



מגמת הנדסת אלקטרוניקה ומחשבים

מדריכים למורה

תוכן עניינים

2	יחידה 1: מערכות ספרתיות.....
56	יחידה 2: יסודות בתורת החשמל.....
67	יחידה 3: אלקטרוניקה תקבילית וספרתית.....
89	יחידה 4: מבוא למערכות משובצות מחשב.....
146	יחידה 5: יסודות התכנות בשפת C #.....

יחידה 1: מערכות ספרתיות

היקף: 60 שעות עיוני; 30 שעות מעבדה

פרק 1

ראציונל

מערכות ספרתיות הינן תחום לימוד מרתק שמהווה בסיס משמעותי להמשך לימודי האלקטרוניקה והמחשבים. עיקר הלימוד מתמקד בלוגיקה וברכיבים המהווים את ה"מוח" של מערכת אלקטרונית.

הפרק הראשון קצת יוצא דופן מאחר ואופיו מתמטי "יבש" שלעיתים עשוי לשעמם תלמידים צעירים, המצפים לעבודה עם רכיבים והבנתם. יחד עם זאת, חשוב מאוד להסביר את הקשר בין נושאי הלימוד בפרק זה לשאר המקצוע בכל אחד מסעיפי הלימוד. הסבר המשמעות והשימוש בכל אחד מבסיסי הספירה ושיטות ייצוג המספרים הבינאריים ייתן מוטיבציה להתמודד עם הפרק תוך כדי ציפייה להמשך. חשוב להכין מראש את התלמידים לכך שהפרק הראשון הוא קצת מתמטי ולא מייצג את המשך הנושאים הנלמדים במקצוע זה.

מטרות

פרק זה נועד בעיקרו לתיאור השיטה הבינרית וללימוד האריתמטיקה שלה. במהלך הפרק נכיר שיטות ספירה נוספות, הקרובות לשיטה הבינרית. במהלך הפרק ילמד התלמיד את הסיבה לקיום בסיס הספירה הבינארי, בסיסי ספירה נוספים ואת דרכי המעבר בין הבסיסים.

עקרונות

כל שיטת ספירה מאופיינת על-ידי מספר המכונה **בסיס השיטה** או **בסיס הספירה** ומסומן באות b . למשל: בסיס השיטה העשרונית הוא 10; בסיס השיטה הבינרית הוא 2. על משמעות המספר b נדון בהמשך.

חשוב להציג לתלמיד את ההיגיון והעיקרון החוזר על עצמו במעבר בין בסיסי הספירה ובכך להקל עליו את החשש מהצורך לזכור את כל שיטות המעבר.

מושגים

נהוג לסמן מצבים אפשריים אלה בסמלים 0 או 1. הסמלים אלו הם הסמלים היחידים של שיטת הספירה הבינרית. קבוצת ספרות בינריות עשויה להיראות למשל כך: 101101.

הספרה המשמעותית ביותר (*MSD – Most significant Digit*)

הספרה הפחות משמעותית (*LSD – Least significant Digit*)

הנחיות לפתרון התרגילים בגוף הפרק

הפרק בנוי כך שבתחילת לימוד כל נושא ישנן דוגמאות לפתרון תרגילים המציגות את הדרך המלאה לפתרון והסברים נלווים.

חשוב להקפיד שהתלמיד ניצמד לשלבי עבודה נכונים, עובד לפי סדר פעולות חשבון ומתרגל לסדר בכתיבה שעיקרו כתיבת כל סיבית או כל ספרה במשבצת בתוך מחברת או דפדפת משבצות.

פרק 2

ראציונל

בפרק זה ייחשפו התלמידים בפעם הראשונה לרכיבים אלקטרוניים ותפקידם בחלק ה"חושב" של מערכת אלקטרונית. חשוב לשלב את לימוד הפסוקים הלוגיים והשערים ביחד עם דוגמאות ברורות מחיי היומיום שבהם אנו נותנים הוראות והנחיות לביצוע פעולות מותנות. מומלץ לשים לב לדרך שבה משרטטים התלמידים במחברתם את השערים ולהקפיד על שרטוט נכון וברור. לא מעט תלמידים מתקשים בהתחלה להבין בעצמם מה הם שרטטו וזה מקשה ומעכב להם את הלמידה העצמית והחזרה מהמחברת.

מטרות

מטרות הפרק הן להקנות לתלמיד ידע לגבי הקשרים הלוגיים הבסיסיים ותיאורם באמצעות סימול, טבלת אמת וביטוי לוגי.

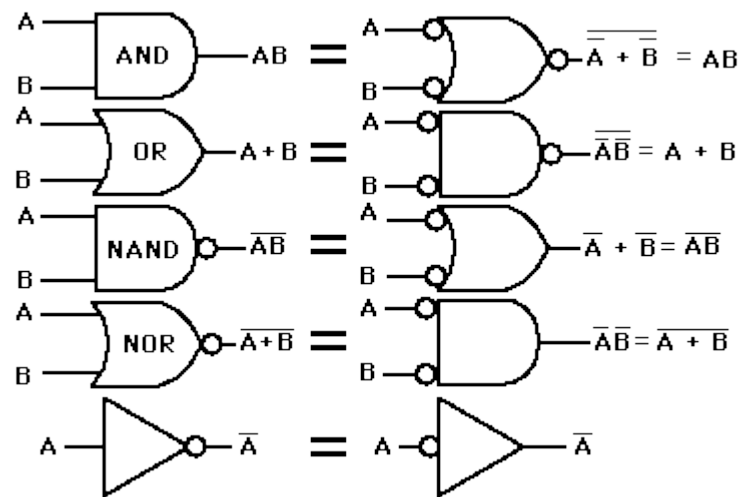
עקרונות

קשר לוגי הוא פונקציה היוצרת מקבוצה של פסוקים לוגיים או תבניות לוגיות פסוק חדש או תבנית חדשה בהתאמה. כל קשר לוגי מייצג פעולה בוליאנית על ערכי האמת של האיברים המתקבלים כקלט, ומקושרת אליו טבלת אמת ספציפית. כדי לחשב את ערך האמת של הפסוק או התבנית הנוצרים על ידי הקשר, משתמשים בערכי האמת של הפסוקים או התבניות המתקבלים כקלט, ובטבלת האמת המשויכת לקשר. באופן מסורתי, מוגדרים בלוגיקה מתמטית שלושה קשרים "בסיסיים":

- לא (NOT)
- או (OR)
- וגם (AND)

מושגים:

- ניתן לייחס לפסוק את הערך **אמת** או **שקר** ורק אחד מהם. **אמת** נהוג לסמן על-ידי האות **T** ו**שקר** - על-ידי האות **F**.
- פסוק מורכב בנוי מצירוף של כמה פסוקים פשוטים. הקשר בין הפסוקים נעשה באמצעות קשרים.



פרק 3

ראציונל

פרק זה עוסק בכללי האלגברה הבולאנית והשימוש בהם לצורך פישוט/צמצום של פונקציות בולאניות מורכבות. יש להדגיש את התרגול המשמעותי בסעיף במהלכו. כל הדרכים בהן נדרש התלמיד בשלב זה לדעת להציג פעולה של מערכת צירופית הם:

- מימוש בשערים
- טבלת אמת
- סיגמה ופאי
- ביטוי לוגי כסכום מכפלות קנוני וכמכפלת סכומים
- ביטוי לוגי מצומצם והעברתו למינימום שערים.

האלגברה הבולאנית מהווה אבן דרך משמעותית בהבנת המקצוע ומומלץ להמחיש באמצעות תוכנת הסימולציה את משמעות פישוט הפונקציות על ידי הוכחה שביטוי לוגי קנוני וביטוי מפושט עושים את אותה הפעולה. מומלץ לממש על אותן הכניסות של המערכת, גם את הביטוי המפושט למוצא אחד וגם את הביטוי הקנוני למוצא אחר. בנוסף, יש להדגים בדיאגרמת זמן שלכל האפשרויות בין הכניסות, המערכת מבצעת את אותה הפעולה.

מטרות

- הקניית ידע לגבי מגוון כללי האלגברה הבולאנית, כללי דמורגן ושערים לוגים מורכבים.
- לימוד של פישוט מערכות/פונקציות בולאניות מורכבות ומימושן במינימום שערים.
- לימוד מימוש מערכות באמצעות שערי NOR בלבד או שערי NAND בלבד.

מושגים

$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$	כללי ההיפוך:
	$\overline{\bar{A}} = A$	כלל השלילה הכפולה:
$A \cdot A = A$	$A + A = A$	כללי הכפילות:
$A \cdot 0 = 0$	$A + 0 = A$	כללי האפס:
$A \cdot 1 = A$	$A + 1 = 1$	כללי היחידה:
$A \cdot B = B \cdot A$	$A + B = B + A$	כללי החילוף:

$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = A + (B + C)$	כללי הקיבוץ:
$A + (B \cdot C) = (A + B) \cdot (A + C)$	[שונה ממתמטיקה רגילה]	כללי הפילוג:
ולכן חשוב לזכור:		
$A \cdot (B + C) = A \cdot B + A \cdot C$		
$A + A \cdot B = A$	(ניתן להוכחה בעזרת כלל הפילוג הראשון).	כלל צמצום ראשון:
$A \cdot (A + B) = A$		כלל צמצום שני:
$A + \bar{A} \cdot B = A + B$	(ניתן להוכחה בעזרת כלל הפילוג הראשון).	כלל צמצום שלישי:
$A \cdot (\bar{A} + B) = A \cdot B$		כלל צמצום רביעי:
		כללי דה-מורגן:
$\overline{A + B} = \bar{A} \cdot \bar{B}$		כלל היפוך OR (כלל NOR)
$\overline{A \cdot B} = \bar{A} + \bar{B}$		כלל היפוך AND (כלל NAND)

פרק 4

ראציונל

הפרק מתמקד בדרכים השונות לייצוג פונקציות בולאניות ובפישוטן באלגברה בולאנית ובמפות קרנו. חשוב לתרגל פישוט של פונקציות בשתי השיטות ולהוכיח שהביטוי המפושט הסופי הוא זהה.

בלימוד צירופי הברירה חשוב לתת דוגמאות שבהן קל להבין שאין משמעות לכניסות ולכן מצב המוצא לא משנה למתכנן. השימוש הנכון של צירופי הברירה במפות הקרנו יוביל לביטויים מפושטים יותר ובדרך זאת יבינו התלמידים את הצורך והמשמעות בשימוש בצירופי הברירה.

מטרות

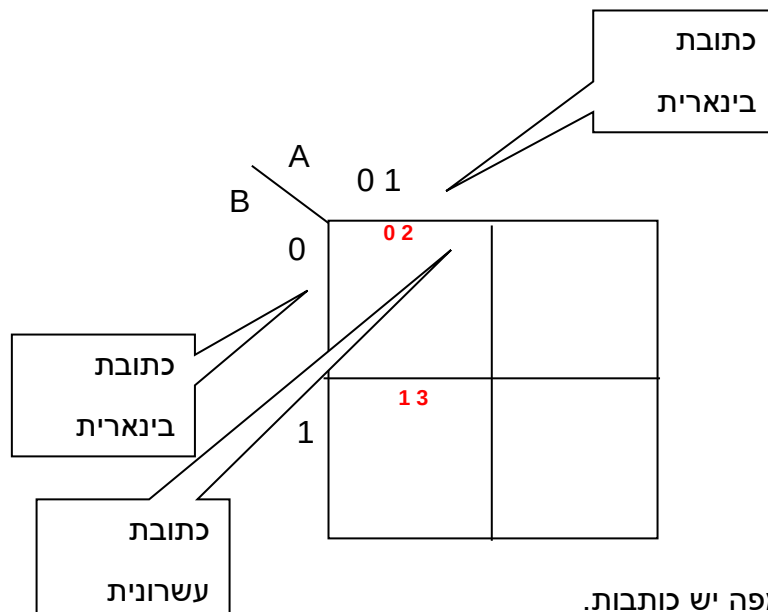
הקניית ידע לגבי שימוש במפות קרנו לטובת פישוט/צמצום פונקציות המשמעות והשימוש בצירופי הברירה וייצוג קנוני של פונקציות.

מושגים

- הבסיס לבניית פונקציה בצורה קנונית הוא ביטויים הקרויים "מכפלה סטנדרטית" ו"סכום סטנדרטי"
- מפת קרנו היא שיטה לצמצום ביטויים הנהוגה בבעיות באלגברה בוליאנית.
- מפת קרנו משתמשת ביכולת זיהוי התבניות המובנית של המוח האנושי, כדי למצוא את הייצוג המצומצם ביותר לפונקציות בוליאניות. עם זאת, מורכבות המפה גדלה מהותית ככל שמספר המשתנים של הפונקציה הבוליאנית גדל, כך שהשימוש בה אינו מעשי לפונקציות של יותר מחמישה משתנים.
- השימוש במפת קרנו מתבצע על ידי שרטוט טבלה, כאשר כותרות הטורים מכילות יחס כפל (וגם) של חלק מהמשתנים, בעוד כותרות השורות מכילות יחס כפל של המשתנים שאינם מופיעים בטורים.
- השימוש בשיטה מאפשר להוכיח זהויות התקפות באלגברה בוליאנית כלשהי, מכיוון שניתן להוכיח כי זהות מסוימת תקפה באלגברה בוליאנית כלשהי אם ורק אם היא תקפה בכל אלגברה בוליאנית.

הנחיות לפתרון התרגילים בגוף הפרק

מפה לשני משתנים -

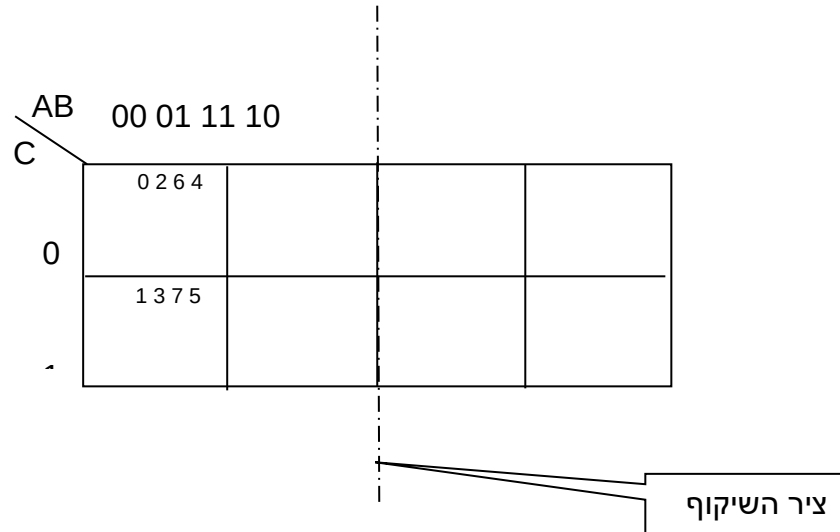


לכל משבצת במפה יש כותבות.

הכותבות יכולה להיות כמספר עשרוני, או בינארי מותאם המשבצת השמאלית העליונה
 כתובת הבינארית היא 00 והכותבת עשרונית 0.

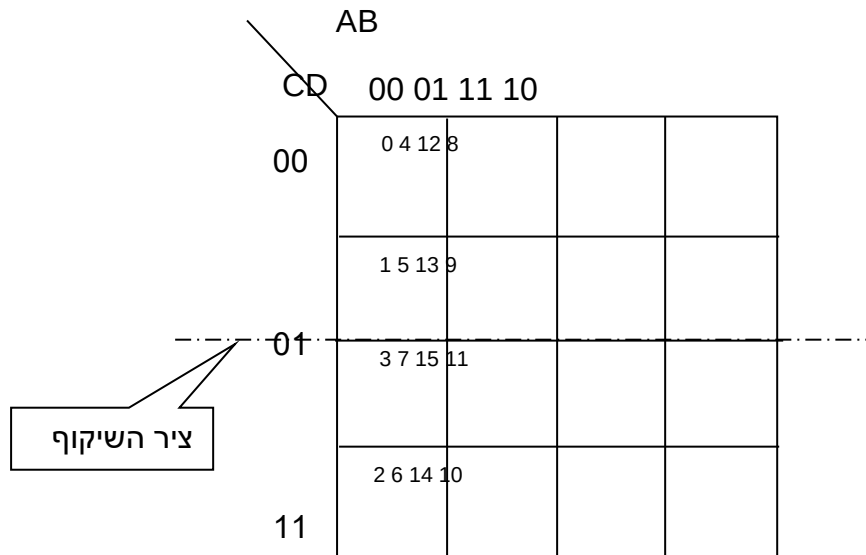
מפה לשלושה משתנים

המפה מבוססת על המפה למעלה (שני משתנים). לקבלת מפה לשלושה משתנים, מבצעים שיקוף של המפה הקיימת על הציר האנכי הימני. למשבצות שהיו לפני השיקוף מוספים אפס מוביל; ולמשבצות אחרי השיקוף מוספים 1 מוביל.



מפה לארבעה משתנים

באותה השיטה ניתן ליצור גם מפה לארבעה משתנים, כאשר משקפים את מפת שלושת המשתנים על הציר האופקי התחתון.



משבצת המוגדרת כ- Don't care, ערכה נתון לשיקולו של מבצע הצמצום. ברצונו יחשיב את המשבצת כ- 1 לוגי, וברצונו יחשיב את המשבצת כ- 0 לוגי. הכל בהתאם לנוחות הצמצום. הסימון המקובל הוא Φ .

פרק 5

ראציונל

הפרק מתמקד בנושא תכנון מערכות צירופיות. חשוב להקפיד על שלבי עבודה נכונים בדרך למימוש מערכת ולבדיקתה:

- הבנת הדרישה – בשלב זה חלק מהתלמידים עשויים להתקשות, עקב בעיות הבנת הנקרא וניסוח הבעיה.
- שרטוט מלבן המערכת עם כניסות ויציאות והגדרת שמות המשתנים.
- מילוי טבלת אמת על פי הדרישה.
- הוצאת ביטוי לוגי וסיגמה.
- פישוט באלגברה בולאנית או בקרנו.
- הגעה למינימום שערים.
- מימוש המערכת.
- בדיקת המערכת.

בפרק זה נדרשים התלמידים גם להשתמש בתוכנה QUARTUS לצורך מימוש המערכות ובדיקתן בדיאגרמות זמן. לפיכך, חשוב מאוד לתרגל שרטוט דיאגרמות זמן במחברת ולהשוותן לתוצאות הסימולציה בתוכנה.

מטרות

- הקניית ידע לגבי תכנון מערכות צירופיות ושלבי העבודה בדרך למימושן.
- לימוד השערים המורכבים והשימוש בהם.
- לימוד השימוש בתוכנת הפיתוח QUARTUS, אשר מאפשרת לשרטט ולבדוק את פעולת המערכות הממומשות ובהמשך גם לצרוב אותן על רכיב בר תכנות.

עקרונות

- קיימת נטייה להשתמש בסוג אחד של שערים NOR או $NAND$ בלבד למימוש פונקציות לוגיות. כדי לממש פונקציה לוגית בעזרת NOR או $NAND$ אנו מציגים את הפונקציה בצורה הקנונית שלה ומממשים אותה.
- מערכת צירופים מהווה רשת לוגית, המורכבת **משערים בלבד**. הפונקציה המתקבלת במוצא המערכת תלויה אך ורק בערכים ה**נוכחיים** של משתני המבוא.

מושגים

- **שער לוגי** (logic gate) הוא רכיב כלשהו המממש קשר לוגי בין משתנים. קיימים שערים מסוגים שונים (מכאניים, פניאומטיים, אלקטרוניים). הנפוצים ביותר הם השערים האלקטרוניים.
 - בשערים אלקטרוניים קיום זרם או מתח מיוצג על-ידי הערך **אמת** (T) והיעדר זרם או מתח מיוצג על-ידי הערך **שקר** (F).
 - ערכי המשתנים שבמבוא של השער הלוגי מוגדרים, כאמור, על-ידי שתי רמות מתח: רמת מתח אחת מייצגת '1' לוגי, ואילו רמת מתח אחרת מייצגת '0' לוגי.
 - ערכי המשתנים המתקבלים ב**מוצא** (output) של השער מיוצגים אף הם על-ידי אותן שתי רמות מתח.
- במסגרת זו נעסוק במערכות הפועלות בלוגיקה חיובית בלבד.

הנחיות לפתרון התרגילים בגוף הפרק

מומלץ מאוד בתרגולים הראשונים להיצמד לשלבי העבודה עם תוכנת QUARTUS המוסברים בנספח.

פרק 6

ראציונל

המערכות הצירופיות השימושיות הן אבן בסיס נוסף לשערים, המאפשרות תכנון מערכות לוגיות מורכבות. חשוב שהתלמידים יכירו את השימושים ואת התכונות של המערכות, ידעו לממשן בשערים לוגיים כדי לייצר אותם בעצמם בשלב זה גם בתוכנת QUARTUS. בנוסף לתירגולים הקיימים בתוכנה שיש בפרק זה, מומלץ שלכל תלמיד תהיה תיקית רכיבים בתוכנה שהוא יצר בעצמו.

מטרות

בפרק זה נתאר כמה מערכות צירופיות נפוצות. מערכות אלה מתחלקות לשני סוגים עיקריים: מערכות חישוב ומערכות לניתוב נתונים. נציג כאן את אופן פעולתן של מערכות אלה ואת המבנה העקרוני שלהן. התלמידים ידעו לממש את המבנה הפנימי של כל מערכת צירופית בתוכנת QUARTUS, לבדוק את פעולתה בדיאגרמת זמן וליצור ממנה מיבנית SYMBOL לטובת שימוש עתידי.

עקרונות

בפרק זה ייחשפו התלמידים למערכות צירופיות סטנדרטיות אשר מאפשרות למתכנן מערכת להשתמש בפעולות נפוצות שימושיות בתהליך הפיתוח. אלה מסייעות לפשט את תהליכי הפיתוח של מערכת בפעולות שחוזרות על עצמן בין מערכות שונות. התלמידים נדרשים להפנים כי יש מספר רב של דרכים למימוש כל מערכת והכרת המערכות הצירופיות הסטנדרטיות מאפשרים להם להשתמש ברכיבים שכבר קיימים בנוסף לשערים הלוגיים לטובת מימוש מערכות.

מושגים

- **מסכם למחצה** הוא מערכת המחברת שתי סיביות, ומפיקה סיבית סכום וסיבית נשא.
- **מסכם מלא** הוא מערכת המחברת שתי סיביות - תוך התחשבות בנשא מבוא - ומפיקה סיבית סכום וסיבית נשא.
- ניתן לחבר שני מספרים בינריים בעלי n סיביות באמצעות **מסכם גלי**, המורכב ממסכם למחצה אחד ומ- $1-n$ מסכמים מלאים. חסרונו העיקרי הוא זמן ההשהיה הארוך שלו.
- **מחסר למחצה** הוא מערכת המחסרת שתי סיביות זו מזו, ומפיקה סיבית הפרש וסיבית לווה.
- **המחסר המלא** הוא מערכת המחסרת שתי סיביות זו מזו - תוך התחשבות בלווה מבוא - ומפיקה סיבית הפרש וסיבית לווה.
- ניתן לחסר שני מספרים בינריים בעלי n סיביות באמצעות **מחסר גלי**, המורכב ממחסר למחצה אחד ומ- $1-n$ מסכמים מלאים. חסרונו העיקרי הוא זמן ההשהיה הארוך שלו.
- מערכת המבצעת השוואה בין שתי סיביות נקראת **משווה לסיבית אחת**. ניתן לבנות באמצעותה - לאחר הרחבה מסוימת - משווה ל- n סיביות. חסרונו העיקרי הוא זמן ההשהיה הארוך שלו.
- **מרבב** הוא מערכת הבוררת - לפי מצב מבואות הברירה - **מבוא נתונים אחד מבין מספר מבואות**, ומעבירה למוצא את ערכו של המבוא הנבחר.
- **מפלג** הוא מערכת הבוררת - לפי מצב מבואות הברירה - **מוצא אחד מבין מספר מוצאים**, ומעבירה אליו את ערכו של המבוא.
- **מפענח** הוא מערכת המשמשת לפענוח הנתונים שבמבואותיה מהצופן הבינרי שבו הוא מוצפן, לרמה לוגית מסוימת באחד ממוצאיה. המפענח שקול למפלג שאין לו מבוא נתונים.

פרק 7

ראציונל

בפרק זה אנו עוסקים ברכיבי הזיכרון, בדלגלים, באוגרים ובמונים. חשוב להבין את השימוש בכל אחד מסוגי רכיבי הזיכרון ולתרגל תכנון אוגרים ומונים וגם מעקב אחר פעולתם בטבלאות מעקב ודיאגרמות זמן. בניית כל רכיב בתוכנת QUARTUS תאפשר יכולת בדיקה עצמית של התלמידים לחקירת המערכת שביצעו על הדף.

מטרות

- הקניית ידע לגבי מערכות הזיכרון הבסיסיות שהן דלגלים, אוגרים ומונים.
- לימוד סוגי הדלגלים והשימושים העיקריים בכל אחד מהם ואת תפקידי הכניסיות הבסיסיות והאופציונליות שלהם.
- לימוד התובנה שכדי לממש אוגרים ומונים יש לבצע חיבור נכון בין דלגלים לטובת מימוש מערכת סינכרונית או אסינכרונית.

עקרונות

בפרקים הקודמים הצגנו כמה דוגמאות של מערכות צירופים, למשל: מסכמים ומרבבים. במערכת צירופים, המוצאים ברגע מסוים תלויים במבואות באותו רגע – ובהם בלבד. מערכות הצירופים הן מרכיב חשוב במערכות הספרתיות, אך לא המרכיב היחיד. מערכות ספרתיות רבות כוללות מרכיב חשוב נוסף: התקני זיכרון המאפשרים אגירת נתונים. בכל רגע ורגע תלויים המוצאים של מערכת ספרתית כזו (כוללת את שני המרכיבים שלעיל) הן במבואות המערכת והן בנתונים האגורים בה באותו רגע כתוצאה מפעולות קודמות. התלמידים נדרשים להפנים את העובדה כי יש הבדל משמעותי בין פעולה יחסית פשוטה של מערכת צירופית ללא זיכרון לבין מערכת אשר משולבים בה גם רכיבי זיכרון אשר מאפשרים יכולות מורכבות ומשמעותיות יותר למערכת.

מושגים

- התקן הזיכרון משמש לזכירת מצב המערכת. מוצאי מערכת הצירופים הם פונקציה בוליאנית של מצב המערכת ומבאותיה.
- מערכת עקיבה נקראת גם מערכת סדרתית (*Sequential System*).

- מצב של מערכת ספרתית הוא האוסף המזערי של נתונים המתארים את השינויים שהתרחשו במערכת בעבר. אוסף זה צריך להספיק לקביעת התנהגות המערכת בעתיד, לפי המבואות שיינתנו.
- המלה סינכרוני מקורה ביוונית ופירושה מתוזמן (כרונוס = זמן). התוספת א למלה סינכרוני (אסינכרוני) מציינת שלילה; ומכאן אסינכרוני - שאינו מתוזמן.
- מבוא האפשר (E) נקרא גם מבוא השער - (G) Gate.
- t , T מציינות זמן ($time$); w מציינת רוחב ($width$).
- f מציינת תדר ($frequency$); יש המציינים שניה (s) כך: sec
- לעתים נקראים דלגלים מדורבנים במעבר גם דלגלים מדורבני קצה. לפי הגדרה זו, גם דלגלים מסוג אדון-עבד הם מדורבני קצה. אנו העדפנו לכלול בספר זה את דלגלי אדון-עבד ומדורבני הקצה תחת הכותרת דלגלים מדורבנים במעבר.
- *Clear, PRESET* - מבואות ישירים לקביעת מצבו ההתחלתי של הנועל.
- *SI* – מבוא טורי
- *SO* – מוצא טורי
- *R* – ימינה
- *L* – שמאלה

נספח 1: ניסויים מעשיים בערכת DE1

ערכת פיתוח מאפשרת לנו לראות את תוצאות התכנות שלנו מבוצעות בפועל על רכיבים לאחר שנבדקו בסימולציה.

בערכת הפיתוח יש מגוון רחב של רכיבים אשר נמצאים סביב רכיב בר התכנות שאליו אנו מורידים בפעולת צריבה/טעינה את התכנות שלנו. הרכיבים העיקריים בהם נשתמש הם: אות שעון, לדים, לחצנים, מפסקים. תצוגות שבעה מקטעים.

בערכת הפיתוח אפשר להשתמש לצורך ביצוע ניסויים ובדיקת קוד, בניית פרויקטונים ופרויקטים.

- מומלץ להיעזר בסרטוני YOUTUBE כגון: <https://youtu.be/lf4iiz4l8Vk>

ניסויים ללימוד עבודה מול ערכת הפיתוח DE1:

- הניסויים הבאים מתקדמים לפי רמת קושי\מורכבות.
- חשוב להתקדם לפי סדר הניסויים כדי ללמוד את הפעולות הבסיסיות שיפורטו בניסויים הראשונים בלבד.

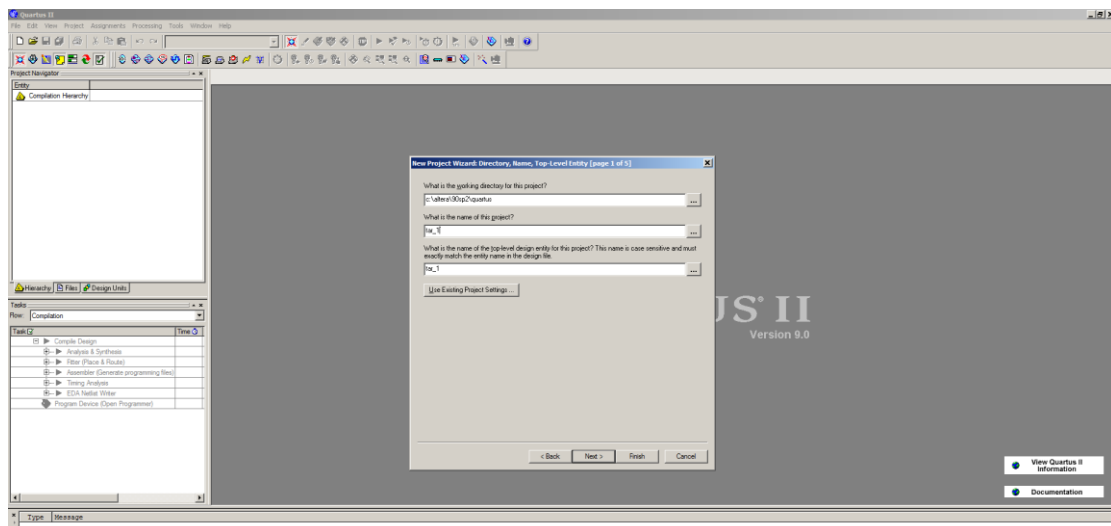
ניסוי מספר 1

תכנן מערכת המקבלת מידע משלושה לחצנים, המערכת תדליק לד רק כאשר שניים מהלחצנים לחוצים (מספקים '1' = לחוצים).

- תכן וממש לאחר פישוט את המערכת בשערים בתוכנת QUARTUS
- בדוק את המערכת בדיאגרמת זמן.
- בצע צריבת המערכת לרכיב הנמצא על ערכת הפיתוח ובדוק מעשית את פעולת המערכת.
- סדר פעולות הצריבה ומהלכן יפורטו בהסבר תהליך הניסוי הזה.

אופן ביצוע ניסוי מספר 1:

- לאחר התכנון והפישוט הגענו לביטוי $F=AB+AC+BC$
- נפתח פרויקט חדש בתוכנת QUARTUS בשם TAR_1

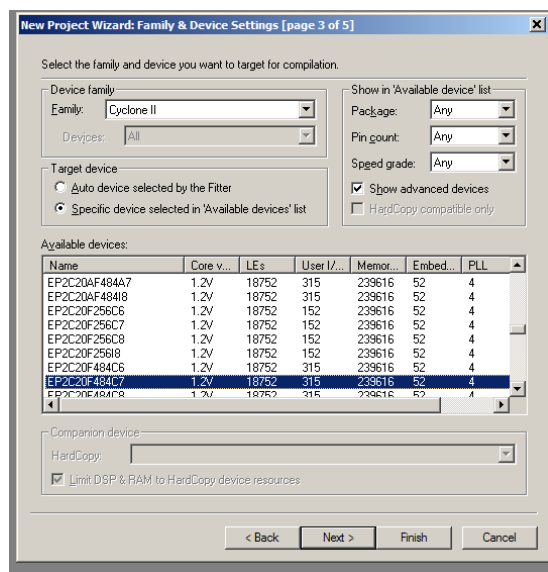


נלחץ על NEXT ולאחר מכן על NO כדי לא לפתוח תיקיה חדשה בשם הפרויקט

- נבחר את הרכיב שעליו נבצע בסוף המימוש את הצריבה

FAMILY "Cyclone II"

DEVICE "EP2C20F484C7"



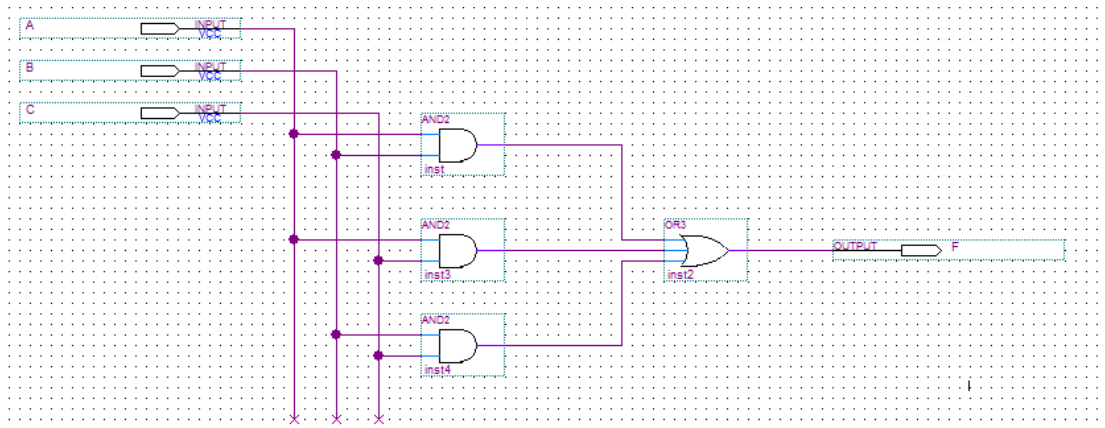
NEXT

NEXT

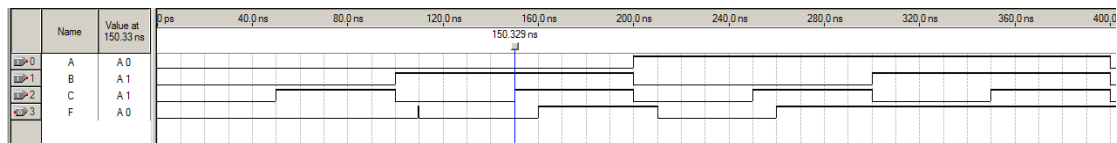
FINISH

- נפתח קובץ חדש מסוג BLOCK DIAGRAM בשם TAR_1

- נייבא אליו כניסות, יציאות ואת שערי המערכת ונממש על פי התכנון.

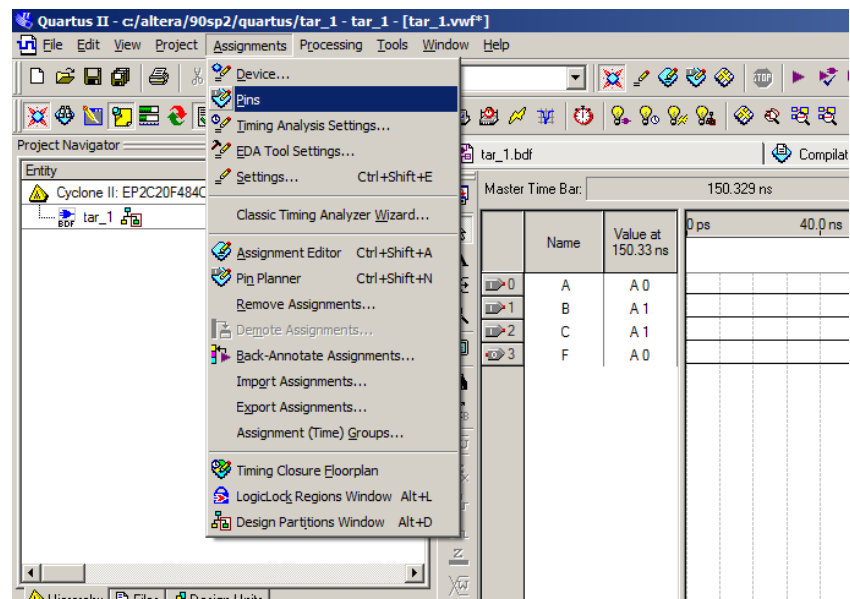


• נבצע הידור ובדיקה בדיאגרמת זמן:

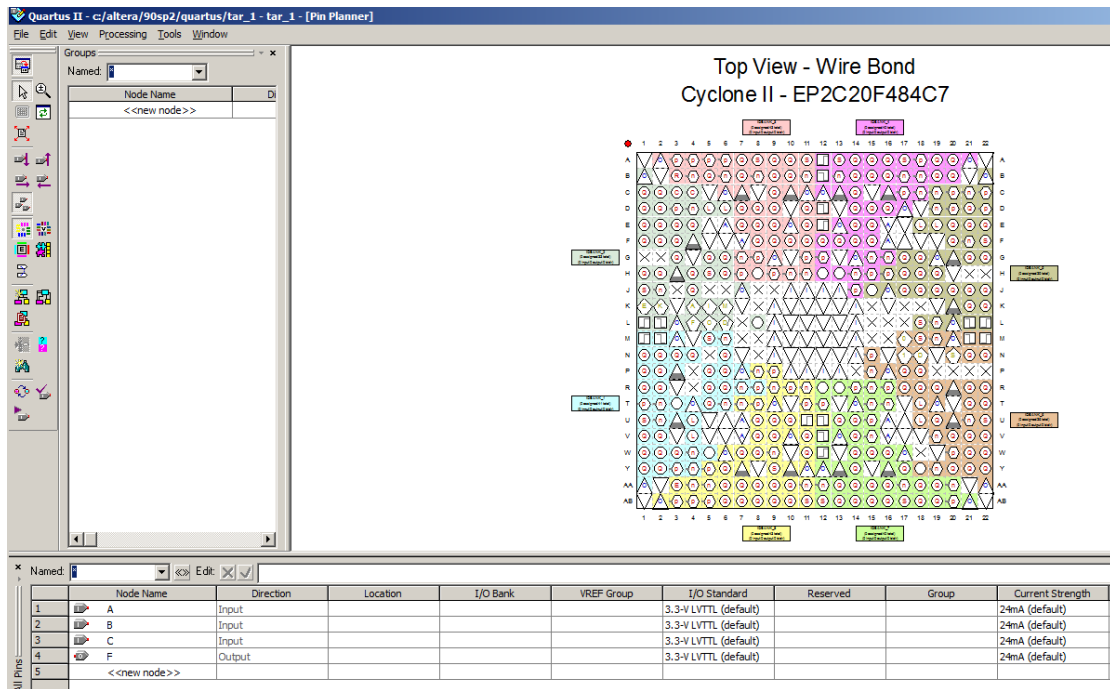


לאחר שבדקנו בדיאגרמת זמן שהמערכת שתכננו עונה לדרישות נתחיל את תהליך הצריבה.

• נבצע התאמת מיקום לכניסות ויציאות – ASSIGNMENT PINS



ונקבל את המסך הבא:



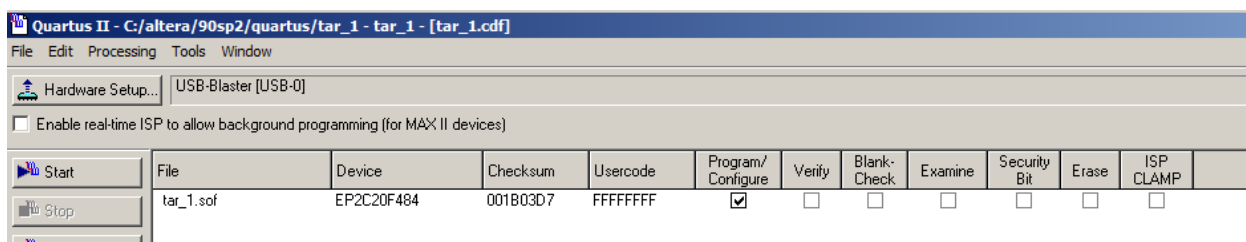
- עפ"י הנספח DE1TCL נתאים את כניסות A B C אל SW0 SW1 SW2 ואת המוצא אל GREEN LED 0
- נלחץ על LOCATION ונכתוב את מיקום הרגל המתאימה למתגים והנורה.

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Group	Current Strength	PCB layer
A	Input	PIN_U22	5	B5_N1	3.3-V LVTTTL (default)			24mA (default)	
B	Input	PIN_U21	5	B5_N1	3.3-V LVTTTL (default)			24mA (default)	
C	Input	PIN_M22	6	B6_N0	3.3-V LVTTTL (default)			24mA (default)	
F	Output	PIN_U22	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	

- נבצע הידור מחדש
- נלחץ על לחצן PROGRAMMER



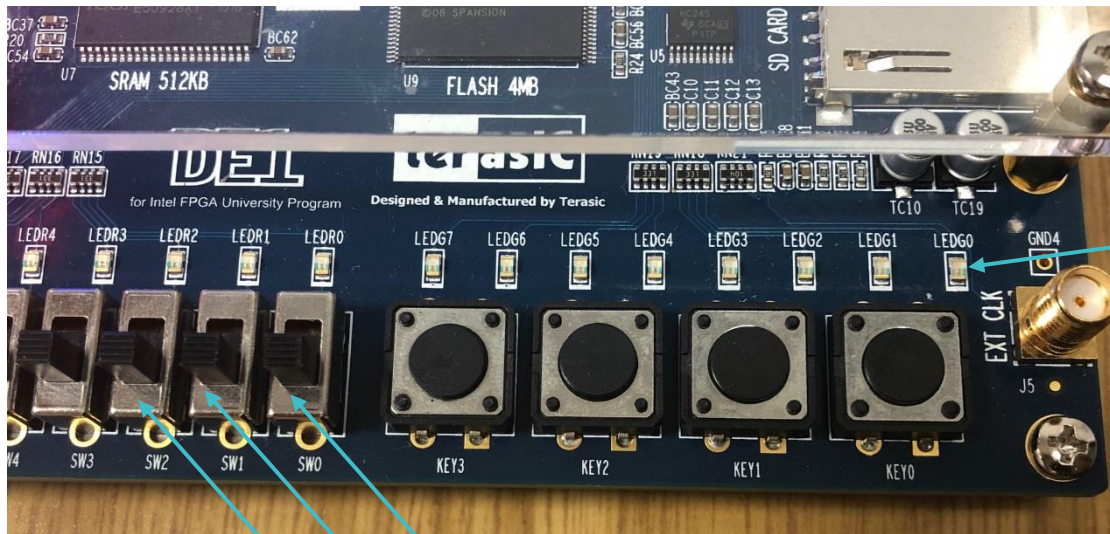
- נוודא שחומרת הצריבה המוגדרת היא USB_BLAZER



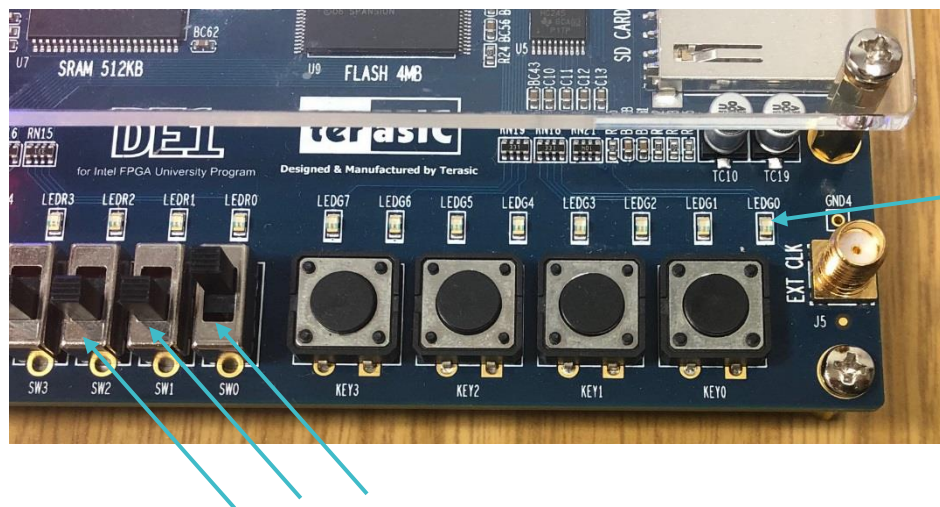
- ברגע שנלחץ על START תבוצע הצריבה והרכיב יתפקד על פי התכנון

ניתן לראות שכאשר שלושת המתגים למטה (א) או כאשר רק אחד למעלה (ב) הנורה
הירוקה לא דולקת

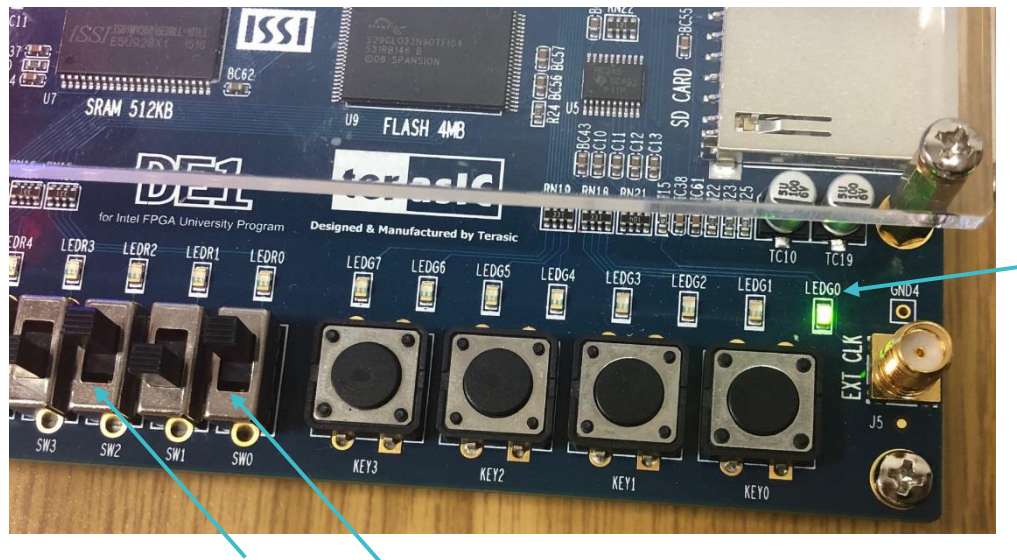
א



ב



וכאשר יש שני לחצנים למעלה ('1') הנורה דולקת.

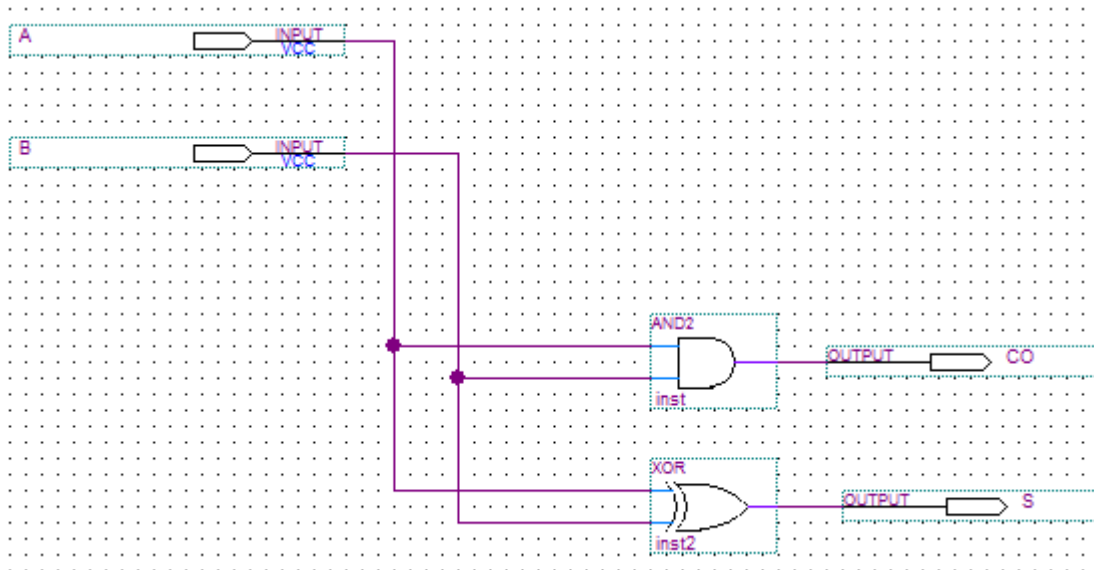


ניסוי מספר 2

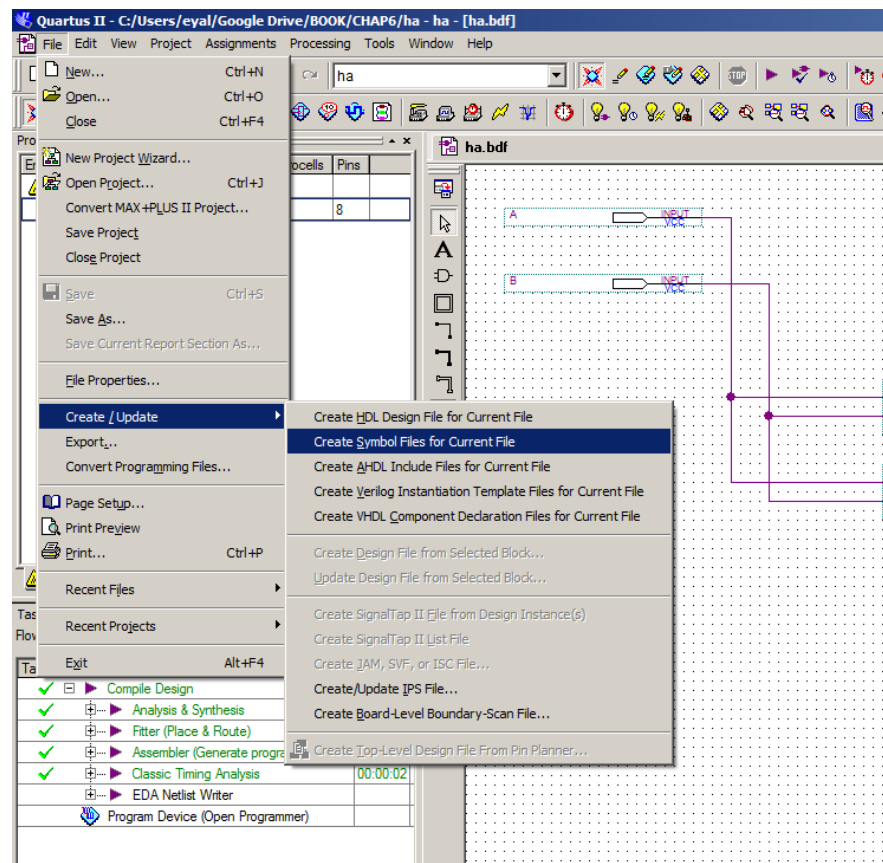
- א. תכנן מחבר חסר H.A וצור ממנו מבנית/SYMBOL לאחר בדיקתו בדיאגרמת זמן.
- ב. תכנן מחבר מלא F.A וצור ממנו מבנית/SYMBOL לאחר בדיקתו בדיאגרמת זמן.
- ג. צור תוכנית חדשה בשם BIG_ADDER שתפקידה לחבר בין שני מספרים בעלי שלוש סיביות כל אחד HA, FA.
- ד. לאחר בדיקת התוכנית BIG_ADDER בדיאגרמת זמן, צרוב את התוכנית לרכיב כך שכניסות המספרים יגיעו מהלחצנים (SWITCH) ויציאות התוצאה יופנו אל הלדים.

אופן ביצוע ניסוי מספר 2:

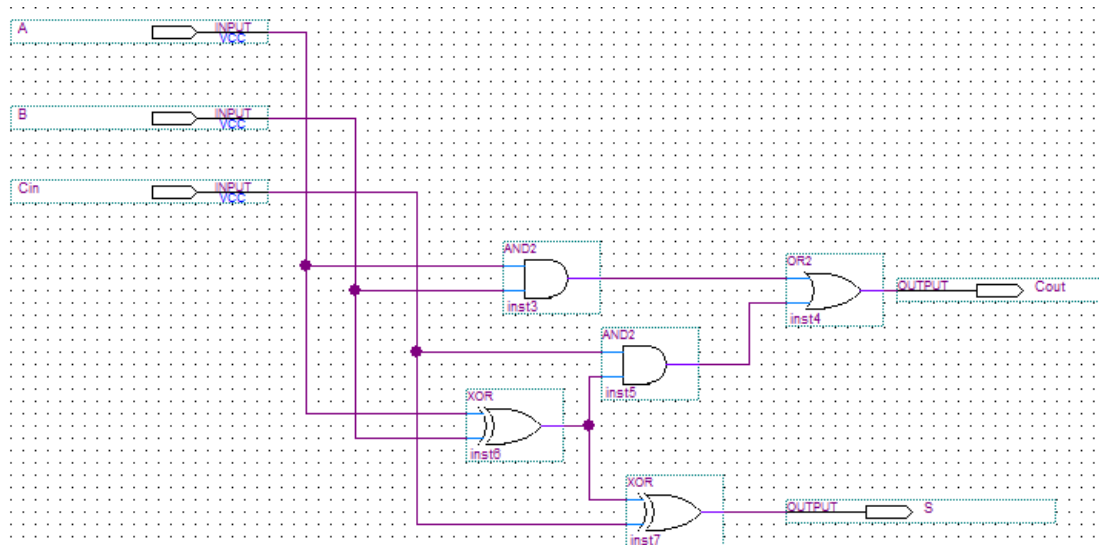
נבצע מימוש HA בשערים:



לאחר בדיקתו בדיאגרמת זמן, ניצור רכיב/SYMBOL בדרך המוצגת:

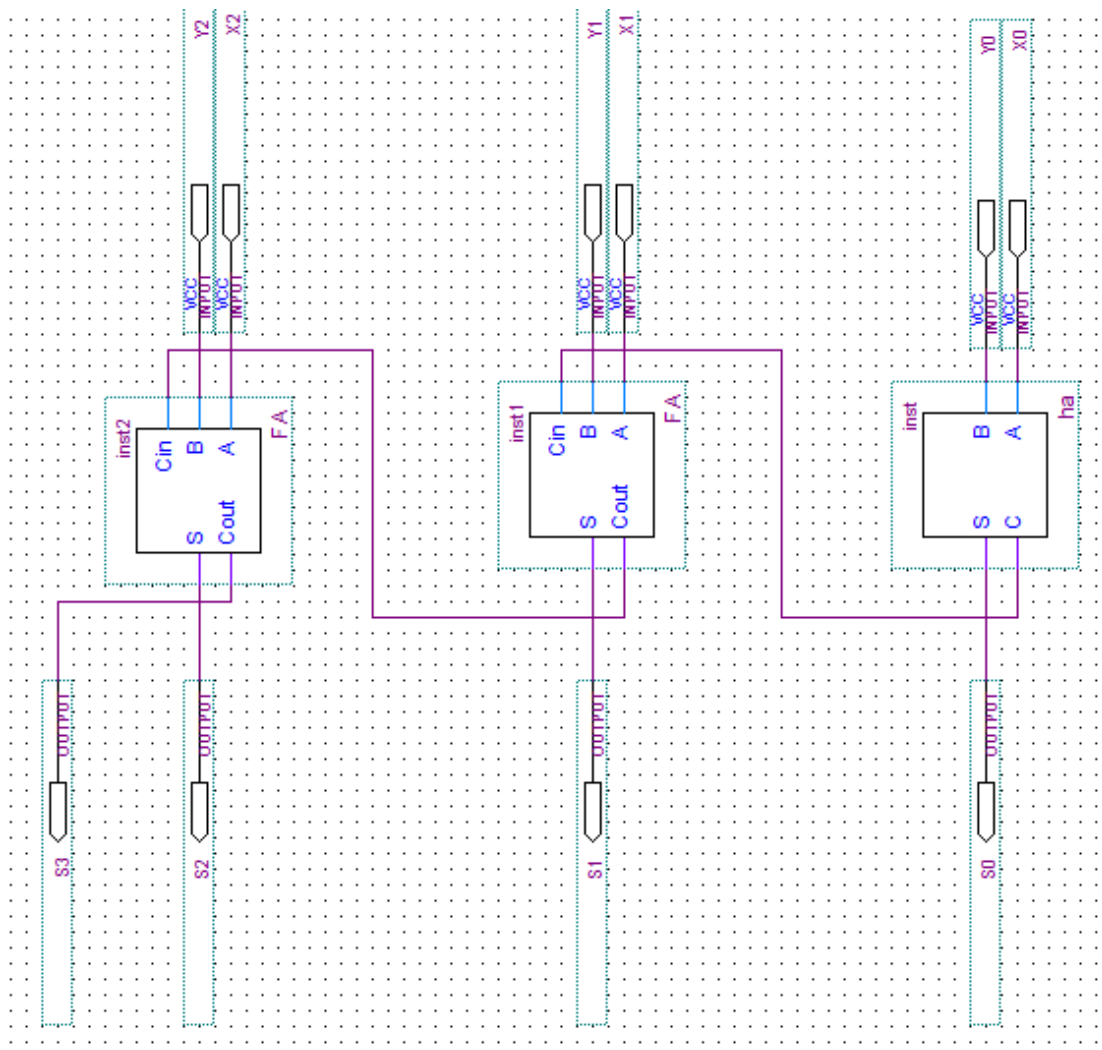


נבצע מימוש FA בשערים:



לאחר בדיקתו בדיאגרמת זמן, ניצור רכיב/SYMBOL.

נפתח קובץ חדש בשם BIG_ADDER ונשתמש ברכיבים שיצרנו לטובת חיבור המערכת:

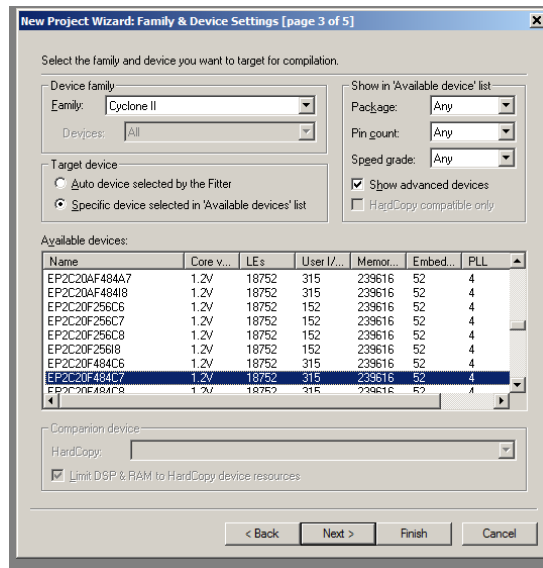


לאחר בדיקת המערכת בדיאגרמת זמן נתחיל את תהליך ההכנה לצריבה.

- נבחר את הרכיב שעליו נבצע את הצריבה

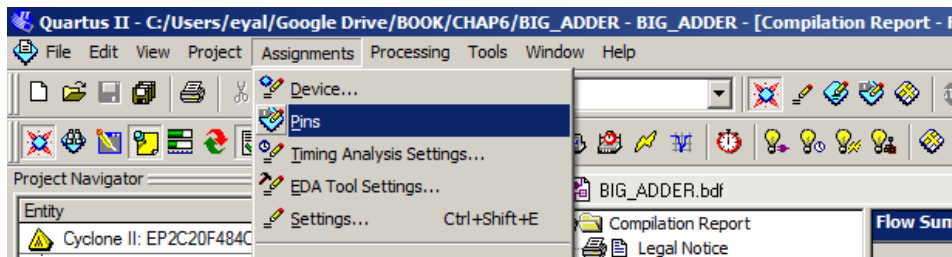
FAMILY "Cyclone II"

DEVICE "EP2C20F484C7"

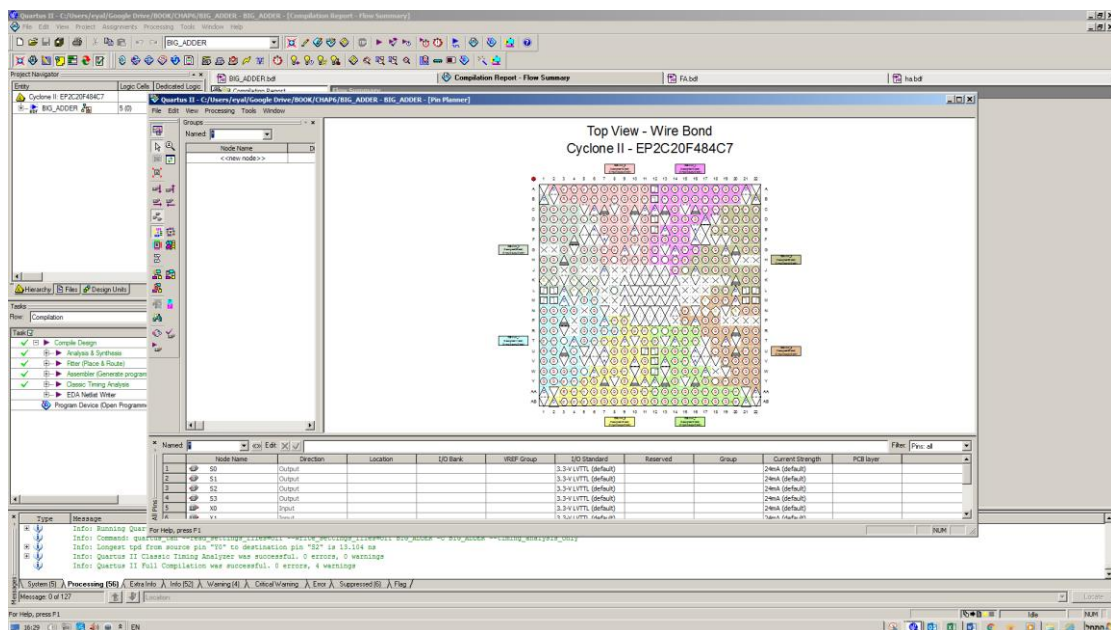


ונהדר מחדש.

- נבצע התאמת מיקום לכניסות ויציאות – ASSIGNMENT PINS



ונקבל את המסך הבא:



- על פי הנספח DE1TCL נתאים את הכניסות של מספר X אל SW0, SW1, SW2 ואת הכניסות של מספר Y את הלחצנים הרחוקים SW7, SW8, SW9 ואת המוצאים אל הלדים 0-3.
- נלחץ על LOCATION ונכתוב את מיקום הרגלים:
 - # set_location_assignment PIN_L22 -to SW[0] ;# SW0
 - # set_location_assignment PIN_L21 -to SW[1] ;# SW1
 - # set_location_assignment PIN_M22 -to SW[2] ;# SW2
 - # set_location_assignment PIN_M2 -to SW[7] ;# SW7
 - # set_location_assignment PIN_M1 -to SW[8] ;# SW8
 - # set_location_assignment PIN_L2 -to SW[9] ;# SW9
 -
 - ;#####
 - ;# 8 Green LEDs #
 - ;#####
 -
 - # set_location_assignment PIN_U22 -to LEDG[0] ;# LEDG[0]
 - # set_location_assignment PIN_U21 -to LEDG[1] ;# LEDG[1]
 - # set_location_assignment PIN_V22 -to LEDG[2] ;# LEDG[2]
 - # set_location_assignment PIN_V21 -to LEDG[3] ;# LEDG[3]

Quartus II - C:/Users/cyal/Google Drive/BOOK/CHAP6/BIG_ADDER - BIG_ADDER - [Pin Planner]

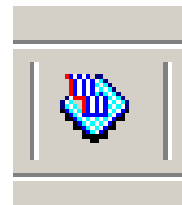
Top View - Wire Bond
Cyclone II - EP2C20F484C7

Named	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Group	Current Strength	PCB layer
1	S0	Output	PIN_L22	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
2	S1	Output	PIN_U21	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
3	S2	Output	PIN_V22	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
4	S3	Output	PIN_V21	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
5	X0	Input	PIN_L22	5	B5_N1	3.3-V LVTTTL (default)			24mA (default)	
6	X1	Input	PIN_L21	5	B5_N1	3.3-V LVTTTL (default)			24mA (default)	
7	X2	Input	PIN_M22	6	B6_N0	3.3-V LVTTTL (default)			24mA (default)	
8	Y0	Input	PIN_M2	1	B1_N0	3.3-V LVTTTL (default)			24mA (default)	
9	Y1	Input	PIN_M1	1	B1_N0	3.3-V LVTTTL (default)			24mA (default)	
10	Y2	Input	PIN_L2	2	B2_N1	3.3-V LVTTTL (default)			24mA (default)	
11	<new node>>									

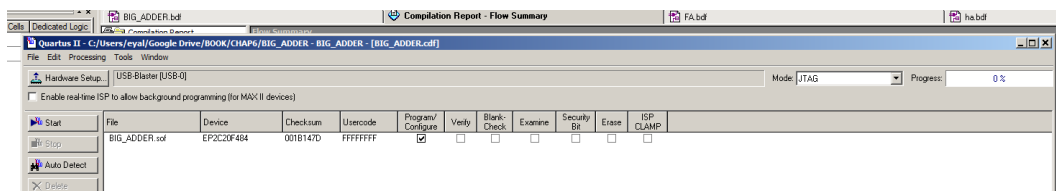
- נבצע הידור מחדש



- נלחץ על לחצן PROGRAMMER



- נוודא שחומרת הצריבה המוגדרת היא USB_BLAZER



לאחר הצריבה נציב מספרים בכניסות X, Y ע"י SW שקבענו:
נבדוק חיבור בין המספרים 6+7 ונקבל את התוצאה על הלדים "1101".



נבדוק חיבור בין המספרים 4+5 ונקבל את התוצאה על הלדים "1001".



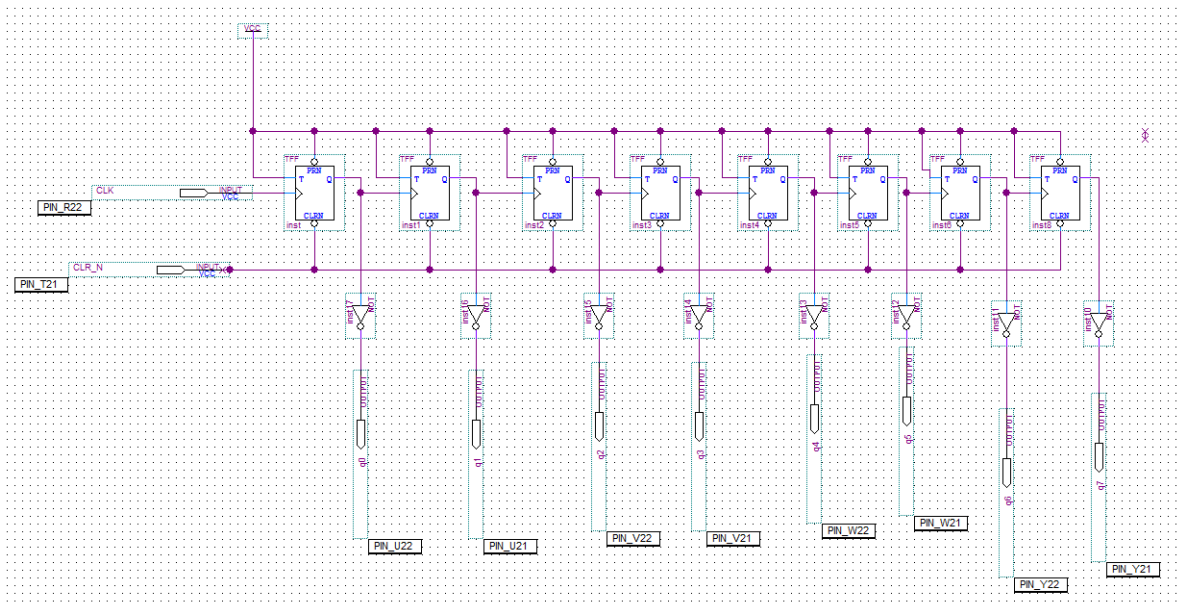
ניסוי מספר 3:

תכנון וצריבת מונה מעלה אסינכרוני לשמונה סיביות.

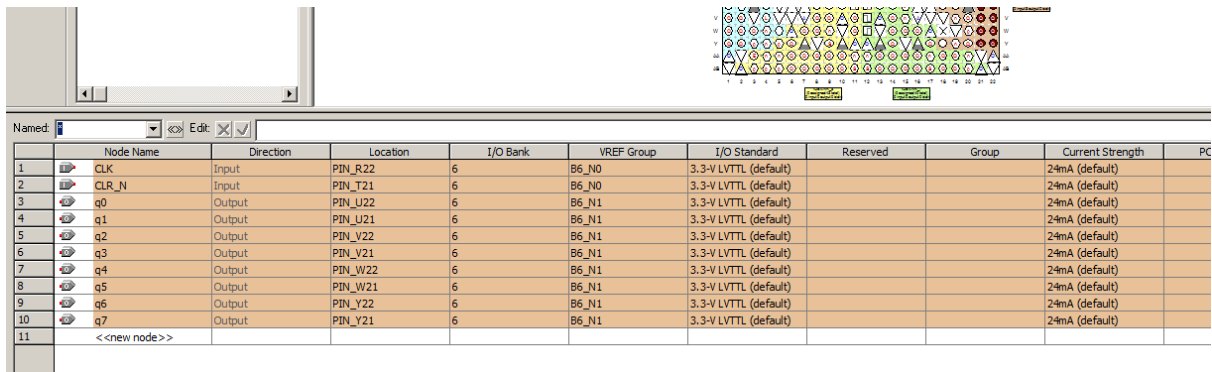
- צור פרויקט חדש בשם UP_COUNTER ובנה מונה מעלה אסינכרוני לשמונה סיביות על ידי ייבוא דלגלי TFF מספריית הרכיבים.
- לאחר בדיקתו בדיאגרמת זמן, התאם את כניסת CLK ואת כניסת CLR אל לחצני KEY ואת המוצאים Q7 – Q0 אל הLEDים הירוקים.

אופן ביצוע ניסוי מספר 3:

בניית המונה בעורך הגרפי:



התאמת הכניסות והיציאות אל הלחצנים והLEDים הירוקים:



Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Group	Current Strength	PC
1 CLK	Input	PIN_R22	6	B6_N0	3.3-V LVTTTL (default)			24mA (default)	
2 CLR_N	Input	PIN_T21	6	B6_N0	3.3-V LVTTTL (default)			24mA (default)	
3 q0	Output	PIN_U22	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
4 q1	Output	PIN_U21	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
5 q2	Output	PIN_V22	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
6 q3	Output	PIN_V21	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
7 q4	Output	PIN_W22	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
8 q5	Output	PIN_W21	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
9 q6	Output	PIN_Y22	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
10 q7	Output	PIN_Y21	6	B6_N1	3.3-V LVTTTL (default)			24mA (default)	
11	<new node>								

לאחר הצריבה ניתן יהיה לראות שבכל לחיצה על KEY0, המספר במוצא המונה עולה באחד והמספר הבינארי המוצג על הלדים מתקדם.
בלחיצה ראשונה



בלחיצה עשירית



ניסוי מספר 4:

יצירת מחלק תדר על ידי שרשור מונים.

בערכת הפיתוח DE1 יש גביש המספק לרכיב בר תכנות תדר 50 מגה הרץ.

בניסוי זה נשתמש במונה שתוכנן בניסוי מספר 3 כדי לבצע חלוקת תדר של 50 מגה הרץ

לתדר של כ 1.5 הרץ.

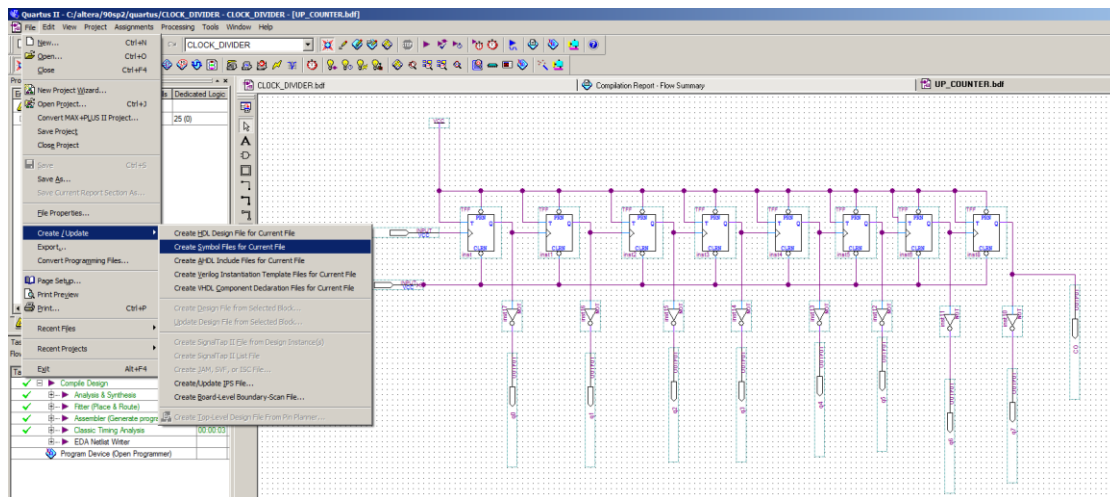
על מנת להגיע לתדר קרוב ל 1.5 הרץ נצטרך לחלק פי 2 את תדר המקור 25 פעמים.
מכיוון שהמונה שתוכנן בניסוי קודם כולל רק 8 דרגות חלוקת תדר, נבצע שירשור בין 4 מונים
לטובת הגעה לתדר הרצוי:

מספר חלוקות	חלוקה פי	50,000,000	תדר מקור
1	2	25000000	
2	4	12500000	
3	8	6250000	
4	16	3125000	
5	32	1562500	
6	64	781250	
7	128	390625	
8	256	195312.5	
9	512	97656.25	
10	1024	48828.13	
11	2048	24414.06	
12	4096	12207.03	
13	8192	6103.516	
14	16384	3051.758	
15	32768	1525.879	
16	65536	762.9395	
17	131072	381.4697	
18	262144	190.7349	
19	524288	95.36743	
20	1048576	47.68372	
21	2097152	23.84186	
22	4194304	11.92093	
23	8388608	5.960464	
24	16777216	2.980232	
25	33554432	1.490116	

מהלך הניסוי:

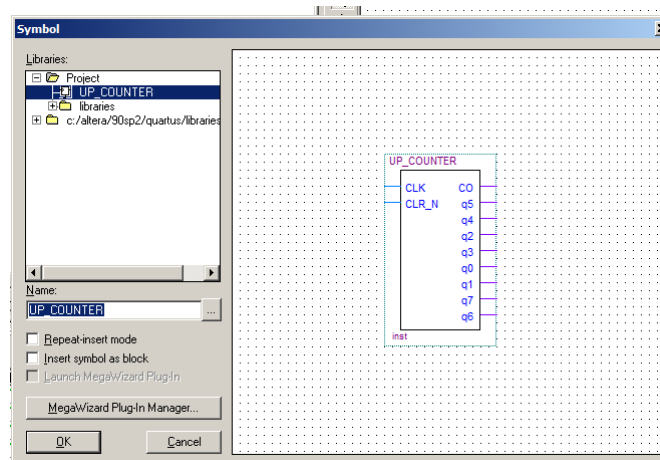
1. פתח את פרויקט המונה שתוכנן בניסוי מספר 3, הוסף לו יציאת CO שתאפשר פעולה זהה בין המונים בדירבון קצה חיובי וצור ממנו מיבנית/רכיב SYMBOL.
2. בצע שירשור בין 4 מונים בעלי 8 יציאות (8 דרגות חלוקת תדר פי 2)
3. את כניסת השעון נפנה לרגל הרכיב המקבלת את התדר מהגביש המספק 50 מגה הרץ.
4. את מוצאו ה 25 של המונה נפנה אל לד ירוק ונראה אותו מהבהב 1.5 פעמים בשניה.

נייצר SYMBOL/רכיב מהמונה שתוכנן בניסוי 3 לאחר הוספת יציאת CO:



שירשור המונים:

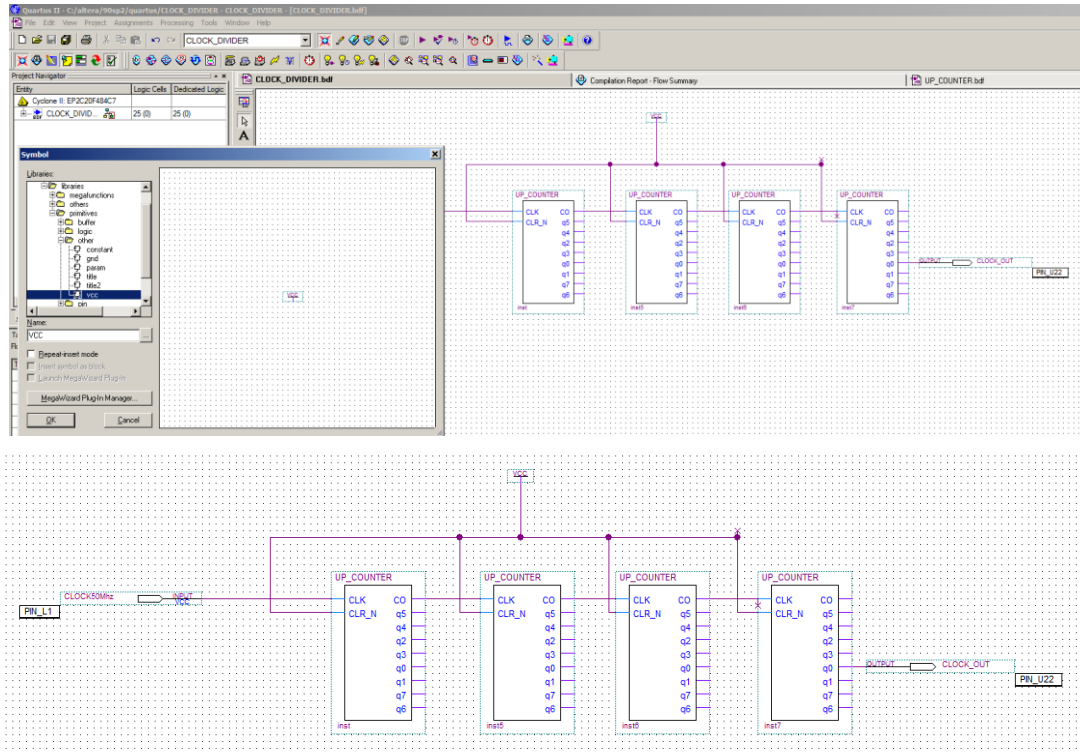
נפתח פרויקט חדש בשם CLOCK_DIVIDER
נפתח קובץ גרפי חדש בשם הפרויקט CLOCK_DIVIDER
ונייבא אליו את רכיב המונה שתוכנן בניסוי 3:



נשרשר בין 4 מונים:

אנו לא זקוקים לאיפוס אשר פעיל בנמוך ולכן נייבא מספריות הרכיבים VCC כדי לנטרל את פעולת האיפוס.

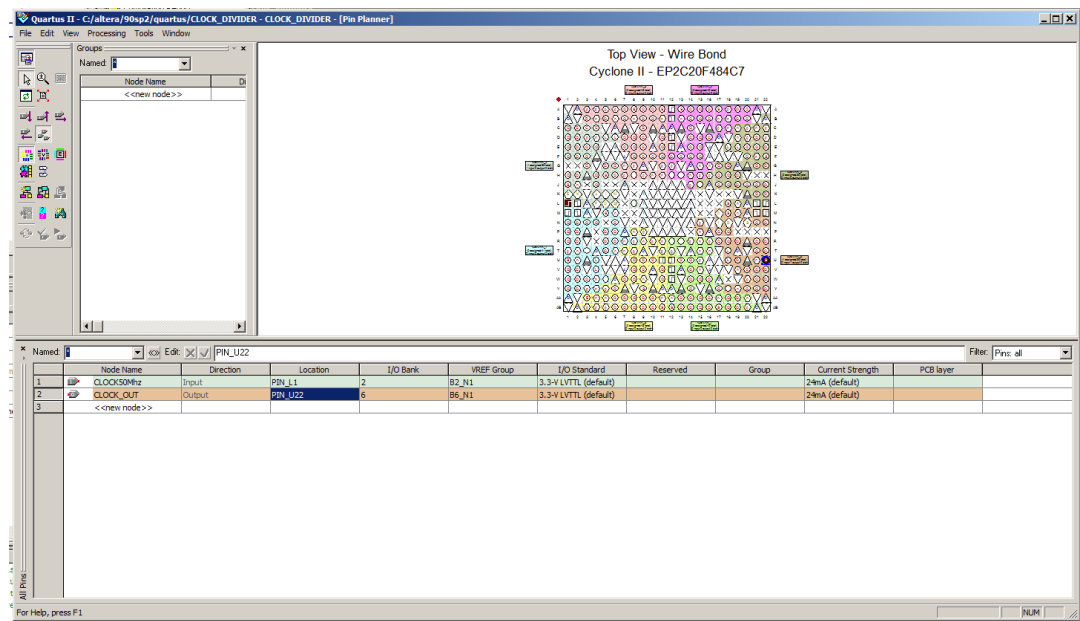
נחבר מוצא אל היציאה ה 25 של המונים שהיא יציאה 7 מהמונה הרביעי ובכך נייצר חלוקת תדר פי 2, 25 פעמים כדי להגיע לתדר הנמוך הרצוי.



נפנה את כניסת השעון ואת המוצא לרגליים המתאימות ובצע הידור.

set_location_assignment PIN_L1 -to CLOCK_50#

set_location_assignment PIN_U22 -to LEDG[0] ;# LEDG[0]



לאחר ביצוע פעולת הצבת המוצאים והכניסות לרגליים המתאימות לנו (PINLOCATION) נבצע הידור מחדש לפני צריבה.
הלד הירוק אמור להבהב לאחר צריבה בערך 1.5 פעמים בשניה.

נספח 2: DE1TCL

```

#####
;# Clock Signals #
#####

# set_location_assignment PIN_L1 -to CLk
# set_location_assignment PIN_L1 -to CLOCK_50#

# set_location_assignment PIN_D12 -to CLOCK_27[0]
# set_location_assignment PIN_E12 -to CLOCK_27[1]
# set_location_assignment PIN_B12 -to CLOCK_24[0]
# set_location_assignment PIN_A12 -to CLOCK_24[1]

# set_location_assignment PIN_M21 -to EXT_CLOCK

#####
;# Debounced Push-Buttons ('0' when pressed) #
#####

# set_location_assignment PIN_R22 -to KEY[0] ;# KEY[0]
# set_location_assignment PIN_R21 -to KEY[1] ;# KEY[1]
# set_location_assignment PIN_T22 -to KEY[2] ;# KEY[2]
# set_location_assignment PIN_T21 -to KEY[3] ;# KEY[3]

#####
;# Slide Switches ('1' when up) #
#####

# set_location_assignment PIN_L22 -to SW[0] ;# SW0
# set_location_assignment PIN_L21 -to SW[1] ;# SW1
# set_location_assignment PIN_M22 -to SW[2] ;# SW2
# set_location_assignment PIN_V12 -to SW[3] ;# SW3
# set_location_assignment PIN_W12 -to SW[4] ;# SW4
# set_location_assignment PIN_U12 -to SW[5] ;# SW5
# set_location_assignment PIN_U11 -to SW[6] ;# SW6
# set_location_assignment PIN_M2 -to SW[7] ;# SW7
# set_location_assignment PIN_M1 -to SW[8] ;# SW8
# set_location_assignment PIN_L2 -to SW[9] ;# SW9

```

```

;#####
;# 8 Green LEDs #
;#####

# set_location_assignment PIN_U22 -to LEDG[0] ;# LEDG[0]
# set_location_assignment PIN_U21 -to LEDG[1] ;# LEDG[1]
# set_location_assignment PIN_V22 -to LEDG[2] ;# LEDG[2]
# set_location_assignment PIN_V21 -to LEDG[3] ;# LEDG[3]
# set_location_assignment PIN_W22 -to LEDG[4] ;# LEDG[4]
# set_location_assignment PIN_W21 -to LEDG[5] ;# LEDG[5]
# set_location_assignment PIN_Y22 -to LEDG[6] ;# LEDG[6]
# set_location_assignment PIN_Y21 -to LEDG[7] ;# LEDG[7]

;#####
;# 10 red LEDs #
;#####

# set_location_assignment PIN_R20 -to LEDR[0] ;# LEDR[0]
# set_location_assignment PIN_R19 -to LEDR[1] ;# LEDR[1]
# set_location_assignment PIN_U19 -to LEDR[2] ;# LEDR[2]
# set_location_assignment PIN_Y19 -to LEDR[3] ;# LEDR[3]
# set_location_assignment PIN_T18 -to LEDR[4] ;# LEDR[4]
# set_location_assignment PIN_V19 -to LEDR[5] ;# LEDR[5]
# set_location_assignment PIN_Y18 -to LEDR[6] ;# LEDR[6]
# set_location_assignment PIN_U18 -to LEDR[7] ;# LEDR[7]
# set_location_assignment PIN_R18 -to LEDR[8] ;# LEDR[8]
# set_location_assignment PIN_R17 -to LEDR[9] ;# LEDR[9]

;#####
;# 7 * Seven Segments #
;# a=6 b=5 c=4 d=3 e=2 f=1 g=0 #
;# DP is not connected ! #
;# Active low (ON when '0') #
;#####

# set_location_assignment PIN_E2 -to HEX0S[6] ;# a HEX0S[6]
# set_location_assignment PIN_F1 -to HEX0S[5] ;# b HEX0S[5]
# set_location_assignment PIN_F2 -to HEX0S[4] ;# c HEX0S[4]
# set_location_assignment PIN_H1 -to HEX0S[3] ;# d HEX0S[3]
# set_location_assignment PIN_H2 -to HEX0S[2] ;# e HEX0S[2]
# set_location_assignment PIN_J1 -to HEX0S[1] ;# f HEX0S[1]

```

```

# set_location_assignment PIN_J2 -to HEX0S[0] ;# g HEX0S[0]

# set_location_assignment PIN_D1 -to HEX1S[6] ;# a HEX1S[6]
# set_location_assignment PIN_D2 -to HEX1S[5] ;# b HEX1S[5]
# set_location_assignment PIN_G3 -to HEX1S[4] ;# c HEX1S[4]
# set_location_assignment PIN_H4 -to HEX1S[3] ;# d HEX1S[3]
# set_location_assignment PIN_H5 -to HEX1S[2] ;# e HEX1S[2]
# set_location_assignment PIN_H6 -to HEX1S[1] ;# f HEX1S[1]
# set_location_assignment PIN_E1 -to HEX1S[0] ;# g HEX1S[0]

# set_location_assignment PIN_D3 -to HEX2S[6] ;# a HEX2S[6]
# set_location_assignment PIN_E4 -to HEX2S[5] ;# b HEX2S[5]
# set_location_assignment PIN_E3 -to HEX2S[4] ;# c HEX2S[4]
# set_location_assignment PIN_C1 -to HEX2S[3] ;# d HEX2S[3]
# set_location_assignment PIN_C2 -to HEX2S[2] ;# e HEX2S[2]
# set_location_assignment PIN_G6 -to HEX2S[1] ;# f HEX2S[1]
# set_location_assignment PIN_G5 -to HEX2S[0] ;# g HEX2S[0]

# set_location_assignment PIN_D4 -to HEX3[6] ;# a HEX3S[6]
# set_location_assignment PIN_F3 -to HEX3[5] ;# b HEX3S[5]
# set_location_assignment PIN_L8 -to HEX3[4] ;# c HEX3S[4]
# set_location_assignment PIN_J4 -to HEX3[3] ;# d HEX3S[3]
# set_location_assignment PIN_D6 -to HEX3[2] ;# e HEX3S[2]
# set_location_assignment PIN_D5 -to HEX3[1] ;# f HEX3S[1]
# set_location_assignment PIN_F4 -to HEX3[0] ;# g HEX3S[0]

;#####
;# UART (serial port) pin signals #
;#####

# set_location_assignment PIN_F14 -to RX
# set_location_assignment PIN_G12 -to TX

;#####
;## Audio CODEC ##
;#####

# set_location_assignment PIN_A6 -to AUD_ADCLRCK
# set_location_assignment PIN_B6 -to AUD_ADCDAT
# set_location_assignment PIN_A5 -to AUD_DACLK
# set_location_assignment PIN_B5 -to AUD_DACDAT

```

```

# set_location_assignment PIN_B4 -to AUD_XCK
# set_location_assignment PIN_A4 -to AUD_BCLK

#####
;# I2C control lines #
#####

# set_location_assignment PIN_A3 -to I2C_SCLK
# set_location_assignment PIN_B3 -to I2C_SDAT

#####
;# VGA interface #
#####

# set_location_assignment PIN_D9 -to RED[0]
# set_location_assignment PIN_C9 -to RED[1]
# set_location_assignment PIN_A7 -to RED[2]
# set_location_assignment PIN_B7 -to RED[3]
# set_location_assignment PIN_B8 -to GREEN[0]
# set_location_assignment PIN_C10 -to GREEN[1]
# set_location_assignment PIN_B9 -to GREEN[2]
# set_location_assignment PIN_A8 -to GREEN[3]
# set_location_assignment PIN_A9 -to BLUE[0]
# set_location_assignment PIN_D11 -to BLUE[1]
# set_location_assignment PIN_A10 -to BLUE[2]
# set_location_assignment PIN_B10 -to BLUE[3]
# set_location_assignment PIN_A11 -to HORIZ_SYNC
# set_location_assignment PIN_B11 -to VERT_SYNC

#####
;# PS2 interface #
#####

# set_location_assignment PIN_H15 -to PS2_CLK
# set_location_assignment PIN_J14 -to PS2_DAT

#####
;# SD Card interface #
#####

# set_location_assignment PIN_V20 -to SD_CLK

```

```

# set_location_assignment PIN_Y20 -to SD_CMD
# set_location_assignment PIN_W20 -to SD_DAT
# set_location_assignment PIN_U20 -to SD_DAT3

;#####
;# SRAM #
;#####
;# SRAM address
# set_location_assignment PIN_AA3 -to SRAM_ADDR[0]
# set_location_assignment PIN_AB3 -to SRAM_ADDR[1]
# set_location_assignment PIN_AA4 -to SRAM_ADDR[2]
# set_location_assignment PIN_AB4 -to SRAM_ADDR[3]
# set_location_assignment PIN_AA5 -to SRAM_ADDR[4]
# set_location_assignment PIN_AB10 -to SRAM_ADDR[5]
# set_location_assignment PIN_AA11 -to SRAM_ADDR[6]
# set_location_assignment PIN_AB11 -to SRAM_ADDR[7]
# set_location_assignment PIN_V11 -to SRAM_ADDR[8]
# set_location_assignment PIN_W11 -to SRAM_ADDR[9]
# set_location_assignment PIN_R11 -to SRAM_ADDR[10]
# set_location_assignment PIN_T11 -to SRAM_ADDR[11]
# set_location_assignment PIN_Y10 -to SRAM_ADDR[12]
# set_location_assignment PIN_U10 -to SRAM_ADDR[13]
# set_location_assignment PIN_R10 -to SRAM_ADDR[14]
# set_location_assignment PIN_T7 -to SRAM_ADDR[15]
# set_location_assignment PIN_Y6 -to SRAM_ADDR[16]
# set_location_assignment PIN_Y5 -to SRAM_ADDR[17]
;# SRAM data
# set_location_assignment PIN_AA6 -to SRAM_DQ[0]
# set_location_assignment PIN_AB6 -to SRAM_DQ[1]
# set_location_assignment PIN_AA7 -to SRAM_DQ[2]
# set_location_assignment PIN_AB7 -to SRAM_DQ[3]
# set_location_assignment PIN_AA8 -to SRAM_DQ[4]
# set_location_assignment PIN_AB8 -to SRAM_DQ[5]
# set_location_assignment PIN_AA9 -to SRAM_DQ[6]
# set_location_assignment PIN_AB9 -to SRAM_DQ[7]
# set_location_assignment PIN_Y9 -to SRAM_DQ[8]
# set_location_assignment PIN_W9 -to SRAM_DQ[9]
# set_location_assignment PIN_V9 -to SRAM_DQ[10]
# set_location_assignment PIN_U9 -to SRAM_DQ[11]
# set_location_assignment PIN_R9 -to SRAM_DQ[12]
# set_location_assignment PIN_W8 -to SRAM_DQ[13]

```

```

# set_location_assignment PIN_V8 -to SRAM_DQ[14]
# set_location_assignment PIN_U8 -to SRAM_DQ[15]
;# SRAM controls
# set_location_assignment PIN_AB5 -to SRAM_CE_N
# set_location_assignment PIN_Y7 -to SRAM_LB_N
# set_location_assignment PIN_T8 -to SRAM_OE_N
# set_location_assignment PIN_W7 -to SRAM_UB_N
# set_location_assignment PIN_AA10 -to SRAM_WE_N

;#####
;# SDRAM #
;#####
;# SDRAM address
# set_location_assignment PIN_W4 -to DRAM_ADDR[0]
# set_location_assignment PIN_W5 -to DRAM_ADDR[1]
# set_location_assignment PIN_Y3 -to DRAM_ADDR[2]
# set_location_assignment PIN_Y4 -to DRAM_ADDR[3]
# set_location_assignment PIN_R6 -to DRAM_ADDR[4]
# set_location_assignment PIN_R5 -to DRAM_ADDR[5]
# set_location_assignment PIN_P6 -to DRAM_ADDR[6]
# set_location_assignment PIN_P5 -to DRAM_ADDR[7]
# set_location_assignment PIN_P3 -to DRAM_ADDR[8]
# set_location_assignment PIN_N4 -to DRAM_ADDR[9]
# set_location_assignment PIN_W3 -to DRAM_ADDR[10]
# set_location_assignment PIN_N6 -to DRAM_ADDR[11]
;# SDRAM data
# set_location_assignment PIN_U1 -to DRAM_DQ[0]
# set_location_assignment PIN_U2 -to DRAM_DQ[1]
# set_location_assignment PIN_V1 -to DRAM_DQ[2]
# set_location_assignment PIN_V2 -to DRAM_DQ[3]
# set_location_assignment PIN_W1 -to DRAM_DQ[4]
# set_location_assignment PIN_W2 -to DRAM_DQ[5]
# set_location_assignment PIN_Y1 -to DRAM_DQ[6]
# set_location_assignment PIN_Y2 -to DRAM_DQ[7]
# set_location_assignment PIN_N1 -to DRAM_DQ[8]
# set_location_assignment PIN_N2 -to DRAM_DQ[9]
# set_location_assignment PIN_P1 -to DRAM_DQ[10]
# set_location_assignment PIN_P2 -to DRAM_DQ[11]
# set_location_assignment PIN_R1 -to DRAM_DQ[12]
# set_location_assignment PIN_R2 -to DRAM_DQ[13]
# set_location_assignment PIN_T1 -to DRAM_DQ[14]

```



```

# set_location_assignment PIN_T2 -to DRAM_DQ[15]
;# SDRAM controls
# set_location_assignment PIN_U3 -to DRAM_BA_0
# set_location_assignment PIN_V4 -to DRAM_BA_1
# set_location_assignment PIN_T3 -to DRAM_CAS_N
# set_location_assignment PIN_N3 -to DRAM_CKE
# set_location_assignment PIN_U4 -to DRAM_CLK
# set_location_assignment PIN_T6 -to DRAM_CS_N
# set_location_assignment PIN_R7 -to DRAM_LDQM
# set_location_assignment PIN_T5 -to DRAM_RAS_N
# set_location_assignment PIN_M5 -to DRAM_UDQM
# set_location_assignment PIN_R8 -to DRAM_WE_N

;#####
;# FLASH Memory #
;#####
;# FLASH address
# set_location_assignment PIN_AB20 -to FL_ADDR[0]
# set_location_assignment PIN_AA14 -to FL_ADDR[1]
# set_location_assignment PIN_Y16 -to FL_ADDR[2]
# set_location_assignment PIN_R15 -to FL_ADDR[3]
# set_location_assignment PIN_T15 -to FL_ADDR[4]
# set_location_assignment PIN_U15 -to FL_ADDR[5]
# set_location_assignment PIN_V15 -to FL_ADDR[6]
# set_location_assignment PIN_W15 -to FL_ADDR[7]
# set_location_assignment PIN_R14 -to FL_ADDR[8]
# set_location_assignment PIN_Y13 -to FL_ADDR[9]
# set_location_assignment PIN_R12 -to FL_ADDR[10]
# set_location_assignment PIN_T12 -to FL_ADDR[11]
# set_location_assignment PIN_AB14 -to FL_ADDR[12]
# set_location_assignment PIN_AA13 -to FL_ADDR[13]
# set_location_assignment PIN_AB13 -to FL_ADDR[14]
# set_location_assignment PIN_AA12 -to FL_ADDR[15]
# set_location_assignment PIN_AB12 -to FL_ADDR[16]
# set_location_assignment PIN_AA20 -to FL_ADDR[17]
# set_location_assignment PIN_U14 -to FL_ADDR[18]
# set_location_assignment PIN_V14 -to FL_ADDR[19]
# set_location_assignment PIN_U13 -to FL_ADDR[20]
# set_location_assignment PIN_R13 -to FL_ADDR[21]
# set_location_assignment PIN_AB16 -to FL_DQ[0]
# set_location_assignment PIN_AA16 -to FL_DQ[1]

```

```

# set_location_assignment PIN_AB17 -to FL_DQ[2]
# set_location_assignment PIN_AA17 -to FL_DQ[3]
# set_location_assignment PIN_AB18 -to FL_DQ[4]
# set_location_assignment PIN_AA18 -to FL_DQ[5]
# set_location_assignment PIN_AB19 -to FL_DQ[6]
# set_location_assignment PIN_AA19 -to FL_DQ[7]
# set_location_assignment PIN_AA15 -to FL_OE_N
# set_location_assignment PIN_W14 -to FL_RST_N
# set_location_assignment PIN_Y14 -to FL_WE_N

#####
;# JP1 - the left connector (GPIO 0) #
#####

# set_location_assignment PIN_A13 -to GPIO_0[0]
# set_location_assignment PIN_B13 -to GPIO_0[1]
# set_location_assignment PIN_A14 -to GPIO_0[2]
# set_location_assignment PIN_B14 -to GPIO_0[3]
# set_location_assignment PIN_A15 -to GPIO_0[4]
# set_location_assignment PIN_B15 -to GPIO_0[5]
# set_location_assignment PIN_A16 -to GPIO_0[6]
# set_location_assignment PIN_B16 -to GPIO_0[7]
# set_location_assignment PIN_A17 -to GPIO_0[8]
# set_location_assignment PIN_B17 -to GPIO_0[9]
# set_location_assignment PIN_A18 -to GPIO_0[10]
# set_location_assignment PIN_B18 -to GPIO_0[11]
# set_location_assignment PIN_A19 -to GPIO_0[12]
# set_location_assignment PIN_B19 -to GPIO_0[13]
# set_location_assignment PIN_A20 -to GPIO_0[14]
# set_location_assignment PIN_B20 -to GPIO_0[15]
# set_location_assignment PIN_C21 -to GPIO_0[16]
# set_location_assignment PIN_C22 -to GPIO_0[17]
# set_location_assignment PIN_D21 -to GPIO_0[18]
# set_location_assignment PIN_D22 -to GPIO_0[19]
# set_location_assignment PIN_E21 -to GPIO_0[20]
# set_location_assignment PIN_E22 -to GPIO_0[21]
# set_location_assignment PIN_F21 -to GPIO_0[22]
# set_location_assignment PIN_F22 -to GPIO_0[23]
# set_location_assignment PIN_G21 -to GPIO_0[24]
# set_location_assignment PIN_G22 -to GPIO_0[25]
# set_location_assignment PIN_J21 -to GPIO_0[26]

```

```
# set_location_assignment PIN_J22 -to GPIO_0[27]
# set_location_assignment PIN_K21 -to GPIO_0[28]
# set_location_assignment PIN_K22 -to GPIO_0[29]
# set_location_assignment PIN_J19 -to GPIO_0[30]
# set_location_assignment PIN_J20 -to GPIO_0[31]
# set_location_assignment PIN_J18 -to GPIO_0[32]
# set_location_assignment PIN_K20 -to GPIO_0[33]
# set_location_assignment PIN_L19 -to GPIO_0[34]
# set_location_assignment PIN_L18 -to GPIO_0[35]
```

```
#####
;# JP2 - the right connector (GPIO 1) #
#####
```

```
# set_location_assignment PIN_H12 -to GPIO_1[0]
# set_location_assignment PIN_H13 -to GPIO_1[1]
# set_location_assignment PIN_H14 -to GPIO_1[2]
# set_location_assignment PIN_G15 -to GPIO_1[3]
# set_location_assignment PIN_E14 -to GPIO_1[4]
# set_location_assignment PIN_E15 -to GPIO_1[5]
# set_location_assignment PIN_F15 -to GPIO_1[6]
# set_location_assignment PIN_G16 -to GPIO_1[7]
# set_location_assignment PIN_F12 -to GPIO_1[8]
# set_location_assignment PIN_F13 -to GPIO_1[9]
# set_location_assignment PIN_C14 -to GPIO_1[10]
# set_location_assignment PIN_D14 -to GPIO_1[11]
# set_location_assignment PIN_D15 -to GPIO_1[12]
# set_location_assignment PIN_D16 -to GPIO_1[13]
# set_location_assignment PIN_C17 -to GPIO_1[14]
# set_location_assignment PIN_C18 -to GPIO_1[15]
# set_location_assignment PIN_C19 -to GPIO_1[16]
# set_location_assignment PIN_C20 -to GPIO_1[17]
# set_location_assignment PIN_D19 -to GPIO_1[18]
# set_location_assignment PIN_D20 -to GPIO_1[19]
# set_location_assignment PIN_E20 -to GPIO_1[20]
# set_location_assignment PIN_F20 -to GPIO_1[21]
# set_location_assignment PIN_E19 -to GPIO_1[22]
# set_location_assignment PIN_E18 -to GPIO_1[23]
# set_location_assignment PIN_G20 -to GPIO_1[24]
# set_location_assignment PIN_G18 -to GPIO_1[25]
# set_location_assignment PIN_G17 -to GPIO_1[26]
```

```

# set_location_assignment PIN_H17 -to GPIO_1[27]
# set_location_assignment PIN_J15 -to GPIO_1[28]
# set_location_assignment PIN_H18 -to GPIO_1[29]
# set_location_assignment PIN_N22 -to GPIO_1[30]
# set_location_assignment PIN_N21 -to GPIO_1[31]
# set_location_assignment PIN_P15 -to GPIO_1[32]
# set_location_assignment PIN_N15 -to GPIO_1[33]
# set_location_assignment PIN_P17 -to GPIO_1[34]
# set_location_assignment PIN_P18 -to GPIO_1[35]

;#####
;# JTAG pins #
;#####

# set_location_assignment PIN_E8 -to TDI
# set_location_assignment PIN_D8 -to TCS
# set_location_assignment PIN_C7 -to TCK
# set_location_assignment PIN_D7 -to TDO

;#####
;# Device selection & Making connections safe #
;#####
;# Analysis & Synthesis Assignments
set_global_assignment -name FAMILY "Cyclone II"
set_global_assignment -name DEVICE_FILTER_PACKAGE FBGA
set_global_assignment -name DEVICE_FILTER_PIN_COUNT 484
set_global_assignment -name DEVICE_FILTER_SPEED_GRADE 7
;# fitter assignments
set_global_assignment -name DEVICE "EP2C20F484C7"
set_global_assignment -name RESERVE_ALL_UNUSED_PINS "AS INPUT TRI-
STATED"

set_global_assignment -name EDA_SIMULATION_TOOL "ModelSim-Altera
(VHDL) "
set_global_assignment -name EDA_OUTPUT_DATA_FORMAT VHDL -section_id
eda_simulation

```

נספח 3: שלבי עבודה מול תוכנת הפיתוח QUARTUS של חברת ALTERA

הקדמה

בעשורים הקודמים נהגו להרכיב מערכות לוגיות בעזרת רכיבים בדידים. דבר זה היה כרוך בעבודת חיווט רבה וגזל שטח עבודה גדול, וקשה לשינויים. כיום, ישנן תוכנות פיתוח המאפשרות לשרטט ולבדוק תקינות של מערכות לוגיות המכילות מאות אלפי שערים ורכיבי זיכרון על גבי מסך מחשב ואף לצרוב אותן לאחר בדיקה לרכיב בר תכנות אחד. היכולות של תוכנות הפיתוח ורכיבים ברי תכנות, וסכום זמן פיתוח בדיקה ויצור של מערכות מורכבות ואף מאפשרות מיזעור משמעותי של מערכות, עלויות פיתוח ויצור, מאפשרות עבודה בלוחות זמנים קצרים יחסית, משפרות זמני ביצוע של מערכת והגעה לוודאיות מוחלטת של פעולת מערכת. זאת מכיוון שהיא נבדקת על גבי מסך מחשב בכל שלב בפיתוחה.

אחת מהחברות בעולם המייצרות תוכנות פיתוח ורכיבים ברי תכנות היא חברת ALTERA ואנו נשתמש בתוכנת הפיתוח QUARTUS מטעמה אשר תאפשר לנו:

- א. לשרטט מערכות לוגיות
- ב. לבצע בדיקת הפעולה הלוגית של המערכת בדיאגרמת זמן.
- ג. לבצע צריבת המערכת המכילה שערים רבים ורכיבי זיכרון על רכיב בר תכנות אחד.

שלבי עבודה:

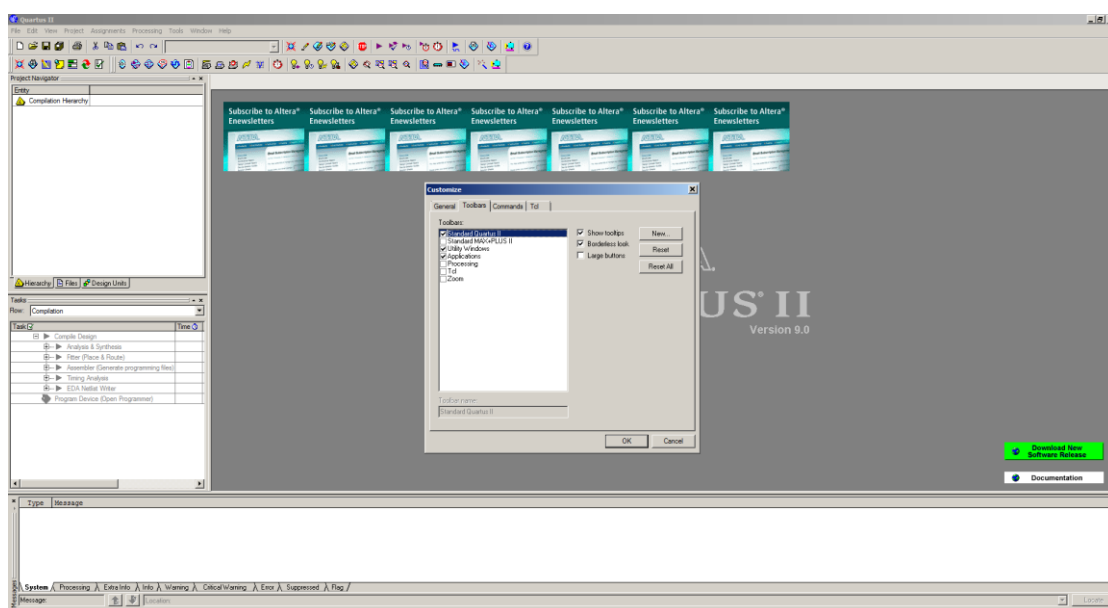
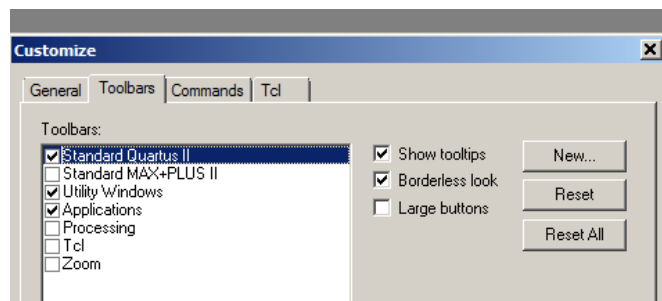
נלחץ על צלמית הפעלת התוכנה לאחר התקנתה



פעם אחת נבצע לפני תחילת עבודה ראשונה את הפעולות הבאות אשר יקלו ויזרזו את העבודה בהמשך:

נלחץ על לשונית TOOLS - < CUSTOMIZE

ונוודא V באפשרויות הבאות ונלחץ OK



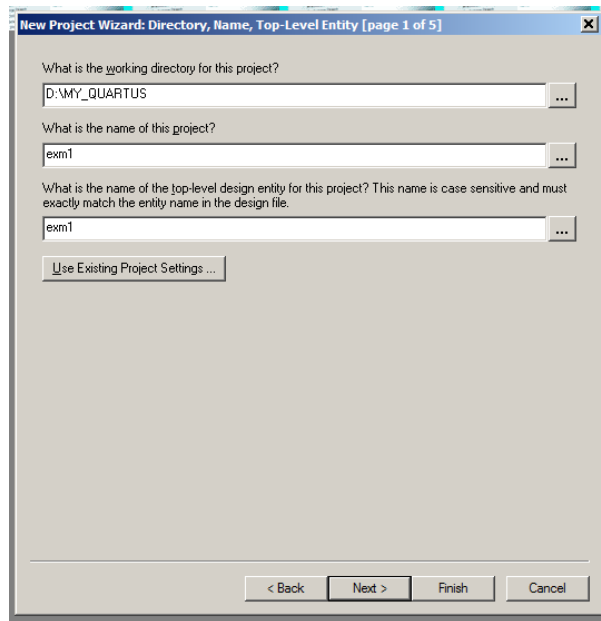
הדגמת שלבי העבודה על בדיקת פעולתו של שער OR ע"י

סימולציה בדיאגרמת זמן - WAVRFORM:

1. כל שרטוט אשר נרצה לבצע איתו גם סימולציה ובהמשך גם צריבה לרכיב צריך להיות מוגדר כפרוייקט.

נפתח פרויקט חדש על ידי לשונית FILE -> NEW PROJECT WIZARD <- NEXT
נגדיר את ספריית העבודה שלנו איפה שנוח לנו לשמור אותה במחשב בשם באנגלית,
וניתן שם לפרוייקט שלנו.

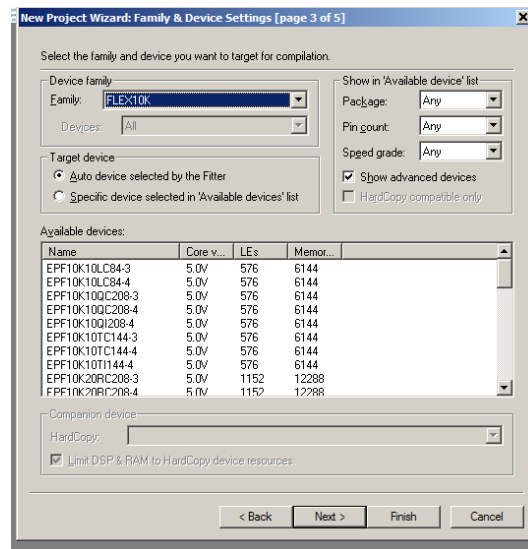
שם הפרוייקט הנוכחי הוא exm1.



ונילחץ על NEXT.

ומאחר וכרגע אין לנו רכיבים\תוכניות נוספות להוסיף לפרוייקט אז נילחץ שוב על NEXT.
בשלב זה נגדיר רכיב שעל תכונותיו אנו רוצים לעבוד, לבצע סימולציה ובהמשך גם לצרוב.

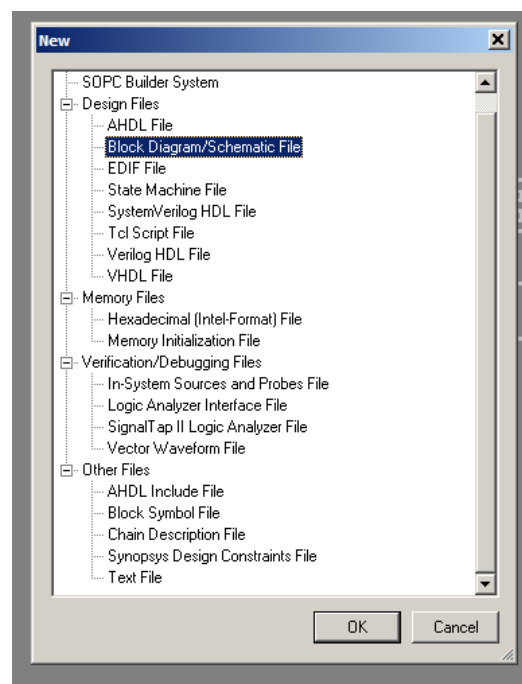
כיוון שבשלב הראשון נרצה לבצע הדמייה בלבד, סימולציה, ניבחר רכיב אשר זמן ההידור עליו מהיר FLEX10K, ונילחץ על NEXT.



מכיוון שאין כלים חיצוניים נוספים שנרצה להשתמש בהם, נילחץ שוב על NEXT ואז .FINISH.

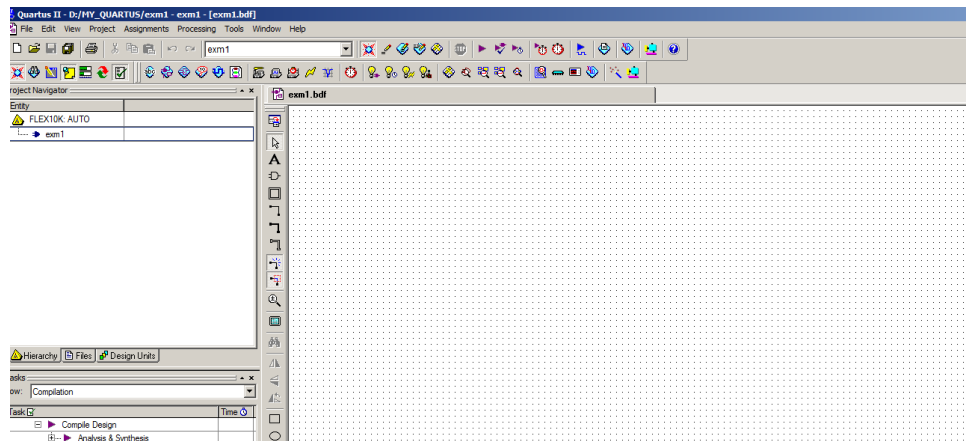
נכון לעכשיו נוצרה לנו תשתית לביצוע פרויקט ונבצע פתיחה של קובץ חדש ומומלץ לתת לו את שם הפרויקט עצמו.

נילחץ על לשונית FILE < NEW וניבחר את אפשרות BLOCK DIAGRAM

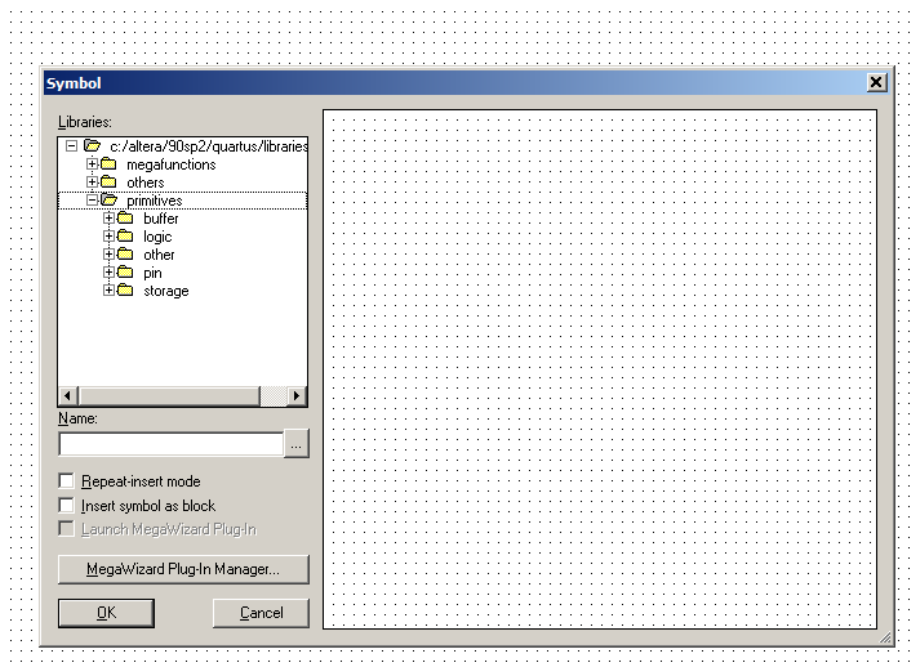


נילחץ על OK.

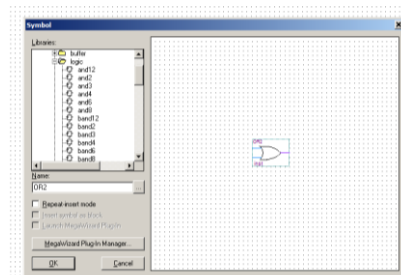
קיבלנו מסך שהוא "שולחן עבודה" עם שם קובץ אקראי block1.bdf ואנו נשנה לו את השם כרצוננו לשם הפרויקט על ידי FILE < SAVE AS < SAVE כעת מסך המחשב נראה כך:



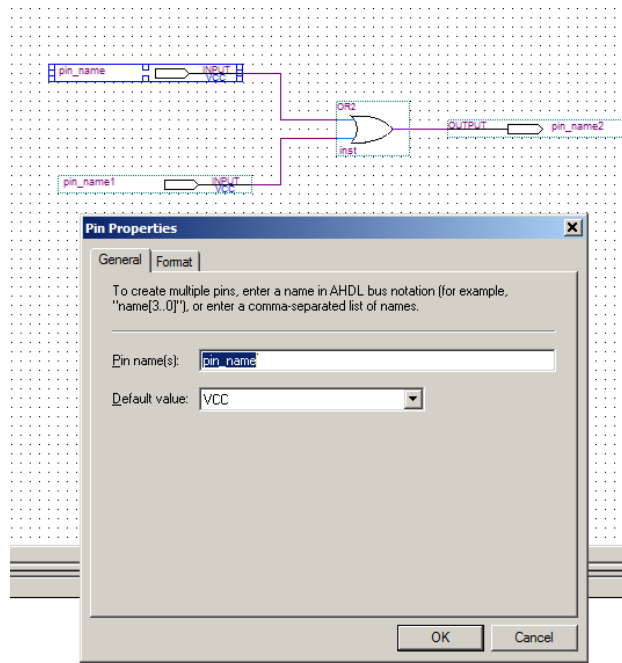
בלחיצה כפולה עם העכבר על שולחן העבודה אשר מסומן בחץ, נקבל תפריט רכיבים, SYMBIL, אשר ניתן יהיה להוסיף לשרטוט, ויש ללחוץ על הפלוס הקטן כדי לקבל את האפשרויות השונות, ואז ללחוץ שוב על הפלוס של האפשרויות הבסיסיות, PRIMITIVES.



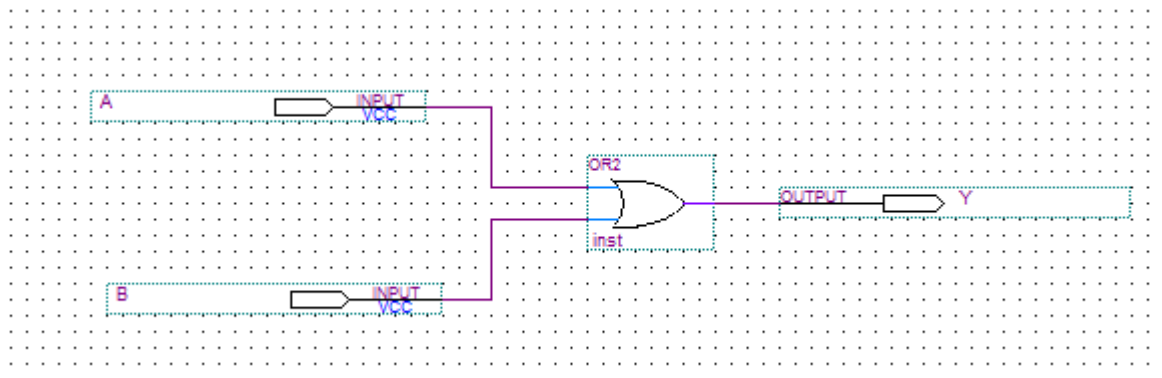
כעת נוסיף שער OR עם שתי כניסות לשרטוט על ידי לחיצה על הפלוס של LOGIC וחפוש השער או כתיבת שמו בתיבת טקסט NAME:



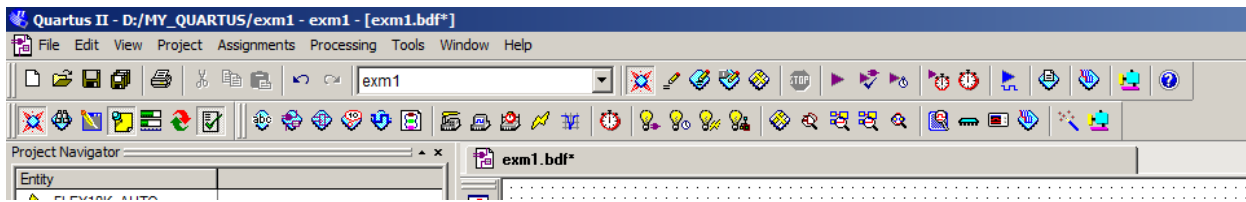
כשנלחץ על OK נמקם אותו כרצוננו בשולחן העבודה.
נוסיף כניסות ויציאות לשער על ידי לחיצה כפולה על שולחן העבודה, ומיתוך האפשרויות שפתחות לנו לאחר הלחיצה על הפלוס של PIN, נייבא INPUT ו OUTPUT.
נחבר אותם לשער בהתאם על ידי גרירת חיווט עם העכבר, ונלחץ פעמיים על כל אחד מהם כדי לתת להם שמות:



ניתן לכניסות את השמות A B ולמוצא את השם Y.



השרטוט מוכן וכעת ניתן לבצע עליו הידור (קומפילציה) אשר הופכת אותו לשפת מכונה, שמאפשרת לבדוק את תקינות החיווט ואת התקינות הלוגית שלו.
את ההידור נבצע על ידי הלחיצה על צלמית המשולש הסגול שבארגז הכלים START :COMPILATION

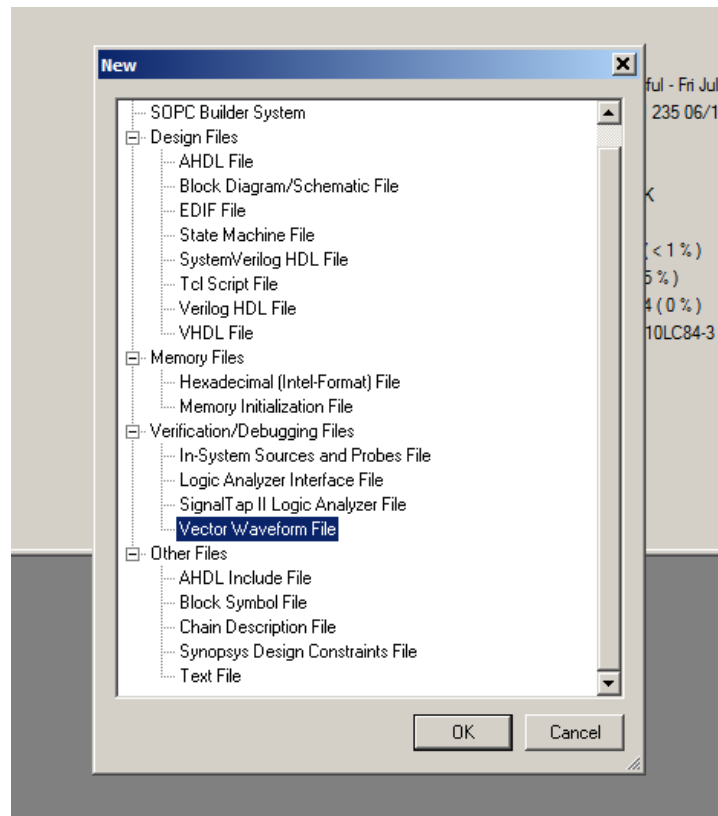


לאחר לחיצה על OK נקבל את תוצאות ההידור.

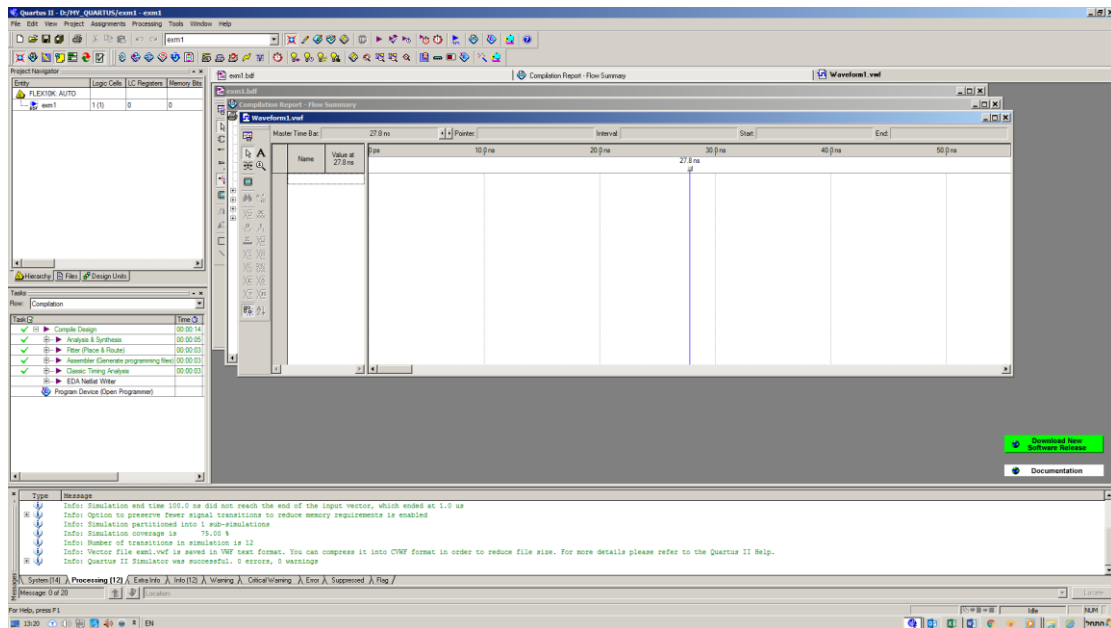
במידה שהמערכת תדווח על שגיאות, נלמד בהמשך איך לאתר לחזור ולתקן אותן עד לביצוע הידור מושלם.

כעת חשוב לבדוק שהמערכת שלנו עובדת לפי הציפיות. נבצע זאת על ידי בדיקה בדיאגרמת זמן:

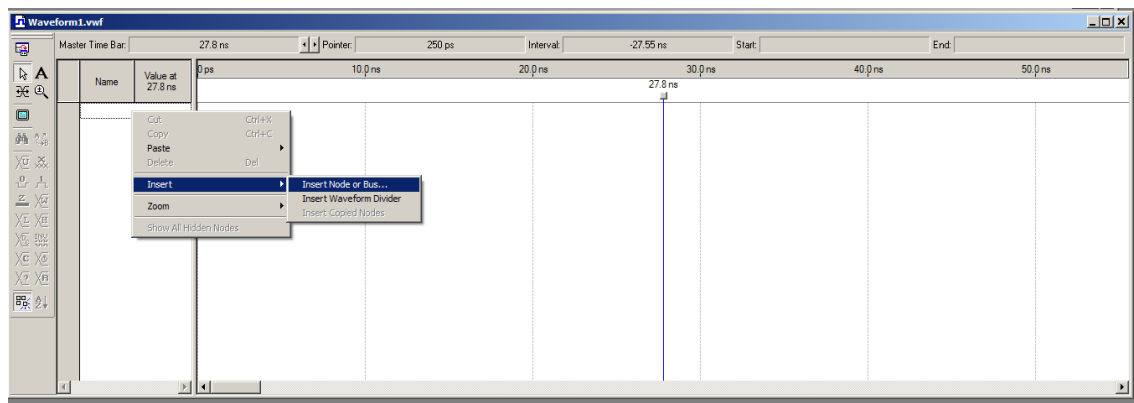
נפתח קובץ חדש לסימולציה על ידי FILE > NEW > VECTOR WAVEFORM FILE



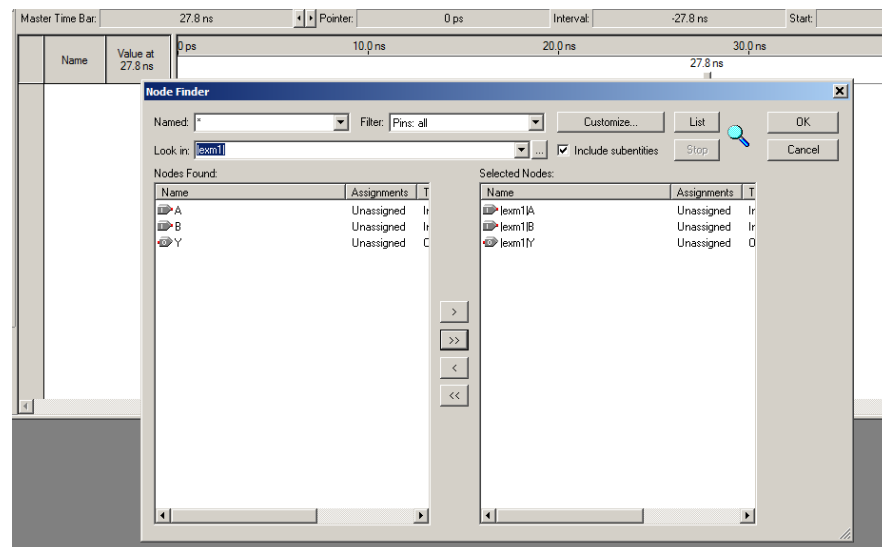
נלחץ על OK ונקבל את המסך הבא:



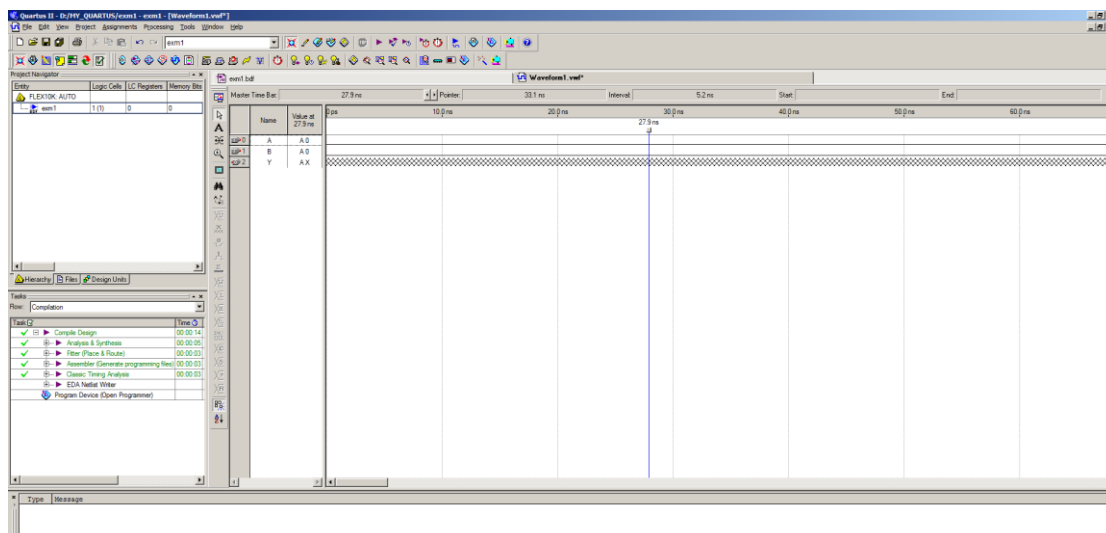
כדי להוסיף את הכניסות והיציאות אל דיאגרמת הזמן, נילחץ עם הלחצן הימני של העכבר מתחת ל NAME שמעמודה השמאלית של דיאגרמת הזמן - INSERT <- INSETRT <- NODES OR BUS



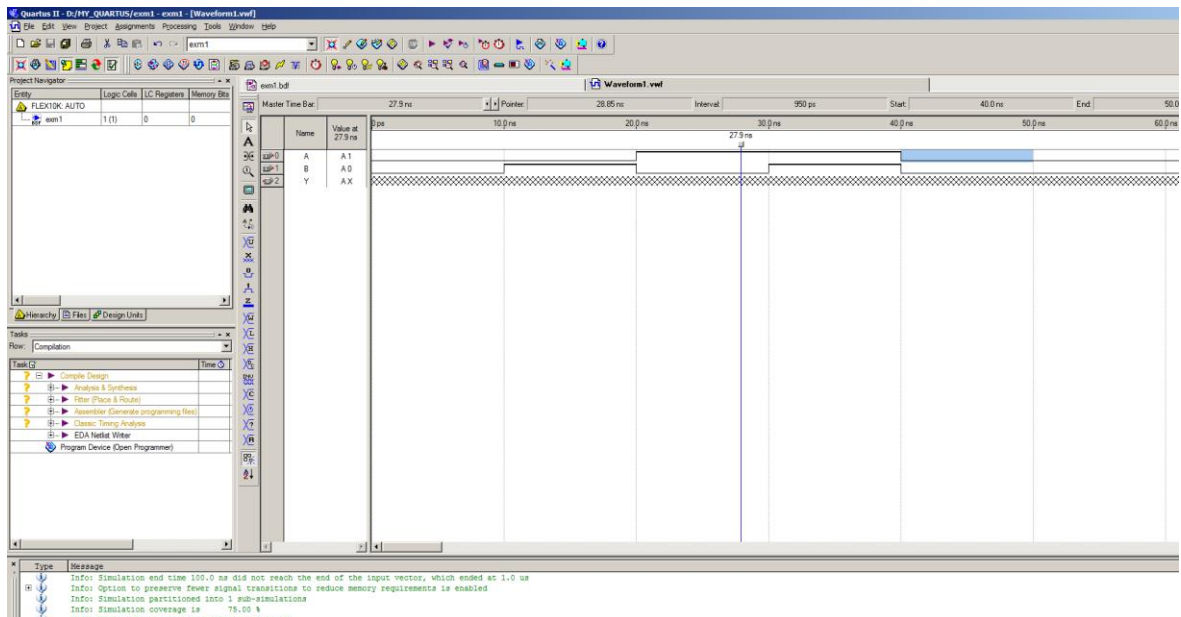
נלחץ על NODE FINDER ולאחר מכן נוודא שב FILTER נבחר PINS:ALL
נלחץ על LIST ובחץ הכפול ימינה נעביר את כל הכניסות והיציאות לצד ימין
ונילחץ על OK <- OK



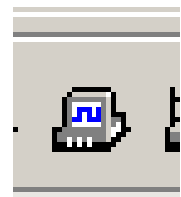
התקבל המסך הבא שבו יש את הכניסות והיציאות של המערכת המשורטטת.



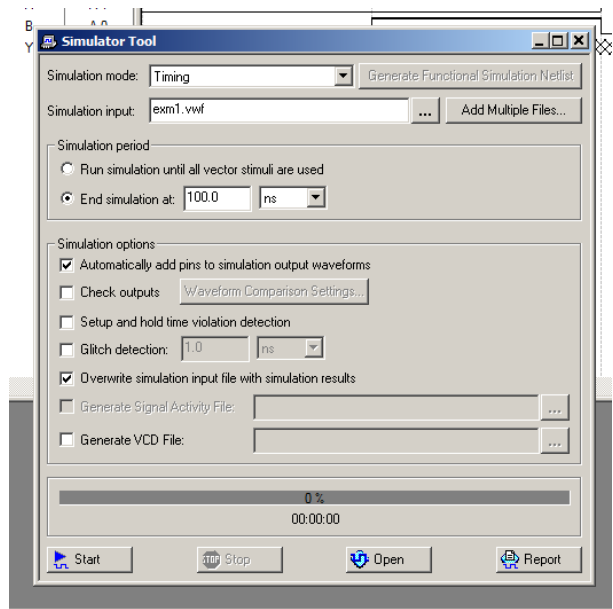
כעת נשרטט בעזרת ארגז הכלים שמופיע בצד שמאל את כל האפשרויות בין הכניסות. נסמן וגרור בעזרת העכבר את המקומות שבהם נזין '1' ונלחץ על הלחצן המתאים בארגז הכלים שמשמאל עד שנקבל את המצב הבא:



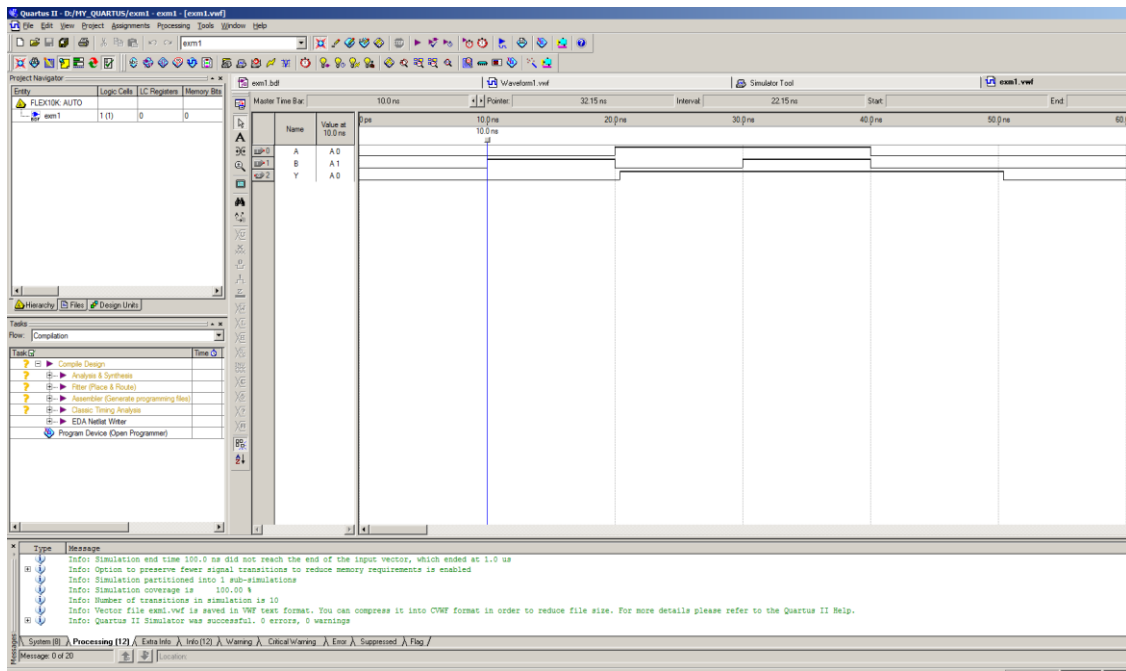
אפשר לראות שכל האפשרויות בין הכניסות שורטטו בדיאגרמת הזמן והמוצא ממתיין לפקודה שלנו להתעדכן על פי המערכת המשורטטת.
על מנת לעדכן את המוצא נילחץ על צלמית כלי הסימולטור המסומנת בחץ:



ונקבל את המסך הבא:

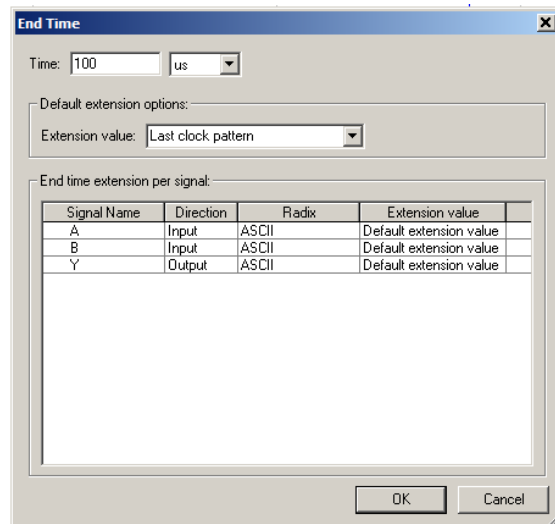


נילחץ על START <- אישור <- OPEN
ונקבל את מצב המוצא:

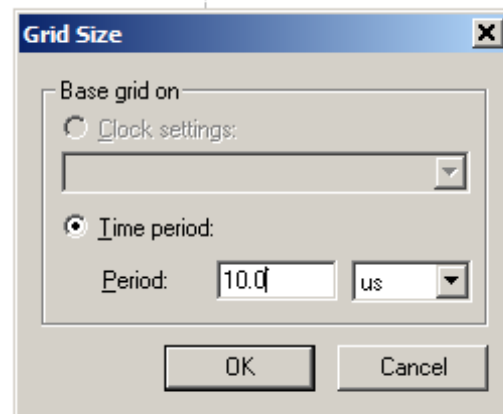


התוצאה שהתקבלה במסך לוקחת בחשבון את ה TPD של הרכיב ולכן המוצא מתעדכן בעיקוב.

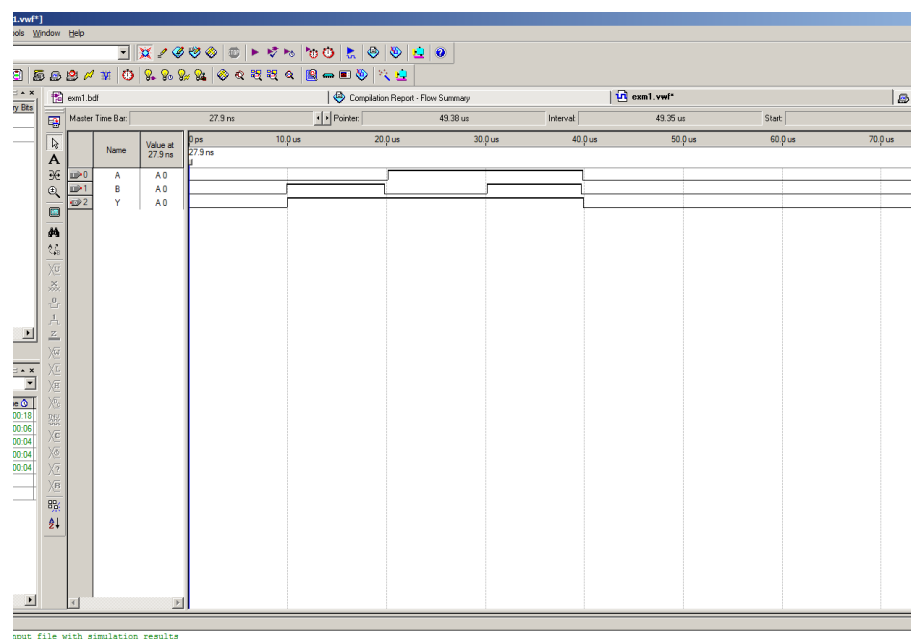
מומלץ לבצע שינויים בבסיסי הזמן של הסימולציה כדי לקבל תוצאות ברורות יותר. נאריך את זמן הסימולציה בלשונית FILE > END TIME כפי שמצוין בחלון הבא:



נשלוט על חלקי הסימולציה לשרטוט על ידי FILE > GRID SIZE כפי שמצוין בחלון הבא:



ניתן כעת לבצע ZOOM OUT על ידי לחיצה על CTRL ביחד עם גלילת גלגלת העכבר לאחור.
נשרטט שוב את האותות כניסה A, B כפי שביצענו בשלבים הקודמים ונריץ את הסימולציה על ידי הצלמת SIMULATOR TOOL ונקבל:



ניתן לראות פעולה ברורה יותר מאחר וזמני ה TPD קצרים בהרבה מזמני שרטוט האותות בסימולציה.

יחידה 2: יסודות בתורת החשמל

היקף: 60 שעות עיוני; 30 שעות מעבדה

פרק 1: מטען, כוח ושדה חשמלי

רציונל

זהו הפרק ראשון בספר הווירטואלי "יסודות בתורת החשמל". הפרק מכיל מושגים בסיסיים החיוניים להמשך לימוד הנושא. לימוד זה, מאפשר לתלמידים להתנסות במעבדות וירטואליות תוך כדי למידה עיונית וזאת על מנת שיוכלו לקבל המחשה מיטבית וביסוס טוב לחומר הנלמד. במהלך הלימוד מופעים תחת תתי הפרקים תרגולים נוספים, סרטונים להמחשה ואף בוחן מסכם. תרגול במנות קטנות, יאפשר למורה לבסס את הבנת החומר, לאתר קשיים ולסייע לתלמידים בטרם ילמד הנושא הבא.

מטרות

הפרק יעסוק בנושאים הבאים:

- מטען, כוח ושדה חשמלי לרבות יחידות .
- מבנה החומר
- הכוח החשמלי וחוק קולון
- השדה החשמלי
- התלמידים יידעו לחשב מטען אלקטרון, כוח על יחידת מטען במרחב השדה החשמלי.

עקרונות

- בספר קיים חומר תיאורטי להקניית המושגים. יש ללמד על פי הסדר, כיוון שהם מהווים תשתית לקראת הנושאים הבאים.
- יש לבצע את התרגולים המוצעים בספר, כדי להבטיח את ההבנה של הנושאים הנלמדים
- יש ללמד את החומר העיוני בליווי המחשה וביצוע מעבדות וירטואליות לצורך העמקה ולמידה משמעותית. לכל "מעבדה וירטואלית" מצורף דוח ניסוי ושלבי ביצוע הניסוי.

פעילויות

- סרטונים ככלי עזר להמחשת המושגים בפרק – כוחות ופוטנציאל חשמלי

- שאלות חזרה – קובץ מבדק פנימי לתלמידים.
- סימולציות (מעבדה וירטואלית) למטען חשמלי ויצירת השדה החשמלי
- משחק הוקי
- סימולציית תא חשמלי
- סימולציה לפעילות בנושא מוליכות חשמלית
- שאלות חזרה
- סימולציה בנושא זרם חשמלי
- סרטון המחשה לזרם החשמלי
- סימולציית סיכום לזרם החשמלי

פרק 2: מתחים וזרמים חשמליים

רציונל

פרק זה, מבסס ומעמיק את הידע בנושא המתחים והזרמים במעגל החשמלי. לאחר לימוד הידע הבסיסי בפרק הראשון, ייחשפו התלמידים לחיבור התיאורטי ושילוב הידע במעגל החשמלי. יחידת הוראה זו מורכבת ממושגים בסיסיים והכרחיים להרכבת המעגל והבנת אופן פעילותו.

מטרות

הפרק יעסוק בנושאים הבאים:

- פוטנציאל ומתח לרבות אנרגיה פוטנציאלית חשמלית.
- שדה חשמלי ואופן פעילותו.
- התא החשמלי כמקור מתח.
- הזרם החשמלי הגדרה, יחידות וחישוב במוליך.
- מוליכים ומבודדים.
- צפיפות הזרם.

עקרונות

- בפרק זה ייחשפו התלמידים לחיבור בין ההגדרות הבסיסיות במעגל החשמלי. יש ללמד ולהטמיע ידע בסדר המופיע בפרקים הראשונים.
- יש לתרגל את נושאי הפרק במנות קטנות.
- מיקום התרגילים והמבדקים נבחרו כך שהתלמידים יוכלו לבסס את הבנתם באופן הדרגתי.
- הפעילויות בספר, יאפשרו לתלמידים למידה משמעותית אשר מחברת בין הוראת החומר באופן תיאורטי לבין תרגול להטמעת הידע באופן וירטואלי.

פעילויות

- סרטון המציג את עיקרי המעגל החשמלי
- סרטון מימוש מתגים במעגל החשמלי
- שאלות מופשטות לתרגול
- מעבדה וירטואלית בנושא מעגל חשמלי ומכשירי מדידה
- סימולציית סכנת התחשמלות

- מבדק מסכם לפרק
- משימות מסכמות בקיט מעגלים חשמליים

פרק 3: התנגדות חשמלית

רציונל

פרק זה, אשר משלים את שני הפרקים הראשונים בספר. לאחר שלמדו התלמידים את נושא הפוטנציאל החשמלי, וזרם במעגל, נותר להשלים את המושג השלישי – התנגדות חשמלית. לאחר הטמעת שלושת המושגים המרכזיים במעגל החשמלי, יוכלו התלמידים לרכוש מיומנות במעגלים מורכבים יותר ובעלי סוגים שונים של חיבורים.

מטרות

בפרק זה נעסוק בנושאים הבאים:

- התנגדות מוליכים
- הגדרת התנגדות באמצעות מתח וזרם.
- התנגדות סגולית
- תכונות המבודדים

עקרונות

- התלמידים ילמדו את המושג התנגדות באמצעות שני מושגים אשר נלמדו בפרקים הקודמים.
- התלמידים יטמיעו את הידע בנושא המעגל החשמלי, הכולל בתוכו, מוליכים, נגדים, מקור מתח וזרם הזורם דרכו.
- הלמידה תתבצע במנות קטנות, ישנה חשיבות רבה לתרגל כל נושא וליצור תשתית להמשך למידה במעגלים מורכבים יותר.

פעילויות

- סימולציה "ג'ון טרבולטה"
- סרטון להמחשת מושגי זרם מתח ונגד.
- סימולציה להדמיית מעגל בעל צרכן, מוליך, זרם ומתח.
- סימולציה בנושא התנגדות סגולית
- סרטון בנושא התנגדות סגולית
- תרגול תיאורטי בנושא התנגדות סגולית התלויה בשטח חתך.
- סרטון המחשה למבודדים ותכונותיהם.

פרק 4: המעגל החשמלי

רציונל

אם ידועים הערכים של שניים מתוך שלושת הגדלים - מתח, זרם והתנגדות - נוכל לחשב את הגודל השלישי. אם נדע את שלושת הגדלים האלה. פרק זה נועד להעמיק למידה על סוגים שונים של נגדים ושילובם במעגל. כמו כן, לפרק זה תפקיד להטמיע למידה בנושא פתרון מעגלים חשמליים בעזרת מציאת זרם, מתח והתנגדות.

מטרות

בפרק זה נעסוק בנושאים הבאים:

- הגדרת מעגל חשמלי
- סוגי צרכנים
- מתג חשמלי
- סוגי נגדים ותכונותיהם.
- מדידת זרם ומתח במעגל החשמלי בעזרת מכשירי מדידה.
- סכנות התחשמלות

עקרונות

- התלמידים ישפרו את המיומנות של פתרון מעגלים חשמליים, אשר מצריכים שימוש בחוק אוהם.
- יש ללמד את החומר העיוני בליווי המחשה וביצוע מעבדות וירטואליות לצורך העמקה ולמידה משמעותית.
- לכל "מעבדה וירטואלית" מצורף דוח ניסויי ושלבי ביצוע הניסוי.

פעילויות

- סרטון בנושא המעגל החשמלי בעל שלושת האלמנטים- מתח, זרם ונגד.
- שאלות מופשטות לתרגול
- סרטון בנושא המתג החשמלי ותפקידו
- סימולציה להרכבת מעגל חשמלי – המעבדה הווירטואלית
- סימולציה בנושא התחשמלות
- פעילות מסכמת- מבדק.

פרק 5: חוק אוהם

רציונל

בפרקים הקודמים הגדרנו את ההתנגדות של מוליך כיחס בין המתח על המוליך לזרם דרכו. כמו כן למדנו שההתנגדות של מוליך תלויה באורכו, בשטח החתך שלו ובחומר שממנו הוא עשוי (כלומר, בהתנגדות הסגולית). ראינו גם כי הטמפרטורה משפיעה על ההתנגדות. עכשיו ברצוננו לבדוק אם ההתנגדות של מוליך תלויה גם במתח שבין קצותיו, ואם כתוצאה מכך היא תלויה גם בעוצמת הזרם העובר במוליך

מטרות

בפרק זה נעסוק בנושאים הבאים:

- חוק אוהם לרבות תיאור גרפי.
- הקשר בין המתח לזרם.
- מתגים במעגל החשמלי

עקרונות

- בפרק זה יתמודדו התלמידים עם חיבור של מוליכים, מקור מתח וצרכנים.
- התלמידים יחזרו על מושגים מפרקים קודמים כאשר הם משולבים במעגל החשמלי.
- יש ללמד את החומר העיוני בליווי המחשה וביצוע מעבדות וירטואליות לצורך העמקה ולמידה משמעותית.
- לכל "מעבדה וירטואלית" מצורף דוח ניסוי ושלבי ביצוע הניסוי.
- יש לבצע תרגול לפי נושא כדי לייצר תשתית יציבה להמשך לימוד של מעגליים חשמליים מורכבים.

פעילויות

- דף הסבר למעבדה קיט מעגל חשמלי כולל רכיבים.
- שאלות מופשטות לתרגול
- מעבדה וירטואלית בנושא מתח וסוללה
- מעבדה בנושא מוליכות חשמלית.
- סרטונים להמחשת חוק אוהם במעגל החשמלי.
- סרטון להמחשת תפקיד המתג במעגל החשמלי.
- פעילות במעבדה הווירטואלית כיצד להדליק נורה
- יצירת מעגלים טוריים ומקבילים בקיט – המעבדה הווירטואלית.

פרק 6: ההספק החשמלי

רציונל

בפרקים הקודמים למדנו על חיבור נגד למקור מתח. בפרק זה נרחיב את ידיעת התלמידים בנושא. המתח בין הדקי המקור גורם לכך שעל האלקטרונים החופשיים בנגד, המחובר למקור, פועל כוח חשמלי. אלקטרונים אלה נעים בנגד, רוכשים אנרגיה הנוצרת מתנועה. במהלך תנועתם מתנגשים ביוני הסורג. כתוצאה מהתנגשויות אלה, הנגד מתחמם. אנו רואים שבתהליך זה יש הפיכת אנרגיה מצורה אחת לצורה אחרת – אנרגיה חשמלית הופכת לחום.

מטרות

בפרק זה נעסוק בנושאים הבאים:

- הגדרות ההספק
- צריכת אנרגיה חשמלית ושימוש ביחידות אנרגיה

עקרונות

- תהליך פיסיקאלי אשר מתקיים במעגל החשמלי, בעת זרימת זרם על נגד, גורם להתחממותו. התופעה עליה ילמדו התלמידים בפרק זה, הינה תופעת מעבר אנרגיה המופקת במעגל, כלומר מעגל חשמלי לאנרגיית חום. המושג נקרא "הספק".
- התלמידים ילמדו תופעות פיסיקליות המתרחשות בעת הפעלת המעגל החשמלי

פעילויות

- סימולציה בנושא הספק חשמלי
- סרטון להמחשת המושג ההספק החשמלי
- תרגול מופשט למציאת ההספק במעגל החשמלי
- פעילות בנושא צריכת חשמל
- פעילות מסכמת במעבדה הווירטואלית בנושא הספק חשמלי

פרק 7: חוקי קירכהוף

רציונל

אם ידועים הערכים של שניים מתוך שלושת הגדלים - מתח, זרם והתנגדות - נוכל לחשב את הגודל השלישי. אם נדע את שלושת הגדלים האלה, נוכל לחשב גודל נוסף – ההספק. אם נדע את ארבעת הגדלים האלה, נאמר שפתרנו את המעגל. אך לא תמיד. בפרק זה נראה כיצד להתיר מעגלים חשמליים המחוברים באופן שאינו מאפשר פתרון על ידי שימוש פשוט בחוק אוהם בלבד.

מטרות

בפרק זה נעסוק בנושאים הבאים:

- חוק הזרמים של קירכהוף
- חוק המתחים של קירכהוף
- סימון מתחים במעגלים חשמליים
- הצורה הכללית של חוק המתחים של קירכהוף

עקרונות

- בפרק זה ילמדו שני חוקים, שיעזרו לתלמידים לפתור מעגלים, המכילים כמה מקורות מתח ונגדים. חוקים אלה נקראים חוקי קירכהוף, על שם המדען שניסח אותם לראשונה.
- התלמידים ישפרו את המיומנות של פתרון מעגלים חשמליים, אשר מצריכים שימוש בחוקים נוספים.
- יש ללמד את החומר העיוני בליווי המחשה וביצוע מעבדות וירטואליות לצורך העמקה ולמידה משמעותית.
- לכל "מעבדה וירטואלית" מצורף דוח ניסויי ושלבי ביצוע הניסוי.
- קיט המעגל החשמלי מהפרק קודם, יכול לשמש את תלמידכם גם בפרק זה.

פעילויות

- משחק בנושא חוק המתחים
- שאלות מופשטות לתרגול
- פעילות מסכמת בחוק המתחים והזרמים של קירכהוף

פרק 8: מעגלים טוריים, מקבילים ומעורבים

רציונל

בפרק הקודם למדנו את שני חוקי קירכהוף. בפרק זה ניעזר בחוקי קירכהוף, כדי לפתור מעגלים, המכילים מקור מתח אחד וכמה נגדים. בהמשך ניעזר בחוקים אלה, כדי לפתור מעגלים מורכבים, המכילים כמה חוגים.

מטרות

בפרק זה נעסוק בנושאים הבאים:

- מעגל טורי
- התנגדות הקו
- חיבור נגדים במקביל
- מעגל מעורב
- נגדים משתנים
- הספק ונצילות
- הספק מרבי לצרכן
- כא"מ ומקורות מתח
- התא החשמלי וצורות החיבור במעגל.

עקרונות

- בפרק זה מתווספים מספר גדול יותר של צרכנים ומקורות מתח.
- התלמידים יחזרו על מושגים מפרקים קודמים כאשר הם משולבים במעגל החשמלי
- התלמידים ילמדו ויעמיקו ידע, בפתרון מעגלים חשמליים מורכבים.
- יש ללמד את החומר העיוני בליווי המחשה וביצוע מעבדות וירטואליות לצורך העמקה ולמידה משמעותית.
- לכל "מעבדה וירטואלית" מצורף דוח ניסויי ושלבי ביצוע הניסוי.

פעילויות

- דף הסבר למעבדה קיט מעגל חשמלי כולל רכיבים.
- משימות למעגלים טורי ומקבילי בקיט.
- שאלות מופשטות לתרגול
- משחק בנושא מעגלים טוריים, מקבילים ומעורבים
- מעבדה וירטואלית בנושא מתח וסוללה

פרק 9: שיטות לפתרון מעגלים

רציונל

עד כה למדנו לפתור מעגלים טוריים, מקביליים או מעורבים, כאשר כל מעגל הכיל מקור מתח יחיד. לצורך הפתרון, חישבנו את ההתנגדות השקולה של הנגדים במעגל, ובאמצעות חוק אום מצאנו את הזרמים והמתחים השונים במעגל. בפרק זה, ילמדו התלמידים שתי שיטות לפתרון מעגלים חשמליים מורכבים.

מטרות

- שיטת זרמי החוגים
- משפט תבנין

עקרונות

- מעגלים חשמליים הצורכים שימוש בשיטות אלו, הנם מעגלים מורכבים. הבסיס לשיטות אלו נבנה מחוק האוהם, ולפיכך יש לוודא הבנה של הפרק הקודם (מעגלים טוריים, מקבילים ומעורבים).
- יש לבצע הקנייה ותרגול לכל שיטה בנפרד.
- יש לתרגל במנות קטנות לאחר כל חלק מיחידת ההוראה, כיוון שהמעגלים בפרק זה הינם מעגלים חשמליים מורכבים.

פעילויות

- דף הסבר למעבדה קיט מעגל חשמלי כולל רכיבים.
- משימות למעגלים טורי ומקבילי בקיט.
- שאלות מופשטות לתרגול
- משחק בנושא מעגלים טוריים, מקבילים ומעורבים
- מעבדה וירטואלית בנושא מתח וסוללה

יחידה 3: אלקטרוניקה תקבילית וספרתית

היקף: 90 שעות עיון; 60 שעות מעבדה

רציונל ומבנה המדריך למורה

במדריך זה ניתן למצוא את תוכניות הלימוד, מטרות הלימוד של כל פרק ומה נדרש מן התלמיד להיות מסוגל לבצע בסיום כל פרק.

בתחילת המדריך, ניתן למצוא את תכנית הלימודים כפי שהוגדרה על-ידי משרד החינוך ולאחר מן עבור כל פרק ניתן תיאור קצר של הרציונל, מטרות ביצועיות, הפניות למקורות עזר במידת הצורך, שימוש באסטרטגיות הוראה שונות ומגוונות, קשיים צפויים וכיצד להתמודד עימם, זאת כדי לאפשר לתלמידים להבין טוב יותר את נושאי הלימוד.

פרק 1 - אותות תקביליים וספרתיים

רציונל

בפרק זו ירכוש התלמיד שני סוגי ידע המהווים בסיס לפרקים הבאים; הצהרתי ותהליכי. התלמיד ילמד להבחין בין אותות ספרתיים לבין אותות תקביליים על פי תרשימים גרפיים. מתוך התרשימים ידע התלמיד לקבוע מאפיינים שונים: מחזוריות, סימטריות.

התלמיד ילמד לקרוא תרשימים גרפים של אותות מחזוריים ולחשב את זמן המחזור ואת התדירות שלהם. בעיקר יכיר התלמיד את האות המחזורי הסינוסואידלי וידע לבטא אותו באופן מתמטי. התלמיד ידע להתאים תרשים גרפי של אות מחזורי סינוסואידלי לתיאור המתמטי שלו ולהיפך, להתאים אות מתמטי לתיאור גרפי נתון.

יעדים לימודיים

בסוף יחידת הלימוד התלמיד יהיה מסוגל לעשות את הפעולות הבאות:

1. הגדרת המושגים: אות, אות מחזורי, אות חילופין.
2. אפיון וסיווג אותות לפי:
 - א. מחזורי / לא מחזורי.
 - ב. תקבילי / ספרתי.
 - ג. סימטרי / אנטי-סימטרי / ללא סימטריה.
 - ד. חילופין / לא חילופין.
3. מציאת הערך המרבי, ערך "שיא לשיא" – על-פי גרף של אות.
4. מציאת זמן המחזור של אות סינוסואידלי (כולל תדירות ותדירות זוויתית) לפי הפונקציה המתמטית או התיאור הגרפי שלו.

5. רישום משוואה של פונקציית סינוס על-פי גרף.
6. סרטוט תרשים עקרוני של אות סינוסואידלי לפי פונקציית האות.
7. מציאת הזמנים שבהם האות הסינוסואידלי מקבל ערך מרבי, ערך מזערי או ערך של אפס.
8. סרטוט אות סינוסואידלי באמצעות מחשב (אקסל / MATLAB / MULTISIM / אחר) – על-פי: תדירות האות, תנופתו וזווית המופע שלו.
9. סרטוט אות ספרתי בינארי באמצעות תוכנת הדמיה.
10. מעבדה: הצגת תמונה של אות סינוסואידלי על-גבי סקופ ומציאת הפרמטרים שלו.
11. מעבדה: הצגת תמונה של אות ריבועי (ספרתי) על-גבי סקופ, מציאת התדר שלו וערכי רמות המתח.

אסטרטגיות הוראה

אמצעי המחשה

מומלץ לקיים את השיעור בחדר הכולל לפחות מחשב ומטול עבור המורה על מנת לאפשר המחשה של החומר הנלמד. בחלק מהשיעורים רצוי להוסיף לעמדת המורה מחולל אותות ומשקף תנודות כדי להמחיש את החומר הנלמד.

למידה מבוססת פתרון בעיות

במידת האפשר כדאי לקיים את השיעור בחדר לימוד הכולל מחשבים על מנת להשתמש בספר זה הכולל דוגמאות ובעיות לפתרון. בחלק מהבעיות, לפי שיקול המורה, מומלץ לחלק את הכיתה לקבוצות בהן יתקיים דיון על הבעיה ופתרונה.

למידה תוך התנסות

- מומלץ לקיים חלק מהשיעורים במעבדה הכוללת את המכשירים מחולל אותות ומשקף תנודות והטיל על התלמיד מטלות של יצירה וקריאת אותות על פי המכשירים.
- תפיסות שגויות וקשיים צפויים
- השימוש בפונקציה המתמטית $\sin$ דורשת התייחסות מיוחדת כאשר משתמשים בפונקציה לפי פרמטר הזמן: $v(t) = V_m \cdot \sin(\omega \cdot t)$. כאשר נדרש התלמיד לחשב את ערך הפונקציה בתלות בערכי התדירות הזוויתית והזמן הוא יכול לחשב זאת בשתי שיטות:
1. המרת יחידות המכפלה $\omega \cdot t$ מיחידות rad/s למעלות ולהשתמש במחשבון לחישוב ערך הסינוס.

2. העברת המחשבון ממצב מעלות (deg) למצב רדיאנים (rad) ולהשתמש במחשבון לחישוב ערך הסינוס.

הבחירה אם להציג לתלמיד אחת מהשיטות או שתיהן נתונה למורה. במידה ונבחרת השיטה השנייה עלולה להתקבל תשובה שגויה. מקור השגיאה הוא באופן הפעולה של המחשבון. השגיאה הנפוצה אצל חלק מהתלמידים היא תופעת "לשכוח לעבור" לאופן פעולה הנכון.

פרק 2 – מגברי שרת

רציונל

אחד מאבני היסוד של מעגלים רבים הוא מגבר השרת. קיימים מגברי שרת רבים בעלי מאפיינים ייחודיים המתאימים ליישומים שונים. תוכנית הלימודים מתחילה בפרק זה כדי לספק לתלמיד כלי מערכתי כדי להבין כיצד פועלים מעגלים חשמליים. בשלב מאוחר יותר נחשף התלמיד לרכיב ממנו עשוי מגבר השרת – הטרנזיסטור.

בפרק זו ירכוש התלמיד שני סוגי ידע; הצהרתי ותהליכי. התלמיד ילמד מהו מגבר שרת אידיאלי ומהם תכונותיו. ידע לזהות את שמם של מעגלים בתצורות שונות הכוללות את מגבר השרת: מגבר הופך מופע, מגבר עוקב, מגבר יחידה, מגבר מחבר, מגבר מחסר.

התלמיד ידע לחשב את ההגבר של כל אחת מתצורות השונות של המעגלים לפי עקרונות מובהקים. התלמיד ידע לסרטט תרשימים המתארים את הקשר בין אותות כניסה לאותות יציאה וכן את אופיין המעבר של מגבר שרת בחוג פתוח.

התלמיד ידע לתכנן מעגלים חשמליים המבוססים על המעגלים שנלמדו לפי דרישות מוגדרות.

יעדים לימודיים

בסוף יחידת הלימוד התלמיד יהיה מסוגל לעשות את הפעולות הבאות:

אפיוני מגבר שרת בחוג פתוח

1. סרטוט סימול סכמתי הכולל שני הדקי כניסה, הדק מוצא והדקים של מתחי הזנה.
2. זיהוי ההדקים של מגבר השרת וציון התפקידים שלהם.
3. סרטוט חיבור של שני מקורות הזנה למקורות מתח מעשיים.
4. רישום התכונות של מגבר שרת אידיאלי (התנגדות כניסה – ∞ , התנגדות מוצא – 0, הגבר חוג פתוח – ∞ , רוחב סרט – ∞).
5. נתינת הסבר מדוע נדרשת התנגדות מבוא אין-סופית ממגבר מתח אידיאלי.
6. נתינת הסבר מדוע נדרשת התנגדות מוצא אפסית ממגבר מתח אידיאלי.
7. ציון היתרון של הגבר חוג פתוח אין-סופי בתכנון מגבר עם משוב שלילי.
8. הסברת המושג רוחב סרט אין-סופי בתור תכונה של מגבר שרת אידיאלי.

מגבר הופך מופע, מגבר עוקב מופע ומגבר יחידה

1. סרטוט תרשים חשמלי של מגבר הופך מופע, של מגבר עוקב מופע ושל מגבר יחידה.
2. בניית מעגל חשמלי עם מגבר שרת מעשי על-פי הסרטוט החשמלי.
3. חישוב הגבר המתח במגבר הופך מופע, במגבר עוקב מופע ובמגבר יחידה.
4. תכנון מגבר הופך / עוקב מופע וחישוב ערכי הנגדים שיש לחבר אליו לפי גודל ההגבר.
5. חישוב מתח המוצא על-פי מתח המבוא וערכי הנגדים במגבר הופך מופע, במגבר עוקב מופע ובמגבר יחידה.
6. סרטוט אות המוצא המתקבל לפי מתח המבוא הסינוסואידלי וערכי הנגדים במגבר הופך מופע ובמגבר עוקב מופע.
7. חישוב הזרמים בנגדי המשוב לפי מתח המבוא במגבר הופך מופע ובמגבר עוקב מופע.
8. רישום התכונות של מגבר יחידה (הגבר מתח, התנגדות מבוא, התנגדות מוצא).

מגבר מסכם, מגבר מחסר ומעגלים של מגברי שרת כלליים

1. זיהוי תצורת החיבור של המגבר: מחבר / מחסר / מסכם משוקלל / מחסר משוקלל.
2. תכנון מגבר מחבר / מחסר וחישוב ערכי הנגדים על-פי: $v_o(t) = v_1(t) \pm v_2(t)$.
3. תכנון מגבר מסכם / מחסר משוקלל וחישוב ערכי הנגדים על-פי: $v_o(t) = A \cdot v_1(t) \pm B \cdot v_2(t)$.
4. חישוב מתחים וזרמים הפועלים על הנגדים במגבר מחבר או מחסר.
5. חישוב וסרטוט של אות המוצא המתקבל במגבר מחבר או מחסר, אם אותות המבוא מורכבים ממתח DC ומתח AC.

אסטרטגיות הוראה

למידה מבוססת פתרון בעיות

במידת האפשר כדאי לקיים את השיעור בחדר לימוד הכולל מחשבים על מנת להשתמש בספר זה הכולל דוגמאות ובעיות לפתרון.

בחלק מהבעיות, לפי שיקול המורה, מומלץ לחלק את הכיתה לקבוצות בהן יתקיים דיון על הבעיה ופתרונה.

למידה תוך התנסות

מומלץ לקיים חלק מהשיעורים במעבדה הכוללת מחשבים ומכשירים המתאימים להתנסות. על התלמיד להתנסות בתוכנת סימולציה בה יוכל לבחון ביתר קלות את ההשפעה של פרמטרים שונים במעגל על אופי תגובתו. ההתנסות במעגלים מעשיים תמחיש לתלמיד את ההבדל בין מעגל להלכה לבין מעגל למשעה.

אמצעי הערכה חלופיים

ניתן לבקש מהתלמידים לבצע מטלת חקר כהערכה חלופית. התלמיד יקבל דרישות המתייחסות לתכנון מגבר. על התלמיד יהיה להציג חלופות למימוש המגבר תוך כדי השוואה בין המימושים השונים (מורכבות, עלות וכדומה). לאחר הבחירה התלמיד יבדוק את הפתרון בתוכנת סימולציה ויבחן את התכנון. המטלה תוגש כמסמך הכולל את הדרישות, החלופות לתכנון, נימוקים לחלופה שנבחרה, תוצאות ביצוע הסימולציה, סיכום והסקת מסקנות. כדאי להבהיר לתלמיד שמטרת המטלה היא התהליך עצמו והמסקנות הסופיות מהתהליך גם אם לא הצליח לקיים את כל הדרישות.

תפיסות שגויות וקשיים צפויים

השימוש במונח "אדמה מדומה" עלול לגרום לתלמיד לחשוב בטעות כי קיים מתח אפס (אדמה = אפס) בנקודה, ולא כך היא. "אדמה מדומה" היא מקרה פרטי ויש להשתמש במונח זה במקרים בהם אכן יש מתח אפס בנקודה אחת המאלץ מתח אפס בנקודה האחרת. כמו כן, להבהיר באופן חד משמעי, שמונח זה אינו קובע שאכן בנקודה יש מתח אפס באופן אמיתי. ומכאן אם נחבר נקודה זו למתח אפס המעגל לא יפעל בצורה תקינה.

המונח "קצר מדומה" הוא מונח מדויק יותר המציין כי מתח בנקודה אחת "משתקף" לנקודה השנייה. לתיאור מצב זה ניתן להשתמש במונח "מתח משתקף".

פרק 3 – חיישנים ומתמרים

רציונל

כיום, כמעט כל אדם נחשף למונח חיישן הזמין בכל מכשיר טלפון חכם או במזגן בבית. החשיבות של חיישן גדולה ועל כן על התלמיד להכיר את עקרונות החישה והמדידה. בפרק זו ירכוש התלמיד שני סוגי ידע; הצהרתי ותהליכי. התלמיד יכיר את המושגים חיישן ומתמר ויבחין בין השניים. כן ידע לזהות סוגים שונים של חיישנים: קול, חום ואור ושימושים אפשריים שלהם.

התלמיד ידע לחשב את ההתנגדות של חיישן RTD על פי הטמפרטורה ולהיפך, את הטמפרטורה על פי ההתנגדות. התלמיד ידע להתייחס לחיישן האור על פי התיאור הגרפי ויכול לקבוע את אופן התנהגות בעוצמות שונות.

בפרק זה נעשה שימוש בחומר הנלמד בפרק הקודם, מגברי שרת. מצופה מהתלמיד להשתמש בידע אותו צבר בפרק הקודם על מנת למתח מעגלים הכוללים חיישנים ומגברי שרת בתצורות השונות.

יעדים לימודיים

בסוף יחידת הלימוד התלמיד יהיה מסוגל לעשות את הפעולות הבאות:

המרת אותות פיזיקליים

1. ציון דוגמה אחת לפחות של חיישן (או מתמר) הממיר תנועה לאות חשמלי ולהפך.
2. ציון דוגמה אחת לפחות של חיישן (או מתמר) הממיר אות אופטי לאות חשמלי ולהפך.
3. ציון דוגמה אחת לפחות של חיישן (או מתמר) הממיר אות תרמי לאות חשמלי ולהפך.
4. ציון דוגמה של חיישן (או מתמר) הממיר אות אקוסטי (קול) לאות חשמלי ולהפך.

חיישן RTD: אופיין המתאר את תלות ההתנגדות בטמפרטורה, מעגלי יישום.

1. חילוץ התנגדות ה-RTD בטמפרטורה הנתונה מתוך אופיין תלות ההתנגדות בטמפרטורה.
2. ניתוח מעגל הכולל חיישן RTD ומשווה בחוג פתוח.
3. ניתוח מעגל הכולל חיישן RTD ומגבר הופך מופע (או לא הופך מופע).

חיישן LDR: אופיין המתאר את תלות ההתנגדות בעוצמת האור, מעגלי יישום.

1. חילוץ התנגדות ה-LDR בתאורה הנתונה מתוך אופיין תלות ההתנגדות בעוצמת האור.
2. ניתוח מעגל הכולל חיישן LDR ומשווה בחוג פתוח.
3. ניתוח מעגל הכולל חיישן LDR ומגבר הופך מופע (או לא הופך מופע).

אסטרטגיות הוראה

למידה מבוססת פתרון בעיות

במידת האפשר כדאי לקיים את השיעור בחדר לימוד הכולל מחשבים על מנת להשתמש בספר זה הכולל דוגמאות ובעיות לפתרון.

בחלק מהבעיות, לפי שיקול המורה, מומלץ לחלק את הכיתה לקבוצות בהן יתקיים דיון על הבעיה ופתרונה.

למידה תוך התנסות

מומלץ לקיים חלק מהשיעורים במעבדה הכוללת מחשבים ומכשירים המתאימים להתנסות. על התלמיד להתנסות בתוכנת סימולציה בה יוכל לבחון ביתר קלות את ההשפעה של פרמטרים שונים במעגל על אופי תגובתו. ההתנסות במעגלים מעשיים תמחיש לתלמיד את ההבדל בין מעגל להלכה לבין מעגל למשעה.

אמצעי הערכה חלופיים

ניתן לבקש מהתלמידים לבצע מטלת חקר כהערכה חלופית. לדוגמה, התלמיד יחקור מהו עיקרון הפעולה של חיישן רברס ברכב. התלמיד יתאר את העקרונות הפיזיקליים של המערכת ויסביר את עיקרון פעולתה. לא יידרש מהתלמיד לבצע חישובים מתחום החשמל והאלקטרוניקה. התלמיד יוכל להציג לכיתה את תוצאות עבודתו במהלך שיעור.

פרק 4 – משוואה בחוג פתוח

רציונל

מגבר השרת המשמש כמשוואה בסיסי הוא אבן יסוד במעגלי גילוי מצב כמו גלאי אור או גלאי טמפרטורה. בלימודים מתקדמים מרחיבים את השימוש במשוואה (משוואה שמידט).

בפרק זו ירכוש התלמיד שני סוגי ידע; הצהרתי ותהליכי. התלמיד ילמד מהו משוואה בסיסי ומהם תכונותיו. ידע לזהות את מתח הייחוס.

התלמיד ידע לסרטט את אופיין המעבר של המשוואה הבסיסי ולחשב את ערך נקודת המפנה באופיין. כן ידע התלמיד לקבוע את ערכי המוצא של המשוואה על פי מתחי ההזנה המסופקים אליו.

התלמיד ידע לנתח מעגל משוואה בסיסי ולקבוע את צורת אות המוצא המתקבל מאות מבוא תקבילי מחזורי ואות שאינו מחזורי.

התלמיד ידע לנתח את התגובה של מעגל משוואה בסיסי הכולל חיישן ולהגדיר את הקשר בין הגודל הפיזיקלי הנמדד על ידי החיישן לבין מתח המוצא שלו.

יעדים לימודיים

בסוף יחידת הלימוד התלמיד יהיה מסוגל לעשות את הפעולות הבאות:

1. זיהוי מתח הסף של משוואה בחוג פתוח.
2. ציון מתחי הרוויה של משוואה בחוג פתוח אידאלי לפי מתחי ההזנה.
3. סרטוט אופיין מעבר של מעגל נתון: מעגל שכניסת אות המבוא שלו בהדק (+) ומעגל שכניסת אות המבוא שלו בהדק (-).
4. סרטוט אות המוצא של משוואה בחוג פתוח על-פי אות המבוא ומתח הייחוס.
5. חישוב מתח הייחוס של משוואה בחוג פתוח שיש לו מקור מתח ושני נגדים (או פוטנציומטר – נגד משתנה).
6. עריכת כמה חישובים במעגלים עם משוואה בחוג פתוח וחיישן כלשהו.
7. זיהוי של משוואה חלון.
8. עריכת חישובים במשוואה חלון.
9. בניית מעגל משוואה באמצעות תוכנת הדמיה וחיבור מכשירים אליו, כמו מחולל אותות ומשקף תנודות.

אסטרטגיות הוראה

למידה מבוססת פתרון בעיות

במידת האפשר כדאי לקיים את השיעור בחדר לימוד הכולל מחשבים על מנת להשתמש בספר זה הכולל דוגמאות ובעיות לפתרון.

בחלק מהבעיות, לפי שיקול המורה, מומלץ לחלק את הכיתה לקבוצות בהן יתקיים דיון על הבעיה ופתרונה.

למידה תוך התנסות

מומלץ לקיים חלק מהשיעורים במעבדה הכוללת מחשבים ומכשירים המתאימים להתנסות. על התלמיד להתנסות בתוכנת סימולציה בה יוכל לבחון ביתר קלות את ההשפעה של פרמטרים שונים במעגל על אופי תגובתו. ההתנסות במעגלים מעשיים תמחיש לתלמיד את ההבדל בין מעגל להלכה לבין מעגל למשעה.

אמצעי הערכה חלופיים

ניתן לבקש מהתלמידים לבצע מטלת חקר כהערכה חלופית. התלמיד יקבל מעגל שלא נלמד בכיתה כדוגמת משוואה חלון הכולל כניסה אחת המתקבלת מחיישן טמפרטורה או אור ושתי יציאות המחוברות אל דיודת led. התלמיד יערוך, על פי הנחייה, סדרת מדידות בתוכנת סימולציה על מנת למצוא את הקשר בין היציאה לכניסות. את התוצאות יערך בטבלה באמצעותה יסביר את התוצאות שהתקבלו. בנוסף יציג התלמיד את הקשר בין הערכים הפיזיקליים של החיישן לבין מצב דיודות ה- led.

תפיסות שגויות וקשיים צפויים

בעיה נפוצה בקרב התלמידים היא סרטוט של אופיין המעבר של המשוואה הבסיסי. על מנת למנוע שגיאות על התלמיד לציין את מתחי ההזנה ואת המתח בהדק שאינו תלוי במתח המוגדר כמבוא – נקודת המפנה באופיין. את הערכים שהתקבלו לסמן במערכת צירים של מתח מבוא ומתח מוצא. בשלב האחרון על התלמיד לבחור ערך אות כניסה מימין או משמאל לנקודת המפנה ולבחון את השפעת מתח המבוא על מתח המוצא.

פרק 5 – דיודות

רציונל

כידוע, הדיודה היא רכיב נפוץ מאוד במעגלים חשמליים כגון מיישרים, גלאים, שיכור. פרק זה מציג את יסודות השימוש בדיודה כבסיס לפרקי לימוד מתקדמים.

בפרק זו ירכוש התלמיד שני סוגי ידע; הצהרתי ותהליכי. התלמיד ילמד לזהות דיודה גרילה ודיודה פולטת אור, את כיוון ההולכה שלהן ואת מתחי הפעולה הטיפוסיים.

התלמיד ידע לסרטט את אופיין הדיודה האידאלית, את אופיין הדיודה עם מודל מפל המתח הקבוע ואת המודל עם מפל מתח קבוע והתנגדות פנימית. התלמיד ידע לחשב את נקודת העבודה של הדיודה. כלומר, את הזרם העובר בדיודה בזמן פעולתה.

התלמיד ידע את השימוש של דיודות במעגלי יישור חצי גל וגל ושלם.

התלמיד ידע לתכנן מעגל הפעלה לדפ"א (led) ולקבוע את ערך ההתנגדות עבור נקודת העבודה של הדפ"א.

יעדים לימודיים

בסוף יחידת הלימוד התלמיד יהיה מסוגל לעשות את הפעולות הבאות:

1. סרטוט הסימול של הדיודה, ציון ההדקים שלה וציון כיוון ההולכה.
2. רישום מצבי הפעולה של דיודה אידאלית.
3. סרטוט אופיין זרם-מתח של דיודה אידאלית.
4. סרטוט אופיין זרם-מתח של דיודה מעשית.
5. סרטוט מודל מקורב של דיודה – מודל מפל מתח קבוע.
6. חישוב נקודת העבודה של הדיודה במעגל טורי שיש בו מקור מתח, נגד ודיודה – לפי מודל מפל מתח קבוע.
7. שימושי דיודה:
 - מיישר חד-דרכי – סרטוט מעגל וצורות גלים בעבור דיודה אידאלית ולפי מודל מפל מתח קבוע.
 - מיישר דו-דרכי (גשר דיודות) – סרטוט מעגל וצורות גלים בעבור דיודה אידאלית ולפי מודל מפל מתח קבוע.
8. חישוב נקודת העבודה של דפ"א (LED) במעגל טורי שיש בו מקור מתח, נגד ודיודה.

9. תכנון הערכים של מעגל בסיסי הנחוץ לשם הפעלה תקינה של דפ"א (LED).

אסטרטגיות הוראה

למידה מבוססת פתרון בעיות

במידת האפשר כדאי לקיים את השיעור בחדר לימוד הכולל מחשבים על מנת להשתמש בספר זה הכולל דוגמאות ובעיות לפתרון.

בחלק מהבעיות, לפי שיקול המורה, מומלץ לחלק את הכיתה לקבוצות בהן יתקיים דיון על הבעיה ופתרונה.

למידה תוך התנסות

מומלץ לקיים חלק מהשיעורים במעבדה הכוללת מחשבים ומכשירים המתאימים להתנסות. על התלמיד להתנסות בתוכנת סימולציה בה יוכל לבחון ביתר קלות את ההשפעה של פרמטרים שונים במעגל על אופי תגובתו. ההתנסות במעגלים מעשיים תמחיש לתלמיד את ההבדל בין מעגל להלכה לבין מעגל למשעה.

אמצעי הערכה חלופיים

ניתן לבקש מהתלמידים לבצע מטלת חקר כהערכה חלופית. התלמיד יקבל מעגל שלא נלמד בכיתה כדוגמת משווא חלון הכולל כניסה אחת המתקבלת מחיישן טמפרטורה או אור ושתי יציאות המחוברות אל דיודת led. התלמיד יערוך, על פי הנחייה, סדרת מדידות בתוכנת סימולציה על מנת למצוא את הקשר בין היציאה לכניסות. את התוצאות יערך בטבלה באמצעותה יסביר את התוצאות שהתקבלו. בנוסף יציג התלמיד את הקשר בין הערכים הפיזיקליים של החיישן לבין מצב דיודות ה- led.

תפיסות שגויות וקשיים צפויים

בעיה נפוצה בקרב התלמידים היא סרטוט של אופיין המעבר של המשוואה הבסיסי. על מנת למנוע שגיאות על התלמיד לציין את מתחי ההזנה ואת המתח בהדק שאינו תלוי במתח המוגדר כמבוא – נקודת המפנה באופיין. את הערכים שהתקבלו לסמן במערכת צירים של מתח מבוא ומתח מוצא. בשלב האחרון על התלמיד לבחור ערך אות כניסה מימין או משמאל לנקודת המפנה ולבחון את השפעת מתח המבוא על מתח המוצא.

פרק 6 – טרנזיסטור דו-נושאי

רציונל

כידוע, הטרנזיסטור הוא אחד מאבני היסוד של מעגלים אלקטרוניים רבים. עם השנים הטרנזיסטור הדו-נושאי תופס פחות מקום במעגלים ומפנה את מקומו למשפחה המבוססת על טרנזיסטור תוצא השדה (fet). על אף זאת פרק זה מציג את אבן היסוד הראשונה שפותחה ועדיין משמשת במעגלים מסוימים או כחלק ממשפחת כלאיים (למשל igbt).

בפרק זו ירכוש התלמיד שני סוגי ידע; הצהרתי ותהליכי. התלמיד ילמד לזהות את שמות ההדקים של טרנזיסטור דו-נושאי מסוג NPN ו- PNP. לציין את כיווני הזרמים והמתחים בכל אחד מההדקים. התלמיד ידע לזהות עבור איזה ערכי מתח V_{BE} יהיה הטרנזיסטור במצב הולכה או במצב קטעון. התלמיד ידע את שימושי הטרנזיסטור כמגבר וכמתג ואת מצבי הפעולה שלו בכל אחד משימושי הטרנזיסטור תוך שימוש באופייני המוצא $I_C = f(V_{CE})$ של הטרנזיסטור.

התלמיד ידע לרשום את משוואות המתח והזרם לפי **חוקי קירכהוף** בעבור שני מעגלי הטרנזיסטור, את **חוק אוהם** עבור כל אחד מהנגדים (ידע שנרכש בתורת החשמל) ואת יחס הזרמים בטרנזיסטור. התלמיד יחשב מתוך המשוואות את נקודת העבודה (I_C, V_{CE}) של הטרנזיסטור. על פי תוצאת החישוב על התלמיד להסיק מהו מצב הפעולה בו נמצא הטרנזיסטור: פעיל או רוויה.

התלמיד ידע להוסיף את קו העבודה של הטרנזיסטור על מערכת הצירים של אופייני המוצא $I_C = f(V_{CE})$ שלו ולסמן את נקודת העבודה.

התלמיד ידע לנתח מעגל הכולל טרנזיסטור במקדם עצמי תוך שימוש **במשפט תבנין** (ידע הנרכש בתורת החשמל). השימוש במשפט תבנין מפשט את המעגל לצורה הבסיסית בעלת שני מעגלים שתוארו לעיל.

התלמיד ידע להסביר את פעולת הטרנזיסטור באות קטן ואת התיאור הגרפי של זרם הקולט והמתח בין הקולט לפולט בתלות בזרם הבסיס. התלמיד ידע לסרטט את מעגל התמורה של הטרנזיסטור לאות קטן לפי פרמטרי h ולחשב את הגבר המתח והזרם של הטרנזיסטור.

יעדים לימודיים

בסוף יחידת הלימוד התלמיד יהיה מסוגל לעשות את הפעולות הבאות:

1. סרטוט הסימולים הסכמתיים של טרנזיסטור מסוג NPN ושל טרנזיסטור מסוג PNP.
2. זיהוי ההדקים של הטרנזיסטור: קולט, פולט ובסיס.
3. רישום הקשרים המתמטיים במצב פעיל בין הזרמים: I_B , I_C , I_E .
4. זיהוי שלושת מצבי הפעולה של הטרנזיסטור: פעיל, רוויה וקיטעון.
5. חישוב נקודת העבודה של מעגל עם טרנזיסטור במצב פעיל בזמן חיבור C.E עם מקדם עצמי (עם נגד R_E ובלי נגד R_E).
6. סרטוט קו העבודה של מעגל עם טרנזיסטור בזמן חיבור C.E (עם נגד R_E ובלי נגד R_E).
7. מציאת פתרון גרפי לנקודת עבודה באמצעות קו העבודה ואופייני הטרנזיסטור.
8. סרטוט מעגל תמורה מופשט (h_{ie} , h_{fe}) של טרנזיסטור לאות חילופין.
9. חישוב הגבר המתח בזמן חיבור C.E.
10. חישוב הגבר הזרם בזמן חיבור C.E.
11. הסבר התפקיד של קבל הצימוד (בין המגבר למקור ובין המגבר לעומס) ושל קבל העקיפה (קבל המקביל לנגד R_E).
12. סרטוט מתח המוצא של מגבר עם טרנזיסטור בזמן חיבור C.E (עם קבל צימוד ובלי קבל צימוד).
13. בדיקה אם תנאי הרוויה מתקיים בטרנזיסטור ($\beta I_B > I_C$).
14. סרטוט מעגל בסיסי של טרנזיסטור הפועל בתור מתג.
15. תכנון מעגל של טרנזיסטור הפועל בתור מתג (הערך של ספק הכוח והנגדים R_C , R_B).
16. ניתוח מעגל הפעלה של דפ"א (LED) באמצעות טרנזיסטור הפועל בתור מתג.

אסטרטגיות הוראה

למידה מבוססת פתרון בעיות

במידת האפשר כדאי לקיים את השיעור בחדר לימוד הכולל מחשבים על מנת להשתמש בספר זה הכולל דוגמאות ובעיות לפתרון.

בחלק מהבעיות, לפי שיקול המורה, מומלץ לחלק את הכיתה לקבוצות בהן יתקיים דיון על הבעיה ופתרונה.

באמצעות תוכנת סימולציה יוכל התלמיד לבדוק את תוצאות החישוב של בעיות ולהסיק מכך על דרך הפתרון.

למידה תוך התנסות

מומלץ לקיים חלק מהשיעורים במעבדה הכוללת מחשבים ומכשירים המתאימים להתנסות. על התלמיד להתנסות בתוכנת סימולציה בה יוכל לבחון ביתר קלות את ההשפעה של פרמטרים שונים במעגל על אופי תגובתו. ההתנסות במעגלים מעשיים תמחיש לתלמיד את ההבדל בין מעגל להלכה לבין מעגל למשעה.

אמצעי הערכה חלופיים

ניתן לבקש מהתלמידים לבצע מטלת חקר כהערכה חלופית. התלמיד יקבל מעגל המשמש כמגבר או כמתג. התלמיד יצטרך לתכנן את מעגל המתאים לדרישות המורה תוך שימוש בכלים שרכש עד כה: אותות (פרק 1), מגברי שרת (פרק 2), חיישנים ומתמרים (פרק 4) והדיודה (פרק 5). מומלץ לעודד את התלמיד להשתמש במנועי חיפוש על מנת למצוא מגוון דוגמאות היכולות לסייע לו בפתרון המטלה. התלמיד יבצע את המטלה בתוכנת סימולציה ויגיש למורה מסמך תיעוד של כל התהליך.

תפיסות שגויות וקשיים צפויים

אחד מהקשיים הנפוצים במעגלי טרנזיסטור הוא המעבר ממעגל הכולל מקורות מתח ישר, מקור מתח חילופים (אות קטן) וקבלים (עקיפה או צימוד) למעגל תמורה של אות קטן. את תהליך ההמרה יש להציג בצורה אלגוריתמית. חובה לנמק את שלבי האלגוריתם כך שלתלמיד תהיה בקרה עצמית.

פרק 7 – ספקים ומייצבי מתח

רציונל

בפרק זה מוצג חומר לימודי של ספק כוח המושתת על שימוש בשנאי. על אף שמקובל היום להשתמש בספקי כוח ממותגים שלהם יתרונות רבים, עדיין חשוב להציג את מבנה הספק המסורתי מהטעמים של התבוננות מערכתית על מכלול המרת אותות וחשיפה לרכיב שנאי. יתרה מכך ניתן בשלב זה להסביר ביתר קלות את ספק הכוח המסורתי בהתבסס על פרקי הלימוד הקודמים מאשר את ספק הכוח הממותג.

בפרק זו ירכוש התלמיד שני סוגי ידע; הצהרתי ותהליכי. התלמיד יכיר את התרשים של ספק הכוח המתחיל במקור מתח החילופים ומסתיים במתח ישר ומייצב. התלמיד ידע לסרטט את צורות אות המתח בכל אחד מחלקי התרשים. התלמיד יכיר את הרכיבים קבל, שנאי ודיוודת זנר וידע לזהות את תפקידם במעגל. כמו כן יכיר את המבנה של מעגל ייצוב טורי וידע לתאר את מרכיביו ותפקידם. בסוף הפרק ילמד התלמיד לקרוא דפי מפרט של מייצב מתח ממשפחת 78xxx.

התלמיד ידע לחשב מתחים וזרמים על פי משוואות המתח והזרם של שנאי אידיאלי ולסרטט את המתחים בנקודות חיבור של מערכת הכוללת שנאי אידיאלי ומיישר מתח שלם המבוסס על גשר דיוודת (דיוודת, פרק 5).

כמו כן התלמיד יתאר את ההשפעה של חיבור קבל במוצא המיישר ותאר את ההשפעה בצורה גרפית וחישובית.

בחלקו הבא של הפרק יכיר התלמיד את המונח מייצב מתח טורי ויכיר מעגלים שונים המשמשים כמייצבי המתח ובכללם רכיב ממשפחת 78xxx.

יעדים לימודיים

בסוף יחידת הלימוד התלמיד יהיה מסוגל לעשות את הפעולות הבאות:

1. לסרטט תרשים עקרוני של ספק כח (שנאי, מיישר, מסנן ומייצב)
2. לסרטט, סרטוט עקרוני, את המתח במוצא של כל אחת מהיחידות העקרונית של ספק הכח.
3. להגדיר את המושגים: ספק כוח מייצב וספק כוח לא מייצב
4. שנאי אידיאלי
 - א. לסרטט את הסימול החשמלי.
 - ב. לחשב את יחס ההשנאה בהנתן מתחי המבוא והמוצא.

- ג. לחשב את יחס ההשנאה בהנתן זרמי המבוא והמוצא.
- ד. לציין את ההבדל בין שנאי אידיאלי לשנאי מעשי.

5. מיישרים (נושא זה מופיע גם בפרק דיודות)

- א. לסרטט מעגל חשמלי למימוש מיישר חד-דרכי
- ב. לסרטט את האות במוצא מיישר חד-דרכי
- ג. לסרטט מעגל חשמלי למימוש מיישר דו-דרכי באמצעות גשר דיודות.
- ד. לסרטט את האות במוצא מיישר דו-דרכי
- ה. לבצע הדמיה ו/או ניסוי מעשי של מיישר חד-דרכי.

6. מעגל הסינון

- א. להסביר את תפקידו של המסנן במוצא המיישר
- ב. לציין את הגורמים המשפיעים על ערכו של מתח האדווה במוצא המסנן
- ג. לחשב את גודל מתח האדווה עבור נתוני מעגל.

7. מעגל הייצוב

- א. להסביר את הצורך במייצב מתח
- ב. לסרטט אופיין של דיודת זנר.
- ג. לרשום את התכונות הנדרשות ממייצב מקבילי.
- ד. לסרטט מייצב מתח מקבילי הכולל דיודת זנר.
- ה. לחשב את תחום ערכי ההתנגדויות הנדרש לפעולה תקינה של המייצב.

8. מייצב מתח טורי (לא בתכנית הלימודים – העשרה)

- א. לסרטט תרשים חשמלי עקרוני של מייצב טורי הכולל מגבר שרת, טרנזיסטור דיודת זנר ונגדים.
- ב. לחשב את מתח המוצא המיוצב.
- ג. לחשב את ההספק המירבי מתפתח על הטרנזיסטור

9. הרכיב המוכלל 78XX

- א. לאתר דף נתונים של הרכיב מהאינטרנט.
- ב. למצוא את מתחי וזרמי העבודה מתוך דף נתונים.

10. ביצוע ניסוי הכולל את הרכיב המוכלל 78XX

- א. הרכבת מעגל ניסוי בריקם ומדידת מתח מוצא.
- ב. מדידת מתחים וזרמים עבור עומסים שונים.

אסטרטגיות הוראה

למידה מבוססת פתרון בעיות

במידת האפשר כדאי לקיים את השיעור בחדר לימוד הכולל מחשבים על מנת להשתמש בספר זה הכולל דוגמאות ובעיות לפתרון.

בחלק מהבעיות, לפי שיקול המורה, מומלץ לחלק את הכיתה לקבוצות בהן יתקיים דיון על הבעיה ופתרונה.

באמצעות תוכנת סימולציה יוכל התלמיד לבדוק את תוצאות החישוב של בעיות ולהסיק מכך על דרך הפתרון.

למידה תוך התנסות

מומלץ לקיים חלק מהשיעורים במעבדה הכוללת מחשבים ומכשירים המתאימים להתנסות. על התלמיד להתנסות בתוכנת סימולציה בה יוכל לבחון ביתר קלות את ההשפעה של פרמטרים שונים במעגל על אופי תגובתו. ההתנסות במעגלים מעשיים תמחיש לתלמיד את ההבדל בין מעגל להלכה לבין מעגל למשעה.

מומלץ לקיים מעבדה עם רכיב מעשי 7805 או אחר כדי לחקור את פעולתו.

אמצעי הערכה חלופיים

ניתן לבקש מהתלמידים לבצע ניסוי עם רכיב 7805 והגיש דוח מפורט על מהלך הניסוי תוך בדיקות כל מאפייניו מול דפי המפרט שלו. מומלץ שכל זוג תלמידים יבצעו ניסוי זה.

פרק 8 – רשת מעבירה נמוכים וגבוהים

רציונל

בכל רשת חשמלית, גם זו שאינה כוללת באופן מפורש קבל או סליל, מתנהגת במקרים מסוימים כמו רשת הכוללת רכיבים אלה, למשל קווי תמסורת בתדר גבוה. בפרק זה נדון בהשפעת רכיב אוגר אנרגיה (קבל או סליל) על אותות ספרתיים בלבד. אופן ההתייחסות לקבל וסליל באות חילופים נלמד בתורת החשמל.

בפרק זה ירכוש התלמיד שני סוגי ידע; הצהרתי ותהליכי. התלמיד יזהה שני סוגי רשתות: מעבירה נמוכים ומעבירה גבוהים על פי סרטוט המעגל החשמלי. ידע לתאר את התנהגות המתח והזרם במעגל ואת משמעות המונח קבוע זמן.

התלמיד ידע לרשום את משוואות המתח והזרם של מעגלי המסננים תוך שימוש במשוואת הדפקים היסודית. התלמיד ידע לתאר באופן גרפי, תוך שימוש בפעולות, חישוב את התגובה לאות מדרגה ולאות דופק הן עבור מסנן מעביר נמוכים והן עבור מסנן מעביר גבוהים.

בפרק קיימת התייחסות למעגלים המבוססים על קבלים בלבד.

יעדים לימודיים

בסוף יחידת הלימוד התלמיד יהיה מסוגל לעשות את הפעולות הבאות:

1. לחשב קבוע זמן של רשת הכוללת נגד קבל.

2. לחשב קבוע זמן של רשת הכוללת נגד קבל.

רשת מעבירה נמוכים - LP:

1. לסרטוט רשת מעבירה נמוכים הכוללת נגד קבל.

2. לסרטוט רשת מעבירה נמוכים הכוללת נגד וסליל.

3. להגדיר את המושג "רשת מעבירה נמוכים"

4. תגובת רשת מעבירה נמוכים לאות מדרגה:

* הרשת יכולה להיות נגד קבל או סליל נגד.

א. לסרטוט את אות המוצא.

ב. להסביר את המושגים V_{0+} , V_{∞} המופיעים בנוסחת הדפקים.

ג. לרשום את הביטוי המתמטי למתח המוצא בתלות בזמן.

ד. להשתמש בנוסחת הדפקים היסודית לחישוב מתח מוצא בתלות בזמן

ה. להשתמש בנוסחת הדפקים לחישוב זמן בהנתן מתח מוצא.

ו. לחשב את מתח המוצא בזמן $t = \square$

ז. לסרטט באמצעות מחשב (גליון אלקטרוני / הדמיה) את מתח המוצא של הרשת.

5. תגובת רשת מעבירה נמוכים לדופק ריבועי:

* הרשת יכולה להיות נגד קבל או סליל נגד.

א. לסרטט אות מוצא עקרוני בהנתן:

$$(1) t \gg \tau$$

$$(2) t \ll \tau$$

$$(3) t \approx \tau$$

ב. להשתמש בנוסחת הדפקים היסודית לחישוב מתח מוצא בתלות בזמן

ג. להשתמש בנוסחת הדפקים לחישוב זמן בהנתן מתח מוצא.

ד. חישוב פרק הזמן בו המתח גבוה מערך נתון.

6. תגובת רשת מעבירה נמוכים לגל ריבועי (העשרה - לא בתכנית הלימודים)

א. לבצע הדמיה של המעגל החשמלי והצגת אות המוצא.

ב. לבצע ניסוי מעשי, והצגת אות המבוא ואות המוצא על גבי משקף תנודות.

ג. לחשב את מתחי הקיצון של אות המוצא במצב מתמיד.

רשת מעבירה גבוהים - HP:

1. לסרטט רשת מעבירה גבוהים הכוללת נגד קבל.

2. לסרטט רשת מעבירה גבוהים הכוללת נגד וסליל.

3. להגדיר את המושג "רשת מעבירה גבוהים"

4. תגובת רשת מעבירה גבוהים לאות מדרגה:

* הרשת יכולה להיות קבל נגד או נגד סליל.

א. לסרטט את אות המוצא.

ב. להסביר את המושגים V_{0+} , V_{∞} המופיעים בנוסחת הדפקים.

ג. לרשום את הביטוי המתמטי למתח המוצא בתלות בזמן.

ד. להשתמש בנוסחת הדפקים היסודית לחישוב מתח מוצא בתלות בזמן

ה. להשתמש בנוסחת הדפקים לחישוב זמן בהנתן מתח מוצא.

ו. לחשב את מתח המוצא בזמן $t = \square$

5. תגובת רשת מעבירה גבוהים לדופק ריבועי:

* הרשת יכולה להיות קבל נגד או נגד סליל.

א. לסרטט את מוצא עקרוני בהנתן:

$$(1) t \gg \tau$$

$$(2) t \ll \tau$$

$$(3) t \approx \tau$$

ב. להשתמש בנוסחת הדפקים היסודית לחישוב מתח מוצא בתלות בזמן

ג. להשתמש בנוסחת הדפקים לחישוב זמן בהנתן מתח מוצא.

6. תגובת רשת מעבירה נמוכים לגל ריבועי (העשרה - לא בתכנית הלימודים)

א. לבצע הדמיה של המעגל החשמלי והצגת את המוצא.

ב. לבצע ניסוי מעשי, והצגת את המבוא ואת המוצא על גבי משקף תנודות.

ג. לחשב את מתחי הקיצון של את המוצא במצב מתמיד.

תגובת רשת LP ורשת HP במישור התדר - רשות

* נושא זה אינו נלמד בפרק זה, אך ניתן לבצע קישור בין פרק המסננים לפרק של מעגל RC טורי (או RL טורי) בתת נושא: תורת החשמל

1. לסרטט את עקום הענות של רשת LP אידיאלית.

2. לסרטט את עקום הענות של רשת HP אידיאלית.

3. לסרטט את עקום הענות של רשת LP מסדר I (נגד קבל / סליל נגד).

4. לסרטט את עקום הענות של רשת HP מסדר I (קבל נגד / נגד סליל).

5. לחשב את תדר הקטעון בהינתן ערכי רכיבי המעגל.

6. לחשב את מתח המוצא של רשת LP (או HP) בתדר קטעון.

אסטרטגיות הוראה

למידה מבוססת פתרון בעיות

במידת האפשר כדאי לקיים את השיעור בחדר לימוד הכולל מחשבים על מנת להשתמש בספר זה הכולל דוגמאות ובעיות לפתרון.

בחלק מהבעיות, לפי שיקול המורה, מומלץ לחלק את הכיתה לקבוצות בהן יתקיים דיון על הבעיה ופתרונה.

באמצעות תוכנת סימולציה יוכל התלמיד לבדוק את תוצאות החישוב של בעיות ולהסיק מכך על דרך הפתרון.

למידה תוך התנסות

מומלץ לקיים חלק מהשיעורים במעבדה הכוללת מחשבים ומכשירים המתאימים להתנסות. על התלמיד להתנסות בתוכנת סימולציה בה יוכל לבחון ביתר קלות את ההשפעה של פרמטרים שונים במעגל על אופי תגובתו. מכיוון שקיים קושי מסוים ליצור במעבדה דופק בודד ולתעד את התגובה של המעגל, מומלץ לבצע במעגלים אלה סימולציות בלבד.

אמצעי הערכה חלופיים

ניתן לבקש מהתלמידים לבצע ניסוי משווה בסימולציה המתאר את ההשפעה של רוחב דופק על אות המוצא כאשר קבוע הזמן קבוע ו/או את ההשפעה של שינוי קבוע הזמן על אות המוצא עבור אות דופק קבוע בכניסה.

תפיסות שגויות וקשיים צפויים

אחד מהקשיים הנפוצים במעגלים מסוג HP הוא הערך המתקבל בנקודות המעבר של אות הכניסה. אחת הדרכים לפישוט הבעיה היא להסתכל על הרכיב במעגל המשמש כמסנן מסוג LP בו קיימת רציפות בערכי התגובה. לאחר החישובים והסרטוט של האותות ניתן להפיק את התגובה למסנן HP באמצעות פעולת חיסור פשוטה בין אות המקור לבין האות על הקבל.

קושי נוסף הקיים הוא חישוב קבוע הזמן במעגלים הכוללים יותר מנגד אחד. במקרה כזה כדאי להמליץ לתלמיד להשתמש במשפט תבנית ולסרטט את המעגל השקול.

יחידה 4: מבוא למערכות משובצות מחשב

היקף: 60 שעות עיוני; 30 שעות מעבדה

רציונל ודגשים בתכנית הלימודים

מבוא

המטרה העיקרית של תכנית הלימודים **מבוא להנדסת מחשבים** של המגמה **הנדסת אלקטרוניקה ומחשבים** היא הבנה של המבנה והפעולה של מערכות ממוחשבות, ובמיוחד ההדגשה של תהליך הפיתוח ותיעוד התוצר והתהליך. התכנית כוללת את המקצועות **יסודות התכנות בשפת C# לכיתה י'**, **מבוא למערכות משובצות מחשב לכיתה י'** והעמקה **בתכנות בשפת C# לכיתה י"א**.

תכנית הלימודים **תכנות בשפת C#**, הנלמדת בכיתה י' וכן בכיתה יא, מקנה לתלמיד היבטי חשיבה אלגוריתמית דרך עקרונות תכנות מתקדם ועד התנסות פעילה בכתיבת תוכנה. התכנית מעניקה לתלמידים בסיס ידע רחב בתחום התוכנה שתאפשר לו ליישם עבודת גמר במסגרת לימודי ההתמחות בכיתה י"ב בכל טכנולוגיה עדכנית אותה הוא יבחר. תכנית הלימודים **מבוא למערכות משובצות מחשב** נותנת מענה עדכני לטכנולוגיות חדשות תוך מתן דגש על יישום מעשי של טכנולוגיות המשלבות רכיבי חומרה משובצות מחשב. מטרת התוכנית היא לתת בסיס התנסותי רחב ככל שניתן כדי להוות בסיס לפיתוח עבודת גמר או פרויקט גמר במסגרת לימודי ההתמחות בכיתה י"ב. במדריך זה נעסוק בתכנית הלימודים "מבוא למערכות משובצות מחשב" בלבד. אולם מכיוון שבמקביל לתוכנית זו נחשף התלמיד לתכנות בשפת C# נשתמש במונחים הלוקחים מתחום תוכן זה. תכנית הלימודים נכתבה כך שבית הספר יוכל לבחור את סביבת פיתוח החומרה באמצעותה ירצה ללמד.

דוגמאות לסביבות פיתוח מתאימות:

- סביבת הפיתוח לבקרים של Arduino
- ערכת פיתוח לבקרים ממשפחת 8051
- ערכת פיתוח לבקרים ממשפחת PIC
- ערכת כרטיסי raspberry pi
- ערכת netduino

רצוי שסביבת הפיתוח בסביבת Embedded לכיתה י' תהיה ערכת הפיתוח Arduino UNO בגלל הפשטות הזמינות וחומרי ההדרכה וההוראה הזמינים לה. בתכנית נכלל מרכיב התנסותי משמעותי, לפיו נדרשים התלמידים לחבר רכיבי חומרה למיקרו בקרים דרך הדקים תקביליים וספרתיים ועד לחיבור רכיבי חומרה דרך פרוטוקולים

טוריים. מומלץ להקצות, במידת האפשר, את מעבדת המחשבים גם לשיעורים עיוניים. האפשרות לשלב הדגמה באמצעות המחשב וציוד הקרנה מתאים בשיעור עיוני תסייע רבות ללימוד הנושא.

רציונל

Embedded C++ הוא שם כולל לכתיבת תוכנה בשפה עלית למיקרו בקרים. תכנות ב- Embedded C++ מבוסס על תחביר C רגיל אך שונה ממנו בגלל המאפיינים הייחודיים שיש למיקרו בקר בהשוואה לתכנות שפת C למחשבי PC. מטרת התוכנית היא להכיר ללומד את הקשר שבין תכנות בשפה עלית לעבודה עם התקני קלט/פלט המחוברים להתקני חומרה ומגיבים להוראות תוכנה. אין בתוכנית כוונה ללמד לעומק את עקרונות התכנות ב- Embedded C++ אלא לתת ללומד חוויה טכנולוגית המאפשרת חיבור בין עולם התוכנה לעולם החומרה. על כן חשוב בשלב זה לא להיכנס לפרטים טכניים ייחודיים לעולם התוכנה ב- Embedded C++ , אלא להדגיש שני עקרונות מרכזיים:

1. מעבר מתכנות בסביבת C# תחת PC לכתיבת תוכנה Embedded C++ למיקרו בקר.

2. חיבור בין התקני חומרה לתוכנה בסביבת מיקרו בקרים. תכנית לימודים זו נכתבה סביב ערכות הפיתוח של Arduino. ניתן ללמד תכנית לימודים זו סביב כל ערכת פיתוח המשלבת מיקרו בקר ותוכנה.

ארגון חומר הלימוד

סדר הלימוד של הפרקים יהיה כדלקמן:

1. יסודות התכנות בסביבת מיקרו בקרים
בפרק זה ילמדו התלמידים את מבנה המחשב (מעבד, זיכרון והתקני קלט פלט) וייחשפו אל עולם התוכן של לוחות פיתוח מבוססי מיקרו בקרים של היצרנית Arduino וסביבת הפיתוח IDE מתוצרתה. הנחת היסוד של פרק זה היא שהתלמידים ילמדו במקביל את המקצוע תכנות בשפת C# ועל כן רק בסוף הפרק נחשפים התלמידים לסביבת הפתוח וילמדו את מבנה התוכנית והוראות פלט Serial.print.

2. כתיבה וקריאה של מפתחים ספרתיים
בפרק זה התלמידים ילמדו את האופן שבו מגדירים הדקי קלט ופלט כהדק בודד או כקבוצה של הדקים. התלמידים ילמדו שיטת חיבור של נורת LED אל לוח הפתוח תוך חישובים חשמליים הנדרשים להפעלה. כן יפעילו התלמידים יחידת תצוגה מסוג 7-SEGMENT. בפרק זה גם יכירו התלמידים את המונח משתנה מסוג byte וכן את המונח מערך (בחלק זה של תוכנית הלימודים נתייחס אליו כאל רשימה בעלת אינדקס). בחלקו האחרון של הפרק

ילמדו התלמידים כיצד לחבר מפסקים המשמשים כקלט וכיצד להתייחס למצבם. בחלק זה יכירו התלמידים משפט תנאי וישלבו אותו תוך תוכנית הכוללת קלט ופלט.

3. מבואות ומוצאים תקביליים

בפרק זה ילמדו התלמידים מהו ההבדל בין אות ספרתי לאות תקבילי ואת הצורך בהמרת אותות תקביליים לספרתיים ולהיפך, המרת אותות ספרתיים לתקביליים. כמו כן, ילמדו מהם המאפיינים של המרת אות תקבילי לספרתי: כושר הבחנה, זמן המרה וכיצד קוראים את הערך הגולמי לאחר המרה ומתאימים אותו לערך המתאים ליחידות הפיזיקליות הנדרשות.

לבסוף ילמדו על המרת אותות ספרתיים לאותות תקביליים בשיטת PWM ובכלל זה יצירת אותות ריבועיים באמצעות פעולות תכנות.

4. מיקרו בקרים

פרק זה פותח בסקירה עיונית של מבנה מיקרו בקרים ומציג את שתי ארכיטקטורות העיצוב הנפוצות CISC ו-RISC מהם הם מיוצרים. בהמשך הפרק מוצג המבנה הפנימי של המיקרו בקר ATmega328P ובהמשכו מבנה האריזה שלו ובמפרט החשמלי שלו. בחלקו האחרון של הפרק מוצג התרשים החשמלי של לוח הפיתוח UNO תוך ביאור חלקים נבחרים שלו.

5. פסיקות חומרה ומונים

בפרק זה ילמדו התלמידים את גישת התשאול (polling) ואת גישת הפסיקה (interrupt) בעת גישה למקורות המייצרים אירועים הן באופן סדיר והן באופן אקראי. התלמידים יכירו את החסרונות והיתרונות של כל אחת מהשיטות. התלמידים ילמדו להגדיר את אופן קבלת הפסיקה של מבואות פסיקה חיצוניים. התלמידים יכתבו שגרת פסיקה וביצוע קישור בתוכנית לשגרה.

כמו כן נציג שיטה בה תתבצע מנייה של אירועים חיצוניים תוך שימוש במבוא פסיקה חיצוני.

6. תקשורת טורית UART בין מיקרו-בקר למחשב

בפרק זה נסקור את תקן תקשורת טורית RS232: קווי התקשורת, מאפיינים חשמליים, מבנה מסגרת שידור ואץ קצב השידור. נכיר את פעולות המחלקה Serial המטפלת יחידת התקשורת המובנית במיקרו בקר. נחבר שני לוחות פיתוח זה לזה בתקשורת טורית ונעביר ביניהם מידע. כמו כן נחבר ערכת פיתוח עם מחשב PC. נכתוב תכניות הן עבור לוח הפיתוח ב- Embedded C++ וכן תוכנית בשפת C# עבור ה- PC.

פרקי תוכנית הלימודים

פרק 1 - יסודות התכנות בסביבת מיקרו בקרים

יעדים

בפרק זה התלמידים יכירו את סביבת הפיתוח IDE של Arduino תוך כדי התנסות פעילה בכתיבת תכניות העושות שימוש בתקשורת הטורית להצגת טקסט על המסך.

תכנים

1. הכרת המושג מיקרו בקר.
2. היכרות עם מספר לוחות פיתוח של Arduino.
3. המפרט הטכני של לוח Arduino UNO.
4. מאפיינים עיקריים של מיקרו בקר ATmega 168/328.
5. הכרת סביבת הפיתוח IDE של Arduino.
6. מבנה בסיסי של תוכנה הכולל setup ו-loop.
7. שימוש בתקשורת טורית Serial.print לשידור טקסט למסך המסוף (Terminal) בכתיבת הוראות פשוטות.

פרק 2 - כתיבה וקריאה מהמפתחים הספרתיים של הבקר

יעדים

בפרק זה התלמידים יכירו את סביבת הפיתוח IDE של Arduino תוך כדי התנסות פעילה בכתיבת תכניות העושות שימוש ברכיבי חומרה בסיסיים המתחברים דרך המפתחים הספרתיים למיקרו בקר.

תכנים

1. הכרת המושג מפתח קלט ופלט ספרתי.
2. הכרת אופן החיבור בין מפתח הבקר לבין נוריות, LED ותצוגות 7-SEG ומפסקים, תוך כדי הבחנה בין חיבור מבוסס pull up לחיבור pull down.
3. הכרת המאפיינים החשמליים של מפתחי המיקרו כמו מתחי עבודה וזרמים מקסימליים.
4. הגדרת מפתח הפלט בתוכנה תוך שימוש בפעולה pinMode.
5. שימוש בפעולות השהייה כמו delay, delayMicroseconds לתיאום בין מהירות העבודה של הבקר להתקני החומרה.
6. כתיבת תוכנית למיקרו בקר העושה שימוש בהתקני קלט פלט ספרתיים תוך כדי שימוש באוגרים PORTD, DDRD ו-PIND ובפעולות digitalWrite ו-digitalWrite.
7. שימוש בתקשורת טורית Serial.print ככלי למעקב אחר משתנים ומבואות ספרתיים.

פרק 3 - מבואות ומוצאים תקביליים

יעדים

בפרק זה התלמידים יממשו תוך כדי התנסות פעילה שילוב בין כתיבת תכניות בסביבת הפיתוח IDE של Arduino לבין רכיבי חומרה המתחברים דרך המפתחים התקביליים למיקרו בקר. יש לחבר את המפתח התקבלי למכשירי המעבדה (סקופ, רב מודד) ולהמחיש לתלמיד את הפעולה.

תכנים

1. הכרת ההבדל בין מפתח קלט תקבילי למפתח קלט ספרתי.
2. הכרות עם התכונות ומאפיינים של מפתחי הקלט התקביליים של מיקרו בקר ממשפחת Arduino.
3. הכרת אופן החיבור בין מפתח הבקר לבין פוטנציומטר המורכב כמחלק מתח, נגד רגיש לאור LDR ומד טמפרטורה LM35.
4. התלמיד ידע לחשב את הערך הספרתי של מתח אנלוגי.
5. הכרת המאפיינים האלקטרוניים של המבואות התקביליים: זמן המרה, רזולוציה, מספר סיביות וטווח מתחי ההמרה.
6. הגדרת מאפייני מפתח הקלט בתוכנה תוך שימוש בפעולה `analogReference`.
7. כתיבת תוכנית למיקרו בקר העושה שימוש בהתקני קלט תקביליים תוך כדי שימוש בפעולה `analogRead`.
8. שימוש בפעולות השהייה כמו `delay`, `delayMicroseconds` לביצוע השהיות בין דגימות.
9. שימוש בפעולות `analogWrite` ו- `tone` להפקת אותות מוצא מחזוריים בתדר וב- `Duty cycle` משתנה.
10. שימוש בתקשורת טורית `Serial.print` ככלי למעקב אחר משתנים ומבואות תקביליים.

פרק 4 - מיקרו בקרים

יעדים

בפרק זה התלמידים יכירו את המבנה הפנימי של מיקרו בקר באופן כללי ואת המבנה הפנימי של מיקרו בקר ATmega329P.

תכנים

1. הכרת מבנה פנימי של מיקרו בקר ויחידותיו.
2. ארכיטקטורת RISC (Reduced Instruction Set Computing).
3. מבנה הפנימי של מיקרו בקר ATmega328P.

4. מבנה אריזה של הבקר.
5. המפרט הטכני של לוח Arduino UNO.
6. הכרת התרשים החשמלי של כרטיס הפיתוח Arduino Uno.

פרק 5 - פסיקות חומרה ומונים

יעדים

בפרק זה התלמידים יכתבו תוכניות תוך שימוש בפסיקות חומרה חיצוניות ופסיקות מונים.

תכנים

1. הכרות עם שיטת הפסיקה
2. מה היתרונות והחסרונות בין שיטת הפסיקה לשיטת הסריקה.
3. פסיקות חיצוניות.
4. אופן דרבון הפסיקה החיצונית.
5. כתיבת שגרת פסיקה חיצונית.
6. הכרות עם מונים, שימוש במונה כקוצב זמן וכמונה אירועים.
7. כתיבת תוכניות לשימוש במונה כקוצב זמן.
8. כתיבת תוכניות לשימוש במונה כקוצב זמן תוך שימוש בפסיקת מונה.

פרק 6 - תקשורת טורית UART בין מיקרו בקר למחשב

יעדים

בפרק זה התלמידים יכירו תקשורת UART ובאמצעותה יעבירו נתונים בין מיקרו בקר מבוסס Arduino - לבין מחשב PC המריץ תוכנה בשפת C#.

תכנים

1. לימוד פרוטוקול UART מבנה הפרוטוקול וקצב תשדורת הנתונים.
2. כתיבת תוכנה לבקר המבוססת על ספרייה Serial ביצוע תקשורת עם מחשב PC לתוכנת מסוף Terminal חיצונית.
3. כתיבת תוכנה בשפת C# המבוססת על המחלקה SerialPort למימוש תקשורת בסיסית בין מחשב PC לבקר Arduino באמצעות תקשורת טורית.
4. כתיבת תוכנה לבקר המבוססת על הספרייה SoftwareSerial וביצוע תקשורת עם מחשב PC לתוכנת מסוף Terminal חיצונית.

פרק 1: יסודות התכנות בסביבת מיקרו בקרים

ביעדים שנקבעו בתוכנית הלימודים התלמיד נדרש יכיר את סביבת הפיתוח IDE של Arduino תוך כדי התנסות פעילה בכתיבת תכניות העושות שימוש בתקשורת הטורית להצגת טקסט על המסך.

את השיעורים בפרק זה מומלץ לבצע בחדר המחשבים המצויד במקרן באמצעותו יציג המורה את תכני הלימוד. נדרש לוודא שבכל עמדת מחשב תתוקן סביבת הפיתוח של Arduino (נספח א - התקנת סביבת הפתוח).

מומלץ לעודד (לא לחייב) את התלמידים לרכוש ערכות פיתוח אישיות לתרגול בבית (ניתן לרכוש באינטרנט ערכות פיתוח הכוללות לוח תואם ורכיבים שונים בעלות של כמה עשרות שקלים). ערכה אחת יכולה לשרת זוג תלמידים. במידה ונרכש לוח תואם ולא מקורי, ניתן למצוא [כאן](#) דרייבר USB התואם ללוח.

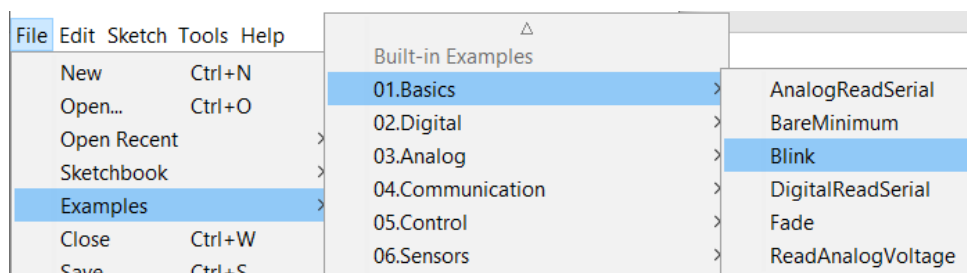
ביחידה הראשונה של הפרק מומלץ למורה לקיים דיון בכיתה לגבי מושג המחשב ולהביא דוגמאות מעולם התלמיד המשתמש מכשירים ממוחשבים. מתוך הדיון להגיע אל מבנה המחשב ולהציג את מרכיביו העיקריים. להציג את המטרות של מערכת משובצת מחשב ומנגד להציג את המטרות של המחשב האישי PC.

בשלב הבא מומלץ לחשוף את התלמידים למחשבי לוח כגון [raspberrypi](#), [lattepanada](#), [netduino](#) וכמובן [Arduino](#). בשלב זה מומלץ לתת לתלמידים להשוות בין הלוחות על פי מאפיינים בסיסיים כמו מחיר, כמות זיכרון וסוג מעבד.

ביחידה השנייה של הפרק יציג המורה את לוח הפיתוח UNO וימנה את תכונותיו של המיקרו בקר המותקן בלוח. בסוף היחידה התלמיד צריך לזהות את מרכיבי הלוח ויכול להשוות את הלוח מול לוחות אחרים של Arduino או של יצרנים אחרים.

ביחידה השלישית יציג המורה את סביבת הפתוח וידגים כיצד לחבר את הלוח למחשב. על המורה להדגיש שעל התלמיד לוודא שאכן נוצר קשר בין הלוח למחשב (לעיתים נדרש זמן מרגע חיבור הלוח ועד שהדרייבר USB של הלוח נטען ומאפשר עבודה מול המחשב).

מומלץ למורה להדגים בדיקה של הלוח באמצעות אחת מתוכניות הדוגמה המסופקות יחד עם סביבת הפתוח. בחלק זה אין צורך להסביר את התוכנית. איור 3.1 מציג בחירה של תוכנית Blink.



טעינת תוכנית דוגמה Blink

ביחידה הרביעית יציג המורה את שפת הפיתוח Embedded C++ ויעמוד על ההבדלים בינה לבין שפת התכנות C# הנלמדת במקביל. מומלץ להשתמש במונחים של שפת C# במקום במונחי שפת C/C++. למשל, להשתמש במונח פעולה המוכר משפת C# במקום במונח פונקציה המקובל בשפת C/C++.

בהמשך היחידה יציג וידגים המורה תוכנית בסיסת המציגה טקסט קצר בחלון המסוף. בחלק זה של ההדגמה יסביר המורה את מרכיבי התוכנית. על המורה להבהיר ולהדגים את השימוש בפעולות setup ו-loop. ולהמחיש באמצעות דוגמה.

הצגת הודעת טקסט פעם אחת בלבד:

```
1 void setup() {
2     Serial.begin(9600);
3     Serial.print("Hello, world");
4 }
5
6 void loop() {
7 }
```

הצגת הודעת טקסט באופן מחזורי (אינסופי):

```
1 void setup() {
2     Serial.begin(9600);
3 }
4
5 void loop() {
6     Serial.print("Hello, world");
7 }
```

לאחר ההדגמה יתרגלו התלמידים מטלות שונות להצגת תבניות טקסט שונות פעם אחת בלבד או באופן מחזורי. (במורה ירחיב להוראת println והוראת delay). מכיוון שבמקביל נלמד המקצוע תכנות בשפת C# יודא המורה כי המושג משתנה שלם (int) כבר מוכר לתלמיד ויוסיף לתוכנית פעולות עם משתנים. בשלב ראשון פעולות חישוב פשוטות (+, -, *, /, %) והצגה של ערכים.

בשלב הבא ילמד המורה (בתיאום עם המורה לשפת C#) את משפט התנאי וישבץ את התנאי בתוך פעולת loop, כפי שמודגם בספר הלימוד.

פתרון לשאלה 1.9

כתבו תוכנית המציגה שעון שניות המתחיל ב-10 ומסתיים ב-99. כל הצגה תהיה בשורה נפרדת ובין הצגה להצגה תהיה השהיה של שנייה אחת.

```

1 int cnt;
2 void setup() {
3     Serial.begin(9600);
4     cnt = 10;
5 }
6
7 void loop() {
8     if (cnt < 100) {
9         Serial.println(cnt);
10        delay(1000);
11        cnt++;
12    }
13 }
```

ביאור התוכנית

שורה 1: הגדרת משתנה שלם בשם cnt

שורות 2-5: פעולת setup

שורה 3: אתחול תקשורת בקצב 9600 סל"ש

שורה 4: אתחול משתנה שלם ל-10

שורות 7-12: פעולת loop

שורה 8: משפט תנאי הבדק אם ערכו של המשתנה נמוך מ-100

שורה 9: הצגת ערך המשתנה במסוף ה-monitor. לאחר ההצגה קיים מעבר שורה.

שורה 10: קידום ערכו של cnt ב-1.

פרק 2: כתיבה וקריאה מהמפתחים הספרתיים של הבקר

ביעדים שנקבעו בתוכנית הלימודים התלמיד נדרש להכיר את סביבת הפיתוח IDE של Arduino תוך כדי התנסות פעילה בכתיבת תכניות העושות שימוש ברכיבי חומרה בסיסיים המתחברים דרך המפתחים הספרתיים למיקרו בקר.

בפרק זה נתמקד בהדקי לוח הפיתוח UNO. הדקי לוח הפיתוח הם חיבורים ישירים אל המיקרו בקר ATmega329P ועל כן תכונות לוח הפיתוח נגזרות מתכונות המיקרו בקר. את דפי המפרט של המיקרו בקר ניתן למצוא [כאן](#).

מפתחי לוח הפיתוח פישטו את השימוש בבקר מתוך כוונה להקל על המשתמש וכן מתוך הצורך המתבקש של מעבר ללוחות פיתוח מדגמים אחרים. במקום לבצע פנייה ישירה אל הדקי המיקרו בקר מתבצעת פנייה אל הדקים באמצעות מספור שניתן להדקים ספרתיים (0-13) ולדקים תקביליים (A0-A5). טבלה 4.1 מתארת את החלוקה של מספור ההדקים ביחס להדקי המיקרו בקר. איור 4.1 מציג את הדקי המיקרו בקר. כפי שניתן לראות באיור 4.1, מוגדרים במיקרו בקר 3 מפתחים המסומנים ב- PCx, PBx ו- PDx. (x – מציין מספור מ-0 עד 7).

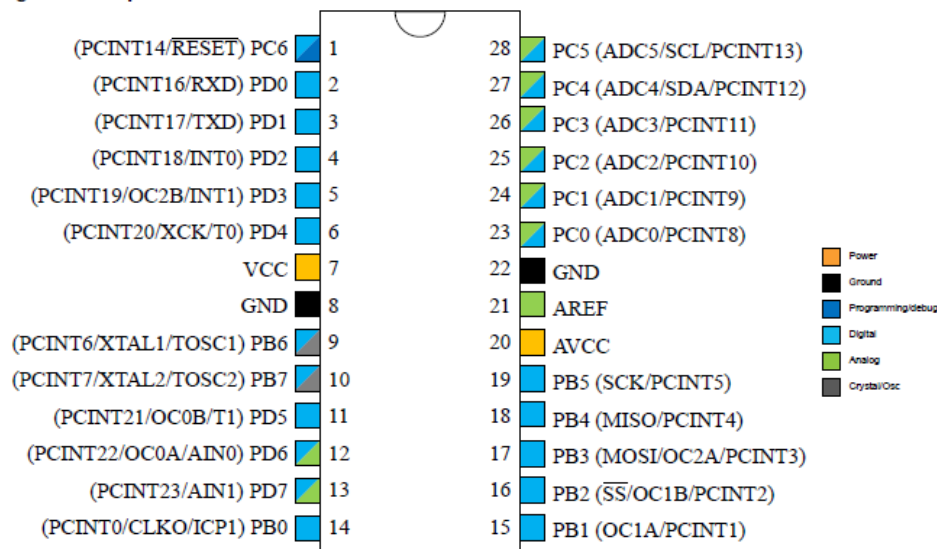
מפתח/סיבית	0	1	2	3	4	5	6	7
PB	8	9	10	11	12	13	C	C
PC	A0	A1	A2	A3	A4	A5	R	
PD	0	1	2	3	4	5	6	7

C – הדקי גביש Crystal, R – הדק איפוס Reset

טבלה 4.1 מיפוי הדקי לוח הפיתוח להדקי המיקרו בקר.

Pin-out

Figure 5-1. 28-pin PDIP



איור 4.1 הדקי המיקרו בקר

באתר שניתן למצוא [כאן](#) מופיעים הסברים ואיורים נוספים המפרטים את חיבורי המיקרו בקר ללוח הפיתוח.

הדקי התקשורת

הדקים 0 ו-1 משמשים לתקשורת טורית. התקשורת הטורית משמשת לשתי מטרות שונות:

- העלאת (UPLOAD) תוכנית למיקרו בקר בשלב הפיתוח.
- משמשת את המחלקה Serial, המציגה מידע במסוף serial monitor בזמן הריצה.

מומלץ להימנע משימוש בהדקים אלה.

במידה ומשתמשים בהדקים אלה (הדק 0 בפרט) ישתבש תהליך העלאת התוכנית למיקרו-בקר ובזמן ריצה השימוש בפעולות של מחלקה Serial עלול לגרום לתגובות לא תקינות. ניתן לנתק את החיבור להדק 0 באופן זמני בזמן העלאת התוכנית.

הדקים תקביליים

יש להבהיר לתלמיד שעל אף שהדקים A0-A5 מסומנים כהדקים תקביליים אפשר להשתמש בהם גם כהדקים ספרתיים.

את המאפיינים החשמליים של המיקרו בקר ניתן למצוא בדפי המפרט שנמצאים [כאן](#). סעיף 28 בדפי המפרט מכיל מידע לגבי המאפיינים החשמליים. להלן קטע מדפי המפרט המתאר את הערכים המרביים:

Parameters	Min.	Typ.	Max.	Unit
Operating temperature	-55		+125	°C
Storage temperature	-65		+150	°C
Voltage on any pin except RESET with respect to ground	-0.5		$V_{CC} + 0.5$	V
Voltage on RESET with respect to ground	-0.5		+13.0	V
Maximum operating voltage		6.0		V
DC current per I/O pin		40.0		mA
DC current V_{CC} and GND pins		200.0		mA
Injection current at $V_{CC} = 0V$		$\pm 5.0^{(1)}$		mA
Injection current at $V_{CC} = 5V$		± 1.0		mA

בפרק זה על התלמיד להכיר מושגי יסוד בתורת החשמל: מתח, זרם, התנגדות וכן את חוק אוהם וחוקי המתח והזרם של קירכהוף.

בפרק זה יכיר התלמיד לראשונה רכיב אלקטרוני מסוג דיודת LED. לרוב בשלב זה התלמיד עדיין לא מכיר רכיבים אלקטרוניים ולכן נציג את דיודת ה-LED (נורת LED) כרכיב בעל תכונות ומפרט חשמלי בלבד (ללא אזכור דיודה רגילה).

יש להבהיר לתלמיד שנורת ה-LED אינה מתנהגת כנגד אלא כרכיב הפועל עם עוצמת מתח קבועה ללא תלות בעוצמת הזרם דרכו.

התלמיד נדרש לחשב את התנגדות הנגד הטורי לפי מאפייני מתח נורת ה-LED, מתח המקור וזרם העבודה של הנורה הן במעגל חשמלי עצמאי והן בחיבור ללוח הפיתוח לפי שיטות החיבור Pull Up ו-Pull Down.

תוכנות עזר

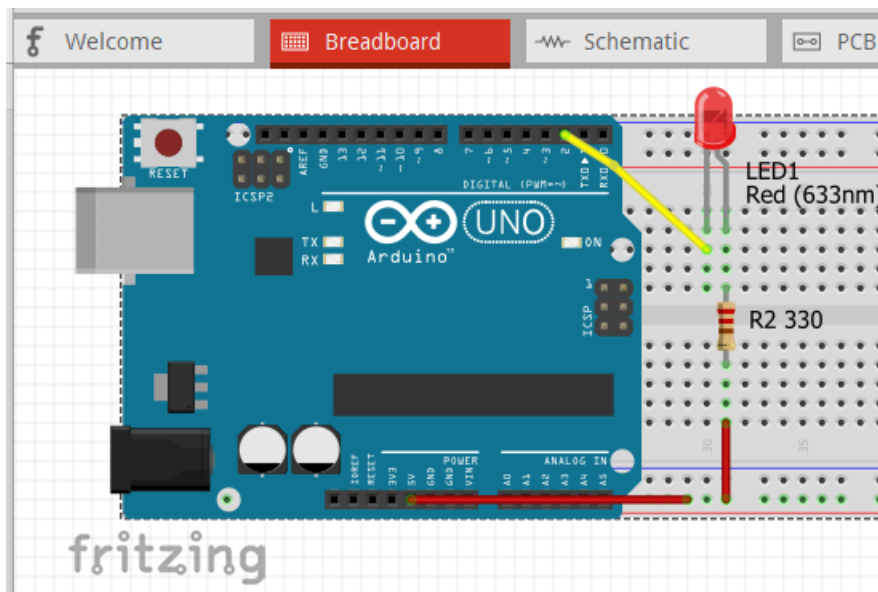
בחלק זה נציג שתי תוכנות חינוכיות המאפשרות עריכה ויזואלית ותרשים חשמלי מתאים למעגלים חשמליים ובתוכם לוח הפיתוח UNO. את התוכנה הראשונה ניתן להתקין במחשב ואילו התוכנה השנייה מופעלת בדפדפן בלבד.

1. Fritzing

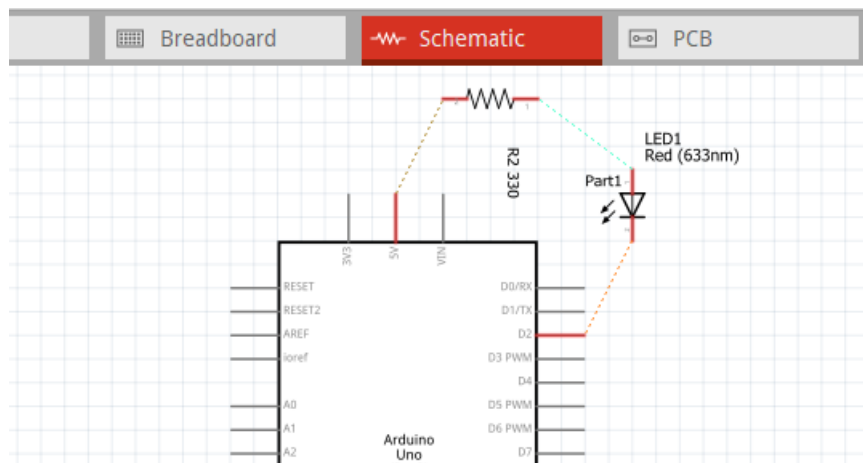
את התוכנה ניתן להוריד באתר שמופיע [כאן](#) לפי גרסת מערכת ההפעלה המותקנת במחשב.

לאחר ההורדה נחלץ את הקובץ המתקבל בפורמט zip לתיקייה על פי בחירתכם. הפעלת התוכנה באמצעות לחיצה על הקובץ fritzing.exe. הנחיות שימוש בתוכנה ניתן למצוא בקישור שמופיע [כאן](#).

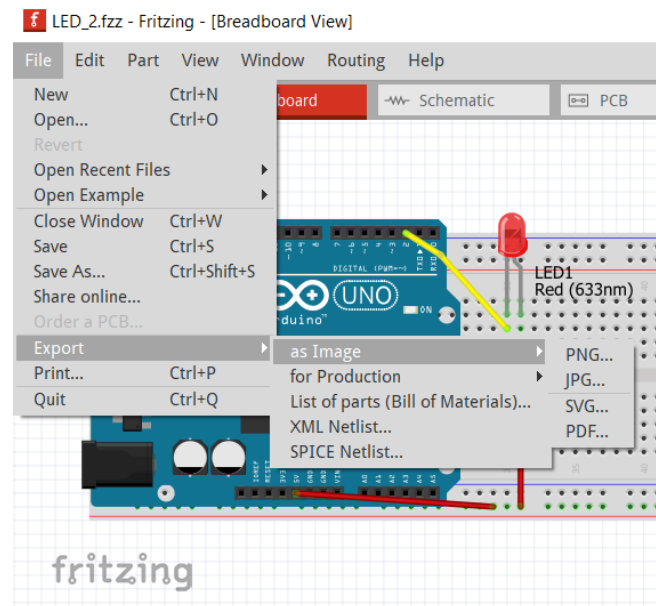
דוגמה למעגל הכולל לוח פיתוח, נורה ונגד



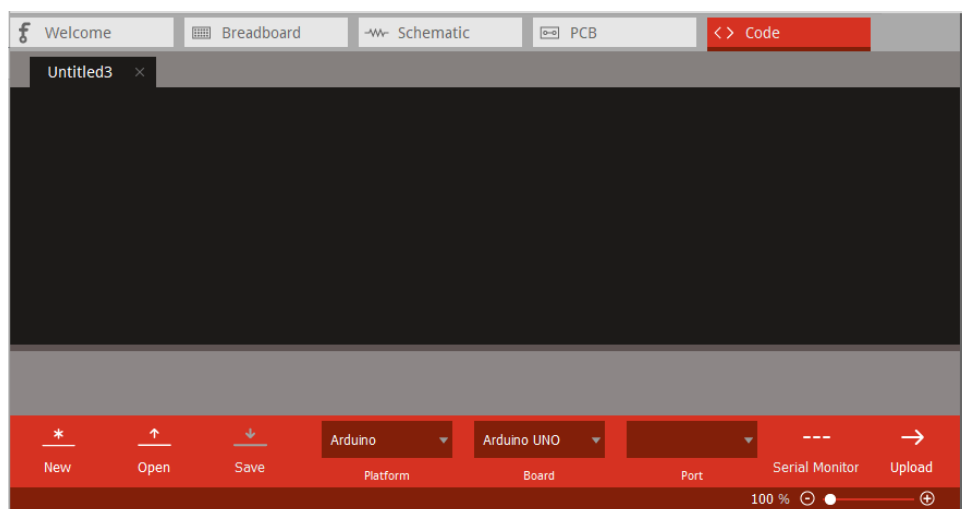
שימו לב, התוכנה נמצאת במצב עריכה של מטריצה (Breadboard).
ניתן לעבור למצב תרשים חשמלי:



שימו לב, התוכנה נמצאת במצב עריכה של תרשים (Schematic).
חלק מהאיורים המופיעים בספר הלימוד נלקחו מתכנה זו המאפשרת ייצוא של תמונות.



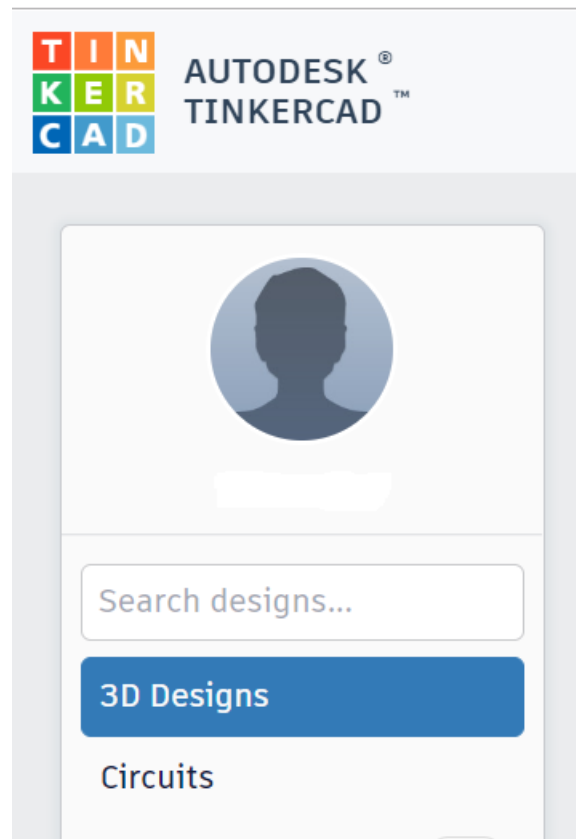
אפשרות נוספת (לא רלוונטית) היא יצירת לוח מודפס (PCB). ניתן להשתמש בתכנה על מנת לכתוב תוכנית (קוד) ללוח הפיתוח ולהעלות אותה למיקרובקר.



- שימו לב, התוכנה נמצאת במצב עריכה של קוד (Code).
- על אף שאפשרות זו קיימת מומלץ לא להשתמש בה מהטעמים הבאים:
- לא להעמיס על התלמיד למידה של סביבה נוספת לשם פיתוח תכניות.
 - להישאר עם סביבת הפתוח של Arduino המקבלת עדכונים באופן שוטף.

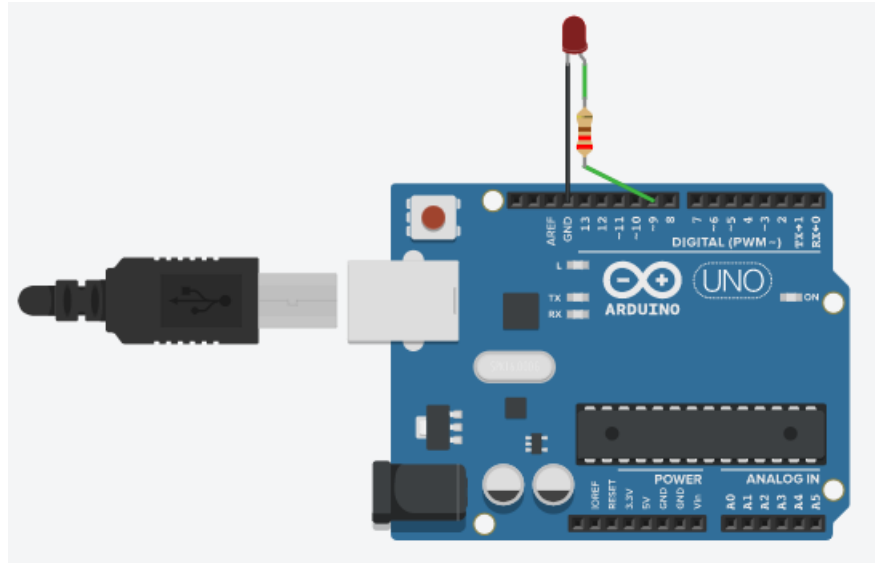
Tinkercad .2

התוכנה מבית היוצר של AUTODESK מופעלת בדפדפן ובכך מאפשרת שימוש ללא קשר לסוג המחשב ולמערכת ההפעלה. התוכנה דורשת רישום כתנאי לשימוש. התוכנה מאפשרת עיצוב של פריטים בתלת מימד לשם הדפסה במדפסת תואמת וכן בנייה של מעגלים חשמליים. ניתן למצוא את התכנה באתר שמופיע [כאן](#). לאחר הרישום ניתן להיכנס לאתר ולהשתמש בתכנה.

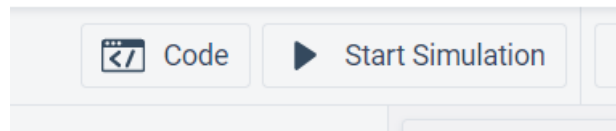


נבחר באפשרות Circuits.

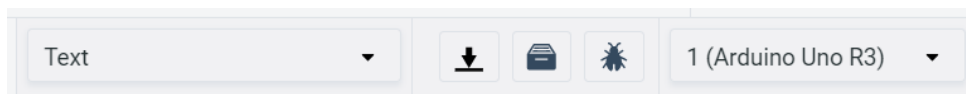
לאחר הבחירה יופיעו מעגלים לדוגמה וכן כל המעגלים הקודמים שנוצרו על ידכם. מעגל מאחת מהדוגמאות:



בחלקו הימני של המסך מופיע התפריט הבא:



לחיצה על האפשרות Code תציג את התוכנית בצורה של טקסט. ניתן לבצע פעולות על התוכנית באמצעות התפריט הבא:



בתפריט הנגלל השמאלי ניתן לשנות את צורת התוכנית לתבנית בלוקים לפי מבניות הדומות לתוכנת scratch של MIT.

כמו כן, ניתן לבחור באפשרות של שילוב בין טקסט לבלוקים.

ההעדפה היא כמובן להישאר בכתיבה והצגה של תוכנית בצורת טקסט.

כמו כן, ניתן להוריד את התוכנית למחשב, לשלב ספריות שונות בתוכנית, לבצע הרצה בסימולציה מבוקרת לניפוי שגיאות (debug) ולבחור את הלוח המתאים (במידה ויש יותר מאחד).

היתרון בתכנה זו היא היכולת לבצע סימולציות למערכות אלקטרוניות בשילוב לוח הפיתוח ומכשירי מדידה זמינים בסימולטור.

באמצעות תכנה זו ניתנת אפשרות לתלמיד לתרגל בבית את הנושאים הנלמדים ללא צורך בהתקנת סביבת הפיתוח, ללא לוח פיתוח וכן ללא רכיבים מעשיים. בנוסף ניתן להשתמש במכשירי המדידה כדי למדוד אותות חשמליים מסוגים שונים.

קבועים

קבוע (constant) הוא ערך מספרי או אחר שאינו בר שינוי.

ניתן להגדיר שני סוגי קבועים:

הגדרה המתבצעת על ידי הקדם מהדר באמצעות מילת המפתח `define`. כל הוראות הקדם מהדר נתחום בסימן `#`. בהגדרה זו מתאימים לכל ערך סמל טקסטואלי. הקדם מהדר מחליף את הסמל בערך בכל מקום שהוא מופיע. הגדרה זו הינה חסרת טיפוס (type) ולכן ייתכנו שגיאות בעת ההגדרה.

דוגמה:

```
#define ZERO 0
```

הסמל ZERO מייצג את הערך 0.

הגדרה המתבצעת באמצעות טיפוס נתונים בעל מאפיינים מובהקים. ההגדרה דומה להגדרת משתנה פרט לעובדה שמלת המפתח `const` נרשמת לפני תחילת ההצהרה. הקבוע תופס מקום בזיכרון ובכל פעם שנעשה בו שימוש מתבצעת פנייה לזיכרון שם מאוחסן הקבוע.

דוגמה:

```
const int zero = 0;
```

הקבוע zero הוא מטיפוס שלם וערכו אפס.

שימוש שגוי

בגלל הדמיון להגדרת משתנה, תלמידים עלולים להתייחס אליו כאל משתנה ולבצע הוראה לא חוקית.
למשל; `zero = 9;` הינה לא חוקית מכיוון שקבוע אינו בר שינוי!

קבועים מספריים

כידוע בשפת C כמו בשפת C# ניתן לכתוב מספרים עשרוניים או מספרים לפי בסיס 16. מספרים עשרוניים נרשמים בצורה רגילה ואילו מספרים בבסיס 16 נרשמים בתוספת של `0x` בתחילתם.

כך למשל הערך 10 הוא עשרוני ואילו הערך `0x10` הוא מספר בבסיס 16. סביבת הפתוח מאפשרת לרשום גם מספרים בבסיס 2 (בינארי) המורכבים מהספרות 0 או 1 בלבד. נוח להשתמש בייצוג בינארי כאשר פונים לקבוצת סיביות (לרוב בגודל בית – 8 סיביות).

הצירוף הבא `01010101` המייצג רמות לוגיות ניתן לייצוג באופן הבא:

`B01010101`

מעריך חד ממדי

בפרק זה הוקדם השימוש במעריך חד ממדי על מנת לפשט את השימוש בריבוי נתונים בעלי סדר. אל המעריך נתייחס כאל טבלה שבה שתי עמודות. עמודה אחת היא מספר סידורי והעמודה השנייה ערך, בדומה לרשימה ממוספרת. המספור מתחיל מאפס עד לערכו של התא האחרון במעריך, והפנייה לעמודת הערך מתבצעת באמצעות המספר הסידורי. בחלק של תוכנית הלימודים נתייחס למעריך כבעל ערכים קבועים בלבד. השימוש במעריך בצורתו המלאה נשאיר לשלב מאוחר יותר שבו ילמד המעריך במסגרת תכנות בשפת C#.

פתרון לשאלה 2.17

כתבו תוכנית המפיקה גל ריבועי בתדר $f=2\text{KHz}$ בהדק 10 במיקרו-בקר. היעזרו באלגוריתם שלהלן:

א. הפיקו 1 בהדק 10 למשך מחצית זמן המחזור.

ב. המתינו מחצית זמן מחזור.

ג. הפיקו 0 בהדק 10 למשך מחצית זמן מחזור.

ד. המתינו מחצית זמן מחזור.

כדי להמתין השתמשו בהוראה `delayMicroseconds`.

בדקו באמצעות משקף תנודות את האות המתקבל בהדק 10 במיקרו-בקר.

$$T = \frac{1}{f} = \frac{1}{2000} = 500\mu s$$

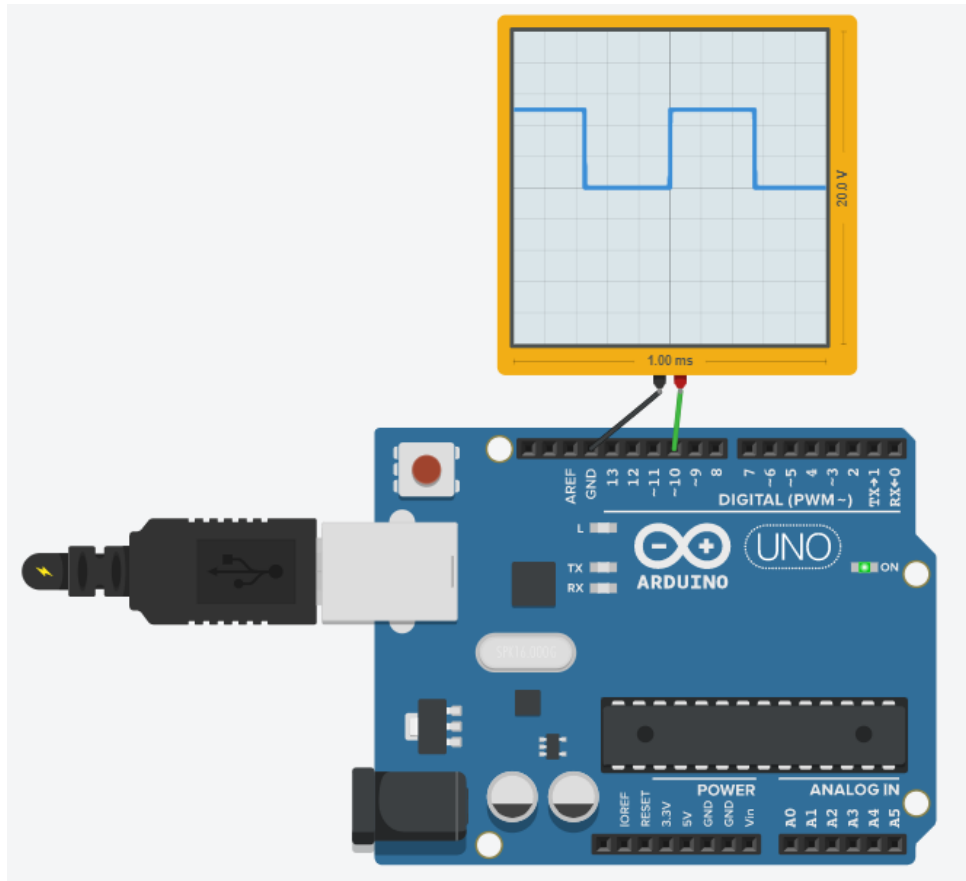
נחשב תחילה את זמן המחזור: $500\mu s$

מחצית זמן המחזור היא: $250\mu s$

ההוראה `delayMicroseconds` מקבלת פרמטר ביחידות של מיקרו-שניות ולכן נזמן השהייה

`delayMicroseconds(250)`.

הבנייה וההרצה של המעגל בוצעה באמצעות התכנה Tinkercad.



הקוד של התוכנית מוצג להלן:

```

1  const int pin = 10;
2  void setup()
3  {
4      pinMode(pin, OUTPUT);
5  }
6
7  void loop()
8  {
9      digitalWrite(pin, HIGH);
10     delayMicroseconds(250);
11     digitalWrite(pin, LOW);
12     delayMicroseconds(250);
13 }

```

פרק 3: מבואות ומוצאים תקביליים

ביעדים שנקבעו בתוכנית הלימודים התלמיד נדרש לממש תוך כדי התנסות פעילה שילוב בין כתיבת תכניות בסביבת הפיתוח IDE של Arduino לבין רכיבי חומרה המתחברים דרך המפתחים התקביליים למיקרו בקר. יש לחבר את המפתח התקבילי למכשירי המעבדה (סקופ, רב מודד) ולהמחיש לתלמיד את הפעולה.

המרה אותות

הגדלים מהירות, תאוצה, טמפרטורה, לחות וכדומה הם ערכים תקביליים, כלומר הגדלים הם בעלי ערכים רציפים בתחום מוגדר. כדי שנוכל לעבד אותות אלו במערכת ממוחשבת יש להמירם לאותות ספרתיים וכדי שנוכל להשתמש בהם במערכות בקרה תקביליות, למשל, יש להמירם חזרה לאותות תקביליים.

באיור 5.1 מופיע תרשים מלבנים של מערכת עקרונית הקולטת אות תקבילי ממירה אותו לאות ספרתי, מעבדת את הנתון הספרתי וממירה אותו בחזרה לאות תקבילי.



איור 5.1 תרשים מלבנים של מערכת הממירה אותות תקביליים וספרתיים

במערכת מתרחשים שני תהליכי המרה:

המרת אות תקבילי לספרתי A/D – Analog to Digital, והמרה הפוכה של אות ספרתי לתקבילי D/A – Digital to Analog.

המרת אות תקבילי לספרתי

האות התקבילי הוא אות רציף במישור הזמן והממיר A/D ממיר אותו לערכים ספרתיים בפרקי זמן קצובים. לשם כך יש להגדיר את קצב דגימת האות. הקצב בו נדגם האות נקרא קצב הדגימה Sample rate ומסומן ב- f_s . קצב הדגימה נמדד ביחידות של $sample/s$. זמן הדגימה T_s הוא הזמן בין שתי דגימות עוקבות. במשך זמן זה המערכת משלימה את תהליך ההמרה ומאחסנת את הערך הספרתי המומר. קצב הדגימה נקבע על פי ספקטרום התדרים של האות במבוא לפי משפט הדגימה.

משפט הדגימה

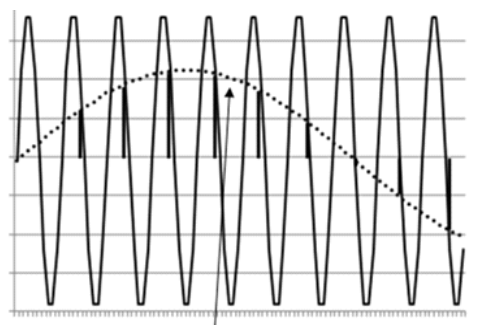
משפט הדגימה קובע כי קצב הדגימה חייב להיות גדול לפחות פי שניים מהתדר הגבוה f_H של אות התקבילי במבוא, כלומר: $f_s \geq 2f_H$.

בחירת קצב דגימה על פי משפט הדגימה מבטיח שחזור מדויק של האות התקבילי מהדגימות.

באתר [Learning By Simulation](#) ניתן למצוא תכנית סימולציה הניתנת להורדה למחשב והמציגה את השפעת קצב הדגימה על האות המשוחזר.

תופעת התחזות

דגימה בקצב הנמוך מקצב הדגימה הדרוש גורם לתופעה נקראת התחזות או קיפול תדרים. בדגימה בקצב נמוך מופיעים אותות בתדרים נמוכים הנובעים מדגימת התדרים גבוהים. באיור 5.2 מתואר אות מבוא תקבילי הנדגם בתדר הנמוך מקצב הדגימה. האות המשוחזר, המופיע כרצף נקודות באיור, הוא בתדר נמוך מזה של האות במבוא.



תופעת התחזות

איור 5.2 הדגמת תופעת ההתחזות

תופעת ההתחזות עלולה להתרחש כאשר האות במבוא כולל רכיב בתדר הגבוה f_H . במקרים אלו כדי לבטל את תופעת ההתחזות ניתן להוסיף למעגל מסנן מגביל פס BPF המסלק את התדרים הלא רצויים.

דוגמה

מניתוח ספקטרלי של אות תקבילי המתקבל מחיישן עוצמת הארה נמצא כי תחום התדרים המשמעותי הוא 0-4200 Hz.

- מהו סוג המסנן שנדרש לחבר בין החיישן לממיר? מהם תדרי הקיטעון של המסנן?
- מהו קצב הדגימה המזערי הדרוש?
- בהנחה שדוגמים את האות בקצב הגבוה פי שניים מקצב הדגימה המזערי. מהו זמן הדגימה?

ד. מהי מגבלת הזמן של הממיר?

פתרון

- המסנן הוא מסוג LPF מסנן מעביר נמוכים. תדר הקיטעון של המסנן הוא 4.2kHz.
- קצב הדגימה המינימלי הוא: $f_{s_min} = 2f_H = 2 \cdot 4.2k = 8.4k \text{ sample/s}$.

ג. קצב הדגימה הוא: $f_s = 2f_{s_min} = 2 \cdot 8.4k = 16.8k \text{ sample/s}$ ולכן זמן הדגימה הוא:

$$T_s = \frac{1}{f_s} = 59.5\mu s$$

ד. תהליך ההמרה וקריאת הנתון חייבים להסתיים במשך זמן הדגימה T_s .

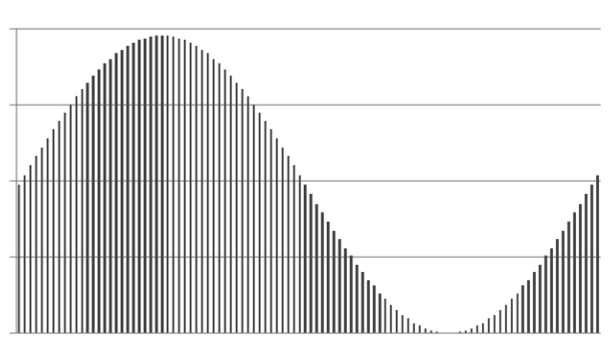
מאפייני ממירים

- שיטות המרה שימושיות: Successive Approximation ADC, flash ADC, ramp ADC
- ממשק: מותאם למעבד או ממשק טורי.
- מספר סיביות: 8, 10, 12, או 16 סיביות נמצא בשימוש בסביבות משולבות מיקרו בקרים.
- זמן המרה: הזמן הדרוש להמרה מרגע הדרבון (נקבע על פי קצב הדגימה הדרוש).
- תחום מתחי מבוא.

המיקרו בקר ATmega329P כולל ממיר ADC מובנה הפועל בשיטת Successive Approximation ובורר תקבילי המאפשר דגימה של מספר מבואות תקביליים.

המרת אות ספרתי לתקבילי

הממיר מספרתי לתקבילי (DAC) מייצר אותות תקביליים של מתח או זרם על פי הצירוף בינארי שבמבואו. ערכים בינאריים, במבוא הממיר, המשתנים לפי הזמן יוצרים במוצאו מתח המשתנה בזמן. באיור 5.3 מתואר אות הסינוס הנוצר מ-100 דגימות שנשלחו ברצף בזו אחר זו אל הממיר. האות המתקבל אינו רציף בזמן. כדי שהאות יהיה רציף נדרש לחבר מסנן מסוג LPF למוצא.



איור 5.3 אות הסינוס הנוצר במוצא ממיר מ אותות בינאריים במבוא

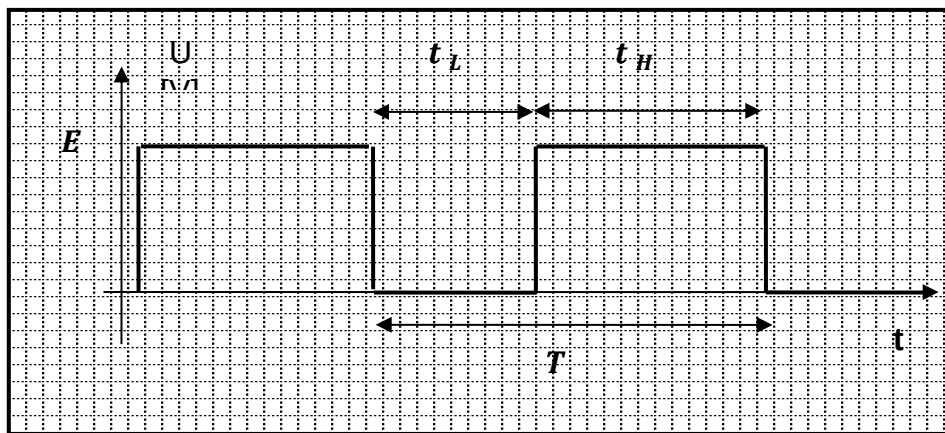
מיקרו בקר ATmega329P לא כולל בתוכו ממיר מסוג DAC. לחילופין מכיל המיקרו בקר יחידה המייצרת אותות PWM ב-6 ערוצים.

שיטת אפנון רוחב הדופק PWM

כאמור, מכיוון שבפועל אין מוצא תקבילי, משתמשים ביציאה **ספרתית** הפועלת על פי העיקרון של אפנון רוחב הדופק (PWM (Pulse Width Modulation. בשיטה זו מייצרים גל ריבועי בתדר f קבוע וגורם מחזור משתנה DC .

מאפייני גל ריבועי

- עוצמת הגל – מתח $0V$ או E
- תדירות הגל $f [Hz]$
- זמן המחזור $T [sec]$
- גורם המחזור DC (Duty Cycle) (מקובל לציין באחוזים).



איור 5.4 גל ריבועי

גורם המחזור

הוא היחס בין פרק הזמן שבו עוצמת הגל הריבועי E לבין זמן המחזור T :

$$DC = \frac{t_H}{T}$$

$$T = t_H + t_L \quad \text{ניתן לראות כי}$$

מתח ממוצע

$$V_{AV} = \frac{1}{T} \int_{t=0}^T U(t) dt = \frac{1}{T} \int_{t=0}^{t_L} 0 dt + \frac{1}{T} \int_{t=t_L}^T E dt = 0 + \frac{1}{T} E \cdot t|_{t=t_L}^T = \frac{1}{T} E (T - t_L) =$$

$$\frac{1}{T} E \cdot t_H = \frac{t_H}{T} E = DC \cdot E$$

על פי הנוסחה המתח הממוצע תלוי בגורם המחזור DC בלבד!

במיקרו בקר $E=5V$ ולכן

$$V_{AV} = 5V \cdot DC$$

ערכו של גורם המחזור הוא בתחום: $0\% \leq DC \leq 100\%$. מכאן שהמתח הממוצע יהיה

$$0 \leq V_{AV} \leq 5V$$

כדי לייצר אות PWM בסביבת הפתוח נשתמש במספר המייצג את גורם המחזור. תחום המספר הוא מ-0 עד 255. ההתאמה בין גורם המחזור, המתחים והערך המספרי להלן:

$$0\% \leq DC \leq 100\%$$

$$0 \leq V_{AV} \leq 5V$$

$$0 \leq Value \leq 255$$

ממיר ADC במיקרו בקר ATmega329P

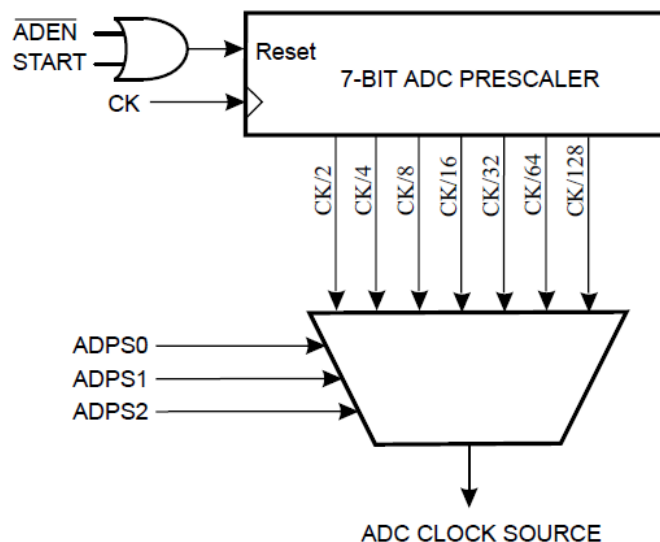
מאפיינים נבחרים

- רזולוציה של 10 סיביות
- דיוק של $\pm 2 \text{ LSB}$
- זמן המרה $13\text{-}260\mu\text{s}$
- 6 ערוצים מבוא תקביליים מרובבים
- ערוץ חיישן טמפרטורה מובנה
- תחום מתחי מבוא $V_{CC} - 0$
- מקור מתח ייחוס פנימי של 1.1V

הממיר יכול לפעול בשני אופני פעולה: המרה יחידה או המרה מתמשכת. נדון בהמרה יחידה בלבד.

קביעת תדר אות השעון של הממיר

ניתן לבחור את תדר הפעולה של הממיר באמצעות מחלק תדר של אות השעון של המיקרו בקר. איור 5.4 מתאר את המבנה של מחלק התדר.



איור 5.4 מבנה מחלק התדר

זמן ההמרה של הממיר הוא 13 מחזורי שעון.
 בלוח הפתוח קיים מתנד גבישי של 16MHz ובסביבת הפיתוח משתמשים במחלק תדר של 128. אות השעון של הממיר ADC הוא:

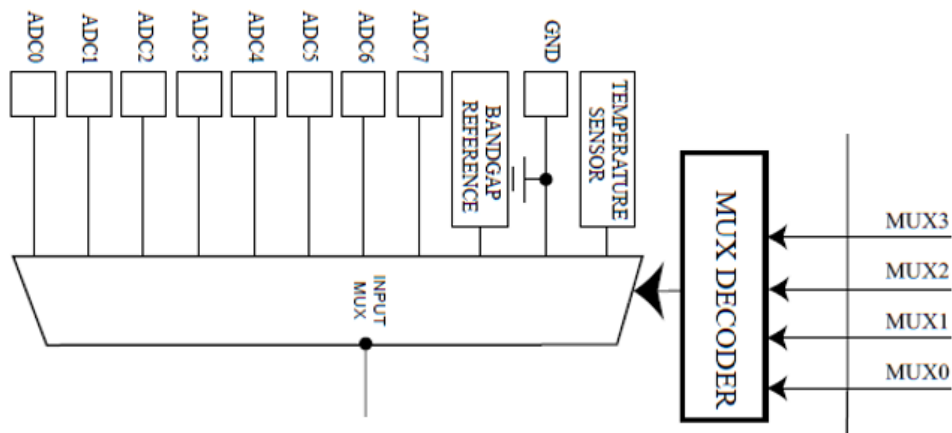
$$f_{ADC} = \frac{f_{osc}}{128} = 125000Hz$$

$$t_{conv} = \frac{1}{f_{ADC}} \cdot 13 = 104\mu s$$

ומכאן שזמן ההמרה הוא $104\mu s$

בחירת הערוץ

ניתן לבחור את ערוץ המבוא להמרה באמצעות קביעת הערך במבוא של המרבה התקבילי בממיר. איור 5.5 מתאר את המבנה העקרוני של המרבה.



איור 5.5 מבנה מרבה בממיר

הערה: במיקרו בקר ATmega329P ההדקים ADC6 ו-ADC7 לא קיימים.
 הפנייה לערוצים ADC0 – ADC5 יכולה להתבצע באמצעות הסימונים A0 – A5 או 0 – 5 בהתאמה בסביבת הפיתוח.
 לחייושן הטמפרטורה לא ניתן לפנות בסביבת הפתוח. ערכו במרבה הוא 8.

מידת טמפרטורה של המיקרו בקר

פעולה זו אינה מתאימה לשימוש על ידי התלמיד מכיוון שנדרשת התייחסות לאוגרי הממיר ADC כפי שמופיעים בדפי המפרט של המיקרו בקר. לא ניתן לגשת אל חיושן הטמפרטורה שלא באמצעות אוגרי המיקרו בקר.

```
void setup(){
  Serial.begin(9600);
  Serial.println(F("Internal Temperature Sensor"));
}

void loop(){
  Serial.println(GetTemp(),1);
}
```

```

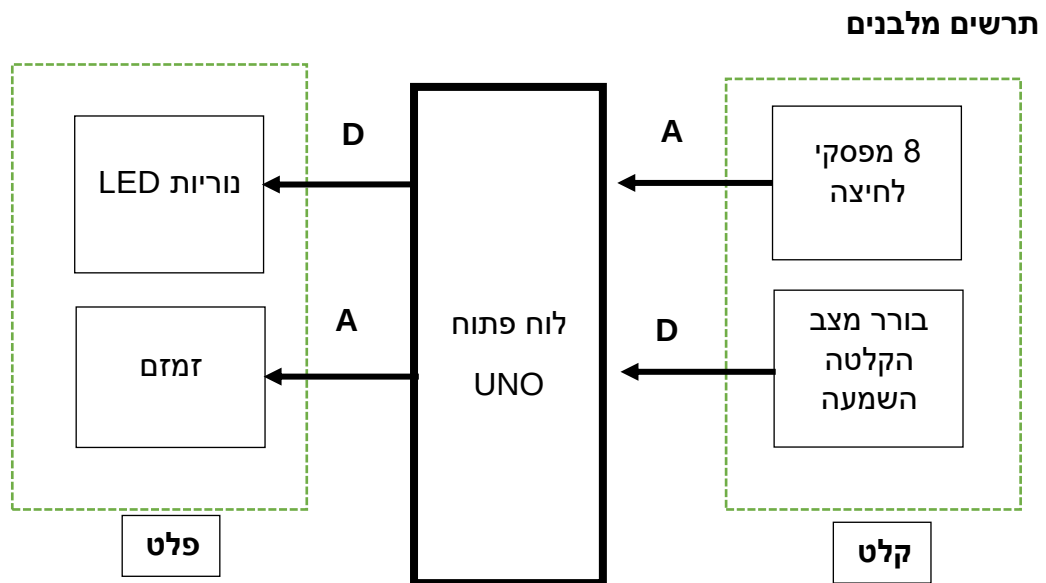
delay(1000);
}
double GetTemp(void){
  unsigned int wADC;
  double t;
  // קביעת הייחוס הפנימי של המרבב לחיישן הטמפרטורה
  ADMUX = (_BV(REFS1) | _BV(REFS0) | _BV(MUX3));
  ADCSRA |= _BV(ADEN); // אפשרור הממיר
  delay(20); // המתנה להתייצבות המתח בממיר
  ADCSRA |= _BV(ADSC); // אתחול ההמרה
  // זיהוי סוף המרה
  while (bit_is_set(ADCSRA,ADSC));
  wADC = ADCW; // קריאת ערך ההמרה בהתייחסות כאל מילה שלמה
  // קביעת ההיסט והתאמה לערכי צלזיוס
  t = (wADC - 324.31) / 1.22;
  return (t);
}

```

פרויקט לדוגמה - פסנתר מבוסס Arduino עם הקלטה והשמעה

מבוא

בפרויקט זה נבנה פסנתר זעיר תוך שימוש בערכת הפתוח Arduino-UNO. לפסנתר 8 מפסקי לחיצה (push button switch) וזמזם (buzzer). כדי להפוך את הפרויקט למעניין יותר נוסיף לו יכולת הקלטה (record) והשמעה (play) חוזרת. הבחירה בין מצבי ההקלטה להשמעה תתבצע תוך שימוש במפסק דו-מצבי (toggle switch). על מנת להבחין בין מצבי הפעולה נוסיף שתי נורות LED; צבוע אדום להקלטה, וירוק להשמעה.



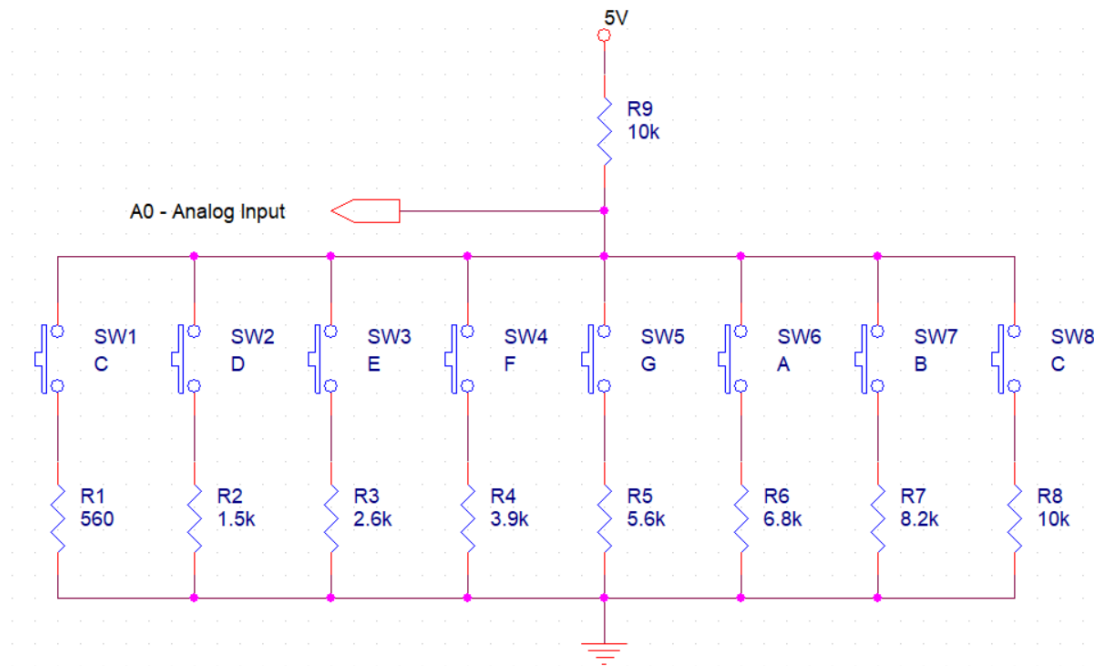
איור 5.6 תרשים מלבנים

תרשים המלבנים מראה את עקרונות החיבור של היחידות השונות אל המיקרו בקר. **יחידות הקלט** המחוברות אל המיקרו בקר הן 8 מפסקי הלחיצה המחוברים אל **מבוא אנלוגי** יחיד ובורר מצב ההקלטה וההשמעה המחובר אל **כניסה ספרתית**. **יחידות הפלט** מחוברות אל המיקרו בקר **ביציאות ספרתיות** הן נוריות ה-LED ויחידת הפלט מחוברת אל **יציאה אנלוגית**.

תיאור החומרה

מפסקי הלחיצה

כאמור, 8 מפסקי הלחיצה יחוברו במבנה המאפשר חיבור של כל המפסקים אל כניסה אנלוגית אחת. באיור 5.7 מתואר התרשים החשמלי של המפסקים.



איור 5.7 מפסקי הלחיצה

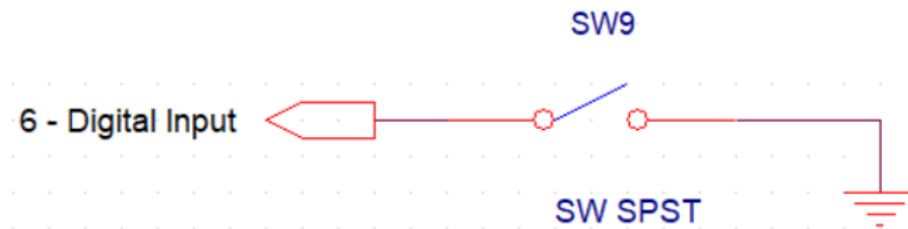
כאשר נלחץ אחד מהמפסקים נוצר מחלק מתח בין הנגד R9 לבין הנגד המחובר בטור למפסק. למשל, בלחיצה על מפסק לחיצה SW8 נוצר מחלק מתח בין הנגד R8 לנגד R9. מכיוון שהנגדים R1 עד R8 שונים, נקבל מתח שונה בהדק A0.

שאלות מומלצות לתלמידים

- א. מהו המתח שיימדד בהדק A0 כאשר אין מפסק לחוץ?
- ב. חשבו את המתח המתקבל בהדק A0 כאשר בכל פעם נלחץ מקש אחד בלבד. הציגו את התוצאה בטבלה.
- ג. חשבו את המתח כאשר לוחצים בו זמנית על מפסק SW1 ומפסק SW8. האם על סמך התוצאה ניתן לקבוע איזה מפסק נלחץ?
- ד. חשבו את המתח כאשר לוחצים בו זמנית על מפסק SW8 ומפסק SW7. האם על סמך התוצאה ניתן לקבוע איזה מפסק נלחץ?
- ה. מהי המסקנה המסתמנת הנוגעת ללחיצה בו זמנית על שני לחצנים או יותר?
- ו. כיצד ניתן לזהות אם אחד במקשים נלחץ?

בורר מצב השמעה הקלטה

בורר מצבי הפעולה של המערכת מספק להדק המבוא הספרתי אחד משתי רמות; HIGH – מצב השמעה ו- LOW עבור מצב הקלטה.

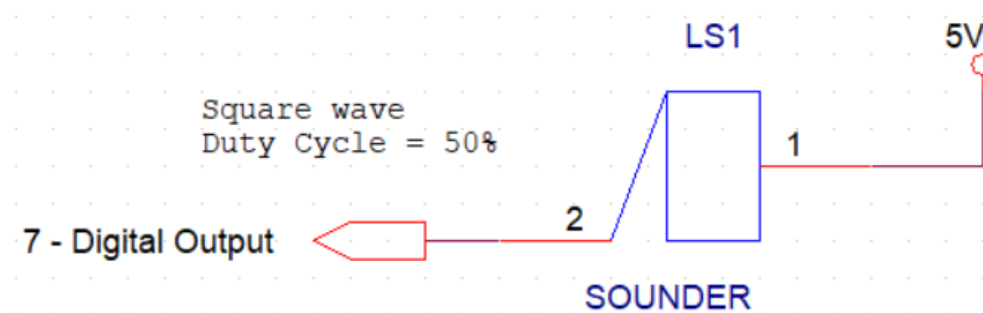


איור 5.8 מפסק בחירת מצב השמעה או הקלטה

בורר זה יחובר ללוח הפיתוח ללא נגד pull-up. כאן נשתמש בנגד pull-up מובנה¹ במיקרו בקר.

רמקול

הרמקול² הינו התקן המקבל אות גל ריבועי בתדר הנקבע באמצעות הפעולה tone. גורם המחזור של הגל הריבועי הוא 50%.



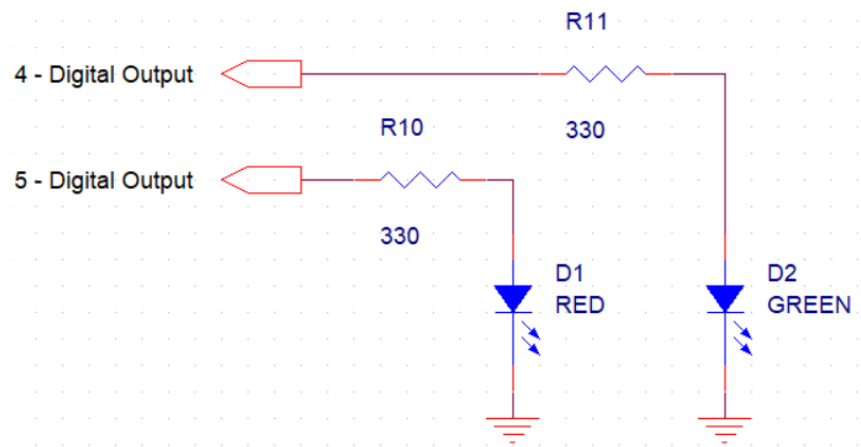
איור 5.9 חיבור רמקול

נוריות חיווי

נוריות החיווי משמשות לציין מצב הקלטה (RED) או השמעה (GREEN). הנוריות מופעלות ברמה לוגית גבוהה.

¹ ל מנת להשתמש בנגד pull-up מובנה נשתמש בהגדרה: pinMode(6, INPUT_PULLUP)

² רכיב sounder ולא רכיב buzzer



איור 5.10 נוריות חיווי

רשימת רכיבים

פריט	כמות	מזהה רכיב	רכיב
1	1	D1	LED RED
2	1	D2	LED GREEN
3	1	LS1	SOUNDER
4	1	R1	560
5	1	R2	1.5k
6	1	R3	2.6k
7	1	R4	3.9k
8	1	R5	5.6k
9	1	R6	6.8k
10	1	R7	8.2k
11	3	R8	10k
		R9	10k
		R12	10k
12	2	R10	330
		R11	330
13	2	SW1	C
		SW8	C
14	1	SW2	D
15	1	SW3	E
16	1	SW4	F
17	1	SW5	G
18	1	SW6	A
19	1	SW7	B
20	1	SW9	PLAY/RECORD

תכנה

קריאת 8 מפסקי לחיצה

כפי שניתחנו בסעיף מפסקי לחיצה הערך המתקבל בהדק A0 משתנה לפי מצבי המפסקים. כדי לקרוא את הערך בהדק זה נשתמש בפעולה `analogRead(A0)` המחזירה ערך מספרי המייצג את המתח בהדק A0. הערך המתקבל הוא בתחום בין 0 לבין 1023 עבור מתחים

בתחום בין 0V לבין 5V. על מנת לחשב את המתח המתקבל בהדק על פי הערך המספרי נשתמש בחישוב הבא:

$$V = \frac{5}{1024} value = 0.0048828125 * value$$

למשל, אם הערך שהוחזר על ידי הפעולה `value = analogRead(A0)` הוא 500 הרי שהמתח הקיים בהדק A0 הוא: 2.44V.

התנסות 1 – זיהוי לחצן

א. בנו את המעגל הכולל את 8 המפסקים וחברו את הדק המוצא שלהם אל הדק A0.
ב. כתבו תוכנית הקוראת כל 5 שניות את המתח בהדק A0 ומציגה בחלון המוניטור את הערך המספרי `value` ואת ערך המתח V לאחר החישוב.

תזכורת: לא לשכוח את זימון הפעולה `Serial.begin(9600)` ואת השימוש

בפעולה `Serial.print()` להצגת הערכים מתאימים בשלב ניפוי השגיאות בפיתוח.

ג. תעדו את תוצאות המדידה עבור כל מצבי המפסקים (9 מצבים שונים).

הפעלת רמקול

על מנת להפעיל את הרמקול בתדרים שונים נשתמש בפעולה tone המייצרת גל ריבועי.
תחביר הפעולה:

pin – מספר הדק בו יופק הגל הריבועי.	tone(pin, frequency)
בזמן נתון ניתן להפעיל יציאה אחת בלבד.	tone(pin, frequency, duration)
frequency – מספר חיובי שלם המייצג את תדירות האות שיופק בהדק.	
duration – מספר שלם חיובי ארוך המייצג את משך הזמן בו יופק התדר בהדק. הזמן נמדד ביחידות של מילי שנייה.	

למשל, הזימון tone(7, 500, 250) יפיק גל ריבועי בתדר 500Hz למשך רבע שנייה בהדק 7.

הפעולה noTone(pin) מסיימת את הפקת הגל הריבועי בהדק pin.

התנסות – 2 השמעת צליל

א. חברו את הרמקול אל לוח הפתוח.

ב. כתבו תוכנית להשמעת צליל בתדר של 1kHz למשך 1 שנייה.

סולם תווים:

C4	D4	E4	F4	G4	A4	B4	C5
262	294	330	349	392	440	494	523

ג. כתבו תוכנית המשמיעה את סולם הצלילים, תו אחר תו ברציפות ללא הפסקה. ממשו את התוכנית ללא מערכים ועם מערכים.

התנסות 3 – השמעת צליל לפי לחצן

כתבו תוכנית המשייכת צליל ללחצן. כאשר נלחץ לחצן SW1 יושמע הצליל C4, וכאשר נלחץ המקש SW2 יושמע הצליל D4 וכן הלאה.

התנסות 4 – בורר מצב הקלטה השמעה והפעלת ה-LED המתאים

א. חברו את מפסק הבורר והנוריות RED-LED ו-LED-GREEN.

ב. כתבו תוכנית המפעילה את הנוריות לפי מצב המפסק:

LED-GREEN	RED-LED	ערך הבורר – SW9
דולק	כבוי	HIGH
כבוי	דולק	LOW

התכנית הכוללת (הצעה)

```
// Set frequency for C4, D4, E4, F4, G4, A4, B4, C5
int notes[] = {262, 294, 330, 349, 392, 440, 494, 523};
const int btn = 6, spk = 7, greenLED = 4, redLED = 5;

char button = 0;
int analogVal;
char REC = 0;

int recorded_button[200];
int pev_button;

int recorded_time [200];
char time_index;

char button_index = 0;

unsigned long start_time;
int note_time;

void setup() {
  Serial.begin(9600);
  pinMode (btn, INPUT);
  pinMode(redLED, OUTPUT);
```

```
pinMode(greenLED, OUTPUT);
delay(2000); //Wait for display to show info
}
void loop() {
  while (digitalRead(btn) == 0) { //If the toggle switch is set in
recording mode
    digitalWrite(greenLED, LOW);
    digitalWrite(redLED, HIGH);
    Detect_button();
    Play_tone();
  }
  while (digitalRead(btn) == 1) { //If the toggle switch is set in Playing
mode
    digitalWrite(greenLED, HIGH);
    digitalWrite(redLED, LOW);

    for (int i = 0; i < sizeof(recorded_button) / 2; i++){
      delay((recorded_time[i] * 10); //Wait for before paying next tune
      if (recorded_button[i] == 0)
        noTone(spkr); //user didnt touch any button
      else
        tone(spkr, notes[(recorded_button[i] - 1)]); //play the sound
        corresponding to the button touched by the user
    }
  }
}

void Detect_button()
{
  analogVal = analogRead(A0); //read the analog voltag on pin A0
  pev_button = button; //remember the previous button pressed by
the user
```

```
if (analogVal < 550)
  button = 8;
if (analogVal < 500)
  button = 7;
if (analogVal < 450)
  button = 6;
if (analogVal < 400)
  button = 5;
if (analogVal < 300)
  button = 4;
if (analogVal < 250)
  button = 3;
if (analogVal < 150)
  button = 2;
if (analogVal < 100)
  button = 1;
if (analogVal > 1000)
  button = 0;
```

```
/**Rcord the pressed buttons in a array**/
if (button != pev_button && pev_button != 0)
{
  recorded_button[button_index] = pev_button;
  button_index++;
  recorded_button[button_index] = 0;
  button_index++;
}
/**End of Recording program**/
}

void Play_tone(){
  /**Rcord the time delay between each button press in a array**/
  if (button != pev_button) {
    note_time = (millis() - start_time) / 10;
```

```
recorded_time[time_index] = note_time;
time_index++;
start_time = millis();
}
/**End of Recording program**/
if (button == 0){
  noTone(spk);
}
else {
  tone(spk, notes[button-1]);
}
}
```

פרק 4: מיקרו בקרים

ביעדים שנקבעו בתוכנית הלימודים התלמיד נדרש להכיר את המבנה הפנימי של מיקרו בקר באופן כללי ואת המבנה הפנימי של מיקרו בקר ATmega329P.

מה טוב יותר CISC או RISC?

מאמרים לקריאה

- [מעבדי אינטל](#)
- [מעבדי ARM](#)
- [מאמר באנגלית Which one is better?](#)

פרק זה הוא בעיקרו עיוני. ניתן לבקש מהתלמיד לזהות התרשים המלבני של מיקרו בקר באופן כללי ושל ATmega329P בפרט את החלקים שנלמדו: זיכרון תוכנית, זיכרון נתונים, הדקים ספרתיים, הדקים תקביליים, ממיר ADC ויחידת PWM. ניתן בחלק זה לסקור את שיטות התקשורת השונות ולהדגיש את הצורך וחשיבות הזמינות של שיטות שונות במיקרו בקר יחיד. בשלב זה של שנת הלימודים אין בידי התלמיד מספיק כלים בשביל לנתח ממש את המעגל החשמלי של לוח הפיתוח וכל כן יש להסתפק בתיאור כללי בלבד של התרשים החשמלי. ניתן לבקש מהתלמיד להסביר את חיבורי נוריות הלד בתרשים החשמלי ולהציג את התנאים להפעלה של כל אחת מהן.

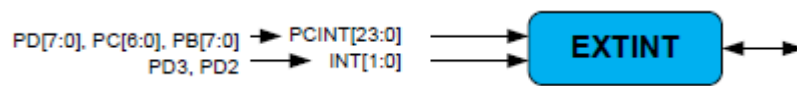
פרק 5: פסיקות חומרה ומונים

ביעדים שנקבעו בתוכנית הלימודים התלמיד נדרש לכתוב תוכניות תוך שימוש בפסיקות חומרה חיצוניות ופסיקות מונים.

בקר הפסיקות במיקרו-בקר ATmega328P

באיור 7.1 מוצגת יחידת הבקרה של פסיקות החומרה החיצוניות במיקרו-בקר. לפי האיור יש שתי קבוצות של מבואות הגורמות לפסיקה:

- INT[1:0] – שני הדקים ייעודיים שבאמצעותם אפשר לדווח על אירוע המחייב תגובה מיידית. כל הדק מפעיל שגרת פסיקה משלו.
- PCINT[23:0] 24 הדקים שאפשר לדווח באמצעותם על אירוע של שינוי ברמה הלוגית. קבוצת הדקים זו מחולקת לשלוש תת-קבוצות הדקים. כל תת-קבוצה כזאת מפעילה שגרת פסיקה אחת המשותפת לכל ההדקים.



איור 7.1 בקר הפסיקות במיקרו-בקר

בקר הפסיקות מסוגל לזהות בקשות פסיקה לפי שלוש אפשרויות:

- עלייה (rising) – האות החשמלי משנה את ערכו מרמה לוגית 0 לרמה לוגית 1.
- ירידה (falling) – האות החשמלי משנה את ערכו מרמה 1 לרמה 0.
- רמה נמוכה (low level) – האות החשמלי מוחזק ברמה נמוכה 0.

אם הדק מוגדר פלט, שינוי במצבו עלול לעורר בקשת פסיקה. אופציה זו מאפשרת הפעלת שגרת פסיקה גם באמצעות הוראה בתוכנה.

בפרק זה נרחיב ונדגים את השימוש בפסיקות שמקורן בשינוי ברמה של הדק מסוים מתוך 24 ההדקים המסומנים PCINT[23:0].

PCINT – Pin Change Interrupt Request

פרק 16 בדפי הנתונים שנית למצוא [כאן](#) מרחיב על נושא הפסיקות.

בקשות הפסיקה לפי הדקים נתונות בטבלה הבאה:

הדק הבקשה	קבוצת הדקים
PCI0	PCINT[7:0], PB7:PB0
PCI1	PCINT[14:8], PC6:PC0
PCI2	PCINT[23:16], PD7:PD0

דוגמה – מניית אירועי שינוי ירידה בהדק A4

בדוגמה זו הנחנו קיומו של מפסק מגנטי המחובר להדק A4 בלוח הפיתוח. כאשר אין בקרבת המפסק חפץ מגנטי הוא במצב פתוח. כאשר בקרבת המפסק נמצא חפץ מגנטי הוא נסגר. המטרה היא למנות כמה פעמים נמצא חפץ מגנטי בקרת המפסק. בדוגמה המפסק מחובר בין הדק A4 לאדמה ולכן הוגדר ההדק ככניסה עם נגד pull-up. להלן התוכנית:

```
#define MAGSENSOR A4

volatile unsigned long debouncetime;// משתנה לזמן ריטוטים
volatile byte magCount; // מונה מצב המפסק

void setup() {
  Serial.begin(9600);
  pinMode(MAGSENSOR, INPUT_PULLUP);// הגדרת הדק כניסה
  magCount = 0;
  // אפשרור ההדק
  *digitalPinToPCMSK(MAGSENSOR) |= bit
  (digitalPinToPCMSKbit(MAGSENSOR));
  // ניקוי בקשות פסיקה מקבוצה אליה שייך הדק זה
  PCIFR |= bit (digitalPinToPCICRbit(MAGSENSOR));
  // אפשרור בקשות פסיקה מהקבוצה אליה שייך ההדק
  PCICR |= bit (digitalPinToPCICRbit(MAGSENSOR));
  Serial.println("start counting");
}

void loop(){
  Serial.println(magCount);
  delay(1000);
}

// שגרת A0:A5 פסיקה לטיפול בהדקים
ISR (PCINT1_vect) {
  if (digitalRead(MAGSENSOR) == LOW)
    debouncetime = millis();
}
```

```

else if (millis() - debouncetime > 250){
magCount++;
}
}

```

כפי שניתן לראות נעשה שימוש במצביעים בעת ההגדרה של קבוצת ההדקים. התלמיד אינו נחשף לטיפול של מצביעים בתוכנית הלימודים בתיכון.

סביבת הפתוח תומכת בפסיקות חיצוניות INT0 ו-INT1 בצורה של פעולות מותאמות. אין תמיכה זהה בפסיקות של שינוי מצב בהדק אלא באמצעות מצביעים ושימוש באוגרי המיקרו בקר.

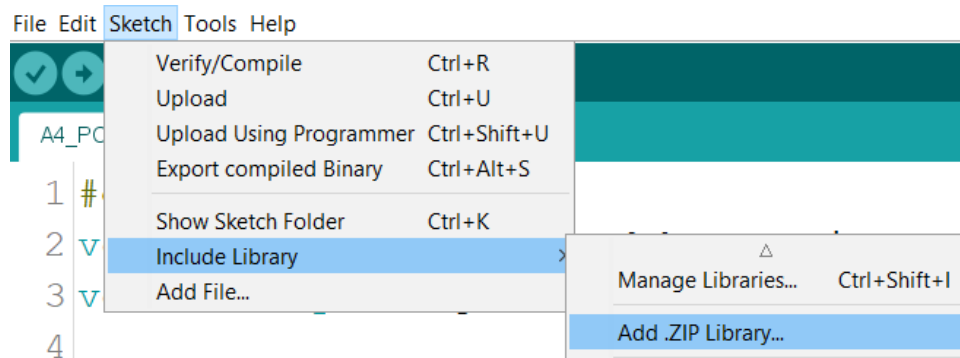
כמו כן סביבת הפתוח לא תומכת בפסיקות של קוצבי זמן (timres) של המיקרו בקר באופן ישיר אלא באמצעות שימוש באוגרי המיקרו בקר.

מחלקת Timer

המחלקה משתמשת בפעולה millis() כך שכושר ההבחנה שלה היא 1 מילי שנייה. הספרייה משתמשת בשיטת התשאול כך שאם מופע המחלקה אותחל לתזמון כל 2 מילי שניות והפעולה המקושרת פעולת 5 מילי שניות, הפעולה המתוזמנת תופעל כל 5 מילי שניות במקום כל 2 מילי שניות.

התקנה

1. הורד את הספרייה [כאן](#). שמור את הקובץ בתיקייה לפי בחירתך. זכור תיקייה זו.
2. הוסף את הספרייה. פעל על פי איור 7.2



איור 7.2 תפריט הוספת ספרייה

פעולות המחלקה

הפעולה	תיאור
<code>int every(long period, callback, context)</code> <code>int every(long period, callback, int repeatCount, context)</code>	הפעולה תזמן את הפעולה callback כל תקופה של period במילי שניות. במידה והוגדר ערכו של המשתנה repeatCount הפעולה callback תתבצע לפי ערכו של משתנה זה. context מייצג את הפרמטר המועבר לפעולה callback. הפעולה מחזירה את המזהה ID של אירוע קוצב הזמן.
<code>int after(long duration, callback, context)</code>	הפעולה תזמן את הפעולה callback <u>לאחר</u> התקופה duration במילי שניות. context מייצג את הפרמטר המועבר לפעולה callback. הפעולה מחזירה את המזהה ID של אירוע קוצב הזמן.
<code>int oscillate(int pin, long period, int startingValue)</code> <code>int oscillate(int pin, long period, int startingValue, int repeatCount)</code>	הפעולה <u>משנה</u> את מצבו של הדק pin <u>כל תקופה</u> של period מילי שניות. ערכו ההתחלתי נקבע לפי startingValue (HIGH or LOW). במידה והוגדר ערכו של המשתנה repeatCount הפעולה תתבצע לפי ערכו של משתנה זה. הפעולה מחזירה את המזהה ID של אירוע קוצב הזמן.
<code>int pulse(int pin, long period, int startingValue)</code>	הפעולה <u>משנה</u> את מצבו של הדק pin <u>פעם אחת בלבד</u> אחרי תקופה של period מילי שניות. ערכו ההתחלתי נקבע לפי

startingValue (HIGH or LOW) הפעולה מחזירה את המזהה ID של אירוע קוצב הזמן.	
הפעולה עוצרת את האירוע לפי המזהה ID. הפעולה מחזירה את המזהה ID של אירוע קוצב הזמן.	int stop(int id)
חייבים לזמן את הפעולה מתוך הפעולה הראשית loop. פעולה זו משרתת את כל האירועים.	int update()

תוכנית דוגמה 1

התוכנית נלקחה מהדוגמאות המצורפות למחלקה. התוכנית נקראת Blink2.
התוכנית גורמת לנורת LED1 להחליף את מצבו כל שנייה ואת נורת LED2 להחליף את
מצבו כל 10 שניות.

```

1) #include "Timer.h"
2) const int LED1 = 8;
3) const int LED2 = 9;
4) const unsigned long PERIOD1 = 1000;
5) const unsigned long PERIOD2 = 10000;
6) עצם של המחלקה // Timer t;
7) void setup(void) {
8) pinMode(LED1, OUTPUT);
9) pinMode(LED2, OUTPUT);
10)t.oscillate(LED1, PERIOD1, HIGH);
11)t.oscillate(LED2, PERIOD2, HIGH);
12)}
13)void loop(void) {
14)t.update();
15)}
```

הסבר

בשורה 1 הכללנו את הספרייה של המחלקה Timer.

בשורה 6 יצרנו עצם בשם t של מחלקת Timer.

בשורות 10 ו- 11 קבענו את הקשר בין הדק לבין מחזור הפעולה שלו.

בשורה 14 זימנו את פעולת העדכון. **חובה לזמן!**

תוכנית דוגמה 2

התוכנית נלקחה מהדוגמאות באתר האינטרנט [כאן](#).

התוכנית תוקנה ועודכנה. התוכנית מדגימה שימוש בפעולות השונות.

```

1) #include "Timer.h"
2) Timer t;
3) int ledEvent; // משתנה מכיל מזהה אירוע להפעלת הנורה
4) void setup() {
5)   Serial.begin(9600);
6)   t.every(2000, doSomething, (void*)2);
7)   pinMode(13, OUTPUT);
8)   ledEvent = t.oscillate(13, 50, HIGH);
9)   t.after(10000, doAfter, (void*)2);
10) }
11) void loop() {
12)   t.update();
13) }
14) void doSomething() {
15)   Serial.print("2 second tick: millis()=");
16)   Serial.println(millis());
17) }
18) void doAfter() {
19)   Serial.println("stop the led event");
20)   t.stop(ledEvent);
21)   t.oscillate(13, 500, HIGH, 5);
22) }
```

הסבר

בשורה 1 הכללנו את הספרייה של המחלקה Timer.

בשורה 2 יצרנו עצם בשם t של מחלקת Timer.

בשורה 6 בוצע זימון של הפעולה every המזמנת את הפעולה doSomething כל 2 שניות.
 בשורה 8 בוצע זימון של הפעולה oscillate הגורמת לנורת המחוברת להדק 13 להחליף את מצבה כל 50 מילי שניות. מזהה אירוע של הפעולה נשמר במשתנה ledEvent.
 בשורה 9 בוצע זימון של הפעולה after הגורמת לזימון הפעולה doAfter לאחר 10 שניות.
 בשורה 12 זימנו את פעולת העדכון. **חובה לזמן!**
 בשורות 14-17 מוגדרת הפעולה doSomething המציגה במסוף הטורי את הזמן ביחידות מילי שניות מרגע הפעלת התוכנית. מכיוון שהפעולה מזמנת כל שתי שניות נראה את הזמן בכפולות של 2000.
 בשורות 18-22 מוגדרת הפעולה doAfter העוצרת את האירוע בשורה 8 ומפעילה אירוע חדש המפעיל את הנורה בהדק 13 כך שיחליף את מצבו כל 500 מילי שנייה במשך 5 פעמים בלבד.
 לסיכום, התוכנית מפעילה את נורת הלד במשך 10 שניות בקצב של 50 מילי שנייה. כעבור 10 שניות הורת הלד תופעל בקצד של חצי שנייה במשך 5 פעמים. לאחר מכן נורת הלד תישאר באותו המצב (דולק).

פתרון שאלה 5.3

- מתכננים שעון שניות חד-ספרתי הסופר מטה מערך 5. בתחילת כל הפעלה השעון מאותחל לערך ההתחלתי (5). באמצעות מפסק לחיצה START תחל ספירה מטה. כאשר השעון מגיע לערך 0 יופעל זמזם למשך שנייה אחת והשעון יאותחל לערכו ההתחלתי.
- כדי לשנות את ערך השעון מספקים למערכת אות גל ריבועי בתדר של 1Hz ממקור חיצוני (מחולל אותות).
 - השעון יוצג באמצעות יחידת תצוגה של שבעה מקטעים מסוג CA.
- א. זהו את סוגי הכניסות והיציאות למערכת המתוארת (ספרתית או תקבילית).
 - ב. תארו את המערכת בתרשים מלבנים.
 - ג. הסבירו את שתי גישות הפתרון האפשריות למערכת המתוארת.
 - ד. כתבו תוכנית למימוש המערכת המתוארת לפי שיטת התשאל. בפתרון התייחסו להקצאת ההדקים.
 - ה. כתבו תוכנית למימוש המערכת המתוארת לפי שיטת הפסיקה. בפתרון התייחסו להקצאת ההדקים. כניסות ספרתיות: מפסק ושעון.
 - ו. יציאות ספרתיות: תצוגת 7 מקטעים וזמזם.
 - ז. תרשים מפורט באיור 7.3.
 - ח. ניתן לממש לפי שיטת התשאל או לפי שיטת הפסיקה.



תוכנית לפי שיטת התשאול

134

```

write7Seg();
while(digitalRead(START)== HIGH);
clkValue = digitalRead(CLK);
}
void loop() {
  int currentClkValue = digitalRead(CLK);
  if (currentClkValue != clkValue){
    if (currentClkValue == LOW) { // falling edge
      counter--;
      write7Seg();
      if (counter == 0){
        digitalWrite(BUZZER, HIGH);
        delay(1000);
        digitalWrite(BUZZER, LOW);
        counter = 5;
        write7Seg();
        while(digitalRead(START)== HIGH);
        currentClkValue = digitalRead(CLK);
      }
    }
    clkValue = currentClkValue;
  }
}

void write7Seg(){
  PORTD = (PIND & B00001111) | (table7Seg[counter] << 4);
  PORTB = (PINB & B11111000) | (table7Seg[counter] >> 4);
}

```

תוכנית לפי פסיקה

```

// 0 1 2 3 4
byte table7Seg[] = {B1000000, B1111001, B0100100, B0110000,
B0011001,

```



```
// 5 6 7 8 9
B0010010, B0000010, B1111000, B0000000, B0010000};
byte counter;
const int BUZZER = 11;
const int CLK = 2;
const int START = 3;
void setup() {
  DDRD = B11110000; //pins 7 6 5 4 3 2 1 0
  DDRB = B00000111; //pins 13 12 11 10 9 8
  Serial.begin(9600);
  pinMode(BUZZER, OUTPUT);
  pinMode(CLK, INPUT);
  pinMode(START, INPUT_PULLUP);
  counter = 5;
  write7Seg();
  while(digitalRead(START)== HIGH);
  attachInterrupt(digitalPinToInterrupt(CLK), updateCounter,
  FALLING);
}

void loop() {
}

void write7Seg(){
  PORTD = (PIND & B00001111) | (table7Seg[counter] << 4);
  PORTB = (PINB & B11111000) | (table7Seg[counter] >> 4);
}

void updateCounter() {
  counter--;
  write7Seg();
  if (counter == 0){
    digitalWrite(BUZZER, HIGH);
```

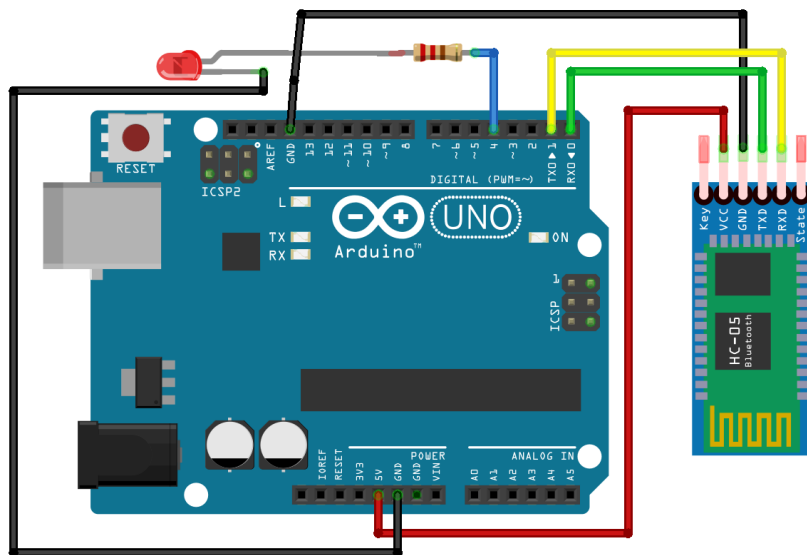
```
delay(1000);  
digitalWrite(BUZZER, LOW);  
counter = 5;  
write7Seg();  
while(digitalRead(START)== HIGH);  
}  
}
```

פרק 6: תקשורת טורית UART בין מיקרו בקר למחשב

ביעדים שנקבעו בתוכנית הלימודים התלמיד נדרש להכיר תקשורת UART ובאמצעותה יעביר נתונים בין מיקרו בקר מבוסס Arduino - לבין מחשב PC המריץ תוכנה בשפת C#. עד לשלב זה התלמידים התנסו בשימוש בפעולות בסיסיות של פלט בלבד באמצעות הפעולה print או println של מחלקת התקשורת Serial למטרות הודעות מצב (status) הקשורות לפיתוח תוכנית. מומלץ להבהיר לתלמידים את חשיבות התקשורת הדו-כיוונית. ניתן להביא דוגמאות מתחום תקשורת האינטרנט. למשל, הפעלת מכשיר המחובר להתקן תקשורת ובהתאם לפקודה המגיעה מטלפון חכם (smartphone) יופעל או ינותק המכשיר תוך כדי משלוח חוזר של הודעת ביצוע. ניתן להדגים באמצעות שימוש במודול תקשורת bluetooth המחובר לערוץ התקשורת הטורית.

תוכנית הדגמה

בתוכנית זו ניצור תקשורת בין טלפון חכם לבין לוח הפיתוח תוך שימוש במודול תקשורת מדגם HC05 (או HC06). באיור 8.1 מוצג תרשים חיבורים של התקן bluetooth המחובר לערות התקשורת הטורית ונורת LED. הנורה תופעל אם תתקבל מחרוזת "ON" ותכובה אם תתקבל מחרוזת "OFF". אם מתקבלת מחרוזת תקינה תוחזר המחרוזת "ON_OK" או "OFF_OK". עבור כל מחרוזת אחר תוחזר המחרוזת "NOK".



איור 8.1 תרשים חיבורים

התוכנית

```

const int LED = 4;
String command = "";

void setup(){
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  digitalWrite(LED, LOW); // כיבוי הנורה
}

void loop(){
  if (Serial.available()>0){
    command = Serial.readStringUntil('\n');
    if(command == "ON"){
      digitalWrite(LED, HIGH); // הדלקת הנורה
      Serial.println("ON_OK"); // אישור ביצוע
    }
    else if (command == "OFF"){
      digitalWrite(LED, LOW); // כיבוי הנורה
      Serial.println("OFF_OK"); // אישור ביצוע
    }
    else {
      Serial.println("NOK");// דחיית בקשה
    }
  }
}

```

אפליקציה לטלפון

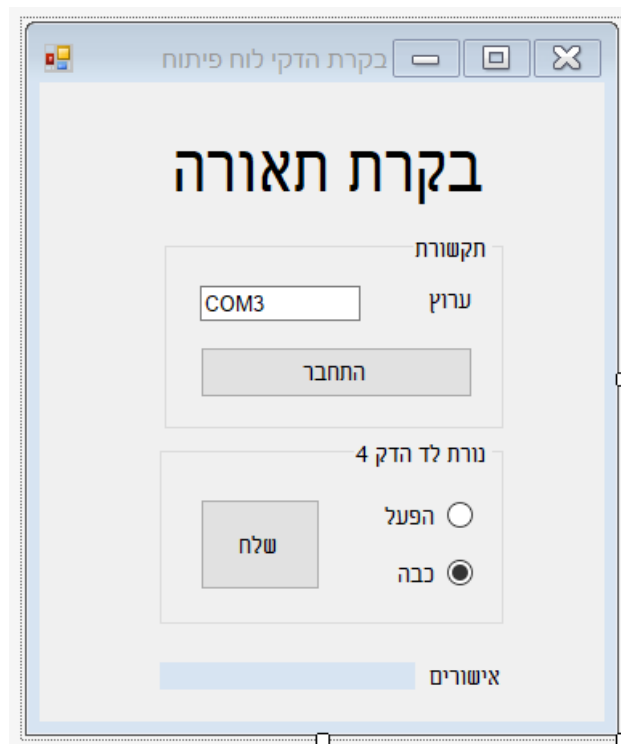
קיימות אפליקציות רבות להתחברות להתקן BT.
[כאן](#) ניתן למצוא אפליקציה לדוגמה.

אפליקציה גרפית למחשב

האפליקציה הגרפית הבאה תפעל באופן דומה לאפליקציה של הטלפון החכם. האפליקציה תשלח מחרוזות הפעלה "ON" ומחרוזת כיבוי "OFF" לפי פקדים המופיעים באפליקציה. התוכנה של לוח הפיתוח תישאר ללא שינוי.

עיצוב גרפי

איור 8.2 מציג הצעה לעיצוב גרפי לטופס.



איור 8.2 העיצוב הגרפי

בטופס נעשה שימוש בפקדים הבאים:

Lable, TextBox, RadioButton, Button, GroupBox

התוכנית

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
```

```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Arduino
{
    public partial class Form1: Form
    {
        public Form1()
        {
            InitializeComponent();
        }

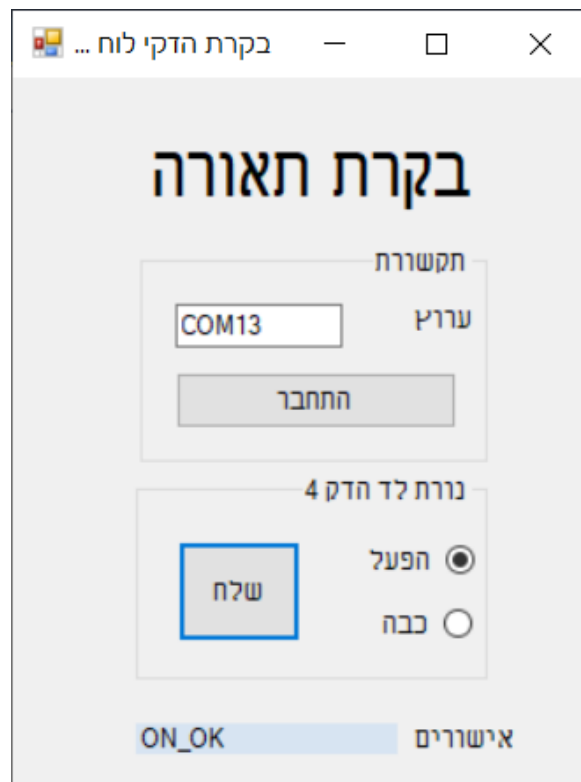
        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (serialPort1.IsOpen)
            {
                serialPort1.Close();
            }
            serialPort1.PortName = textBox1.Text;
            try
            {
                serialPort1.Open();
                serialPort1.ReadTimeout = 100;
                serialPort1.WriteTimeout = 100;
                button1.Enabled = true;
            }
            catch { }
        }
    }
}
```

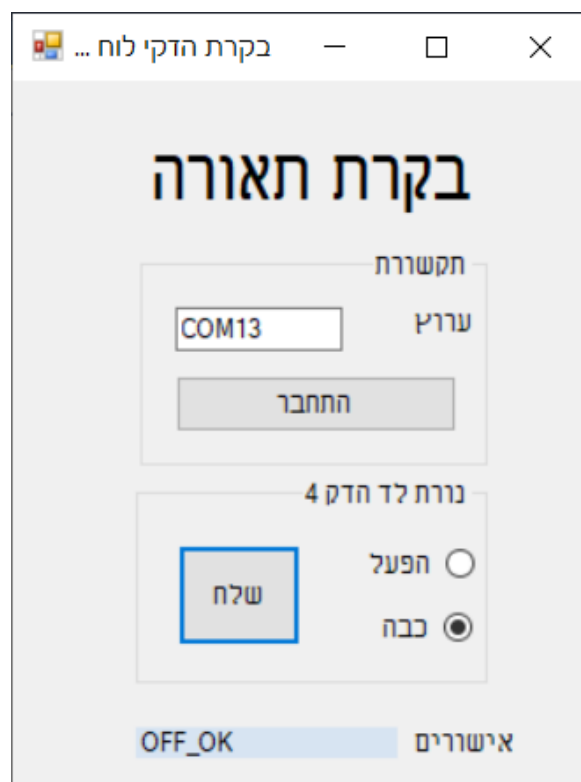
```
}  
catch (Exception)  
{  
    MessageBox.Show("הערוץ של פתיחה שגיאת");  
}  
  
}  
  
private void button1_Click(object sender, EventArgs e)  
{  
    if (radioButton1.Checked)  
        serialPort1.WriteLine("ON");  
    else  
        serialPort1.WriteLine("OFF");  
    timer1.Enabled = true;  
}  
  
private void timer1_Tick(object sender, EventArgs e)  
{  
    serialPort1.ReadLine();  
    string indata = serialPort1.ReadLine();  
    label4.Text = indata;  
    timer1.Enabled = false;  
}  
}  
}
```

מסכי הפעלה

מצב הפעלה



מצב כיבוי



התוכנית דומה מאוד לדוגמה המופיעה בספר הלימוד. הדגש בדוגמה הושם על העצוב הגרפי.

ניתן להכשיר את התלמידים לשימוש בעיצוב הגרפי תוך מספר שיעורים ואחר כך לשלב את התקשורת הטורית.

את התוכנית המלאה ניתן למצוא [כאן](#).

נספח א: התקנת סביבת הפתוח

להורדת סביבת הפתוח הפעל את הקישור הבא:

[Arduino Software Download](#)

המתן לפתיחת הדף המתאים בדפדפן.

גלול את הדף עד לקבלת ההתייחסות לסביבת הפתוח:

Download the Arduino IDE



ייתכן וגרסת סביבת הפתוח תהיה שונה מזו בעת כתיבת הנספח.

מצד ימין מופיעות הגרסאות השונות לפי מפתח של מערכת ההפעלה.

בחרו בגרסה המתאימה למערכת ההפעלה של המחשב בו אתם משתמשים.

למשתמשי Windows מומלץ לבחור באפשרות הראשונה.

לאחר הבחירה יופיע דף בו תוזמנו לתרום במידה ותרצו:

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

בחרו באפשרות המועדפת עליכם.

לאחר הבחירה יורד קובץ התקנה של התכנה: arduino-1.8.5-windows.exe או בשם

דומה על פי הגרסה. שם הקובץ תואם למערכת ההפעלה.

לאחר הורדת הקובץ הפעילו את קובץ ההתקנה ופעלו על פי ההנחיות. בהלך ההתקנה יותקן

גם מנהל התקן (driver) לחיבור לוח Arduino למחשב.

בסיום ההתקנה יתווסף לשולחן העבודה סמל של סביבת הפתוח:



יחידה 5: יסודות התכנות בשפת #C

היקף: 150 שעות עיון; 60 שעות מעבדה

פרק 1

פתח דבר

בפרק זה התלמידים ילמדו מונחי יסוד המשתייכים לכל ההתקנים (Devices) בעולם מבוססי מחשב. עליהם להבין מהם קלטים, פלטים, ביצוע מניפולציות חישוביות ועוד. כמו כן, התלמידים יבינו מהו אלגוריתם, מהי תוכנית מחשב ומדוע הוראה או פקודת מחשב חייבת להיות חד משמעית. הם יכירו את סביבת העבודה Visual Studio וילמדו בצורה יסודית לבנות פרויקט בסיסי, לכתוב תוכנית מחשב ראשונה ולהתנסות בכתיבת קוד. מצופה מהמורה לתמוך בתלמידים בצעדיהם בעולם התכנות ולתת להם את ההזדמנות להתנסות באופן חיובי ולהצליח. כלומר, לקבל פלטים ראשונים ו/או להריץ קובץ exe (בר הרצה). בנוסף מצופה מהמורה בעיקר למנוע התנסויות שליליות בהמשך, להציג התמודדויות עם בעיות קלאסיות (כגון שגיאות תחביר). מאוד חשוב בשלב המוקדם הנ"ל להציג דרך (אחת לפחות) בה יש לשמור פרויקט שלם, וכיצד יש לפתוח פרויקט מחדש. כמו כן, מומלץ, לכל מורה להראות לתלמידיו כיצד יש לגבות את החומרים אותם הם כותבים ברמה האישית (DiskOnKey או Dropbox או GoogleDrive) או ברמת בית הספר (כונן משותף, אם קיים, במעבדה).

סיכום מונחי יסוד

- מחשב (Computer) הוא מכונה חישובית המסוגלת לבצע מטלות שונות. המחשב קולט (Input) נתונים, העיבוד (Processing) מתבצע במחשב ע"י אוסף סדור של הוראות ולבסוף מתבצע פלט (Output).
- חומרה ותוכנה ⇔ מהווים מערכת מחשב:
- חומרה (Hardware) – הרכיבים הפיזיים של מערכת המחשב.
- תוכנה (Software) – תוכניות מחשב המבצעות מדיניות מסוימת (Policy). אין רכיבים פיזיים.

דוגמאות

תוכנה	חומרה
מערכות ההפעלה השונות	מעבד המחשב (CPU)
ממשקי משתמש (גרפי / לא גרפי)	זיכרון ראשי (RAM)
ספריות שונות	אמצעי קלט/פלט
אפליקציות	לוח אם

- מערכת ההפעלה (Operating System) היא תוכנית מחשב, מבוססת מדיניות, המנהלת את החומרה והתוכנה ומאפשרת גישה אליהם על ידי המשתמשים.

דוגמאות נוספות שלא מופיעות בפרק:

דוגמא :1		
המטרה:	ביצוע חישוב מיסים לעובדים בחברה מסוימת.	
קלט:	מידע שונה על העובדים (למשל: דרגת העובד, השכלה, ניסיון וכו').	
עיבוד:	לפי אוסף חוקים ומוסכמות נקבע אחוז המס.	
פלט:	תלוש שכר ובו יופיע (בין השאר) השכר לפני ואחרי הורדת המס.	

דוגמא :2		
המטרה:	גישה לאתר אינטרנט בצורה מהירה.	
קלט:	סריקת בר קוד (באמצעות הפלאפון) מודפס / מוצג.	
עיבוד:	הפלאפון ינסה להגיע על בסיס הרשת להגיע ולהציג את נתוני האתר.	
פלט:	הצגת האתר המבוקש כי גבי מסך הפלאפון.	

פרק 2

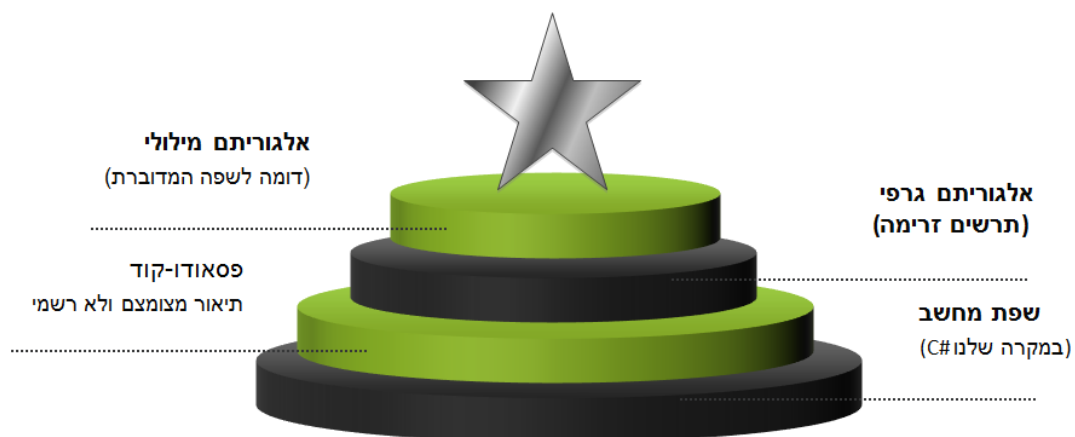
פתח דבר

בפרק זה המורה נדרש להסביר לתלמידים את ההבדלים בין טיפוסים הנתונים השונים – ולהרחיב שזיכרון המחשב, בסופו של דבר, הוא מוגבל ולכן יהא עלינו לחשוב כיצד אפשר לחסוך בזיכרון. כמו כן, עלינו להבדיל בין סוגי טיפוסים הנתונים השונים. על המורה להסביר את השימוש בפונקציה / מטודה ולהדגיש את העבודה של פונקציה / מטודה מסוימת למחלקה מסוימת. לסיום, חשוב מאוד להדגים כיצד לגשת לבעיה אלגוריתמית ולפתור אותה בשלבים מוגדרים.

תיאור דרכים שונות לייצוג אלגוריתם:

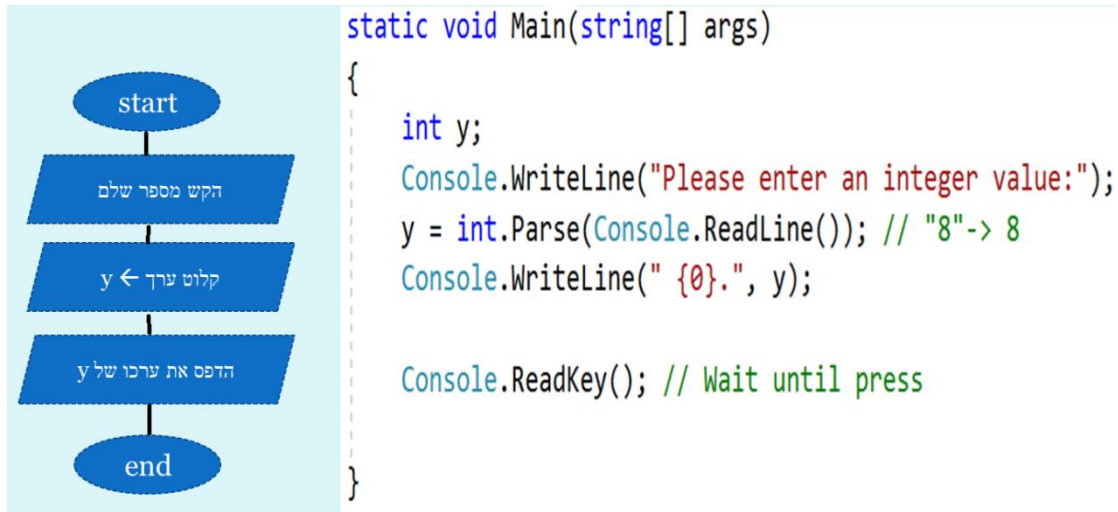
דרכים לייצוג אלגוריתם:

אלגוריתם – אוסף של הוראות חד משמעיות הניתנות לביצוע

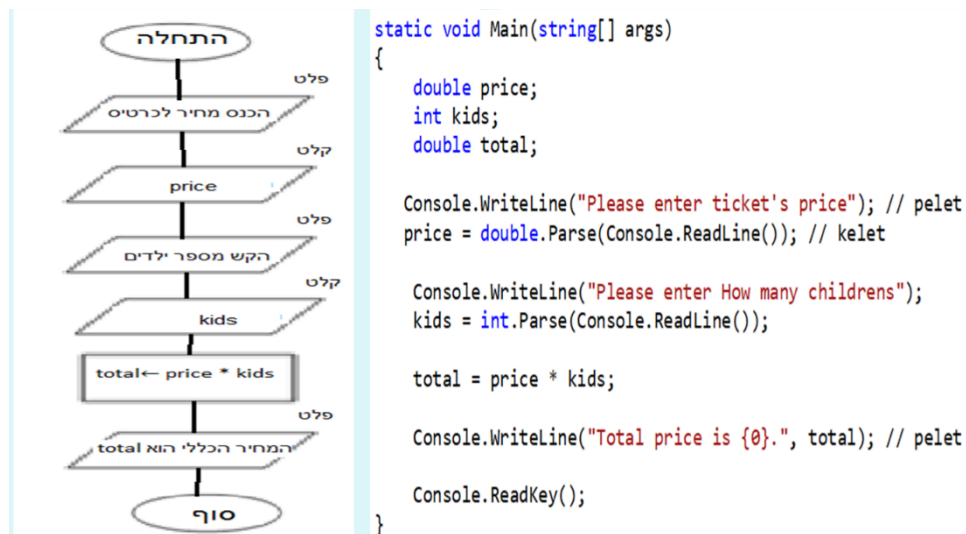


שילוב בין תרשים זרימה לתוכנית מחשב:

א. דוגמה מספר 1 - הקשר בין תרשים זרימה לתוכנית מחשב:



ב. דוגמה מספר 2 - הקשר בין תרשים זרימה לתוכנית מחשב:



כתיבת תוכנית מחשב בשלבים:

השאלה: כתוב תוכנית הקולטת מספר שעות עבודה (int) ותעריך לשעה (double).

לדוגמא: עבור עובד העובד 8 שעות, 50.0 ₪ הפלט יהיה ⇔ 400.0 ₪

שלבי הפתרון:

שלב ראשון – הצהרת משתנים:

```
static void Main(string[] args)
{
    //1)
    int hours; // מספר השעות ביום
    //2)
    double pay; // תשלום לשעה
```

שלב שני – הנחייה למשתמש – פלט למסך
:"אנא הקש...")

```
//3)
Console.WriteLine("Please enter hours and payment per hour:");
```

שלב שלישי – קלט (עבור כל משתנה)

```
//3)
Console.WriteLine("Please enter hours and payment per hour:");
```

שלב רביעי – פלט מסכם למסך

```
Console.Write("{0} X {1} =", hours, pay);
Console.WriteLine(" {0}.", hours * pay);

Console.ReadKey(); // Wait until press
```

דוגמה הרצה מספר 1:

Please enter hours and payment per hour:

10

55.09

10 X 55.09 = 550.9.

דוגמה הרצה מספר 2:

Please enter hours and payment per hour:

8

162.7

$$8 \times 162.7 = 1301.6.$$

טבלאות מעקב:

למורה: אנו ממליצים לבנות טבלאות מעקב ולמלא אותן בשלבים מוגדרים.
דוגמאות על בסיס דוגמא ב': הערה טכנית: המשתנים אמורים להיות באותיות קטנות

הרצה מספר 1:

יצירת טבלת מעקב ריקה:

Price	Kids	Total	פלט (מסך)
-------	------	-------	-----------

ביצוע פקודה ראשונה (פלט ראשון בתוכנית):

Price	Kids	Total	פלט (מסך)
			אנא הקש מחיר לכרטיס

ביצוע פקודה שנייה (קלט ראשון בתוכנית):

Price	Kids	Total	פלט (מסך)
			אנא הקש מחיר לכרטיס
28.5			

ביצוע פקודה מספר 3 (פלט שני בתוכנית):

Price	Kids	Total	פלט (מסך)
			אנא הקש מחיר לכרטיס
28.5			
			אנא הקש מספר תלמידים

ביצוע פקודה מספר 4 (קלט שני בתוכנית):

Price	Kids	Total	פלט (מסך)
			אנא הקש מחיר לכרטיס
28.5			
			אנא הקש מספר תלמידים
	20		

ביצוע פקודה מספר 5 (חישוב והשמה למשתנה total):

Price	Kids	Total	פלט (מסך)
			אנא הקש מחיר לכרטיס
28.5			
			אנא הקש מספר תלמידים
	20		
		570.0	

פקודה מספר 6 – פלט מספר 3 (פקודה אחרונה בתוכנית):

Price	Kids	Total	פלט (מסך)
			אנא הקש מחיר לכרטיס
28.5			
			אנא הקש מספר תלמידים
	20		
		570.0	
			המחיר הכולל הוא 570.

הרצה מספר 2:

יצירת טבלת מעקב ריקה:

Price	Kids	Total	פלט (מסך)
-------	------	-------	-----------

ביצוע פקודה ראשונה (פלט ראשון בתוכנית):

Price	Kids	Total	פלט (מסך)
			אנא הקש מחיר לכרטיס

ביצוע פקודה שניה (קלט ראשון בתוכנית):

Price	Kids	Total	פלט
			אנא הקש מחיר לכרטיס
33.7			

ביצוע פקודה מספר 3 (פלט שני בתוכנית):

Price	Kids	Total	פלט (מסך)
			אנא הקש מחיר לכרטיס
28.5			
			אנא הקש מספר תלמידים

ביצוע פקודה מספר 4 (קלט שני בתוכנית):

Price	Kids	Total	פלט
			אנא הקש מחיר לכרטיס
33.7			
			אנא הקש מספר תלמידים
	12		

ביצוע פקודה מספר 5 (חישוב והשמה למשתנה total):

Price	Kids	Total	פלט
			אנא הקש מחיר לכרטיס
33.7			
			אנא הקש מספר תלמידים
	12		
		404.4	

פקודה מספר 6 – פלט מספר 3 (פקודה אחרונה בתוכנית):

Price	Kids	Total	פלט
			אנא הקש מחיר לכרטיס
33.7			
			אנא הקש מספר תלמידים
	12		
		404.4	
			המחיר הכולל הוא 404.4

פתרון השאלה – "בית החולים":

```

static void Main(string[] args)
{
    int kids; // המאושפדים הילדים מספר
    double price; // כניסה מחיר
    double total; // הכללית העלות
    double prn; // להדפסה אחוז 15
    double cost; // ימבורי'ג עלות אחוז 18
    double rest; // הכסף שאר

    Console.WriteLine("How many kids?");
    kids = int.Parse(Console.ReadLine());

    Console.WriteLine("What is the price for ticket?");
    price = double.Parse(Console.ReadLine());

    total = kids * price;
    prn = total * 0.15;
    cost = total * 0.18;
    rest = total - (prn + cost);

    Console.WriteLine("Printing Cost " + prn + ".");
    Console.WriteLine("Gimbury tickets cost " + cost + ".");
    Console.WriteLine("Rest of the money is {0}.", rest);

    Console.ReadKey();

}

```

דוגמת הרצה מספר 1:

How many kids?

15

What is the price for ticket?

63.5

Printing Cost 142.875.

Gimbury tickets cost 171.45.

Rest of the money is 638.175.

דוגמת הרצה מספר 2:

How many kids?

20

What is the price for ticket?

26.9

Printing Cost 80.7.

Gimbury tickets cost 96.84.

Rest of the money is 360.46.

פרק 3:

פתח דבר

בפרק זה נלמד כיצד לכתוב תוכנית מחשב בה ביצוע הפקודות לא יהיה בהכרח סדרתי. חשוב שהתלמידים יבינו כי בהתאם לקלטים שונים אותם יכניסו המשתמשים יכולים להיות פלטים שונים. בנוסף, יש ללמד מהו ביטוי בוליאני ושעל בסיס התוצאה שלו תתבצענה פקודות שונות.

מבנה בקרה ראשון – if: התלמיד נדרשים לדעת כיצד לבנות או לסווג את בעבודה עם סוגי ה־ז'אנר השונים. אנו ממליצים להציג פתרונות שונים (מתלמידי הכיתה) ולעבור עליהם באמצעות טבלאות מעקב. גם מטבלאות שגויות ניתן ללמוד.

חשוב ללמד כיצד לעקוב אחר תרשים זרימה באמצעות טבלת מעקב או למצוא טעויות לוגיות לפני השלב של כתיבת תוכניות מחשב בשפת C#.

מבנה בקרה שני – switch: התלמידים נדרשים לדעת מהו ההבדל בין if (על סוגיו השונים) לבין switch.

כמו כן, יש לפתח מיומנויות כתיבה לגבי אפשרות משותפות.

If מקוצר: יש להבין את משמעותו. מתאים לביטויים קצרצרים בלבד.

התלמידים נדרשים לדעת לתרגם אלגוריתמים הכתוב באמצעות ז'אנר (על סוגיו השונים) ל – switch והפוך.

יש לחדד את ההבדלים בין הקשרים הלוגיים השונים. על המורה להציג בדומה לבדיקת ציון חוקי בפרק (באמצעות 3 דרכים) - בדיקת חוקיות של ערך מדיד אחת ולהראות יותר מדרך פתרון בודדת.

פתרונות עם משתנה בוליאני לשאלות של הפרק

- א.** כתבו תוכנית מחשב שתקלוט שלושה ציונים. התוכנית תחשב ותדפיס את ממוצע הציונים של התלמיד. התוכנית תדפיס (בצורה בוליאנית) האם התלמיד "עבר" או "נכשל".
- ב.** כתבו תוכנית מחשב הקולטת את מספר הבנים והבנות בכיתה. התוכנית תדפיס (בצורה בוליאנית) האם מספר הבנים שווה למספר הבנות.
- ג.** כתבו תוכנית מחשב שתקלוט מספר שלם ותדפיס (בצורה בוליאנית) האם הוא בין 40 (כולל) ל-65 (כולל).

ד. כתבו תוכנית מחשב שתקלוט שני מספרים שלמים ותדפיס (בצורה בוליאנית) האם ההפרש ביניהם גדול מ-10.

פתרונות:

א. התוכנית:

```
static void Main(string[] args)
{
    int m1, m2, m3; // ציונים שלוש
    double avg; // הציונים ממוצע
    bool pass; // בוליאני משתנה

    Console.WriteLine("Please enter 3 marks (as integer):");
    m1 = int.Parse(Console.ReadLine());
    m2 = int.Parse(Console.ReadLine());
    m3 = int.Parse(Console.ReadLine());
    avg = (m1 + m2 + m3) / (double)3; // Why cast?
    Console.WriteLine("Your average is {0}.", avg);
    pass = (avg >= 55);

    if (pass) // <=> if (pass==True)
        Console.WriteLine("Pass!");
    else
        Console.WriteLine("Fail!");

    Console.ReadKey();
}
```

דוגמת הרצה ראשונה:

```
Please enter 3 marks (as integer):
55
55
54
Your average is 54.6666666666667.
Fail!
```

דוגמת הרצה שנייה:

Please enter 3 marks (as integer):

82

17

94

Your average is 64.33333333333333.

Pass!

ב. התוכנית:

```
static void Main(string[] args)
{
    int boys, girls;

    Console.WriteLine("Please enter the number of boys:");
    boys = int.Parse(Console.ReadLine());
    Console.WriteLine("Please enter the number of girls:");
    girls = int.Parse(Console.ReadLine());

    Console.WriteLine("The number of boys equals to the
    number of girls:" + (boys == girls));

    Console.ReadKey();
}
```

דוגמת הרצה ראשונה:

Please enter the number of boys:

10

Please enter the number of girls:

12

The number of boys equals to the number of girls:False

דוגמת הרצה שנייה:

Please enter the number of boys:

14

Please enter the number of girls:

14

The number of boys equals to the number of girls:True

ג. התוכנית:

```

static void Main(string[] args)
{
    int num40_65;
    bool degel;

    Console.WriteLine("Please enter an integer value:");
    num40_65 = int.Parse(Console.ReadLine());

    degel = (num40_65 >= 40 && num40_65 <= 65); // Limits
    if (!degel) // <=> if (degel==False)
        Console.WriteLine(num40_65+ " is not between 40 and 65.");
    else
        Console.WriteLine(num40_65 + " is between 40 and 65.");

    Console.ReadKey();
}

```

דוגמת הרצה ראשונה:

Please enter an integer value:

40

40 is between 40 and 65.

דוגמת הרצה שנייה:

Please enter an integer value:

78

78is not between 40 and 65.

ד. התוכנית:

```

static void Main(string[] args)
{
    int num1,num2;
    bool distance;

    Console.WriteLine("Please enter 2 integer value:");
    num1 = int.Parse(Console.ReadLine());
    num2 = int.Parse(Console.ReadLine());

    distance = Math.Abs(num1 - num2) > 10;

    if (!distance) // <=> if (distance==False)
    Console.WriteLine(distance +
        " => the distance is not bigger than 10.");
    else
    Console.WriteLine(distance +
        " => the distance is bigger than 10.");

    Console.ReadKey();
}

```

דוגמת הרצה ראשונה:

Please enter 2 integer value:

10

20

False => the distance is not bigger than 10.

Please enter 2 integer value:

30

41

True => the distance is bigger than 10.

דוגמת הרצה שנייה:

Please enter 2 integer value:

19

40

True => the distance is bigger than 10.

דוגמת הרצה שלישית:

Please enter 2 integer value:

40

50

False => the distance is not bigger than 10.

פרק 4

פתח דבר

בפרק זה על המורה להסביר את בין סוגי הלולאות השונות בדגש מתי יש להשתמש בכל אחת מהן.

אלו הם הקווים המנחים המומלצים:

ניסוח מילולי א' – **"בצע מספר** מסוים של פעמים..." או

"חזור מספר מסוים של פעמים..."

ניסוח זה מרמז על שימוש בלולאת for.

ניסוח מילולי ב' – **"כל עוד...בצע"** – ניסוח זה מרמז על שימוש בלולאת while.

ניסוח מילולי ג' – **"בצע...כל עוד"** – ניסוח זה מרמז על שימוש

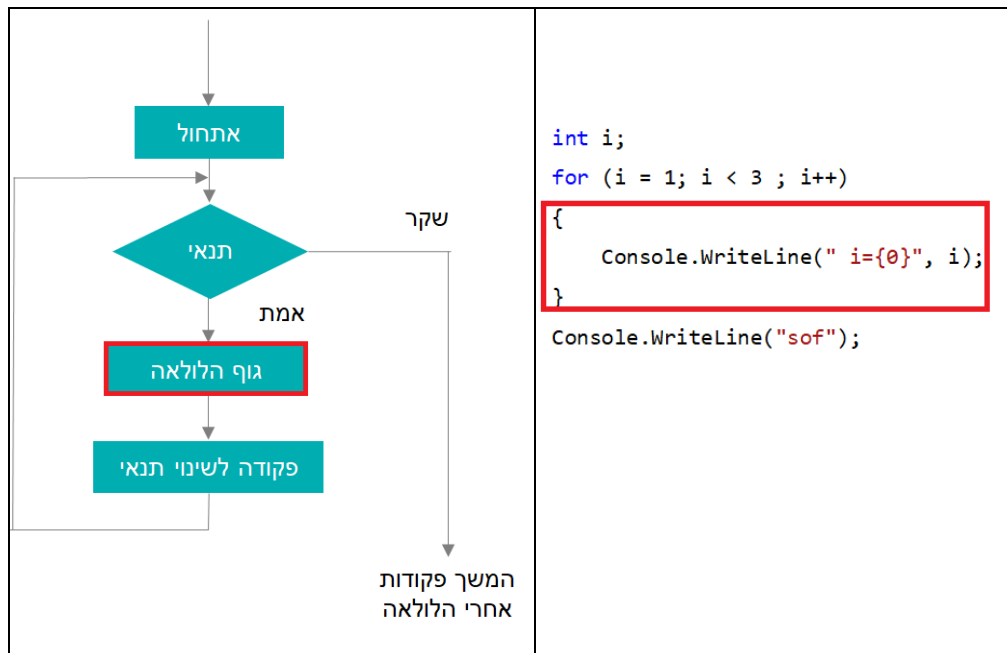
בלולאת do..while.

- על המורה להסביר את הקשר בין ביטוי בוליאני לבין תנאי קיום של כל אחת מהלולאות.
- על המורה להדגים כיצד לגשת לבעיה אלגוריתמית ולפתור אותה בשלבים מוגדרים.
- על המורה לוודא כי התלמידים מבינים את תרגום גוף הלולאה לתרשים זרימה מתאים.
- יש להציג ולתרגל דוגמאות המסבירות טעויות קלאסיות העלולות לגרום ללולאות אין-סופיות.

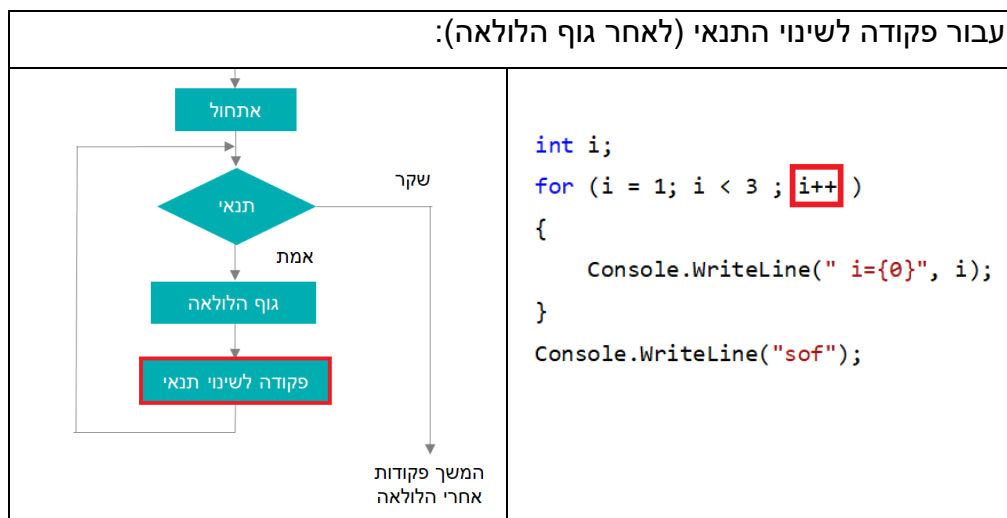
התייחסות ללולאת FOR:

הקשר בין תרשים זרימה לקוד מקור:

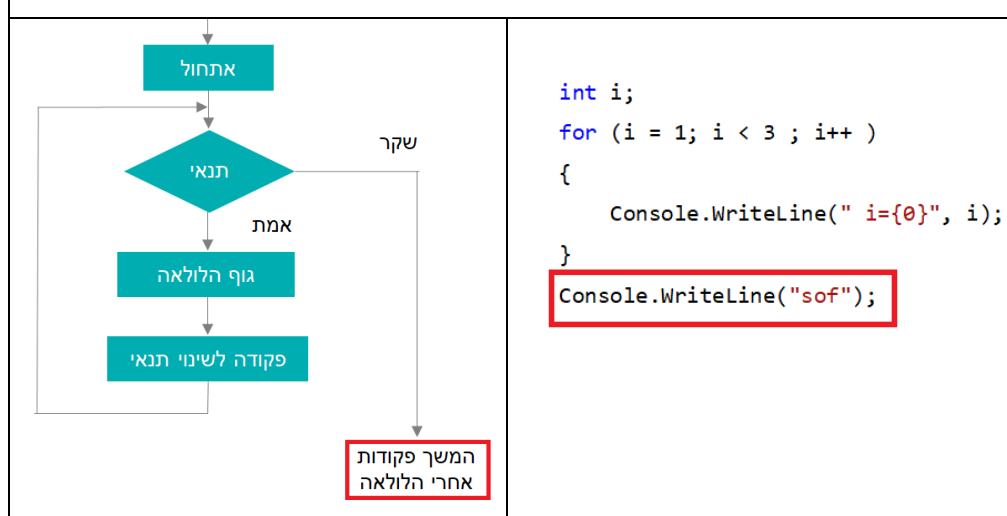
עבור שלב האתחול:	
<pre> graph TD Start(()) --> Init[אתחול] Init --> Cond{תנאי} Cond -- אמת --> Body[גוף הלולאה] Body --> Change[פקודה לשינוי תנאי] Change --> Cond Cond -- שקר --> End(()) </pre>	<pre> int i; for (i = 1; i < 3 ; i++) { Console.WriteLine(" i={0}", i); } Console.WriteLine("sof"); </pre>
עבור תנאי הקיום:	
<pre> graph TD Start(()) --> Init[אתחול] Init --> Cond{תנאי} Cond -- אמת --> Body[גוף הלולאה] Body --> Change[פקודה לשינוי תנאי] Change --> Cond Cond -- שקר --> End(()) </pre>	<pre> int i; for (i = 1; i < 3 ; i++) { Console.WriteLine(" i={0}", i); } Console.WriteLine("sof"); </pre>
עבור גוף הלולאה (כלומר תנאי הקיום החזיר – אמת):	



עבור פקודה לשינוי התנאי (לאחר גוף הלולאה):



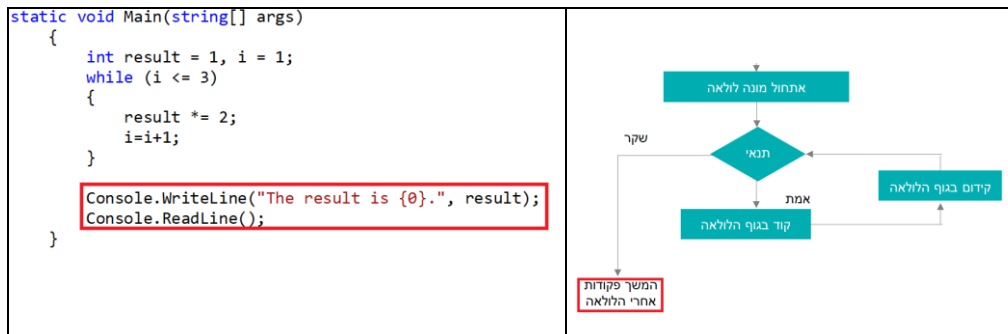
עבור המשך פקודות אחרי הלולאה



התייחסות ללולאת While:

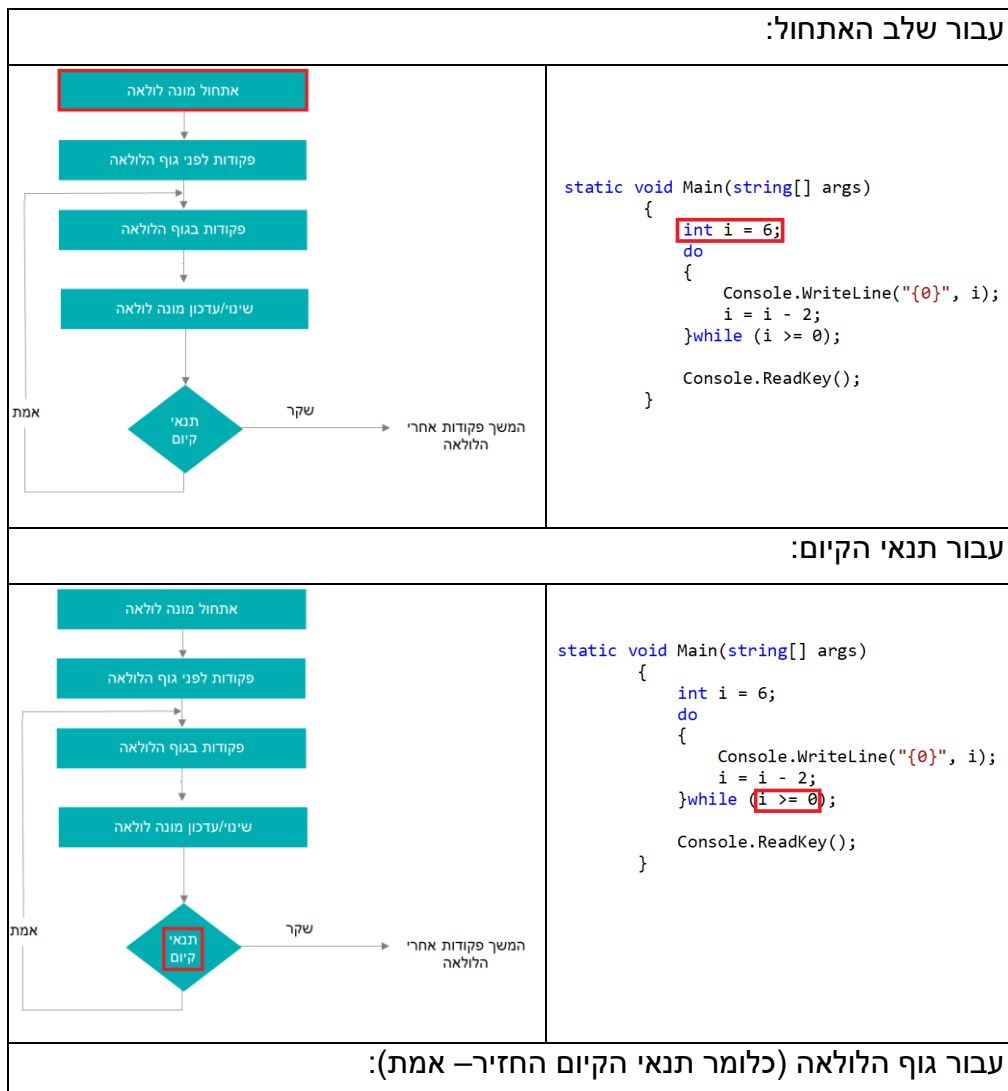
הקשר בין תרשים זרימה לקוד מקור:

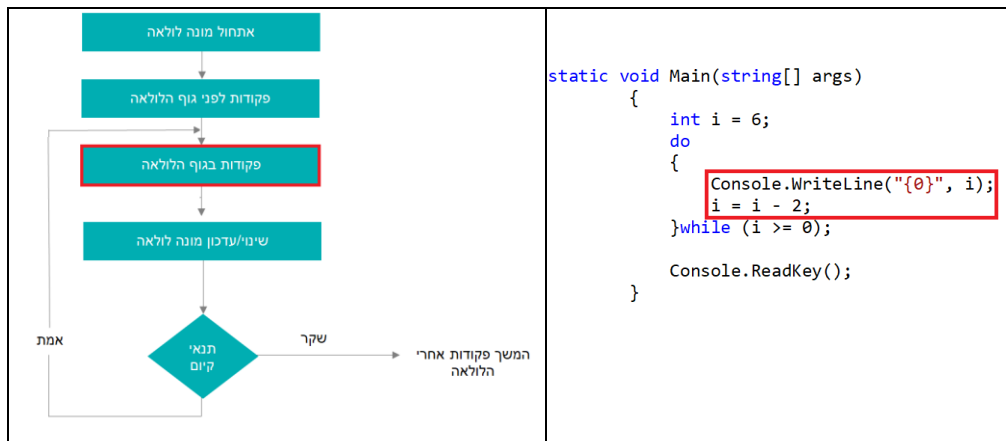
עבור שלב האתחול:	
<pre>static void Main(string[] args) { int result = 1, i = 1; while (i <= 3) { result *= 2; i=i+1; } Console.WriteLine("The result is {0}.", result); Console.ReadLine(); }</pre>	
עבור תנאי הקיום:	
<pre>static void Main(string[] args) { int result = 1, i = 1; while (i <= 3) { result *= 2; i=i+1; } Console.WriteLine("The result is {0}.", result); Console.ReadLine(); }</pre>	
עבור גוף הלולאה (כלומר תנאי הקיום החזיר – אמת):	
<pre>static void Main(string[] args) { int result = 1, i = 1; while (i <= 3) { result *= 2; i=i+1; } Console.WriteLine("The result is {0}.", result); Console.ReadLine(); }</pre>	
עבור פקודה לשינוי התנאי:	
<pre>static void Main(string[] args) { int result = 1, i = 1; while (i <= 3) { result *= 2; i=i+1; } Console.WriteLine("The result is {0}.", result); Console.ReadLine(); }</pre>	
עבור המשך פקודות אחרי הלולאה	



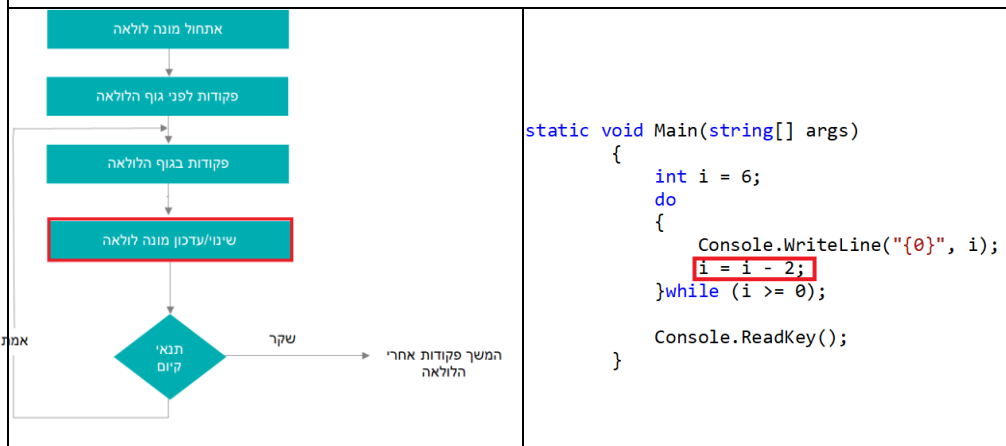
התייחסות ללולאת do..while:

הקשר בין תרשים זרימה לקוד מקור:

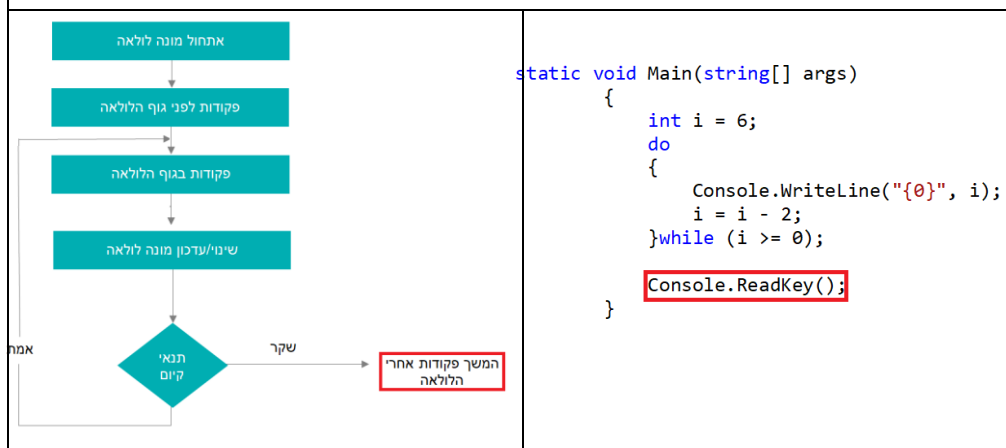




עבור פקודה לשינוי התנאי (לאחר גוף הלולאה):



עבור המשך פקודות אחרי הלולאה



עבודה עם תבניות ⇔ פיתוח מיומנויות לזיהוי דפוס מונוטוני של בעיות אלגוריתמיות:

תבנית מספר 1 – מציאת מינימום או מקסימום בקבוצה סדורה של ערכים.
 נניח כי קיימים 4 תלמידים בכיתה והמורה קולט (באמצעות אחת מ-3 הלולאות) את הציון שלהם. המטרה היא לשמור (=באמצעות משתנה) ולהציג את הציון הגבוה ביותר:

דוגמת פתרון:

```
static void Main(string[] args)
{
    int i = 1; // סידורי מספר של הציון
    int maxgrade=0; // ביותר הגבוה הציון
    int grade_toran; // הנוכחי הנקלט הציון
    int grades; // הציונים מספר
    Console.Write("How many grades? ");
    grades = int.Parse(Console.ReadLine());

    while (i <= grades)
    {
        Console.WriteLine("Please enter grade (number " + i + "):");
        grade_toran = int.Parse(Console.ReadLine());
        if (i == 1)
            maxgrade = grade_toran;
        else
            if (grade_toran > maxgrade) maxgrade = grade_toran;
        i++;
    }

    Console.WriteLine("The higher grade is: " + maxgrade + ".");

    Console.ReadKey();
}
```

דוגמת הרצה:

```
How many grades? 4
Please enter grade (number 1):
85
Please enter grade (number 2):
92
Please enter grade (number 3):
75
Please enter grade (number 4):
68
The higher grade is: 92.
```

תרגול לתלמידים: תרגם את התוכנית הנ"ל לתרשים זרימה ועקוב אחריה באמצעות טבלת מעקב.

תבנית מספר 2 – עבודה עם זקיף

בצורה דומה – שוב, נקלוט מספר לא ידוע מראש של ציונים. כל עוד הציון הנקלט אינו (-1) נמשיך ונקלוט.

אם קבלת הקלט (-1) תיעצר התוכנית ותדפיס את הציון הגבוה ביותר:

דוגמת פתרון:

```
static void Main(string[] args)
{
    int maxgrade=0; // ביותר הגבוה הציון
    int grade_toran; // הנוכחי הנקלט הציון
    Console.WriteLine("Please enter grade:");
    grade_toran = int.Parse(Console.ReadLine());
    while (grade_toran != (-1))
    {
        Console.WriteLine("Please enter grade:");
        grade_toran = int.Parse(Console.ReadLine());
        if (grade_toran > maxgrade) maxgrade = grade_toran;
    }

    Console.WriteLine("The higher grade is: "+ maxgrade+".");

    Console.ReadKey();
}
```

דוגמת ריצה:

```
Please enter grade:
30
Please enter grade:
90
Please enter grade:
66
Please enter grade:
3
Please enter grade:
-1
The higher grade is: 90.
```

תרגול לתלמידים: תרגם את התוכנית הנ"ל לתרשים זרימה ועקוב אחריה באמצעות טבלת מעקב.

תבנית מספר 3 – קיצוץ מספר באורך לא ידוע

נתון מספר, כאמור, באורך לא ידוע. נקלוט את המספר ובכל פעם נחלק את המספר ב-10 עד אשר ערכו יהיה אפס.
בכל שלב שכזה נדפיס את ספרת האחדות התורנית:

דוגמת פתרון:

```
static void Main(string[] args)
{
    int num;
    int digit;
    Console.WriteLine("Please enter number:");
    num = int.Parse(Console.ReadLine());
    do
    {
        digit = num % 10; // האחדות ספרת בידוד
        num = num / 10; // המספר קיצוץ
        Console.WriteLine("The digit is {0},  
the number is {1}.", digit, num);

        } while (num > 0); // מאפס גדול המספר עוד כל ובצע חזור
    Console.ReadKey();
}
```

דוגמת ריצה:

```
Please enter number:
48723
The digit is 3, the number is 4872.
The digit is 2, the number is 487.
The digit is 7, the number is 48.
The digit is 8, the number is 4.
The digit is 4, the number is 0.
```

שאלות:

- א. מה יש לשנות בתוכנית כך שיודפס מספר הספרות של המספר?
- ב. סכום הספרות?
- ג. סכום הספרות הזוגיות?
- ד. מה יקרה אם ננסה להדפיס את המספר הנקלט בסיום התוכנית?

פרק 5

פתח דבר

בפרק זה אנו נכנסים לעולם מופלא של תכנות מונחה עצמים. יש להסביר לתלמידים, באמצעות שימוש בדוגמאות רבות, כי בשונה ממשתנה "רגיל", כאשר אנו מדברים על אובייקט – אנו מדברים גם על "התנהגות" של האובייקט.

מהי מחלקה (Class)?

יש להסביר לתלמידים כי המחלקה היא "בית החרושת" לאובייקטים. המחלקה היא התבנית הכללית שממנה נגזרים אובייקטים שונים.

יש להסביר מהן תכונות מחלקה (בדומה למשתנים), מה הם הקווים המשותפים לכל האובייקטים של המחלקה ומה מייחד כל אובייקט (למשל, כל אחד נוצר בזמן אחר).

הרשאת תכונות:

- יש להסביר שלעיתים אפשר לגשת לנתונים של אובייקט שלא מבתוך המחלקה שלו. מצד שני, לעיתים זו "סכנת נפשות", ואין לאפשר זאת.
- יש להסביר כי גישה לתכונות פרטיות אפשרית באמצעות פונקציות מיוחדות של Setters-ו Getters.
- יש להסביר כי שינוי תכונה פרטית בעולם האמיתי מתבצע באמצעות קוד, כרטיס או התקן אבטחה מסוים.

שיטות:

יש להסביר, שוב – באמצעות דוגמאות שונות, כי השיטות / הפונקציות / המתודות – הן אלו המממשות (implements) את התנהגות האובייקט. לדוגמה, צפוצף או אור-אדום אם הרכב שלנו נוסע מעל המהירות המותרת, או השמעת אזעקה אם הטמפרטורה של החדר עלתה בדקות האחרונות בצורה מסוכנת.

פונקציות בונות:

יש להסביר שגם פונקציות בונות / בנאיות הן פונקציות לכל דבר. אך מה שמבדיל אותן הוא – שהן מייצרות אובייקט. ניסוח חלופי – אפשר לעבוד עם פונקציות "רגילות" (לא בנאים) רק אחרי שנוצר לפחות אובייקט בודד מהמחלקה/ות המסוימת/ות.

פונקציית מחלקה:

לתלמידים קשה להבין, לפחות בהתחלה, איך פונקציה עובדת "על" אובייקט. או אפשר להשוות בין 2 אובייקטים כאשר הפונקציה מקבלת אובייקט בודד. יש להסביר שכל פונקציית מחלקה – עובדת על מושא מסוים – כלומר אובייקט מסוג המחלקה.

אם הפונקציה אינה סטטית – הווה אומר – שהיא חייבת לעבוד על אובייקט מסוים.

משמעות הפקודה this:

← בתוך פונקציה משמש לגישה של תכונת מחלקה.

←בתוך בנאי ⇔ מפעיל למעשה בנאי אחר.

פונקציית השוואה:

```
public override bool Equals(Object obj)
```

כך נוכל להשוות בין אובייקטים על בסיס היגיון מסוים.

למשל נוכל להגדיר כי מלבן א' שווה למלבן ב' אם יש להם את אותו השטח.

נניח כי מלבן א' בנוי מ- 3 X 4 ואילו מלבן ב' בנוי מ- 6 X 2.

לפי ההיגיון הנ"ל – הם שווים.

עבודה עם פונקציית הדפסה:

```
public override string ToString()
```

כל נוכל להדפיס כל אובייקט קיים ע"י שימוש בפקודה:

```
Console.WriteLine
```

דוגמא לעבודה עם מחלקת מלבן:

```
class Rect
{
    private int length;
    private int width;

    public void SetWidth(int w)
    {
        this.width = w;
    }

    public int GetWidth()
    {
        return this.width;
    }

    public void Length(int l)
    {
        this.length = l;
    }

    public int GetLength()
    {
        return this.length;
    }

    public Rect(int length, int width)
    {
        this.length = length;
        this.width = width;
    }
}
```

```

public override string ToString() // (12)
{
    return string.Format("length={0} width={1}.\n", length, width);
}

public override bool Equals(Object obj) //(11)
{
    Rect PObj = obj as Rect;
    if (PObj == null)
        return false;
    if (this == PObj)
    {
        Console.WriteLine("Same Rect !");
        return (false);
    }
    return ((length * width) == (PObj.length * PObj.width));
}
}
class Program
{
    static void Main(string[] args)
    {
        Rect r1 = new Rect(6, 2);
        Rect r2 = new Rect(3, 4);
        Console.WriteLine("r1 is " + r1);
        Console.WriteLine("r2 is " + r2);
        Console.WriteLine(r1.Equals(r2));

        Console.ReadKey();
    }
}

```

פלט ההרצה:

r1 is length=6 width=2.

r2 is length=3 width=4.

True

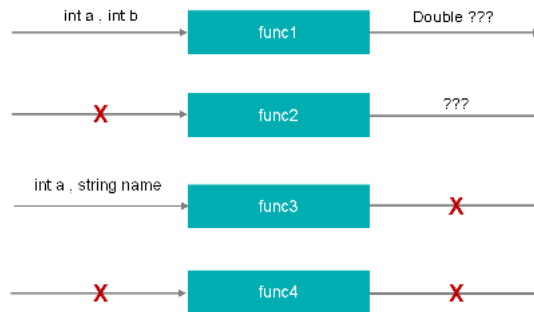
הצעה לתרגילים ב-OOP:

- א. הגדר את מחלקת Person:
- א. תכונות המחלקה:
 - שם פרטי, שם משפחה, שנת לידה ועיר מגורים.
 - ב. כתוב את השיטה ToString.
 - ג. אדם א' יהיה שווה (Equals) לאדם ב' אם מתקיימות שתי התכונות הבאות:
 - $\leq$ יש להם את אותה שנת לידה.
 - $\leq$ הם גרים באותה עיר.
 - ד. בנה 3 בנאים למחלקה.
- ב. הגדר את מחלקת Car:
- א. תכונות המחלקה:
 - מספר רישוי, מספר גלגלים, יצרן ומספר קילומטר שנסעו עד עתה.
 - ב. כתוב את השיטה ToString.
 - ג. רכב א' יהיה שווה (Equals) לרכב ב' אם מתקיימות שתי התכונות הבאות:
 - $\leq$ יש להם את אותו יצרן.
 - $\leq$ יש להם הפרש של 1000 או פחות קילומטר.
 - ד. בנה 3 בנאים למחלקה (אחד מהם חייב להיות בנאי העתקה).
- ג. הוסף למחלקת Car תכונה: מספר בעלים קודמים.
- ג.1. שנה בהתאמה – את פונ' ההדפסה ואת הבנאים השונים.
 - ג.2. שנה בהתאמה – את פונ' ההשוואה והוסף גם אם מספר הבעלים הקודם זהה.
- ד. הסבר במילים מה יש לשנות בשתי המחלקות שכתבת על מנת למנוע השוואה של אובייקט לעצמו?

פרק 6:

פתח דבר

חשוב להסביר מתי יש להשתמש בכל אחת מ-4 הסוגים של הפונקציות הבאות:



הפונקציות הן סטטיות. כלומר – אפשר לעבוד איתן גם אם לא יצרנו אובייקט מסוג מחלקה מסוימת והן לא עובדות על אובייקט.

הרשאת הפונקציה:

רוב הפונקציות שנכתוב יהיו ציבוריות (public) – כלומר יהיה אפשר להפעילן מחוץ למחלקה בהם הן נכתבו. נשתמש בפונקציה עבור חישובים שנעשה בהם שימוש חוזר. אפשרות אחרת – נשתמש בפונקציה על מנת לקצר הליכים ולהשתמש ב"כלי עבודה" קיים ולא נכתוב אחד כזה.

דוגמא: במחלקת Math יש לא מעט פונקציות מוכנות. פונקציה עשויה לקבל מספר ערכים אך לעולם תחזיר ערך בודד.

משתנים מקומיים:

משתנה מקומי הוא משתנה המוגדר בהצהרת המשתנים של הפונקציה. שאר התוכנית (כולל הפונקציה הראשית) לא מכירה את המשתנים האלו.

דוגמה ראשונה:

מה יהיה הפלט של הקוד הבא?

```
static void myFunc()
{
    int x = 6;
    Console.WriteLine(x);
    x++;
    Console.WriteLine(x);
}
static void Main(string[] args)
{
    int x = 10;
    Console.WriteLine(x);

    Console.ReadKey();
}
```

הסבר: הפלט יהיה 10. משתנה בשם x אמנם מוגדר בפונקציה בשם myFunc – אבל אין לו כל קשר למשתנה x בפונקציה Main. כמו כן – הפונקציה הראשית לא זימנה את הפונקציה myFunc(). דוגמה שנייה:

מה יהיה הפלט של הקוד הבא?

הפעם הפונקציה הראשית זימנה את הפונקציה myFunc().

```
static void myFunc()
{
    int x = 6;
    Console.WriteLine(x);
    x++;
    Console.WriteLine(x);
}
static void Main(string[] args)
{
    int x = 10;
    Console.WriteLine(x);
    myFunc();
    Console.WriteLine(x);
    Console.ReadKey();
}
```

פלט:	הסבר:
10	הפונקציה הראשונה שתופעל (תמיד!) היא פונקציית ה-Main .
6	בפקודה הראשונה בה נוצר משתנה לוקלי ומקבל את הערך 10.
7	בשורה השנייה ערכו מודפס.
10	בשורה השלישית ב-Main אנו מזמנים את הפונקציה myFunc(). גם היא יוצרת משתנה לוקלי בשם x – אבל אין לו כל קשר למשתנה x הקיים בפונקציה הראשית. הם נמצאים בכתובות אחרות לגמרי. בשורה הראשונה (בפונקציה myFunc) נוצר, כאמור, המשתנה ומאותחל. בשורה השנייה הוא מודפס. אח"כ הוא מקודם ושוב מודפס. בשלב הזה (עדיין בתוך הפונקציה myFunc) אין כל אפשרות לגשת למשתנה x של הפונקציה הראשית. הפונקציה myFunc הסתיימה וכעת אנו חוזרים לפונקציה הראשית. שוב – משתנה x המקורי נגיש. הוא מודפס שוב והיות ומדובר באותו משתנה (השוכן באותה הכתובת) שוב מודפס 10.

הדפסה של ערך חוזר מהפונקציה:

```
public static string GetTime()
{
    DateTime now = DateTime.Now;

    string output = "";
    output += now.Hour;
    output += ":";
    output += now.Minute;
    output += ":";
    output += now.Second;

    return (output);
}

static void Main(string[] args)
{
    Console.WriteLine(GetTime());
    Console.ReadKey();
}
```

א. שימו לב כי הזמן חוזר לפונקציה הראשית בפורמט(צורה) של מחרוזת.

ב. מה אפשר לעשות על מנת לתקן צורות הדפסה שהתקבלו כך:

23:59:2

0:0:1

המטרה היא שהן יוצגו כך:

23:59:02

00:00:01

מקבץ תרגילים – עבור כל סעיף בנה פונקציה:

א.	מקבלת מספר שעות עבודה לעובד ותעריך לשעה ומחזירה את סך המשכורת.
ב.	מקבלת גובה ובסיס של משולש – ומחזירה את שטחו.
ג.	מקבלת מספר – ומחזירה (bool) – האם הוא ראשוני או לא.
ד.	מקבלת תו (char) – ומחזירה מחרוזת מהתו 'a' עד התו הנקלט.
ה.	המקבלת מספר שלם ומחזירה את מספר ספרותיו.
ו.	פונקציה המגרילה מספר אקראי בין 1 ל-6 (קוביה). את הערך המוחזר יש להדפיס.

פרק 7

פתח דבר

יש להסביר לתלמידים כי מחרוזת היא למעשה סוג של מחלקה. שניצור מחרוזת, ניצור למעשה אובייקט (או עצם/מופע) מטיפוס `string`. יצירת האובייקט כוללת הקצאת מקום בזיכרון (Allocation), הצבת ערך והצבעה על המקום שנוצר. יש להסביר כי כל הפעולות שהוצגו בפרק – מקצות למעשה מחרוזת חדשה ולא עובדות על העצם (מחרוזת) המקורית. בסיום תוכניות על מנת לעכב את הפלט נהגנו לרשום את הפקודה הבאה:

```
Console.ReadLine();
```

מה המשמעות שלה?

המשמעות שלה היא קליטת מחרוזת. שימו לב שאין עבודה עם `Parse` היות ואין צורך בהמרה. כל קלט הוא לפני הכל מחרוזת.

אפשר היה לכתוב כך:

```
static void Main(string[] args)
{
    string s;

    s=Console.ReadLine(); // (1)
    Console.WriteLine(s); // (2)
    Console.ReadLine();
}
```

הפלט הוא:

dani

dani

בשורה (1) אנחנו, כאמור, קולטים מחרוזת.

בשורה (2) אנו מדפיסים אותה.

משימות נוספות לתלמידים:

1. לספור כמה פעמים הופיע תו מסוים במחרוזת.
2. כמה אותיות ניקוד ('a', 'e', 'i', 'o', 'u') הופיעו במחרוזת.
3. קליטת מחרוזת (באמצעות פונקציה) והחזרת מספר המילים המופיעים בה.
4. קליטת מחרוזת המייצגת דוא"ל (באמצעות פונקציה) והחזרת ערך בוליאני –

5. האם מדובר בemail חוקי.

6. "כתב סתרים" – קלוט מחרוזת והחלף תווים נבחרים. נניח האות 'a' תוחלף באות 'q'.

יש להסביר את המונח שרשור. יש להסביר כי הפעולה הזו יוצרת מחרוזת חדשה (בדומה לפעולות האחרות) ומקצה לה מקום בRAM. שרשור מתבצע כאשר מחברים למחרוזת ערך שהוא לא מחרוזת ואז הוא עובר המרה למחרוזת (לטקסט). שרשור של תו למחרוזת:

```
static void Main(string[] args)
{
    string s;
    char a = 'f';

    s=Console.ReadLine();
    Console.WriteLine(s);
    s = s + a;
    Console.WriteLine(s);
    Console.ReadLine();
}
```

הפלט:

```
yose
yose
yosef
```

פרק 8

פתח דבר

בפרק זה יש להדגיש כי למערך יש אוסף של איברים מאותו הסוג. למערך יש שם בודד (ולא אוסף של משתנים) ואפשר לבצע עליו מניפולציות באמצעות ציון index. התלמיד נדרשים לדעת מערך של טיפוסים אטומיים (כגון int או double) וכן מערך של טיפוסים מופשטים (כגון Point). התלמידים נדרשים לדעת להקצות מערך בגודל של משתנה

```
class Program
{
    static void Main(string[] args)
    {
        const int SIZE = 5;
        int i;
        int []arr = new int[SIZE];
        double avg = 0;

        for (i=0 ; i<SIZE; i++)
        {
            Console.WriteLine("Please enter an integer value");
            arr[i] = int.Parse(Console.ReadLine());
            avg += arr[i];

        }
        avg = avg / (double)SIZE;
        Console.WriteLine("avg is: "+avg);
        Console.ReadLine();
    }
}
```

נקלט או קבוע.

דוגמה עבור קבוע:

הפלט:

Please enter an integer value

37

Please enter an integer value

33

Please enter an integer value

98

Please enter an integer value

47

Please enter an integer value

83

avg is: 59.6

על בסיס התבניות במדריך למורה פרק 4:

ד.1. – יש למצוא ולהדפיס את הערך הגבוה ביותר במערך. (תבנית א')

ד.2. – עבודה עם זקיף – קליטת מספר איברים (עד SIZE) ובכל המקומות האחרים נניח 0.

(תבנית ב')

עבור הקלטים הבאים (משמאל לימין) 4,2,5,-1 המערך ייראה כך:

0 1 2 3 4

[4] [2] [5] [0] [0]

ד.3. – פירוק של מספר לתוך מערך. (תבנית ג')

דוגמה לתבנית ד.3.:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace May5_2019
{
    class Program
    {

        static void Show_Array(int[] a)
        {
            int i;
            Console.WriteLine("The array is: ");
            for (i = 0; i < a.Length; i++)
                Console.Write("{0,-5}", a[i]);
            Console.WriteLine();
        }

        static void Main(string[] args)
        {
            int i;
            int size=0; // ידוע לא עוד המספר המרכיבות הספרות מספר
            int num;
            int copy; // הנקלט המספר של גיבוי
            int[] arr; // ידוע לא עוד המערך גודל
            Console.WriteLine("Enter an integer number");
            num = int.Parse(Console.ReadLine());
            copy = num;

            do
            {
                size++;
                copy /= 10;
            } while (copy != 0);

            //Console.WriteLine("size is "+size);
            arr = new int[size]; // הספרות מספר פי על מערך הקצאת
            copy = num; // הלולאה בתוך מקודם אופס הוא
            for (i=0; copy!=0; i++)
            {
                // Console.WriteLine(copy % 10+ " i is "+i);
                arr[i] = copy % 10;
                copy /= 10;
                // Console.WriteLine(arr[i]);
            }
            Show_Array(arr);

            Console.ReadLine();
        }
    }
}

```

פלט לדוגמה:

Enter an integer number

1874

The array is:

4 7 8 1

פעולת אתגר:

נסו ליצור שני מספרים חדשים על בסיס ספרות זוגיות ואי זוגיות.

למשל עבור הקלט הנ"ל יוצרו שני המספרים הבאים:

17

84

קיצוץ באורך לא ידוע

נתון מספר, כאמור, באורך לא ידוע. נקלוט את המספר ובכל פעם נחלק את המספר ב-10

עד אשר ערכו יהיה אפס.

בכל שלב שכזה נדפיס את ספרת האחדות התורנית:

דוגמת פתרון:

```
static void Main(string[] args)
{
    int num;
    int digit;
    Console.WriteLine("Please enter number:");
    num = int.Parse(Console.ReadLine());
    do
    {
        digit = num % 10; // האחדות ספרת בידוד
        num = num / 10; // המספר קיצוץ
        Console.WriteLine("The digit is {0},\nthe number is {1}.", digit, num);
    } while (num > 0); // מאפס גדול המספר עוד כל ובצע חזור
    Console.ReadKey();
}
```

דוגמת ריצה:

Please enter number:

48723

The digit is 3, the number is 4872.

The digit is 2, the number is 487.

The digit is 7, the number is 48.

The digit is 8, the number is 4.

The digit is 4, the number is 0.

שאלות:

א. מה יש לשנות בתוכנית כך שיודפס מספר הספרות של המספר?

ב. סכום הספרות?

ג. סכום הספרות הזוגיות?

ד. מה יקרה אם ננסה להדפיס את המספר הנקלט בסיום התוכנית?

פרק 9

פתח דבר

יש להסביר לתלמידים שכל מחלקה הינה מחלקה עצמאית. כלומר, הקיום שלה לא מותנה במחלקה אחרת. מומלץ לבנות מחלקה אחת מחלקה – ורק אחר כך לשלב בין הדברים.

הצעה למשימות (עם פתרונות) מתגלגלות:

שלב 1:

א.	בנה מחלקת "כתובת" (<code>Address</code>) אשר מורכבת מהתכונות הפרטיות הבאות: עיר (<code>city</code>), רחוב (<code>street</code>) ומספר בית (<code>numOfHouse</code>).
ב.	עבור כל תכונה פרטית – יש לכתוב פונקציית <code>Set</code> וכן פונקציית <code>Get</code> .
ג.	בנה פונקציה בונה (<code>Constructor</code>) מפורש (המקבל את כל התכונות). שים לב כי ההרשאה שלו חייבת להיות ציבורית (<code>public</code>). מדוע?! אם הפונקציה הבונה תהיה פרטית – אי אפשר יהיה ליצור אובייקט מהפונקציה הראשית – הרי היא לא באותה מחלקה.
ד.	בנה פונקציית הדפסה (<code>ToString</code>) על מנת להדפיס (בצורה ברורה) את נתוני הכתובת.
ה.	בפונקציה הראשית צור אובייקט מסוג "כתובת" עם התכונות הבאות: עיר "ירושלים", רחוב "הרצל", מספר בית – 44 והדפס את נתוני האובייקט.

פתרון:

```

class Address
{
    private string city;
    private string street;
    private int numOfHouse;

    public string GetCity()
    {
        return this.city;
    }

    public void SetCity(string s)
    {
        this.city=s;
    }

    public string GetStreet()
    {
        return this.street;
    }

    public void SetStreet(string s)
    {
        this.street = s;
    }

    public int GetNumOfHouse()
    {
        return this.numOfHouse;
    }

    public void SetNumOfHouse(int n)
    {
        this.numOfHouse = n;
    }

    public Address(string c, string s, int num)
    {
        this.city = c;
        this.street = s;
        this.numOfHouse = num;
    }

    public override string ToString()
    {
        return string.Format("\tAddress: city is {0} street is {1} numOfHouse is {2}.", city, street, numOfHouse);
    }
}

static void Main(string[] args)
{
    Address ard1 = new Address("Jerusalem", "Herzel", 44);
    Console.WriteLine(ard1);
}

```

הפלט של ההרצה הזו:

Address: city is Jerusalem street is Herzel numOfHouse is 44.

שלב 2:

א.	בנה מחלקת "סטודנט" (<i>Student</i>) אשר מורכבת מהתכונות הפרטיות הבאות: ת.ז. (<i>id</i>), שם (<i>name</i>), גיל (<i>age</i>) וכתובת (<i>Address</i>).
ב.	כל ארבעת התכונות פרטיות – אך אין לבנות להן פונקציות <i>Setters</i> ו- <i>Getters</i> .
ג.	בנה פונקציה בונה (<i>Constructor</i>) מפורש (המקבל את כל התכונות).
ד.	בנה פונקציית הדפסה (<i>ToString</i>) על מנת להדפיס (בצורה ברורה) את נתוני הסטודנט. שים לב שהיא אמורה לזמן את פונקציית ההדפסה הקודמת (של <i>Address</i>).
ה.	בפונקציה הראשית הוסף את שורות הקוד הבאות: <pre>Address ard1 = new Address("Jerusalem", "Herzel", 44); Console.WriteLine(ard1); Student std1 = new Student(123, "Alex", 22, ard1); Console.WriteLine(std1); Student std2 = new Student(456, "Eran", 27, new Address("Tel_Aviv", "Dan", 27)); Console.WriteLine(std2);</pre>
ו.	בדיקת הפלט: הפלט שאמור להופיע הוא: Address: city is Jerusalem street is Herzel numOfHouse is 44. Student consist of: 123 Alex 22 Address: city is Jerusalem street is Herzel

numOfHouse is 44.	
Student consist of: 456 Eran 27 Address: city is Tel_Aviv street is Dan numOfHouse is 27.	

פתרון:

```

using System;

namespace ConsoleApp4
{
    class Address
    {
        private string city;
        private string street;
        private int numOfHouse;

        public string GetCity()
        {
            return this.city;
        }

        public void SetCity(string s)
        {
            this.city=s;
        }

        public string GetStreet()
        {
            return this.street;
        }

        public void SetStreet(string s)
        {
            this.street = s;
        }

        public int GetNumOfHouse()
        {
            return this.numOfHouse;
        }

        public void SetNumOfHouse(int n)
        {
            this.numOfHouse = n;
        }

        public Address(string c, string s, int num)
        {
            this.city = c;
            this.street = s;
            this.numOfHouse = num;
        }

        public override string ToString()
        {
            return string.Format("\tAddress: city is {0} street is {1} numOfHouse is {2}.", city, street,
numOfHouse);
        }
    }
}

```

```

class Student
{
    private long id;
    private string name;
    private int age;
    private Address address;

    public Student(long i, string n, int a, Address adr)
    {
        id = i;
        name = n;
        age = a;
        this.address = adr;
    }

    public override string ToString()
    {
        return string.Format("Student consist of: " + id + " " + name + " " +
            " " + age + address.ToString());
    }
}

class Program
{
    static void Main(string[] args)
    {
        Address ard1 = new Address("Jerusalem", "Herzel", 44);
        Console.WriteLine(ard1);

        Student std1 = new Student(123, "Alex", 22, ard1);
        Console.WriteLine(std1);

        Student std2 = new Student(456, "Eran", 27,
            new Address("Tel_Aviv", "Dan", 27));
        Console.WriteLine(std2);

        Console.ReadLine();
    }
}

```

שלב 3:

נבנה פונקציית השוואה למחלקת "כתובת".

כתובת א' שווה לכתובת ב' אם ורק אם ההפרש בין מספרי הבתים (בערך מוחלט) הוא מקסימום 10.

שאלה: כיצד תראה הפונקציה?

תשובה:

```
public override bool Equals(Object obj)
{
    Address PObj = obj as Address;
    if (PObj == null)
        return false;
    if (this == PObj)
    {
        Console.WriteLine("same Address!");
        return (false);
    }
    return (Math.Abs(numOfHouse-PObj.numOfHouse)<=10);
}
```

שאלה: מה יהיה הפלט עבור כל אחת מהשורות הבאות ומדוע?

Console.WriteLine(ard1.Equals(ard1));	פלט: same Address! False הסבר: ההשוואה היא על אותו אובייקט (ard1) ולכן יש הודעה שגיאה והחזרת "שקר".
Console.WriteLine(ard2.Equals(ard1));	פלט: True הסבר: מספר הבית של האובייקט עליו אנו עובדים הוא 40 ושל האובייקט המועבר הוא 31. ההפרש הוא 9. 9 (גם בערך מוחלט) קטן או שווה ל10.

<pre>Console.WriteLine(ard1.Equals(ard2));</pre>	פלט: True הסבר: מספר הבית של האובייקט עליו אנו עובדים הוא 40 ושל האובייקט המועבר הוא 31. ההפרש הוא 9-. 9 (גם בערך מוחלט) קטן או שווה ל10.
<pre>Console.WriteLine(ard1.Equals(ard3));</pre>	פלט: False הסבר: ההפרש בין מספרי הבית הוא 35-31 פחות 66 זה החישוב. 35 (בערך מוחלט) הוא גדול 10 ולכן יוחזר "שקר".

שלב 4:

מה יהיה הפלט של קטע הקוד הבא (בהמשך לקוד הקודם)?

```
Student std1 = new Student(123, "Alex", 22, ard1);
Student std2 = new Student(456, "Dana", 23, ard1);
Console.WriteLine(std2.Equals(std1));
```

תשובה:

חשוב להבין (!): למרות שאלכס ודנה שופתים לאותה דירה (הם גרים באותה כתובת)

התשובה היא False.

מדוע?

היות וכעת המהדר מחפש את פונקציית ההשוואה (Equals) של Student. היא אינה קיימת
ולכן מופעלת Equals של המחלקה הקדומה Object. הפונקציה של ההשוואה לגבי כתובת –
לא עובדת.

פונקציית ההשוואה (Equals) של היא עובדת רק לפי כתובות ולא ערכים.

שלב 5:

מה יהיה הפלט של הקוד הבא?

```
Console.WriteLine(std2.Equals(std2));
```

הפלט יהיה True מפני שמדובר באותה כתובת של אובייקט (std2) בזיכרון.

שלב 6:

איך נפתור את הבעיה הזו?

נכתוב את פונקציית השוואה הפעם עבור סטודנט:

```
public override bool Equals(Object obj)
{
    int diss;
    Student PObj = obj as Student;
    if (PObj == null)
        return false;
    if (this == PObj)
    {
        Console.WriteLine("same Student (Student of Address)!");
        return (false);
    }
    diss = Math.Abs(this.address.GetNumOfHouse() - PObj.address.GetNumOfHouse());
    Console.WriteLine("*** Equals of Student ***");

    return (diss<=10);
}
```

פרק 10

פתח דבר

אוסף (Collection) מאגד בתוכו לא מעט סוגים של טיפוסים נתונים מופשטים. אנחנו נתמקד (בהתאם לתוכנית הלימודים) על רשימה בלבד. הפונקציות המובנות המתוארות בפרק (כגון מיון) עובדות רק על רשימה מטיפוס אטומי – כלומר מובנה בשפה. אנו נחזור לעבוד עם מחלקת נקודה (Point) כדוגמא למחלקה לא אטומית. מחלקת Point (הופיעה, כאמור, בפרק מספר 5):

```
class Point
{
    public Point(Point Other)
    {
        x = Other.x;
        this.y = Other.GetY();
    }

    public override bool Equals(Object obj)
    {
        Point PObj = obj as Point;
        if (PObj == null)
            return false;
        if (this == PObj)
        {
            Console.WriteLine("Same Point !");
            return (false);
        }
        return (x.Equals(PObj.x) &&
            y.Equals(PObj.y));
    }

    public override string ToString()
    {
        return string.Format("x={0} y={1}\n", x, y);
    }
}

// ~~~~

public Point(int i): this(i, 300)
{
}

public Point(int x, int y)
{
    this.x = x;
    this.y = y;
}
```

נשנה את הפונקציה של ההצגה מרשימה של int לרשימה של Point:

```
static void showList(List<Point> ls)
{
    Console.WriteLine("Capacity: {0}",
                      ls.Capacity);
    Console.WriteLine("Count: {0}", ls.Count);
    Console.WriteLine("List:");
    foreach (Point item in ls)
    {
        Console.Write(item + " ");
    }
    Console.WriteLine();
}
```

נבנה רשימה של נקודות:

```
static void Main(string[] args)
{
    Point a = new Point(3, 5);
    Point b = new Point(4);
    List<Point> lst1 = new List<Point>();
    lst1.Add(new Point()); lst1.Add(new Point(5,3));
    lst1.Add(b); lst1.Add(a);

    showList(lst1);

    lst1.Sort();
    showList(lst1);

    Console.ReadLine();
}
```

מה תוצאת ההרצה?

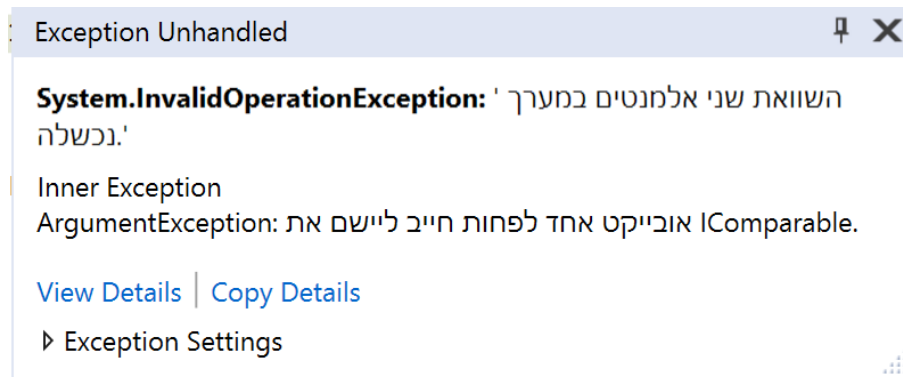
תשובה – שגיאת קומפילציה! מדוע?

המהדר יודע לבצע מניפולציות מובנות (כגון מיון) רק על רשימה אטומית.

השורה הזו:

```
lst1.Sort();
```

מהווה שגיאה:



הבעיה היא שהמהדר לא יודע להשוות (הפעולה "גדול" או "קטן") על בסיס יותר מתכונה אחת.

על מנת לפתור בעיה זו יש לבצע את השלבים:

א.	יש לשנות את כותרת מחלקת "נקודה" לצורה הבאה: <code>class Point: IComparable</code> מה המשמעות? המשמעות היא שמחלקת "נקודה" מקבלת בסיס מסוים – בעל תוכנה אחת יחידה עליו יבצע המיון העתידי. בשלב זה עלינו להחליט לפי מה יש למיין.
ב.	בשלב זה נחליט למיין את רשימת הנקודות שלנו לפי ציר ה-x.
ג.	יש לממש את הפונקציה <code>CompareTo</code> (בדומה להיגיון שמימשנו את <code>Equals</code>).
ד.	נסתכל על גוף הפונקציה (מימוש של מיון עולה): <code>public int CompareTo(object obj)</code> <code>{</code> <code>Point p = (Point)obj; // המרה כלפי מטה</code> <code>if (x < p.x) // התייחסות לתכונת ה-x</code> <code>return -1;</code> <code>if (x > p.x) // התייחסות לתכונת ה-x</code>

<pre>return 1; return 0; } }</pre>	
ה. פלט ההרצה: <pre>Capacity: 4 Count: 4 List: x=512 y=320 x=5 y=3 x=4 y=300 x=3 y=5 Capacity: 4 Count: 4 List: x=3 y=5 x=4 y=300 x=5 y=3 x=512 y=320</pre>	
שימו לב כי בהדפסה השניה (באדום) – הנקודות אכן ממיינות לפני ערך ה-x שלהן.	
ו. אם נרצה לבצע מיון יורד ⇔ יש להפוך את הסימנים גדול וקטן בפונקציה: <pre>public int CompareTo(object obj) { Point p = (Point)obj; if (x > p.x) return -1; if (x < p.x) return 1; return 0; }</pre>	
ז. ואז הפלט יהיה בסדר יורד (לפי ציר ה-x):	

```
Capacity: 4  
Count: 4  
List:  
x=512 y=320  
  x=5 y=3  
  x=4 y=300  
  x=3 y=5
```

```
Capacity: 4  
Count: 4  
List:  
x=512 y=320  
  x=5 y=3  
  x=4 y=300  
  x=3 y=5
```

דוגמת קוד כוללת (מיון עולה לפי ציר ה-א):

```
using System.Collections.Generic; // לשכוח לא
using System;

namespace ConsoleApp4
{
    class Point: IComparable
    {
        private int x;
        private int y;

        public int GetX()
        {
            return this.x;
        }

        public void SetX(int val)
        {
            x = val;
        }

        public int GetY()
        {
            return this.y;
        }

        public void SetY(int val)
        {
            this.y = val;
        }

        public Point()
        {
            this.x = 512;
            y = 320;
        }

        public Point(int i): this(i, 300)
        {
        }

        public Point(int x, int y)
        {
            this.x = x;
            this.y = y;
        }

        public Point(Point Other)
        {
            x = Other.x;
            this.y = Other.GetY();
        }
    }
}
```

```
public override bool Equals(Object obj)
{
    Point PObj = obj as Point;
    if (PObj == null)
        return false;
    if (this == PObj)
    {
        Console.WriteLine("Same Point !");
        return (false);
    }
    return (x.Equals(PObj.x) &&
        y.Equals(PObj.y));
}

public override string ToString()
{
    return string.Format("x={0} y={1}\n", x, y);
}

public int CompareTo(object obj)
{
    Point p = (Point)obj;

    if (x < p.x)
        return -1;

    if (x > p.x)
        return 1;

    return 0;
}
}

class Program
{
    static void showList(List<Point> ls)
    {
        Console.WriteLine("Capacity: {0}",
            ls.Capacity);
        Console.WriteLine("Count: {0}", ls.Count);
        Console.WriteLine("List:");
        foreach (Point item in ls)
        {
            Console.Write(item + " ");
        }
        Console.WriteLine();
    }
}
```

```

static void Main(string[] args)
{
    Point a = new Point(3, 5);
    Point b = new Point(4);
    List<Point> lst1 = new List<Point>();
    lst1.Add(new Point()); lst1.Add(new Point(5,3));
    lst1.Add(b); lst1.Add(a);

    showList(lst1);

    lst1.Sort();

    showList(lst1);

    Console.ReadLine();
}
}
}

```

תרגילים למתקדמים: מבוסס על פרק מספר 5

א.	בנה רשימה של אובייקטים מטיפוס "רכב" (Vehicle).
	א.1. מיין אותם לפי מהירות מירבית. א.2. צור רשימה חדשה של רכבים המסוגלים לנסוע עד 100 קמ"ש. א.3. צור רשימה של רכבים מתוצרת יפנית. רמז – האם יש לנו פונקציה מוכנה במחלקה?
ב.	בנה רשימה של מלבנים (Rect) ומיין אותם בצורה יורדת לפי שטחם.
ג.	בנה רשימה של אנשים (Person) ומיין אותם בצורה לפי שנת לידה ואח"כ שוב מיין אותם לפי עיר מגורים.

פרק 11

פתח דבר

בפרק של המטריצות מומלץ להריץ ולתרגל את כל המניפולציות בנושא. יש להבין ולתרגל כיצד בנייה של מטריצה (לפי שורות ועמודות, מערך של מצביעים וכן מטריצה לא סימטרית). אפשר לחפש מופעים או "תבניות" בתוך מטריצה. למשל, אפשר לבנות פונקציה המקבלת מטריצה ומחפשת "פרחים". בדוגמא הזו תחזיר הפונקציה את הערך 2:

	0	1	2	3	4
0	0	1	0	0	0
1	1	1	1	0	0
2	0	1	0	1	0
3	0	0	1	1	1
4	0	0	0	1	0

שימו כי החיפוש יתבצע מהאינדקס [1,1] ועד יתבצע [3,3]. מדוע?

אם '1' מופיע 5 פעמים ברצף אזי מדובר ב"פרח".

מהם האינדקסים המבוקשים?

פתרון:

```
using System.Collections.Generic; // לשכוח לא
using System;

namespace ConsoleApp4
{
    class Program
    {
        static int Find_Flower(int[,] mat)
        {
            int rowCount = mat.GetLength(0); // השורות מספר
            int colCount = mat.GetLength(1); // העמודות מספר
            int i; // מונה שורה
            int j; // מונה עמודה
            int mone_f = 0; // מונה פרחים
            int mone = 0; // מונה מופעים של התו '1' המבוקש
            for (i = 1; i < rowCount - 1; i++)
            for (j = 1; j < colCount - 1; mone = 0, j++)
            {
                if (mat[i, j] == 1) // אם מדובר במרכז 'פרח'
                {
                    mone++;
                    if (mat[i - 1, j] == 1) mone++;
                    if (mat[i, j - 1] == 1) mone++;
                    if (mat[i, j + 1] == 1) mone++;
                    if (mat[i + 1, j] == 1) mone++;
                    if (mone == 5) mone_f++; // if "flower" has found
                }
                mone = 0; // אפס את מספר מופעי ה-'1' עבור החיפוש הבא
            }
        }
    }
}
```

```

    }
    }
    return (mone_f); // החזר את מספר ה'פרחים' שנמצאו
}

static void Main(string[] args)
{
    int i, j;
    int[,] game = new int[5, 5]
    {
        {0,1,0,0,0},
        {1,1,1,0,0},
        {0,1,0,1,0},
        {0,0,1,1,1},
        {0,0,0,1,0}
    };

    Console.WriteLine(Find_Flower(game));
    Console.ReadLine();
}
}

```

לגבי מיון, מומלץ:

- א. לבצע מיון בועות על מערך של נקודות לפי אחד הצירים.
- ב. לבצע מיון הכנסה על מערך של מלבנים לפי השטח.
- ג. להגדיר מערך של אובייקטים (בתוך המחלקה) יחד עם משתנה בוליאני שיגדיר האם המערך ממוין (לפי היגיון מסוים).
- ד. לבצע חיפוש בינארי (בהמשך בסעיף הקודם) על מערך של מלבנים.
- ה. למזג מערכים של "עובדים" (אחרי שהם ממוינים כמובן).