Top of Form

Bottom of Form

Top of Form

Bottom of Form

**LogicalCircuit**

יומן פעילות

1. שבוע המסתיים ב4/3. בחירת פרויקט – תצוגה גרפית של מעגל בוליאני בחירת תשתית לביצוע הפרויקט – נבחנו מספר אפשרויות: • שימוש בספריית IMGUI – ב C++ או ב Python. – קראתי חומר וכן התחלתי להכיר את הספריה. על פי המלצת המרצה אפשרות זו ירדה שכן אין היא אידיאלית בתור ספריה גרפית. • שימוש ב Schemadraw. היתרון של ספריה זו שהיא מספקת צורות מוכנות למעגלים לוגים וחשמליים. אולם זוהי ספריה סטטית אשר איננה תומכת באירועים ובנוסף חסירה עושר גרפי. • שימוש ב nodeeditor . למדתי Tutorial בנושא וכן צפיתי במספר וידאו אשר יוצר הספרייה סיפק. המסקנה היא כי זוהי חבילה מוגבלת אשר ספק אם מאפשרת את כל הנדרש, בנויה יותר עבור widgets מאשר עבור צורות גרפיות ומצריכה לימוד בהיקף נרחב מאוד. כיסוי הספריה וכן יישומים שלה מצריך צפייה בעשרות קטעי וידאו של כ30 דקות כל אחד וכן בהרבה מאוד דוגמאות קוד וגם אז ספק אם היא תתמוך בכל הנדרש. • שימוש ב PyQt5. גם כאן צפיתי בקטעי קוד והדרכות. אין ספק שספריה עשירה זו יכולה לתת מענה מלא אולם על פי המלצת המרצה בשל גודלה ועומקה של החבילה ויתרתי על אפשרות זו. • שימוש ב tKinter. דומה כי זוהי הפלטפורמה האידיאלית לביצוע המשימה. היא מספקת עושר גרפי כולל תמיכה באירועים דבר אשר יאפשר אינטראקטיביות עם המשתמש. בנוסף דומה כי ניתן להגיע לשליטה ברמה טובה מספיק על מנת לבצע לכל הפחות את החלק הבסיסי של הפרויקט בהיקף השקעה נמוך מאשר PyQt5. מסקנה – הפלטפורמה הנבחרת היא tKinter.

משך זמן עבודה – בחינת שלשת האפשרויות הראשונות ארכה כ 3.5 שעות. בחינת 2 האפשרויות האחרונות כ4.5 שעות, סה"כ 8 שעות עבודה.

1. שבוע שמסתיים ב 11/3 שינוי הבחירה ל חבילה nodeeditor על פי החלטת המרצה

מתברר מלימוד הקבצים כי עמ לשנות את הגרפיקה יש צורך בלימוד QtPy , צורת כל node מנוהלת בשני מקומות. מעטפת חיצונית בקובץ node\_graphics\_node בפונקציה paint. תוכן פנימי ב node\_editor\_widget בפונקציה initUI

בוצע שינוי הכותרות עמ לייצג שערים לוגים – הפונקציה addNodes בקובץ node\_editor\_widget.

הצבת ה nodes ע"ג החלון תתבצע באותו הקובץ בפונקציה addNodes .

טיפול ב socket מתבצע במחלקה Socket בקובץ node\_socket, node\_graphics\_socket

יש לשים לב לפונקצייה initSocket() שנמצאת במחלקה node\_graphics\_node

השליטה במיקום הכניסות והיציאות socket נמצאת בפונקציה node\_graphics\_socket.paint()

לימוד עבודה מול gitHub

סה"כ לחיוב – 6 שעות

1. שבוע המסתיים ב18/3

שליטה בכמות הכניסות והיציאות של כל שער נמצאת בפונקציה addNodes בקובץ node\_editor\_widget.

שליטה במיקום ה socket על גבול השער נמצאת בפונקציה getSocketPosition בקובץ node\_node .

איתחול ה sockets מתבצע ב קובץ node\_node בפונקציה initSocket.

בוצע שינוי של צורת האיתחול. במקור האיתחול הוא קשיח ומתבצע עבור כל הכניסות וכל היציאות בבת אחת, כלומר המשתמש יכול לשלוט במיקום של קבוצת הכניסות ולא של כל כניסה בפני עצמה וכן ביחס ליציאות. שיניתי את הקוד כך שניתן יהיה לשלוט על כל כניסה בפני עצמה

צפייה ב tutorials של דוגמת המחשבון.

סה"כ: 6 שעות.

1. שבוע המסתיים ב 25/3

עיבוד קבצים בפייתון, שילוב עיבוד הקובץ , הוצאת רשימת המעגלים מתוכו וציור השערים על המסך , עדיין לא בסדר הנכון ובלא חיבור ביניהם

שעות: 4 שעות

1. שבוע המסתיים ב 1.4 .

המשך עבודה על עיבוד קובץ ה Verilog. שליפת השערים למערך שערים, יציאות למערך יציאות, כניסות למערך כניסות

תיקון הבאגים וסיום עיבוד הקובץ, כמות שעות: 6 שעות

1. הפיכת השערים לרשימה מקושרת על מנת למיין אותם בצורה קלה ונוחה בהתאם לכניסות והיציאות הלוגיות שלהם.

6 שעות עבודה.

שבוע המסתיים ב 15/4 – ללא התקדמות, נקיונות לפסח

שבוע המסתיים ב 21/4

סידור השערים על פני המסך בהתאם לסדר הלוגי שלהם (יציאות כנגד כניסות מתאימות) ולא לפי סדר הופעתם בקובץ ( Verilog מאפשרת רישום השערים בכל סדר שהוא).- היה אגוז קשה לפיצוח.

הבעיות כרגע – ציור הכניסות הראשוניות המזינות את השערים והיציאות. הספריה הזו תומכת רק בציור של nodes על פני המסך.