

תרגיל בית רטוב מספר 2

מגישים:

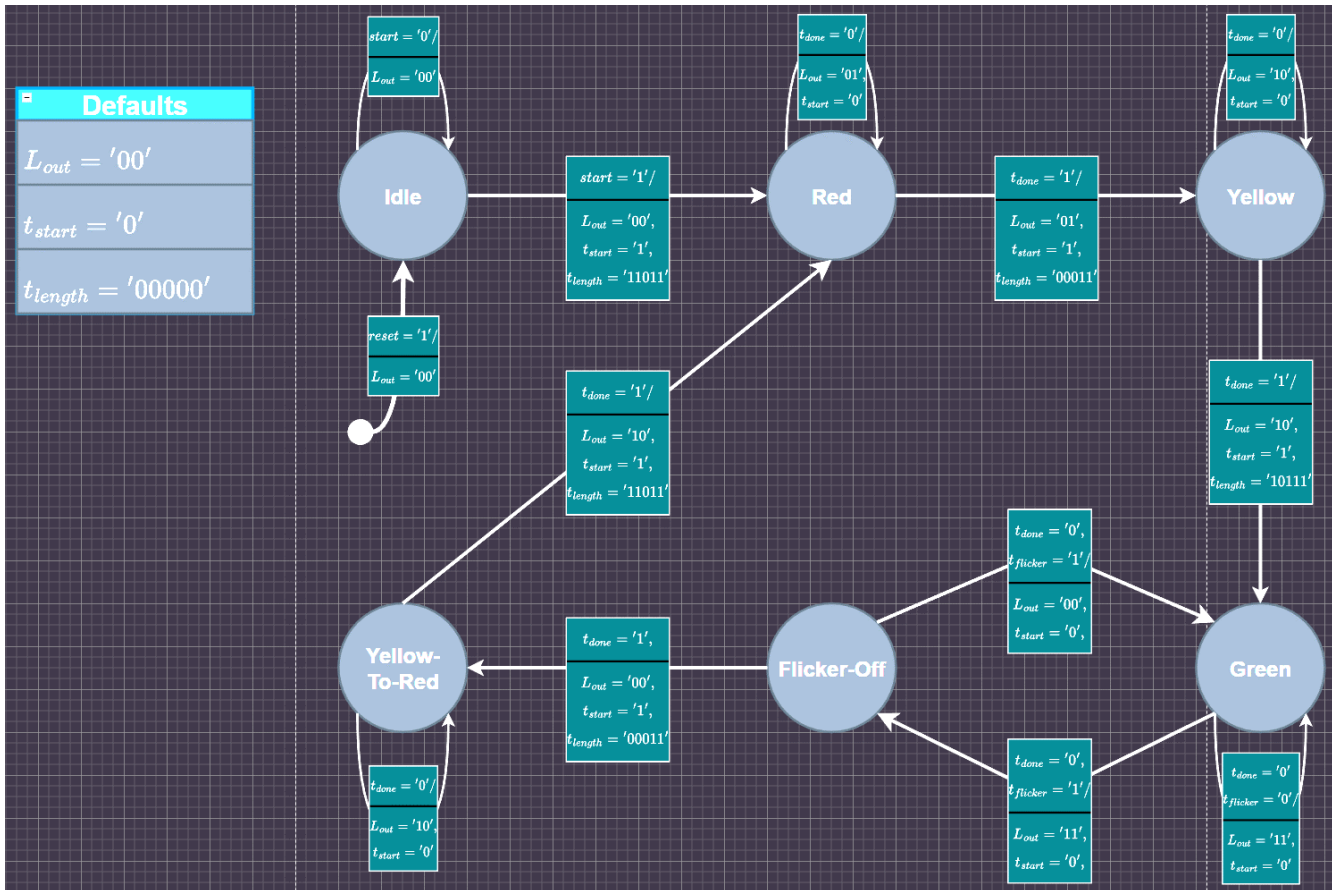
322654617	ינון יהב
327812483	נועם בן שמעון

07/08/2024

תאריך:

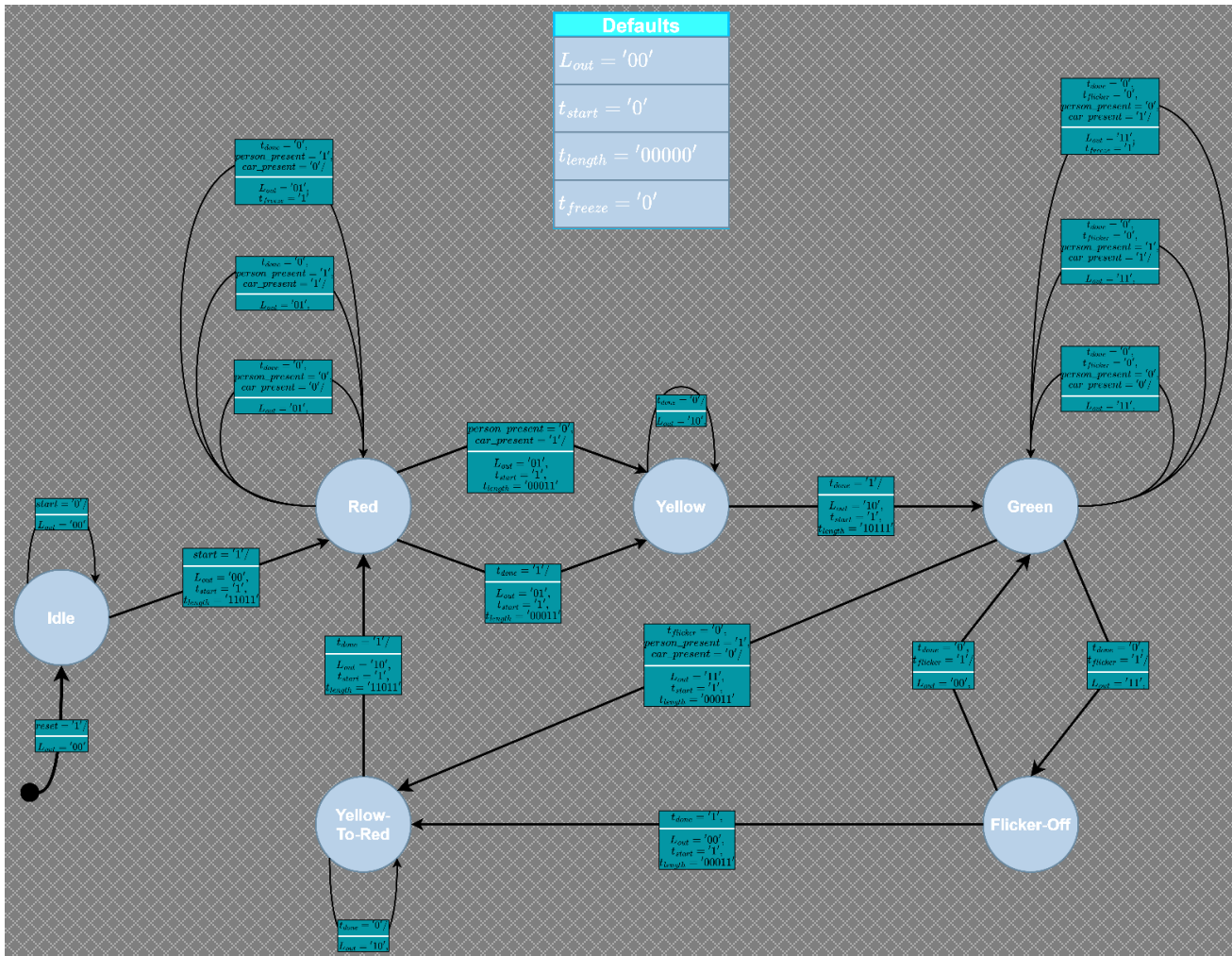
322654617	ינון יזהב
327812483	נועם בן שמעון

2.1 – מכונת מצבים מסוג *Mealy* לרמזור פשוט:



$X = 7$
$Y = 3$

2.2 – מכונת מצבים מסוג Mealy לרמזור חכם:



$X = 7$
$Y = 3$

הרמזור החכם מתפקד בצורה זוהי לגרסה הקודמת במקרים בהם יש גם אנשים וגם מכוניות, או כשאין אנשים ואין מכוניות ($car_present = person_present$), הוא לא מעניק עדיפות לאף וממשיך כרגיל.

הרמזור החכם יעיל יותר בזמן ההמתנה הממוצע לרכבים כאשר יש תנועה מעטה של אנשים, ולכן יש הרבה פרקי זמן בהם אין אנשים כלל המכונים. רמזור זה, באותם המקרים, יעבור לירוק למכוניות וימנע את בזוז הזמן של הרכבים שמחכים שהאנשים יסיימו לחצות כשאין אנשים.

לעומת זאת, זמן ההמתנה גרוע יותר במקרים בהם יש הרבה אנשים שרוצים לחצות, ולא הרבה מכוניות שבאות באופן רציף. זה גורם לכך שהרמזורים יהיו רוב הזמן על מצב להולכי רגל, וכאשר תבוא מכונית, היא תיאלץ לחכות את מלוא זמן הרמזור האדום ואת הרמזור (לעומת שברגיל מעט פחות מחצי מהזמן הרמזור ירוק לרכבים והם יכולים להמשיך במקום לחכות). במקרה זה יאלצו הרכבים לחכות כ-30 מחזורי שעון, לעומת לכל היותר 33 ברגיל (אם היא הגיעה בדיוק כשהירוק עברה נגמר) ובממוצע כחצי ממספר זה.

2.3

טבלת המצבים של "רמזור חכם".

Is_person	Is_car	t_flicker	t_done	t_freeze = z ₁	t_start = z ₂
0	0	0	0	0	0
0	0	0	1	dc	dc
0	0	dc	dc	0	0
0	0	dc	dc	0	0
0	1	dc	dc	0	0
0	1	dc	dc	0	0
0	1	0	1	dc	dc
0	1	0	0	1	0
1	1	0	0	0	0
1	1	0	1	dc	dc
1	1	dc	dc	0	0
1	1	dc	dc	0	0
1	0	dc	dc	0	0
1	0	dc	dc	0	0
1	0	0	1	dc	dc
1	0	0	0	0	1

נבנה טבלת קרנו ובעזרתה נמצא ביטוי מצומצם ל- t_freeze:

t_flicker, t_done is_person, is_car	00	01	11	10
00	0	dc	0	0
01	1	dc	0	0
11	0	dc	0	0
10	0	dc	0	0

$$t_{freeze} = \overline{is_person} * is_car * \overline{t_flicker}$$

נבנה טבלת קרנו ובעזרתה נמצא ביטוי מצומצם ל- t_start :

$t_flicker, t_done$ is_person, is_car	00	01	11	10
00	0	dc	0	0
01	0	dc	0	0
11	0	dc	0	0
10	1	dc	0	0

$$t_start = \overline{is_car} * is_person * \overline{t_flicker}$$

2.4



בין - Start of your code ל - End of your code הקוד שכתבנו הוא:

```

not t3 t3 # not is_car
not t5 t5 # not t_flicker
and t3 t3 t5 # t3 = is_car' and t_flicker'
and a1 t3 t4 # t5 = is_car' and t_flicker' and is_person
  
```

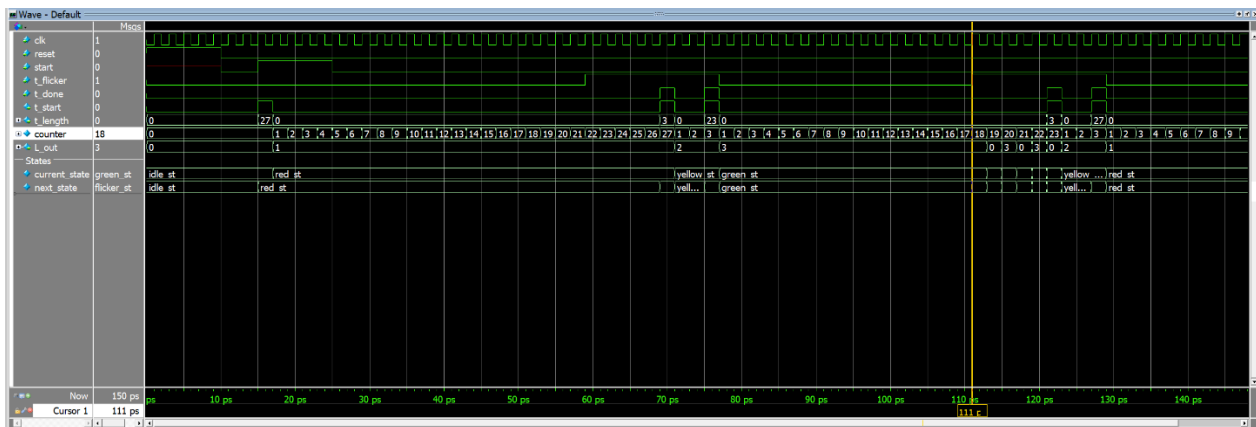
אנחנו מבצעים ארבעה פקודות כאשר נתון כי כל פקודה היא מחזור שעון אחד, לכן בכל לופ הקטע קוד שכתבנו יסתיים לאחר ארבעה מחזורי שעון. בסך הכול בכל התוכנית נעבור ארבעה פעמים על קטע הקוד שכתבנו כך שארבעת הפקודות ירוצו ארבעה פעמים שזה שישה עשר מחזורי זמן.

322654617	ינון יהב
327812483	נועם בן שמעון

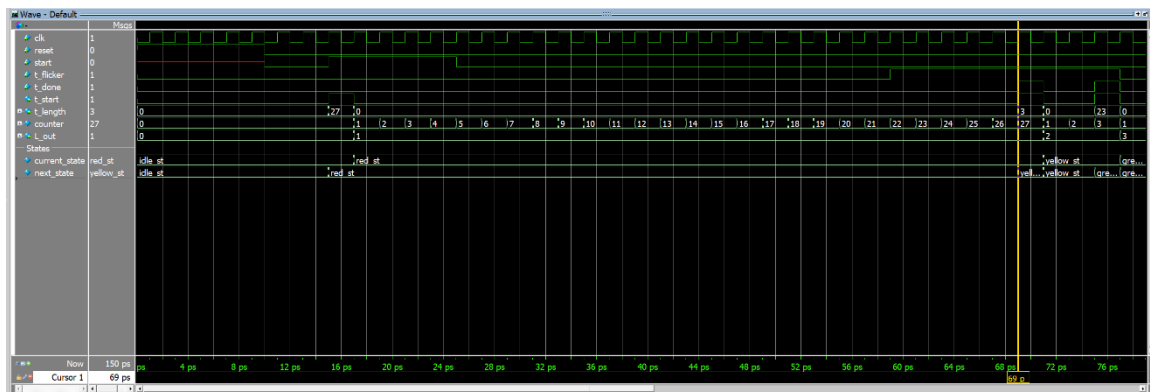
תרשים גלים עבור 3.4 – מימוש רמזור פשוט:

$X = 7$
$Y = 3$

תמונה כוללת של כל תרשים הגלים:



תמונות עם פרטי התרשים:



(1)



נשים לב ונזכור במהלך המעבר על הדיאגרמה כי:

$$time_unit = 1ps, \quad clock_cycle = 2 \cdot time_unit = 2ps$$

לאחר ההשמה, אנו נחכה 5 מחזורי שעת. ממיקום החתוצץ אנו יכולים לראות כי אכן אנו שומרים על הערך יציב למשך בדיוק 5 מחזורי שעת.

(2) בשלב השני אנו שמים $reset = '0', start = '0'$. הערכים בתרשים תואמים.

לאחר ההשמה, אנו נחכה 5 יחידות זמן. ניתן לזהות כי ספירה זו החלה כאשר הקו האדום של *start* הפך לירוק וקיבל את הערך המתאים, שבו נשאר יציב למשך בדיוק 5 יחידות זמן.

(3) בשלב השלישי אנו שמים $start = '1'$. הערך בתרשים תואם.

לאחר ההשמה, אנו נחכה 10 יחידות זמן. מהשלב שבו *start* משתנה בתרשים להיות 1 אנו יכולים לראות כי אכן אנו שומרים על הערכים יציבות למשך בדיוק 10 יחידות זמן.

ניתן לראות כי המערכת החלה לפעול, שכן אנו יכולים לראות כי תוך מחזור שעות אחד אנו עוברים למצב *red_st* שמייצג רמזור אדום, ומתחילים את הטיימר עבורו (*27 clock cycles*) כפי שניתן לראות ב-*counter*.

(4) בשלב החמישי אנו שמים *'0' = start*. הערך בתרשים תואם.

לאחר ההשמה, אנו נרצה להמשיך את הריצה כרגיל, ואלו הם החלקים החל מאחרי כך שהטיימר כאשר אנו במצב האדום כשהטיימר על 4, ובזמן זה אנו עוברים על שארית המצבים במערכת.

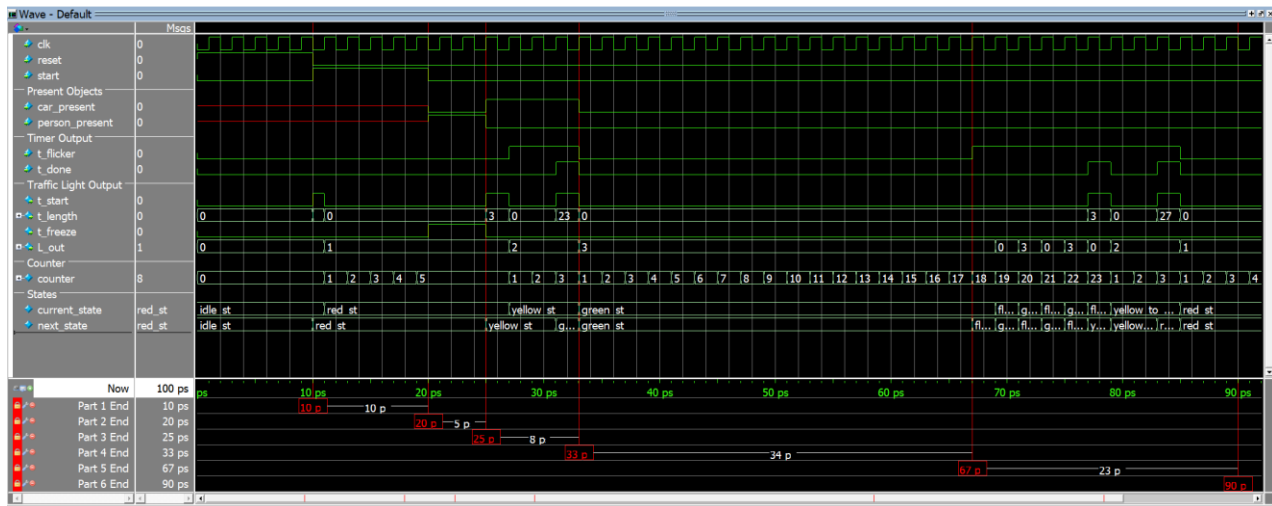
מכאן אנו יכולים לראות כי הריצה ממשיכה כרגיל. ראשית אנו רוצים לסיים את הריצה על המצב האדום, ומחכים את שאר מחזורי השעות הנותרים שאנו צפויים לחכות. לאחר מכן אנו עוברים למצב הצהוב שבו אנו מחכים 3 מחזורי שעות. אחריו אנו עוברים למצב הירוק שבו אנו מחכים 23 מחזורי שעות, כאשר ה-5 האחרונים הם במצב הבהוב. אחרי זה אנו מגיעים למצב "צהוב לאדום" שבו אנו מחכים עוד 3 מחזורי שעות כצפוי. לבסוף אנו מגיעים שוב למצב האדום, שממנו התהליך ימשיך לנצח בצורה חזורתית.

תרשים גלים עבור 3.7 – מימוש רמזור חכם:

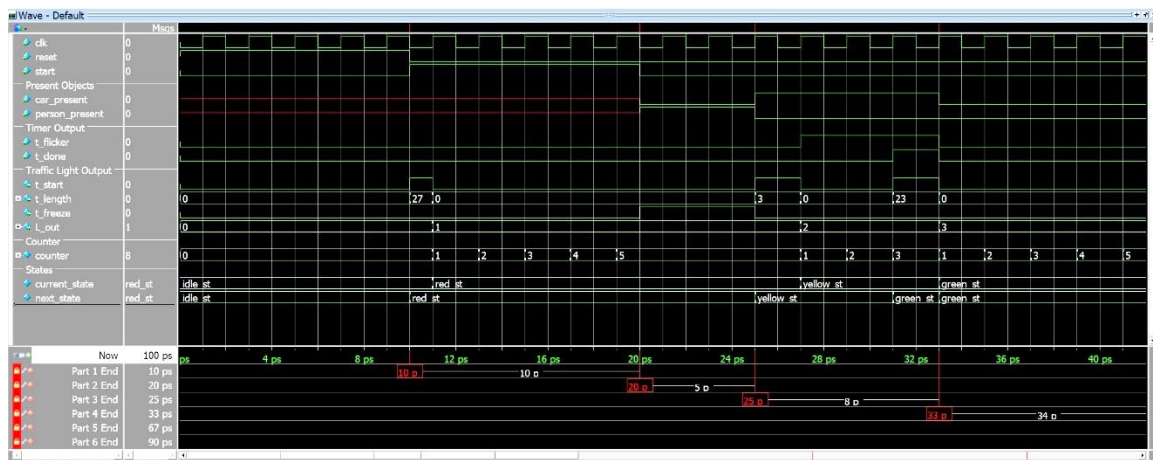
$$X = 7$$

$$Y = 3$$

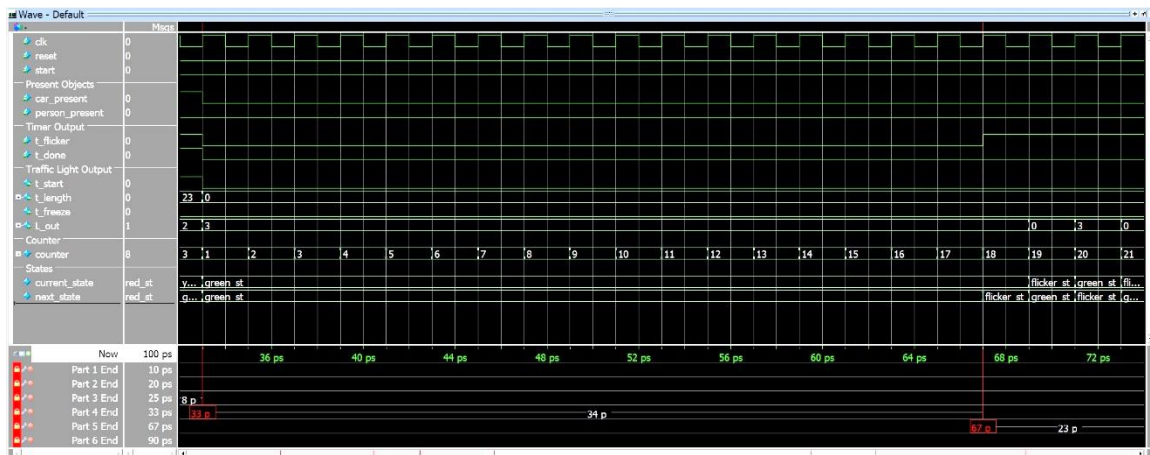
תמונה כוללת של כל תרשים הגלים:



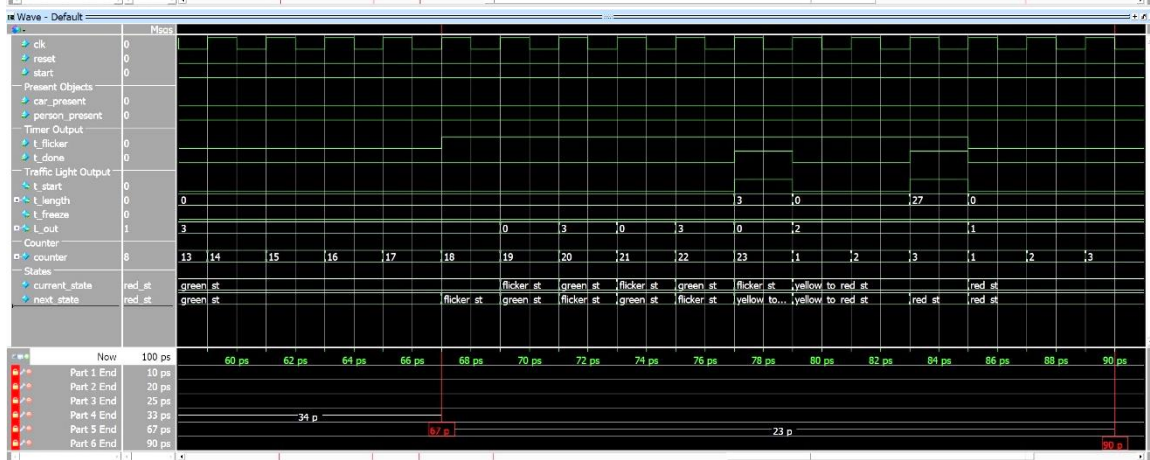
תמונות עם פרטי התרשים:



(1)



(2)



(3)

נסביר כעת את תוצאות הדיאגרמה, ניעזר בחוצצים שהוגדרו מראש והונחו בסוף של כל שלב מההנחיות.

נשים לב ונזכור במהלך המעבר על הדיאגרמה כי:

$$time_unit = 1ps, \text{ clock_cycle} = 2 \cdot time_unit = 2ps$$

(5) בשלב הראשון אנו שמים $reset = '1', start = '0'$. הערכים בתרשים תואמים.

לאחר ההשמה, אנו נחכה 5 מחזורי שעון. ממיקום החוצץ אנו יכולים לראות כי אכן אנו שומרים על הערכים יציבות למשך בדיוק 5 מחזורי שעון.

נשים לב כי $car_present, person_present$ שניהם אינם מאותחלים וניתן לראות כי כך גם בדיאגרמה שכן מסומנים בקווים אדומים.

(6) בשלב השני אנו שמים $reset = '0', start = '1'$. הערכים בתרשים תואמים.

לאחר ההשמה, אנו נחכה 10 יחידות זמן. ממיקום החוצץ אנו יכולים לראות כי אכן אנו שומרים על הערכים יציבות למשך בדיוק 10 יחידות זמן.

ניתן לראות כי המערכת החלה לפעול, שכן אנו יכולים לראות כי תוך מחזור שעון אחד אנו עוברים למצב red_st שמייצג רמזור אדום, ומתחילים את הטיימר עבורו (27 clock cycles) כפי שניתן לראות ב- $counter$.

(7) בשלב השלישי אנו שמים $person_present = '1', car_present = '0', start = '0'$.
הערכים בתרשים תואמים, $car_present$ ו- $person_present$ התייצבו והתקבעו.

לאחר ההשמה, אנו נחכה 5 יחידות זמן. ממוקם החוצץ אנו יכולים לראות כי אכן אנו שומרים על הערכים יציבות למשך בדיוק 5 יחידות זמן.

אנו נרצה במצב זה עם ערכים אלו לקפוא במצב האדום ולתת לאנשים לעבור, ואכן ניתן לראות כי אנו במצב האדום t_freeze עולה ל-1, וכי הטיימר עוצר מלהתקדם ונשאר על 5.

(8) בשלב הרביעי אנו שמים $person_present = '0', car_present = '1'$. הערכים בתרשים תואמים.

לאחר ההשמה, אנו נחכה עד שנגיע למצב הירוק. ברגע אחרי השינוי נשים לב כי כבר מתקבל $next_state = yellow_st$, ובחזור השעון שאחרי אנו עוברים למצב הצהוב ומחכים בו את ה-3 מחזורי שעון (כצפוי), ואז עוברים למצב הירוק. משמע סה"כ אנו מחכים 4 מחזורי שעון. ממוקם החוצץ אנו יכולים לראות כי אכן אנו שומרים על הערכים יציבות למשך בדיוק 4 מחזורי שעון.

אנו נרצה במצב זה עם ערכים אלו לדלג על הטיימר ולהגיע למצב הירוק כדי לתת לרכבים לעבור, ואכן ניתן לראות כי אנו קופצים תוך מחזור שעון אל המצב הצהוב (בהתעלמות מהטיימר) ואז עוברים לירוק. בהתאמה t_freeze יורד ל-0 שכן אנו לא רוצים לעצור במצב זה יותר.

(9) בשלב החמישי אנו שמים $car_present = '0'$. הערך בתרשים תואם.

לאחר ההשמה, אנו נרצה להמשיך את הריצה כרגיל, ואלו הם החלקים החל מאחרי החוצץ הרביעי, שבזמן זה אנו עוברים על שארית המצבים במערכת.

מכאן אנו יכולים לראות כי הריצה ממשיכה כרגיל. ראשית אנו עוברים במצב הירוק, ומחכים את 23 מחזורי השעון שאנו צפויים לחכות, כאשר 5 מהם הם מחזורי הבהוב. לאחר מכן אנו עוברים למצב "צהוב לאדום" שבו אנו מחכים 3 מחזורי שעון. לבסוף אנו מגיעים שוב למצב האדום, שממנו התהליך ימשיך לנצח בצורה חזרתית.