים שלאחר שליחת מילת הבקרה היציאה עוברת ל 1 . מילת הבקרה 1AH אומרת שמתכנתים את מונה 0 לאופן עבודה 5 , הפנייה רק לנמוך והספירה בינארית. לאחר שליחת המספר 3 למונה הוא נטען בפולס השעון שיגיע אחרי שניתן עלייה מ 0 ל 1 בשער ואז המונה מתחיל לספור. כאשר הוא מגיע ל 0 היציאה יורדת ל 0 לפרק זמן של פולס שעון אחד וחוזרת ל 1 . רק עלייה נוספת ב GATE תטען את המספר למונה (בפולס השעון הבא) ונקבל ספירה נוספת. כדאי לשים לב שמספיקה עלייה בשער ואין צורך להשאיר את ה GATE ב 1 כדי שהמונה יספור (כמו באופני עבודה אחרים ).

**קבוצה 2** מתארת את השפעת עלייה מ 0 ל 1 ב GATE על הספירה. רואים שכאשר ה GATE עולה מ 0 ל 1 המספר נטען למונה והוא מתחיל לספור מהמספר 3 שנשלח אליו. לפני שהספירה הסתיימה נשלחה התנעה נוספת ב GATE על ידי עלייה נוספת מ 0 ל 1 . המונה איננו מסיים את ספירתו אלא נטען למספר 3 פעם נוספת בפולס השעון הבא ומתחיל ספירה חדשה.

**קבוצה 3** מתארת מה קורה אם שולחים מספר חדש אל המונה לפני שהספירה הקודמת מסתיימת. ניתן לראות ששלחנו בפעם הראשונה את המספר 3 אל המונה והוא נטען למספר זה בפולס השעון הבא אחרי שיש התנעה בשער על ידי עלייה מ 0 ל 1 . הספירה מתחילה ולפני סיומה שלחנו למונה מספר חדש (את המספר 5) . המונה לא ייטען למספר החדש (5) אלא יסיים את הספירה שלו. בהתנעה הבאה , (בעלייה מ 0 ל 1 בשער) הוא ייטען למספר 5 בפולס השעון הבא ויתחיל לספור מ 5 .

**קריאת סטאטוס וקריאת הערך שנמצא במונה**

ישנן 3 דרכים כדי לקרא את הערך שנמצא במונה:

1. קריאה "פשוטה"
2. נעילת מונה
3. קריאת פקודה חוזרת שבה ניתן גם לדעת מה הסטאטוס של המונה (יוסבר בהמשך)
4. **קריאת מונה פשוטה**

מבצעים קריאה רגילה של המונה. בעזרת ההדקים A1 A0 בוחרים את המונה . יש לעצור את ספירת המונה על ידי הפסקת פולסי השעון או הפסקת השער כי אחרת הקריאה יכולה להתבצע כאשר יש שינוי מצב במונה ונקבל קריאה לא נכונה.

**דוגמא** :

רכיב 8254 מחובר בתחום הכתובות 90h – 93h . מונה 1 תוכנת כך שהוא עובד באופן 0 והפנייה היא גם לנמוך וגם לגבוה. רשום פקודות לקריאת הערך במונה.

**פתרון**

בהנחה שעצרנו את פולסי השעון או הורדנו את מצב השער ל 0 . נבצע 2 קריאות , הקריאה הראשונה תיתן את המספר בחלק הנמוך של המונה והקריאה השנייה את החלק הגבוה שלו. נתחיל עם אסמבלי ונשמור את התוצאה ברגיסטר BX.

In al,91h

Mov bl,al

In al,91h

Mov bh,al

בשפת C : בהנחה שהוגדרו קבצי הכותר המתאימים :

int counter1;

counter1=inp32(0x91); // קריאת החלק הנמוך

counter1=counter1+inp32(0x91)\*256; // קריאת החלק הגבוה וחיבור מתאים עם החלק הנמוך

1. **פקודת נעילת מונה**

בשיטה זו , שולחים מילת בקרה אל רגיסטר הבקרה (A1=A0=1 ) ובעזרת הביטים D4 D5 שבהם שמים 0 ו 0 אומרים שמתבצעת פעולת נעילת מונה. המונה נבחר בעזרת הביטים D7 D6 .

נתאר את מילת הבקרה :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  X |  X |  X |  X |  0  |  0 |  SC0 |  SC1 |

 פקודת נעילת מונה איזה מונה נועלים



הערך שקיים במונה שנבחר מועבר מהמונה עצמו וננעל בנועלים OLl ו OLm . המונה עצמו ממשיך לספור. המספר ננעל בנועלים ועכשיו כשנקרא מהמונה נקבל את הערך שיש במונה ברגע הנעילה. צורת הקריאה היא על פי אופן הפנייה שהמונה תוכנת (רק לנמוך או רק לגבוה או קודם לנמוך ואח"כ לגבוה). המספר שננעל נשאר שם עד שנבצע קריאה של המונה או עד שלרגיסטר הבקרה של המונה תישלח פקודת תכנות חדשה ורק אז "משתחרר" המספר שבנועלים). אם נתנו פקודת נעילה ולא ביצענו קריאה או תכנות חדש של המונה ואח"כ שלחנו פקודת נעילת מונה נוספת (לאותו המונה), אז הנועלים לא מקבלים את המספר הקיים במונה בפקודת הנעילה השנייה אלא נשאר בהם המספר מהנעילה הראשונה !

**דוגמה**:

רכיב 8254 מחובר בתחום הכתובות 68h – 6bh . מונה 2 תוכנת עם אופן פנייה לנמוך בלבד. רשום קטע תכנית שתנעל את מצב מונה 2 ותקרא את הערך שיש במונה.

**פתרון**:

נתחיל עם אסמבלי:

Mov al,80h ; פקודת נעילה למונה 2

Out 6bh,al ; הוצאת הפקודה לרגיסטר הבקרה

In al,6ah ; קריאת הערך שבמונה 2

בשפת C בהנחה שהוצהרו קבצי הכותר המתאימים :

Int counter2;

Out32(0x6b,0x80);

Counter2=Inp32(0x6a);

1. **פקודת קריאה חוזרת – Read Back Command**

פקודה זו מאפשרת לבדוק את הערך שבמונה ו/או את מצב התכנות הנוכחי של המונה, הדק היציאה OUT של המונה ואת דגל "Null Count " שלו. ניתן לשלוח פקודה אחת למספר מונים במקביל.

הפקודה נכתבת לרגיסטר הבקרה ויש לו את הפורמט הבא (שרטוט מספר 13 ) :



שרטוט מספר 13 – פורמט מילת הבקרה בפקודת קריאה חוזרת

בסיביות D7 D6 נשים 1 שאומרים לרגיסטר הבקרה בפקודת קריאה חוזרת.

הביט D5- COUNT - האם רוצים לנעול את הערך שבמונה/מונים . 0 בביט נועל את הערך שבמונה.

הביט D4 - STATUS - האם רוצים לנעול את סטאטוס המונה/מונים. 0 בביט נועל את סטאטוס המונה.

הביטים D3 D2 D1 - קובעים מיהו המונה. CNT0 - מונה 0 . CNT1 – מונה 1 . CNT2 – מונה 2. אם נשים 1 בביט קובעים שהמונה נבחר. ניתן לבחור יותר ממונה אחד.

בביט D0 חייבים לשים 0 .

**דוגמא** :

רשום קטע תכנית שינעל את הערך ואת הסטאטוס של מונה 0 ומונה 2 . הרכיב 8254 נמצא בכתובת 6ch-6fh .

**פתרון** :

המילה שנשלח אל רגיסטר הבקרה תיראה כך:



 0 1 0 1 0 0 1 1

באסמבלי:

Mov al,0cah

Out 6fh,al

בשפת C (בהנחה שהוצהרו קבצי הכותר המתאימים ) :

Out32(0x6f,0xca);

כאשר מבצעים נעילה גם לסטאטוס וגם לערך אז קריאת המונה הראשונה מהמונה היא של הסטאטוס והקריאה או הקריאות הבאות הן של הערך (תלוי באופן הפנייה שהמונה תוכנת ).

כאשר מבצעים קריאה של סטאטוס המונה נקבל ביית בפורמט הבא :



שרטוט מספר 14 – ביית הסטאטוס

הביטים מ D0 ועד D5 הם 6 הביטים שברגיסטר מילת הבקרה ששלחנו לתכנת את המונה. הביט D6 מראה האם המספר ששלחנו למונה הועבר מרגיסטרי המונה (CR) אל המונה עצמו (CE). 0 מתאר שהמספר הועבר. 1 מציין שהמספר ששלחנו עדיין לא הועבר למונה.

הביט D7 מציין מה מצב הדק היציאה OUT של המונה. ניתן לבצע עליו POLLING (שאילתה ) בתוכנה ולדעת מה מצב הרגל מבלי לחבר את הרגל עצמה כפסיקה או כל שימוש חומרה אחר.

**פעולת הרכיב 8254 במחשב PC.**

הרכיב 8253 היה קיים במחשבי XT PC/ והרכיב 8254 הוא שדרוג של ה 8253 והחליף אותו החל ממחשבי PC/AT . ה 8253 עובד עם תדרים עד 2 מגה הרץ וה 8254 עד 10 מגה הרץ. לשניהם אותו סידור הדקים. במחשבים האישיים החדשים יותר הוא חלק מסט הרכיבים של ה southbridge (גשר דרומי) של המחשב הכולל בין השאר רכיבים השולטים על הכונן הקשיח, בקר מסך, מחברי USB ועוד.

הרכיב תופס 4 כתובות במרחב הקלט פלט של המחשב, מכתובת 40h ועד 43H לפי הסדר הבא :

**מונה 0** נמצא בכתובת **40h**, יוצר כ 18.2 פסיקות בשנייה ועסוק בהפעלת שעון ותאריך במחשב.

**מונה 1** נמצא בכתובת **41h**, וקובע את זמני הריענון של הזיכרון הדינמי במחשב.

**מונה 2** נמצא בכתובת **42h**, ומפעיל את הרמקול הפנימי במחשב, ליצירת צפצופים שונים כאותות אזהרה למצבים שונים.

**אוגר הבקרה** נמצא בכתובת **43h**.

מונה 2 הוא היחידי שאנחנו יכולים להשתמש בו מבלי לפגוע בפעולה הרגילה של המחשב.

**הפקת צלילים ברמקול של המחשב האישי**

לכל 3 המונים מתחבר גל מרובע בתדר של 1.193180 מגה הרץ. גל זה מתחבר להדקי CLK של שלושתם. לדוגמה : אם נרצה צליל בתדר של 660 הרץ ביציאת מונה 2 נעבוד עם המונה באופן 3 היוצר גל מרובע ונחלק את התדר /660=18071193180 . נטעין גם את החלק הנמוך ( ל 07 ) וגם את החלק הגבוה ל (18H) ונספור בעשרוני ונקבל ביציאת המונה את תדר ה 660 הרץ שרוצים.

השער – GATE של מונה 2 של ה 8254 נשלט על ידי ביט 0 של פורט61H של המחשב האישי. יציאת OUT2 של המונה מתחברת לביט 1 של פורט 61H . השרטוט הבא מתאר כיצד לחבר את מונה 2 אל רמקול המחשב :

8254

 MHz1.193180 CLK2 =

מונה 2

 GATE2

רמקול

מגבר

 OUT2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  D0 |  D1 |  D2 |  D3 |  D4 |  D5 |  D6 |  D7 |

 פורט 61H

שרטוט מספר 15 : חיבור הרמקול אל מונה 2 של ה 8254 במחשב PC כולל פורט 61H .

מהשרטוט ניתן לראות שכדי לשמוע ברמקול יש להעביר אל הרמקול את הגל המרובע שב OUT2 דרך מגבר השמע. לכן יש לשים '1' בביטים D0 ו D1 של פורט 61H . יש להימנע מלשנות את הביטים 2 עד 7 של הפורט השולטים על רכיבי חומרה שמחשב. בהנחה שמונה 2 תוכנת להוצאת גל מרובע בתדר הרצוי אז התוכנית שתוציא אל הרמקול את הגל המרובע היא :

באסמבלי :

In al,61h ; הכנסת מצב פורט 61 הקסה

Or al,3 ; כדי לקבל בביטים 0 ו 1 '1' ORפעולת

Out 61h,al ; הוצאה לפורט 61 הקסה

כדי להפסיק את ההשמעה ברמקול נרשום :

In al,61h ; הכנסת מצב הפורט

And al,0FCH ; איפוס הביטים 0 ו 1 מבלי לפגוע בביטים הגבוהים

Out 61h,al ; הוצאת הנתון אל הפורט

ובשפת C , בהנחה שהוצהרו קבצי הכותר המתאימים:

חיבור הגל המרובע אל הרמקול :

char port61; // הגדרת משתנה מטיפוס תווי

port61=inp32(0x61); // העברת מצב פורט 61 הקסה אל המשתנה

out32(0x61,port61|3); // הוצאת הנתון שקלטנו מהפורט עם הוספת '1' בביטים 0 ו 1

הפסקת הצליל ברמקול :

port61=inp32(0x61); // העברת מצב פורט 61 הקסה אל המשתנה

out32(0x61,port61&0xfc); // הוצאת הנתון שקלטנו מהפורט עם הוספת '0' בביטים 0 ו 1

**תרגיל מסכם : פסנתר**

יש להפוך את המקלדת לקלידים של פסנתר. לחיצה על הקלידים תשמיע תו מסוים. המטרה היא לנגן עם מקשי המקלדת ולשמוע ברמקול מוזיקה מתאימה.

בהתחלה נהפוך את המקשים 1 עד 8 לתווים דו רה מי פה סול לה סי דו. בהמשך תוכל לבחור איזה מקשים נוספים ייתנו אוקטבות נוספות. במקור "אוקטבה" מציינת את המספר שמונה, (למשל, מ[דו](http://he.wikipedia.org/wiki/%D7%93%D7%95) אחד לדו הבא אחריו. ב[מוזיקה](http://he.wikipedia.org/wiki/%D7%9E%D7%95%D7%96%D7%99%D7%A7%D7%94), **אוקטבה** היא מרווח בין שני [צלילים](http://he.wikipedia.org/wiki/%D7%A6%D7%9C%D7%99%D7%9C) שתדירותו של האחד מהם היא חצי (או פי 2) מתדירותו של השני. מרווח זה שקול ל-6 [טונים](http://he.wikipedia.org/wiki/%D7%98%D7%95%D7%9F_%28%D7%9E%D7%95%D7%96%D7%99%D7%A7%D7%94%29), (או 12 חצאי טונים, 12 צלילים) בין [תו](http://he.wikipedia.org/wiki/%D7%AA%D7%95_%28%D7%9E%D7%95%D7%96%D7%99%D7%A7%D7%94%29) לאותו תו ברצף (במחזור) הבא.

נתונה טבלת צלילים והתדרים המתאימים.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **דו-של האוקטבה הבאה** | **סי** | לה דיאז | **לה** | סול דיאז | **סול** | פה דיאז | **פה** | **מי** | רה דיאז | **רה** | דו דיאז | **דו** | **צליל** |
| **1059** | **992** | 942 | **882** | 837 | **784** | 744 | **706** | **661** | 628 | **588** | 558 | **523** | **תדר בהרץ** |

טבלה מספר 2 : צלילים ותדרים

בטבלה מספר 2 נתונה טבלה המקשרת בין צלילים ותדרים באוקטבה מסוימת. המספרים המודגשים ישמשו כתדרים עבור התווים מדו ועד ה דו הבא, המתאימים למקשים 1 עד 8 במקלדת.

**תרגיל**

רשום תכנית שתבצע את הדברים הבאים :

1. תדפיס במסך תפריט שבו יש את הפרטים הבאים :
2. Play נגן
3. Play nad Save in a file נגן ושמור בקובץ
4. Play from a file נגן מקובץ
5. Quit סיום תכנית

Please Enter Your Choice :

1. על פי בחירת המשתמש תתבצע הפונקציה המתאימה . לדוגמה : אם המשתמש בחר ב 1 , עוברים לפונקציה שבו יהיה רשום : Play with keys 1 to 8 . To end play press Enter . כל מקש שהמשתמש יקיש נשמע ברמקול את הצליל המתאים למקש בעזרת הטבלה שבעמוד הקודם.
2. אם המשתמש בחר בתפריט במספר 2 ייפתח מסך שבו המשתמש יצטרך לבחור באיזה את שם הקובץ שהוא רוצה לשמור את הנגינה שלו. לאחר הכנסת שם הקובץ הוא יקבל את ההודעה Play with keys 1 to 8 . To end play press Enter . התווים אותם ינגן יישמרו בקובץ שאותו רשם המשתמש.
3. אם המשתמש בחר בתפריט במספר 4 הוא יישאל איזה קובץ הוא רוצה לשמוע ואז השיר שניגן והוקלט לקובץ יישמע ברמקול.

**הערות**

כדי לבצע את התרגיל ניתן מספר רמזים.

1. בדיקה האם נלחץ מקש כלשהו : בשפת אסמבלי רושמים את השורות הבאות להפעלת פונקציית שרות מספר 21H :

Mov ah,0bh

Int 21h

אם AL שווה 0 לא נלחץ שום מקש ואם al=ffh נלחץ מקש כלשהו ואז ניתן למשוך את המקש עם פונקציית שרות מספר 1 עם רוצים לראות על המסך מהו התו שנלחץ או פונקציית שרות מספר 7 אם לא רוצים לראות על המסך את המקש :

 בשפת C יש פונקציה הנקראת kbhit() הבודקת האם המקש לחוץ. בדוק ב help מה עושה הפונקציה.

1. כדי שהמנגינה ששומרים בקובץ ואח"כ משמיעים תהיה קרובה ככל האפשר למקור כדאי לשמור בקובץ את הזמן שהתו לחוץ והזמן שהתו לא לחוץ. לשם כך כדאי לבדוק מספר פעמים בשנייה האם התו עדיין לחוץ ואם כן לשמור את ערכו. אם לא אז לבדוק כמה זמן לא לחוץ תו ולתת למצב זה ערך כלשהו...

**בהצלחה**