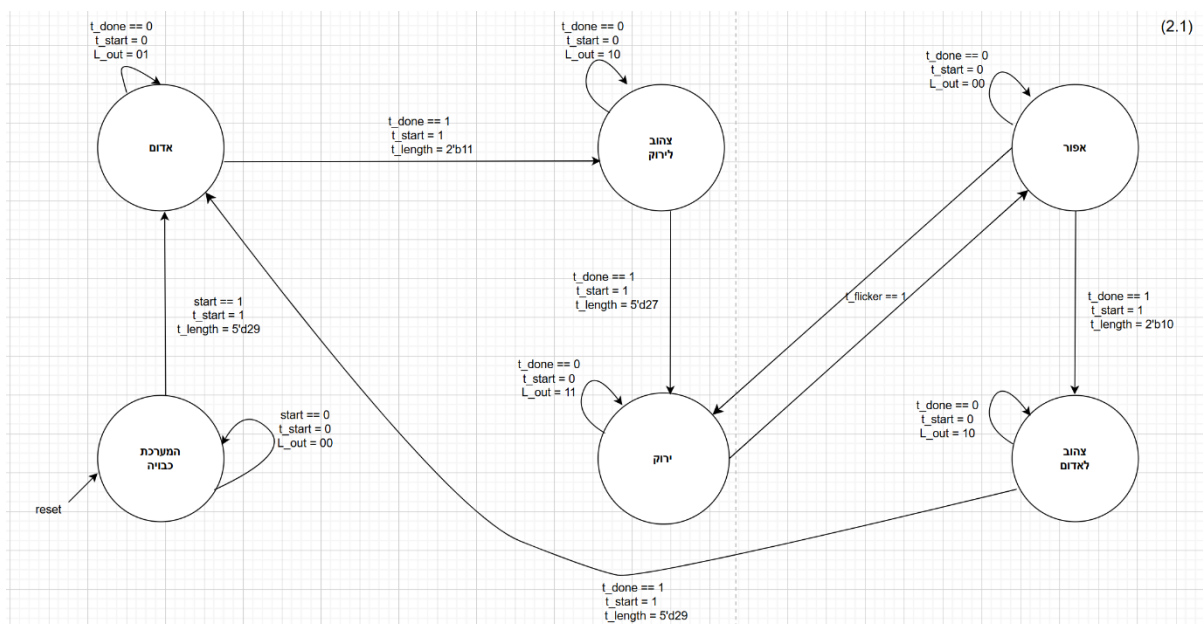
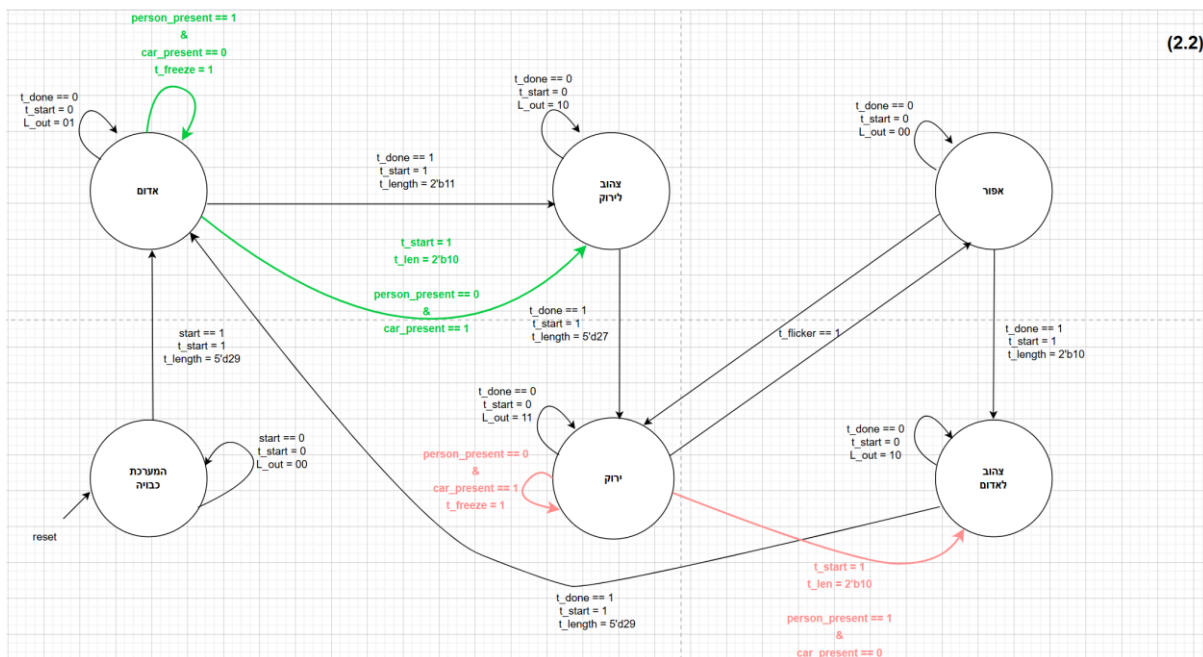


2.1



הערה: לא הוספנו בשרטוט מקרים בהם שינוי בכניסה אינו גורם לשינוי במכונת המצבים. כלומר מקרים כאשר השינוי בכניסה אינו משפיע על אף יציאה ואינו גורם למעבר מצב. למשל: לא קיים חץ $\text{clk} = \emptyset$ במצב אדום "לעצמו".

2.2



הערה: לא הוספנו בשרטוט מקרים בהם שינוי בכניסה אינו גורם לשינוי במכונת המצבים. כלומר מקרים כאשר השינוי בכניסה אינו משפיע על אף יציאה ואינו גורם למעבר מצב. למשל: לא קיים חץ $\text{clk} = \emptyset$ במצב אדום "לעצמו".

המצבים בהם הרמזור החכם דומה לרמזור הרגיל:

1. כאשר אין הולכי רגל ואין רכבים בקרבת
2. כאשר יש גם רכבים וגם הולכי רגל בקרבת הרמזור.

המצב בו זמן ההמתנה (הממוצע) של הרכבים טוב מזה של הרמזור הרגיל הוא מצב בו יש רכבים ואין הולכי רגל. מכיוון שאם סיטואציה זו מתקבלת כאשר הרמזור אדום, מתחילים ישר את תהליך החלפת הרמזור ולא מחכים עד לסוף הטיימר.

המצב בו זמן ההמתנה (הממוצע) של הרכבים גרוע מזה של הרמזור הרגיל הוא מצב בו יש הולכי רגל ואין רכבים כאשר הרמזור ירוק, ומיד לאחר מכן מגיע רכב. תחילה, כאשר יש אנשים ואין רכבים נתחיל את תהליך החלפת הרמזור ונעבור למצב צהוב ואז לאדום. אך ייתכן שבדיקת לאחר שעברנו מצב יגיע רכב, ואז הוא יצטרך לחכות שהרמזור יחזור לירוק כדי לעבור (גם אם אין הולכי רגל הוא עדיין יצטרך לחכות לכל הפחות את משך הזמן בו הרמזור צהוב). לעומת זאת, ברמזור הרגיל בסיטואציה זו (יש הולכי רגל, אין מכוניות והרמזור ירוק) הרמזור לא היה מתחלף ואז הרכב שהגיע היה יכול ישר לעבור.

2.3:

עבור t_start :

מספור	t_done	t_flicker	is_person	is_car	t_start
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

הערה: למעשה עבור טיימר תקין תמיד $t_done = 0$ במצב ירוק, וכניסת t_done כלל לא משפיעה במצב ירוק – לא גוררת לשינוי באף יציאה ולא למעבר מצב.

מפת קרנו:

	$\overline{t_done} * \overline{t_flicker}$	$\overline{t_done} * t_flicker$	$t_done * \overline{t_flicker}$	$t_done * t_flicker$
$\overline{is_person} * \overline{is_car}$	0	0	0	0
$\overline{is_person} * is_car$	0	0	0	0
$is_person * \overline{is_car}$	0	0	0	0
$is_person * is_car$	1	0	0	1

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

עבור t_freeze:

מספור	t_done	t_flicker	is_person	is_car	t_start
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

הערה: למעשה עבור טיימר תקין תמיד $t_done = 0$ במצב ירוק, וכניסת t_done כלל לא משפיעה במצב ירוק – לא גוררת לשינוי באף יציאה ולא למעבר מצב.

מפת קרנו:

	$\overline{t_done} * \overline{t_flicker}$	$\overline{t_done} * t_flicker$	$t_done * \overline{t_flicker}$	$t_done * t_flicker$
$\overline{is_person} * \overline{is_car}$	0	0	0	0
$\overline{is_person} * is_car$	1	0	0	1
$is_person * \overline{is_car}$	0	0	0	0
$is_person * is_car$	1	0	0	0

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

2.4:

Editor Simulator

Run
Step
Prev
Reset
Dump

Machine Code	Basic Code	Original Code
0x10000297	auipc x5 65536	la t0, in # Load address of 'in' into t0
0x00028293	addi x5 x5 0	la t0, in # Load address of 'in' into t0
0x0002a283	lw x5 0(x5)	lw t0, 0(t0) # Load word from memory address in t0
0x00000513	addi x10 x0 0	addi a0, x0, 0 # Initialize output to 0
0x00400393	addi x7 x0 4	li t2, 4 # Loop counter set to 4 for 4 iterations

11

(x7)

s0 0x00000000

(x8)

s1 0x00000000

(x9)

a0 0x0000000a

(x10)

a1 0x0000000b

(x11)

a2 0xffffffff

(x12)

a3 0xffffffff

(x13)

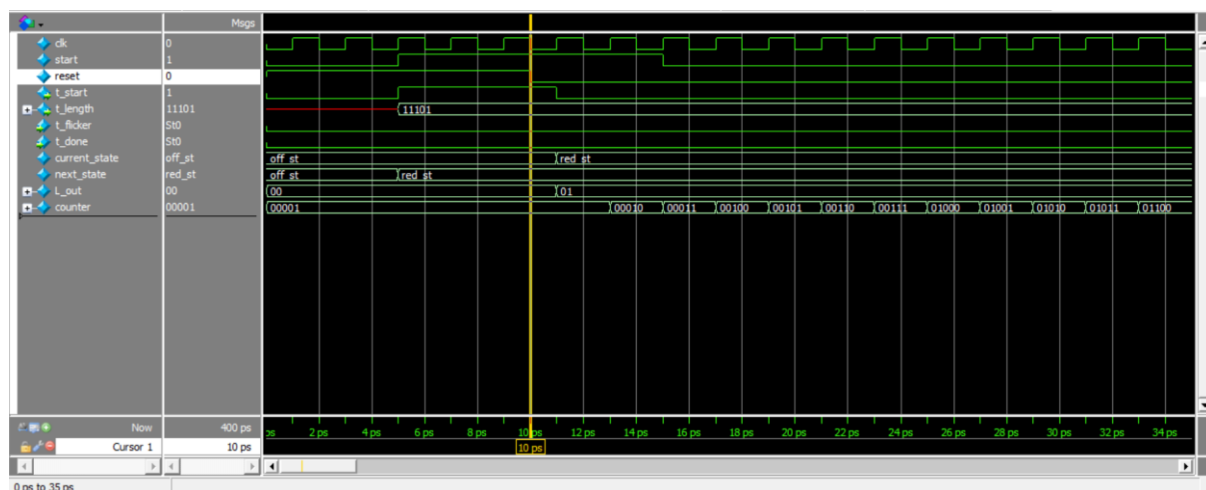
a4 0x00000001

(x14)

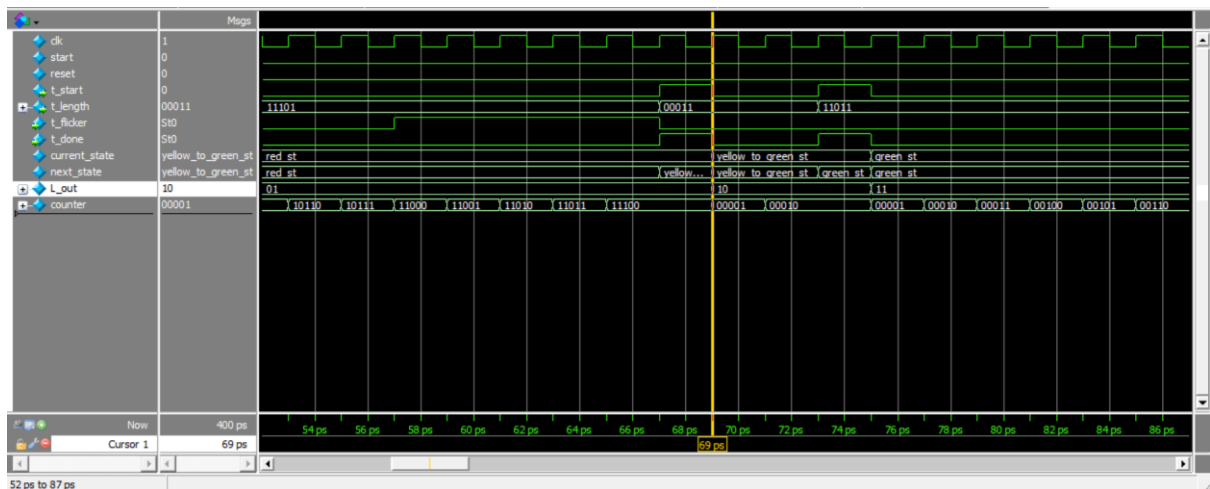
Display Hex

כדי לממש את פעולת החישוב השתמשנו ב4 פקודות (פעמיים not ופעמיים and). בהנחה שכל פקודה לוקחת מחזור שעון אחד, סה"כ כל איטרציה של הלולאה מבצעת חישוב שלוקח 4 מחזורי שעון. יש 4 איטרציות, ולכן פעולת החישוב לוקחת סה"כ 16 מחזורי שעון.

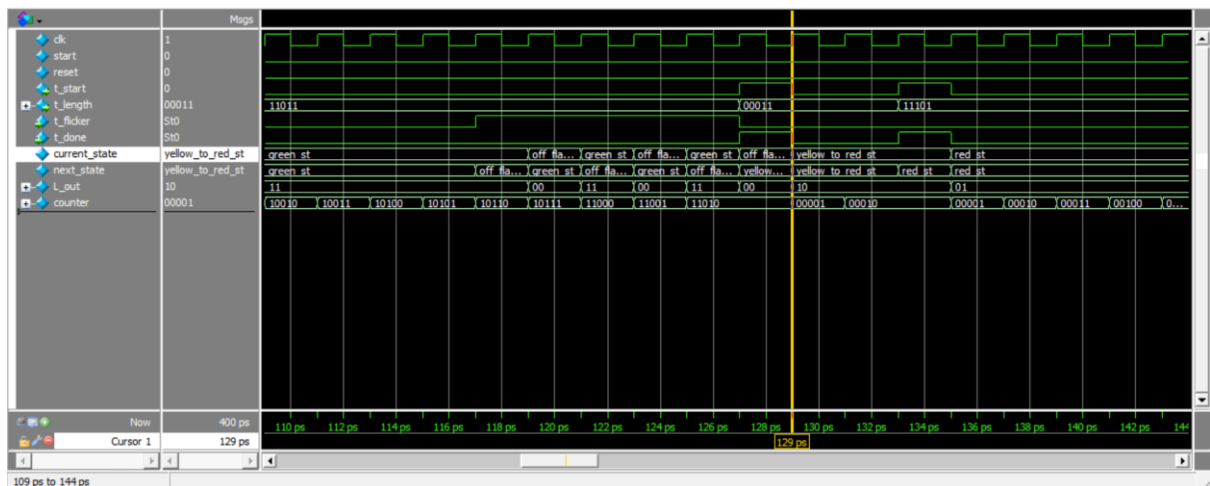
3.4:



Start עלה לאחר 5ps כאשר ה1 = reset. 10ps reset יורד ואכן בעליית השעון הבא הרמזור מתחיל לעבוד. כלומר ב 11ps הרמזור התחלף לאדום ובמקביל L_out השתנתה ל 01 (אדום).

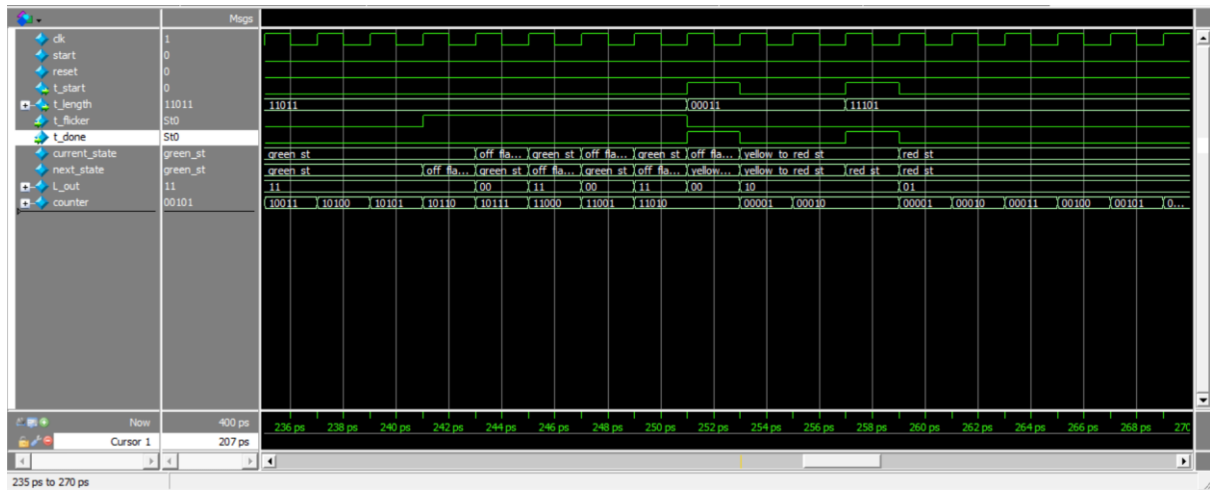
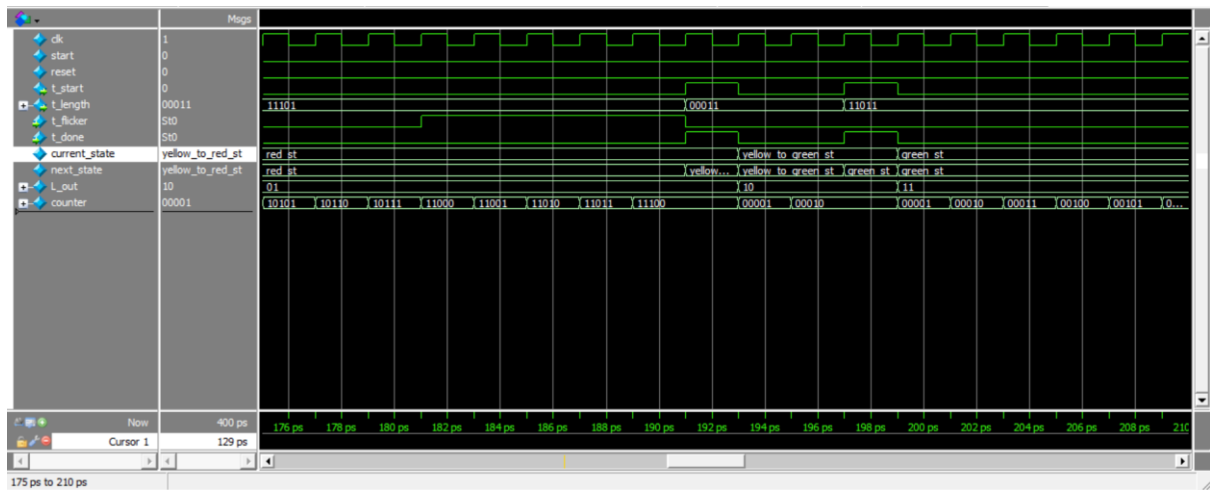


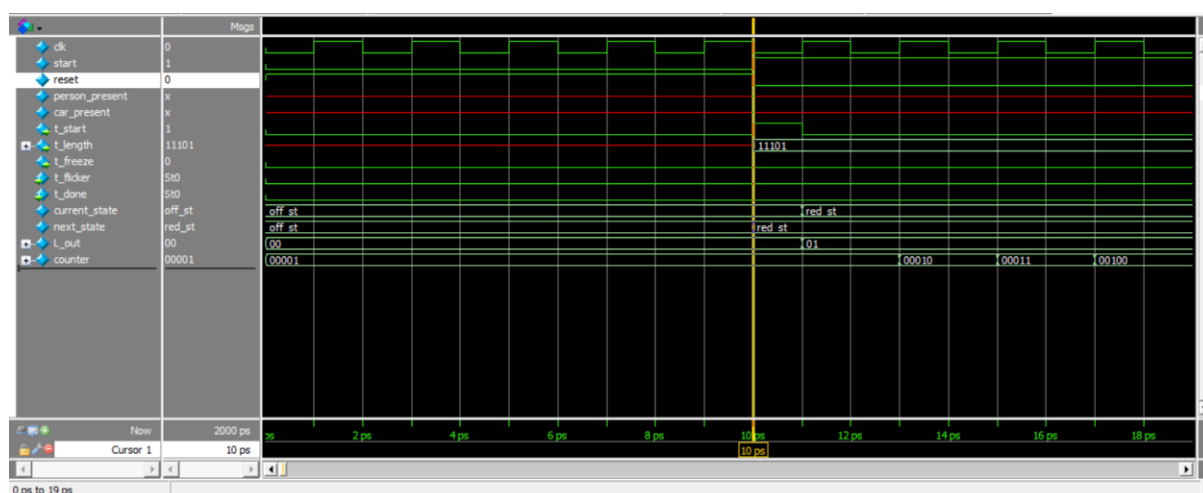
$$\frac{69-11}{2} = 29$$
 מחזורי שעון כנדרש
 L_out השתנתה מ10 ל01 (אדום לצהוב) ב69ps, כלומר שה"כ היה אדום במשך 29
 בחלק היבש. לאחר מכן הרמזור היה צהוב ל3 מחזורי שעון ואז עבר לירוק ב75ps.



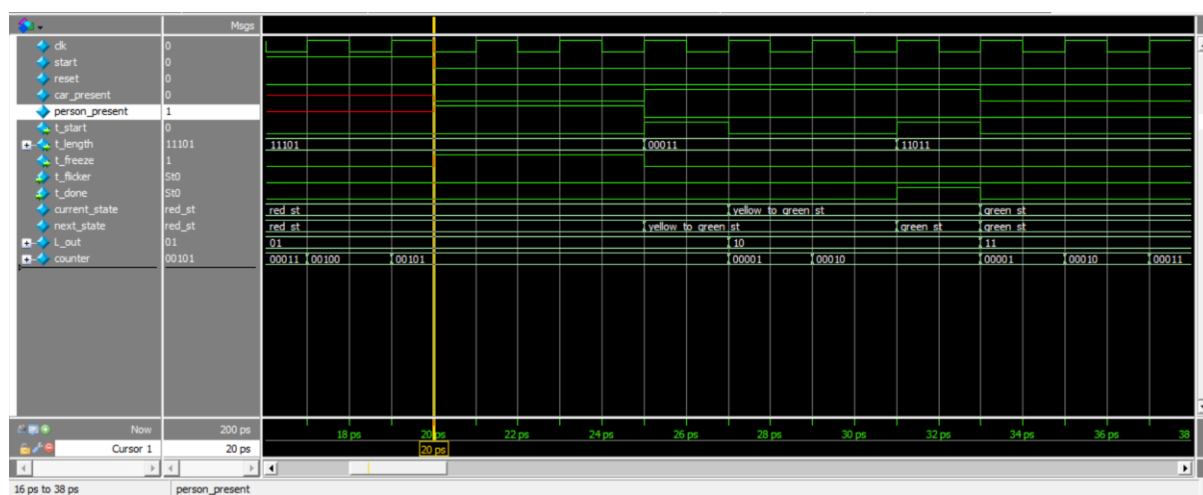
$$\frac{129-75}{2} = 27$$
 מחזורי שעון
 הרמזור עבר מירוק לצהוב ב129ps כלומר סך הכל היה ירוק (כולל הבהוב) למשך 27
 בהתאם לחלק היבש. ניתן לראות שההבהוב עובד כרצוי – הרמזור מהבהב 5 מחזורי שעון לפי הדפוס בחלק היבש
 (כבוי – ירוק – כבוי – ירוק – כבוי). ולאחר מכן יש 3 מחזורי שעון שבהם הרמזור בצהוב. ב135ps הרמזור הופך לאדום
 וחוזר חלילה.

צילומי מסך של מחזור נוסף של פעולת הרמזור:





למשך 5 מחזורי השעון הראשונים ה- $reset = 1$ ו- $start = 0$ כנדרש בטסט. לאחר מכן $start = 1$ והרמזור עובר לאדום ב-11ps (עליית השעון הבאה).



ב-20ps (לאחר 10 יחידות זמן מקבלת $start = 1$) $person_present = 1$ ו- $car_present = 0$, $start = 0$ כנדרש בטסט. ניתן לראות שבזמן זה ה- t_freeze עלה ל-1 בצפוי וה- $counter$ קפא. ב-25ps, $person_present = 0$ ו- $car_present = 1$ וניתן לראות שעקב כך המצב משתנה לצהוב ללא תלות ב- $counter$. ב-33ps (במעבר לירוק) $car_present = 0$ כנדרש בטסט, ומשם הרמזור ממשיך לפעול רגיל.

צילומים של המשך פעילות הרמזור:

