**Win2015a\_sol**

1. 1. שיפוע הפונקציה גדול ב-1 מערכו המוחלט ולכן יכול לממש שער id תחת משטר סטטי מתאים.
   2. שיפוע הפונקציה קטן מ-1 בערכו המוחלט ולכן לא יכול לממש שער NOT. הפונקציה אינה חסומה מאף צד ולכן לא יכולה לממש קבועים. הפונקציה מהפכת ערכים גדולים לקטנים והפוך ולכן כנראה שלא תוכל למממש id עבור אף משטר. לא נשארה אף אופציה אחרת לשער עם משתנה אחד ולכן הפונקציה לא יכולה לממש אף שער בעל כניסה אחת.
2. בפתרון הרשמי.
3. מצורפת תמונה.
4. 1. מצורפת תמונה.
   2. מיזוג יחידות רק מרע את המצב ביחס לקריטריון מילי, אם הגרף לא קיים את קריטריון מילי, לאחר איחוד יחידות לא יכול להיות שהמצב השתפר.
   3. ע"פ משפט שנלמד בקורס- אם המערכת מקיימת את קריטריון מילי עבור חלוקה מסוימת ליחידות הוא אינה מכילה מעגל צירופי. לכן אם הוא מכילה מעגל צירופי לא ייתכן שתקיים את קריטריון מילי עבור אף חלוקה שלה ליחידות.
   4. אם מערכת אינה מקיימת את קריטריון מילי עבור חלוקה מסוימת לא ניתן לדעת אם יש בה מעגל או לא (המשפט הינו רק בכיוון אחד). ראינו זאת בתרגול.
5. נבצע כמה אבחנות קודם:

|  | Throughput | latency | זמן מחזור | תחנות |
| --- | --- | --- | --- | --- |
| A | 1/60 | 180 | 60 | 180/60=3 |
| B | 1/30 | 240 | 30 | 240/30=8 |
| C | 1/90 | 270 | 90 | 270/90=3 |
| D | 1/100 | 200 | 100 | 200/100=2 |

* 1. נדרש שעל כל מסלול יהיה מספר זהה של רגיסטרים ע"פ הגדרת פייפליין.  
     על המסלול הראשון (העליון) יש 11 רגיסטרים.  
     על המסלול השני יש 5 רגיסטרים.  
     על המסלול השלישי יש 5 רגיסטרים.  
       
     -לכן נצטרך להוסיף לקווים ב,ג 6 רגיסטרים בשביל להשוות את מספר הרגיסטרים בכל המסלולים.
  2. המסלול בעל ההשהיה המקסימלית הוא זה שיוצא מ-C, עובר דרך E ונכנס לרגיסטר בכניסה של פייפליין D. השהיית מסלול זה הינה 90+20=110 ולכן זמן המחזור המינימלי הוא 110.
  3. זמן המחזור לא השתנה כי לא הוספנו רגיסטר בין C ל-E (כי לא היתה אופציה כזו). Latency הוא פשוט מספר מחזורי השעון כפול זמן מחזור. ==> 110\*11=1210.

1. נפתח נוסחה ל- ונקבל-  .
2. המסלול הכי ארוך הוא 4. לכן מחזור שעון הוא 4. נתון שהחישוב לוקח 100 מחזורי שעון ולכן זמן החישוב יהיה 4\*100=400.  
   אם נבצע חתכים ככה שכל הקשתות בכיוון אחד נוכל להגיע תוך 5 חתכים למצב בו כל יחידה מבודדת מיחידות אחרות.  
   ביצענו 5 חתכים שרק הוסיפו רגיסטרים לכל המסלולים ולכן עיכבנו את החישוב ב-5 מחזורי שעון והוא ייקח עכשיו 105 מחזורים במקום 100. אבל הקטנו את זמן המחזור ל-1 ולכן זמן החישוב עתה יהיה 105\*1=105.  
     
   איך ידענו שצריכים לבצע רק 5 חתכים בדיוק? שתי אופציות:  
   א. ניסוי וטעייה (פחות מומלץ ולא ניתן להוכחה אבל עובד ;) ).  
   ב. אם נסתכל על המערכת נשים לב שכל מסלול מכניסה ליציאה בין אם צירופי ובין אם לא הוא באורך של 8 יחידות. קיים מסלול כזה שעליו 2 רגיסטרים בין היחידות ולכן נצטרך להוסיף 5 רגיסטרים למסלול הנ"ל (קיבלנו חסם תחתון- 5). מכיוון שאין מסלול שדורש הוספה של יותר מ-5 רגיסטרים, קיבלנו חסם עליון על מספר החתכים שצריכים לעשות (5).  
   (את 5 החתכים נעשה ככה שיתפרו את כל המסלולים).
3. זה מוסבר בסוף התרגול על מיפס, אבל ניתן לבצע pre-fetch רק בפקודות שבמחזור האחרון שלהם אין שימוש ב- MEM **(להוציא SW)** ושהפקודה הבאה לביצוע היא הפקודה שאחרי הפקודה שרצה עכשיו **(להוציא קפיצות- J,BEQ)**.
4. Micro-programming ירד מחומר הקורס ☺.
5. משווים כל מספר לאחד שאחריו (בו זמנית) ומוציאים ביט ששווה ל-1 אמ"ם המספר קטן ממש מהאחד שאחריו - .  
   עושים AND על כל הביטים מהשלב הראשון, נעשה זאת בצורה של עץ - .  
     
   סה"כ: .