

סימולציה 2 – רמזור, מערכות ספרתיות

מגישים:

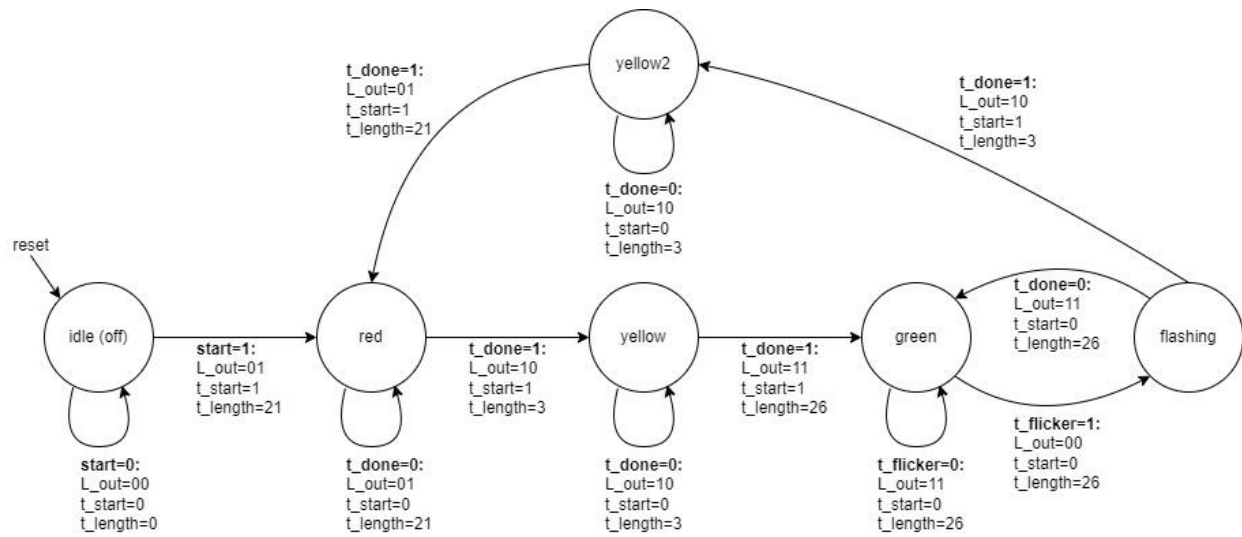
328274386

אביב ריכטר –

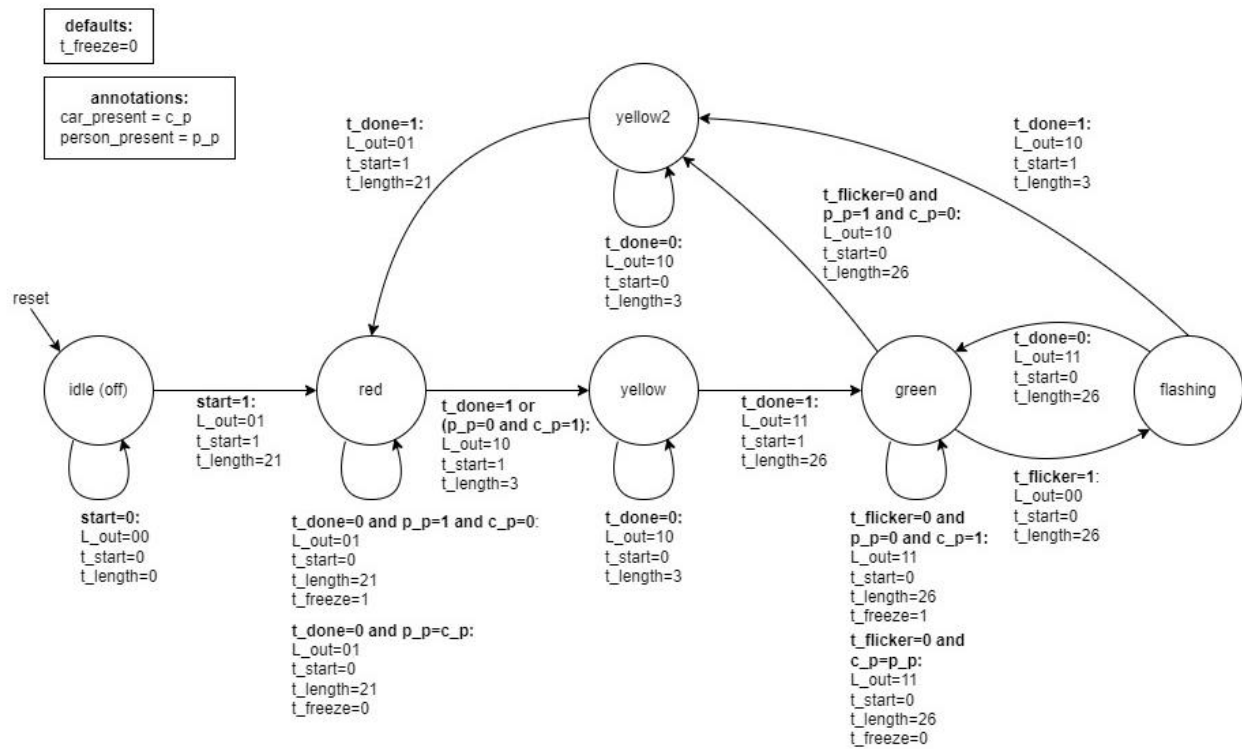
215105321

אורי כשר חיטין –

(2.1)



(2.2)



הרמזור ה-"חכם" דומה לשל הגרסה הקודמת מסעיף 2.1 כאשר לא נתקלים בסיטואציות אשר גורמות לרמזור לפעול בצורה "חכמה". כלומר, כל עוד איננו נתקלים במצב שבו ישנם אנשים אך אין רכבים או להיפך, הרמזור ה-"חכם" יפעל בדיוק כמו קודם.

זמן ההמתנה (הממוצע) של הרכבים טוב יותר מהגרסה הקודמת, כלומר קצר יותר, כאשר אין הולכי רגל. במקרה זה הרמזור החכם מיד מתחיל את תהליך ההחלפה מאדום לירוק. זמן ההמתנה של הרכבים לא יכול להיות ארוך יותר: במקרה שאין הולכי רגל זמן ההמתנה כאמור קצר יותר, ובמקרה שיש – זמן ההמתנה לא ישתנה.

2.3) טבלה ופונקציה עבור t_{start} :

t_{done}	$t_{flicker}$	person_present	car_present	t_{start}
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	\emptyset
0	1	0	1	\emptyset
0	1	1	0	\emptyset
0	1	1	1	\emptyset
1	0	0	0	\emptyset
1	0	0	1	\emptyset
1	0	1	0	\emptyset
1	0	1	1	\emptyset
1	1	0	0	\emptyset
1	1	0	1	\emptyset
1	1	1	0	\emptyset
1	1	1	1	\emptyset

נוכל להציב במוצא \emptyset כאשר $t_{done} = 1$ או כאשר $t_{flicker} = 1$ שכן אנו בונים פונקציה רק עבור מצב בו

הרמזור אינו מהבהב ולכן המקרים $t_{done} = 1, t_{flicker} = 1$ אינם רלוונטיים במקרה זה.

כעת נבנה מפת קרנו.

$t_{done}, t_{flicker}$ person_present, car_present		$t_{done}, t_{flicker}$			
		00	01	11	10
00			\emptyset	\emptyset	\emptyset
01			\emptyset	\emptyset	\emptyset
11			\emptyset	\emptyset	\emptyset
10		1	\emptyset	\emptyset	\emptyset

הפונקציה המצומצמת היא:

$$f(t_{done}, t_{flicker}, person_present, car_present) = person_present * \overline{car_present}$$

טבלה ופונקציה עבור t_freeze:

t_done	t_flicker	person_present	car_present	t_freeze
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	∅
0	1	0	1	∅
0	1	1	0	∅
0	1	1	1	∅
1	0	0	0	∅
1	0	0	1	∅
1	0	1	0	∅
1	0	1	1	∅
1	1	0	0	∅
1	1	0	1	∅
1	1	1	0	∅
1	1	1	1	∅

כעת נבנה מפת קרנו.

person_present, car_present T_done, t_flicker	00	01	11	10
00		1		
01	∅	∅	∅	∅
11	∅	∅	∅	∅
10	∅	∅	∅	∅

הפונקציה המצומצמת היא:

$$f(t_{done}, t_{flicker}, person_present, car_present) = \overline{person_present} * car_present$$

2.4) מימשנו את הפונקציה בעזרת שתי פקודות בלבד (*not*, *and*) בהן השתמשנו פעם אחת בלבד. לפיכך, מכיוון שהרצת כל פקודה נמשכת בדיוק מחזור ששון אחד נקבל כי משך ביצוע החישוב הוא 2 מחזורי ששון.

תוצאת ההרצה:

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
0x04038263	beq x7 x0 68	beq t2, x0, finish
0x00151513	slli x10 x10 1	slli a0,a0,1
0x00f2f313	andi x6 x5 15	andi t1, t0, 0xF
0x00137e13	andi x28 x6 1	andi t3, t1, 1 # is_car
0x00135313	srli x6 x6 1	srli t1, t1, 1
0x00137e93	andi x29 x6 1	andi t4, t1, 1 # is_person
0x00135313	srli x6 x6 1	srli t1, t1, 1
0x00137f13	andi x30 x6 1	andi t5, t1, 1 # t_flicker
0x00135313	srli x6 x6 1	srli t1, t1, 1
0x00137f93	andi x31 x6 1	andi t6, t1, 1 # t_done
0xfffe4e13	xori x28 x28 -1	not t3, t3

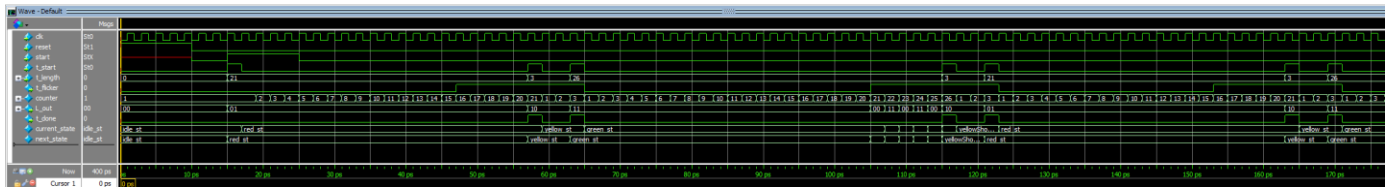
11

Registers
Memory

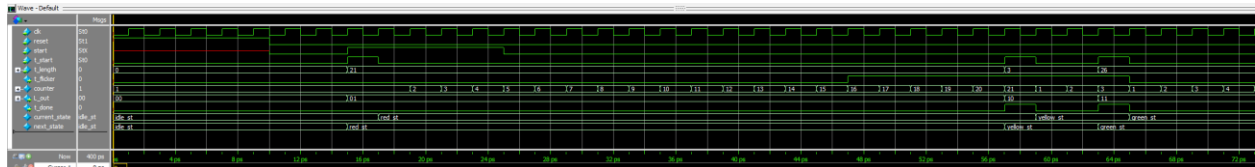
zero	0x00000000
ra (x1)	0x00000000
sp (x2)	0x7fffffff0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000000
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x0000000a
a1 (x11)	0x0000000b
a2 (x12)	0x00000000
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings
Hex

3.4) הרצה:

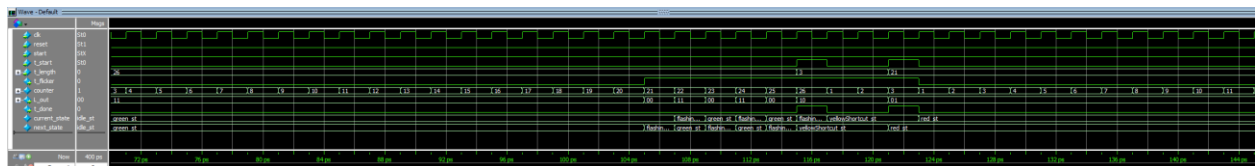


אתחול ומעבר ראשון של הרמזור עד מצב ירוק:



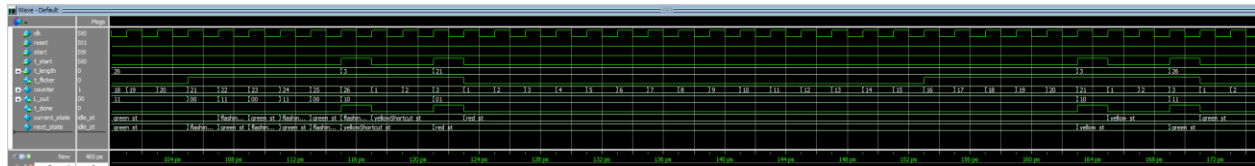
הסבר: לאחר התייצבות המערכת, מתחילה הספירה של ה- counter למשך 21 יחידות זמן. 5 יחידות זמן לפני הסוף, t_flicker עובר מ- 0 ל- 1. בסוף 21 יחידות הזמן, t_done עובר מ- 0 ל- 1 כדי לסמן את סיום שלב האדום, ו- t_start עובר גם הוא מ- 0 ל- 1 כדי לסמן לטיימר להתחיל לספור מחדש. לאחר מכן, עוברים לצהוב וממשיכים באופן דומה ל- 3 יחידות זמן. בפרק זמן זה t_flicker הינו עדיין 1, מכיוון שזמן השלב הצהוב קטן מ- 5 יחידות זמן. שוב ישנן עליות של t_done ו- t_flicker ועוברים לירוק.

מעבר להבהוב עד חזרה לאדום:



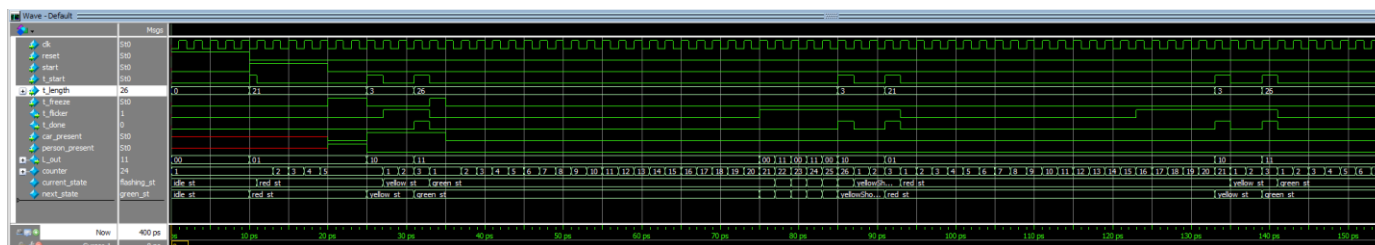
הסבר: ממשיכים בירוק, ושוב ב- 5 יחידות הזמן האחרונות t_flicker הינו 1 – והפעם ישנו הבהוב המתבטא במעבר בין המצבים flashing (בפועל כבוי) לבין green (בפועל ירוק דלוק). לאחר סיום ההבהוב עוברים למצב מעבר צהוב. לאחר מכן חוזרים לאדום.

ושוב בחזרה לירוק:

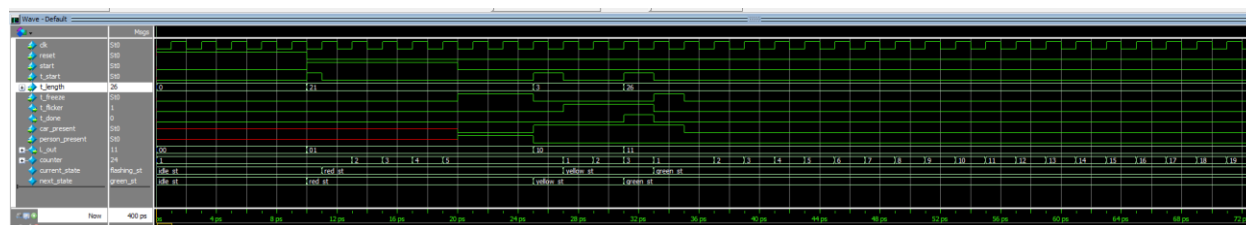


הסבר: ממשיכים באופן זהה כמו קודם, במעבר מאדום לצהוב לירוק.

3.7 הרצה:

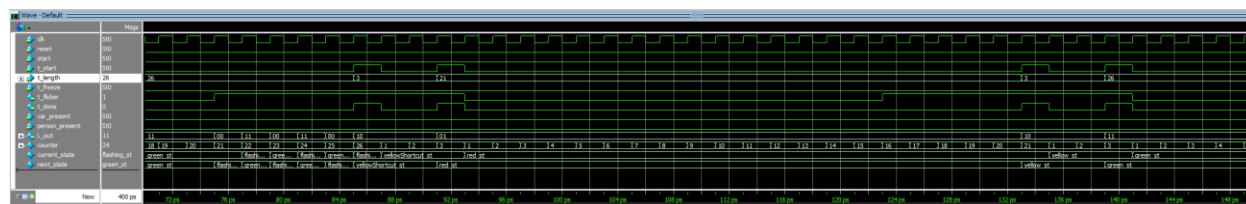


אתחול ומעבר מאדום לירוק:



הסבר: לאחר אתחול הכניסות המתאימות, והפעלת המערכת הגדרנו בהתאם לדרישות כי car_present יהי 0 ו-person_present יהיה 1. ברגע שכניסות אלו מוגדרות בערכים אלו הטיימר קופא ל-5 יחידות זמן עד לשינוי הרמזור נשאר על אור אדום (משתנה כאשר person_present הופך ל-0 ו-car_present הופך ל-1 – כלומר כאשר הולכי הרגל סיימו לחצות והגיעו מכוניות). לאחר מכן הרמזור מתחלף כרגיל לאור צהוב. מיד עם מעבר הרמזור לאור ירוק הרמזור קופא בשנית על אור ירוק עד אשר לא יהיו מכוניות שמחכות ברמזור (כלומר עד אשר תשתנה הכניסה car_present ל-0). לאחר מכן ממשיכה הריצה כרגיל.

מעבר מירוק להבהוב, בחזרה לאדום ושוב לירוק:



הסבר: נשים לב כי t_freeze כבוי מכיוון ש-car_present ו-person_present שניהם 0 (ובסיטואציה זו אין צורך להקפיא את הרמזור). מלבד זאת התהליך זהה למעברים שהוסברו קודם.