

## מיתוג ומערכות ספרתיות תרגיל 5 (תרגיל גדול).

משחק החיים מוגדר כך:

### משחק החיים

תהי מטריצה  $n \times m$ . כל תא יכול להיות חי או מת.  
לכל תא (חוץ מהקצוות) ישנם 8 שכנים

1	2	3
8	*	4
7	6	5

בהינתן דור  $y$  התא בדור  $y+1$  יהיה חי אם מספר השכנים שלו נע בין  $\min$  ל  $\max$ . אם מספר השכנים של התא קטן ממש מ  $\min$  התא ימות מבדידות, אם מספר השכנים של התא גדול ממש מ  $\max$  התא ימות מחנק.

דוגמא (\* תסמן תא חי): כאן  $\min=3$  ו  $\max=6$

דור $y$				דור $y+1$		
*			→			
	*					

*	*	*	→		*	
	*				*	

	*		→		*	
*	*	*		*	*	*
	*				*	

*	*	*	→	*	*	*
*	*	*		*		*
*	*	*		*	*	*

### תרגיל

העבודה בתרגיל זה הינה בזוגות.

ממשו מעגל בעזרת תוכנת TKGate אשר מקבל לוח בגודל  $16 \times 14$  בדור  $y$  ומחזיר לוח בדור  $y+1$ . עליכם להגיש אלקטרונית לאתר הקורס את הקובץ  $v$ . של המעגל יחד עם קובץ README אשר מפרט את האלגוריתם הכללי ותיאור קצר של כל מודול. כמו כן, עליכם להגיש תדפיס של כל מודול לתא ברוס 2- במודול הראשי צריכה להיות אפשרות להגדיר את מצב הלוח (על ידי DIP switches) ולראות את התוצאה (על ידי Led Bars). כמו כן, מספר השכנים המינימאלי ומספר השכנים המקסימאלי (כך שהתא יחיה בדור

הבא) הוא חלק מהפרמטרים של המעגל. אנה הקפידו על תיעוד של כל מודול (גם במעגל על ידי Comment).

## הדרכה

עברו על ה Tutorial אשר מופיע באתר. על מנת לממש את המעגל אנו נזדקק לכמה מודולי עזר:

### ייצוג המטריצה

המטריצה תיוצג על ידי 16 מחרוזות בנות 14 סיביות כל אחת. אולם על מנת לטפל במקרי הקצה נשנה את הקלט במעט (כיצד?) כך שאותם מודולים יטפלו בתא ללא תלות במיקומו (כלומר בקצה או במרכז המפה).

### מודולי עזר

1. בנו מודול בשם Comparator אשר מקבל 2 מספרים בני 4 סיביות,  $B, A$ , ומחזיר 0 אם  $A > B$  ו 1 אם  $A \leq B$ .
2. בנו מודול בשם Subtractor אשר מקבל 2 מספרים בני 4 סיביות  $A, B$ , ומחזיר את  $A - B$ .
3. בנו מודול בשם Decode16 אשר מקבל מספר  $i$  בן 4 סיביות ומחזיר מחרוזת בת 16 סיביות כאשר הסיבית ה  $i$  בעלת ערך 1 וכל השאר בעלי ערך 0. לדוגמא: אם  $i=5$  המודול יחזיר את המחרוזת הבאה 0000100000000000. רמז: השתמשו בשני Decode 1-8.

### מודולים

1. Adjacent\_Alive: מקבל מחרוזת בת 16 סיביות ומספר  $i$  בעל 4 סיביות ומחזיר 1 אם המקום ה  $i$  חי ו 0 אם לא.
2. Adjacent\_Line\_Count: מקבל מחרוזת בת 16 סיביות, מספר  $i$  בעל 4 סיביות וסיבית Center ומחזיר את מספר השכנים מספר בעל 4 סיביות) שיש למקום ה  $i$ . כלומר אם  $Center=1$  השכנים ייספרו במקומות ה  $i-1, i+1$  ואם  $Center=0$  השכנים ייספרו במקומות  $i-1, i, i+1$ .
3. Adjacent\_Count: מקבל 3 שורות בנות 16 סיביות כל אחת
4. LineUp, LineCenter, LineDown ומספר  $i$  בעל 4 סיביות. מחזיר את מספר השכנים (מספר בעל 4 ספרות) שיש לתא בשורה LineCenter במקום ה  $i$ .
4. Cell\_Next\_Generation: מקבל מקבל 3 שורות בנות 16 סיביות כל אחת LineUp, LineCenter, LineDown, מספר  $i$  בעל 4 סיביות ומספר data בעל 8 סיביות. מחזיר 1 אם התא חי בדור הבא ו 0 אם לא (ב data אנו מקודדים את  $max$  ו  $min$  – שני מספרים בעלי 4 סיביות כל אחד).
5. Line\_Next\_Generation: מקבל מקבל 3 שורות בנות 16 סיביות כל אחת LineUp, LineCenter, LineDown, ומספר data בעל 8 סיביות. מחזיר את הדור הבא של השורה LineCenter.

6. Matrix\_Next\_Generation: מקבל 16 שורות של 14 סיביות ומחזיר את הדור הבא של כל השורות.  
שימו לב: אין חובה לממש דווקא את המבנה המתואר כאן.