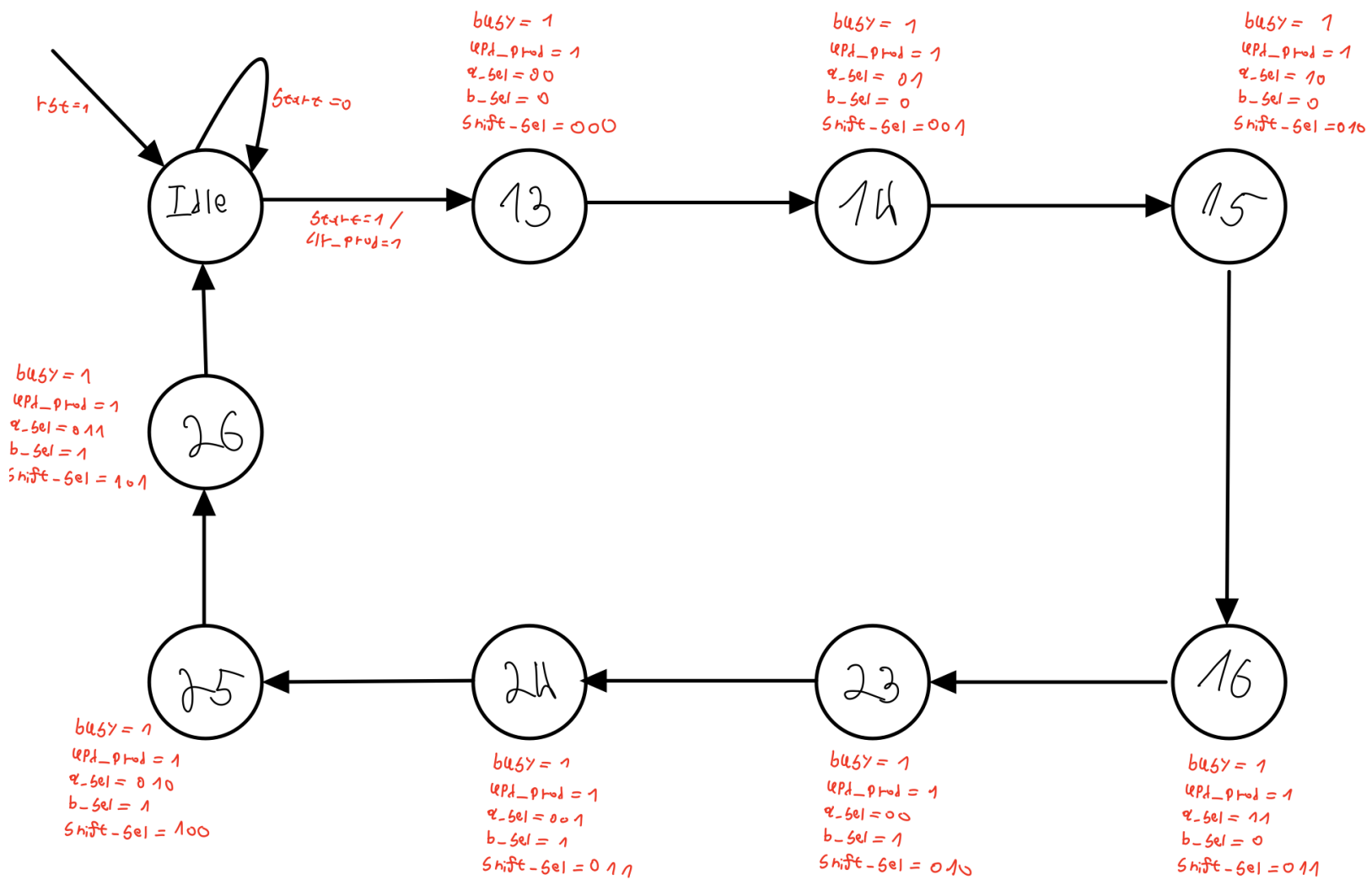


מגשים:

316225523	ליעד מוטורין
207477738	אור באוברג

2.1



למכונת המצבים לוקח 9 מחזורי שעון

נסמן ב- i את הכניסה שנבחרה מהבורר בגודל $1 \rightarrow 2k/8$ כאשר i נע בין 0 ל- $2k-1$.
 נסמן ב- j את הכניסה שנבחרה מהבורר בגודל $1 \rightarrow 2k/16$ כאשר j נע בין 0 ל- $k-1$.
 נאתחל את i ו- j ל- 0 תחילה, ונשתמש בלולאה מקוננת:

```

Int res=0
For(int i=0;i<2k/8;i++)
{
    For(int j=0;j<k/16;j++)
    {
        Int sum=mux2k/8->1[i]* 2k/16->1[j];
        Sum.ssl(<<)(8·i+16·j);
    }
}

```

חישוב סיבוכיות:

בקוד יש לולאה מקוננת שכל אחת מהם תלויה בא,
 ובתוך הלולאה יש פעולות סכימה , חיבור והוזזה שהם מתבצעות בסיבוכיות של $O(1)$,
 לכן זמן ריצת האלגוריתם הוא $O(k^2)$.
 מכיוון ש $N=2k$
 ניתן לראות שסיבוכיות האלגוריתם הוא $O(N^2)$

2.4

RunStepPrevResetDump

Address	Assembly	Comment
0x00829293	slli x5, x5, #8	slli t0, t0, #8
0x0ff2e293	ori x5, x5, #255	ori t0, t0, #0xff
0x0ffe7313	andi x6, x28, #255	andi t1, t3, #0xff
0x03d30fb3	mul x31, x6, x29	mul t6, t1, t4
0x005fffb3	and x31, x31, x5	and t6, t6, t0
0x008e5e13	srli x28, x28, #8	srli t3, t3, #8
0x0ffe7313	andi x6, x28, #255	andi t1, t3, #0xff
0x03d30e33	mul x28, x6, x29	mul t3, t1, t4
0x005e7e33	and x28, x28, x5	and t3, t3, t0
0x008e1e13	slli x28, x28, #8	slli t3, t3, #8
0x01cf8fb3	add x31, x31, x28	add t6, t6, t3
0x00100513	addi x10, x0, #1	finish: addi a0, x0, #1
0x000f8593	addi x11, x31, #0	addi a1, t6, #0
0x00000073	ecall	ecall # print integer

195065129

a5 (x15)0x00000000

a6 (x16)0x00000000

a7 (x17)0x00000000

s2 (x18)0x00000000

s3 (x19)0x00000000

s4 (x20)0x00000000

s5 (x21)0x00000000

s6 (x22)0x00000000

s7 (x23)0x00000000

s8 (x24)0x00000000

s9 (x25)0x00000000

s10 (x26)0x00000000

s11 (x27)0x00000000

t3 (x28)0x0af42f00

t4 (x29)0x0000feed

t5 (x30)0x00000000

t6 (x31)0x0ba07529

Display SettingsHex

הפועלה לוקחת כ-9 מחזורי שעון.

2.5

```
32 #####
33
34 # Start of your code
35
36     andi t1, t3, 0xff
37     beq t1,t3,a_msb0
38     andi t1,t4,0xff
39     beq t1, t4,b_msb0
40
41
42     andi t1, t3, 0xff
43     mul t6, t1, t4
44     and t6, t6, t0
45     srli t3, t3, 8
46     andi t1, t3, 0xff
47     mul t3, t1, t4
48     and t3, t3, t0
49     slli t3, t3, 8
50     add t6, t6, t3
51     j end
52
53 b_msb0:
54 mul t6,t3,t4
55 and t6,t6,t0
56 j end
57
58 a_msb0:
59 mul t6,t4,t3
60 and t6,t6,t0
61
62 end:
```

שינויים בקוד: במידה ושמונת הביטים ה-9-16 באחד הרגיסטרים שווה לאפס, אז ניתן לבצע במקום 2 מכפלות רק כפל אחד שמקצר משמעותית, נפריד למקרים:

מקרה 1: מקרה שבו יש קפיצה בגלל הרגיסטר הראשון, רק 5 הפקודות מתבצעות

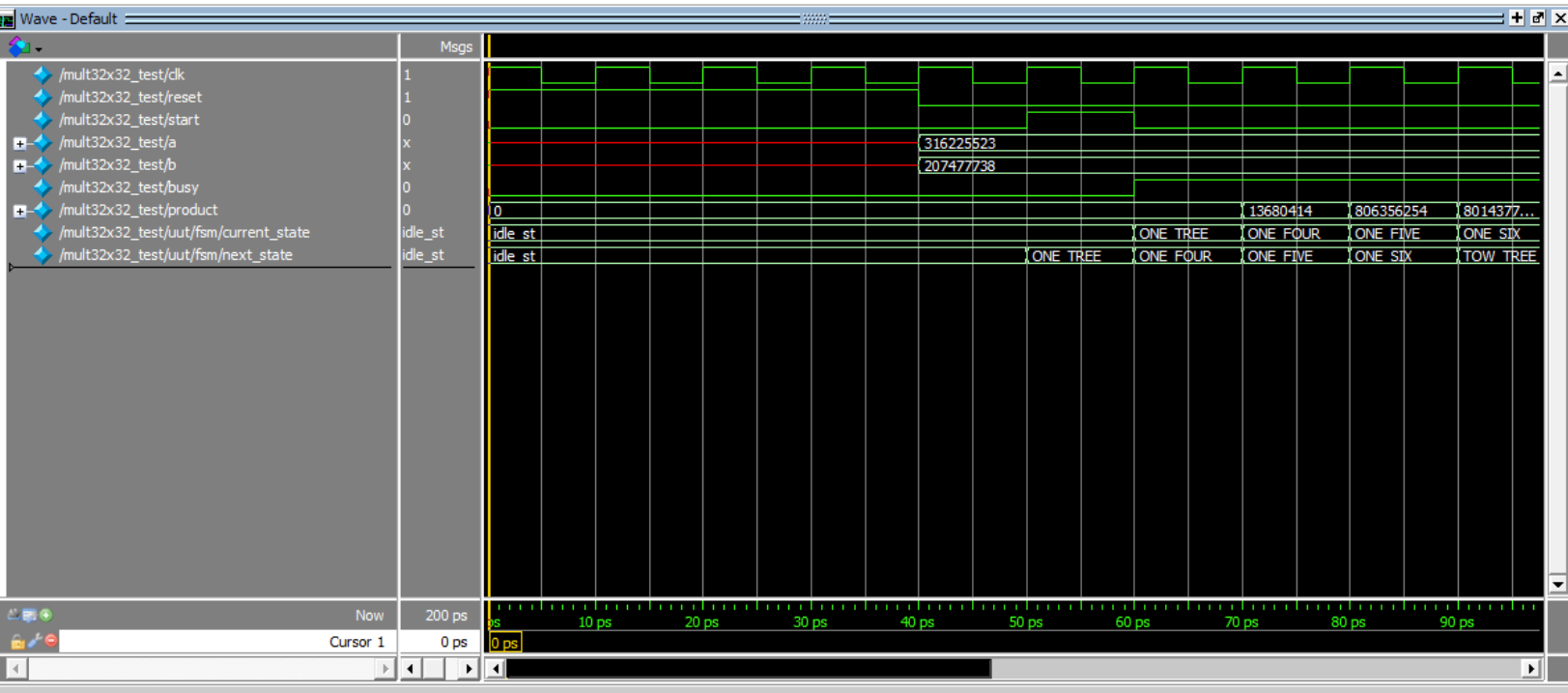
מקרה 2: במידה ויש קפיצה בגלל הרגיסטר השני, 7 פקודות מתבצעות.

מקרה 3: במידה ואף אחד מהם לא נכון, אז בגלל הבדיקות הנוספות 14 פקודות מתבצעות.

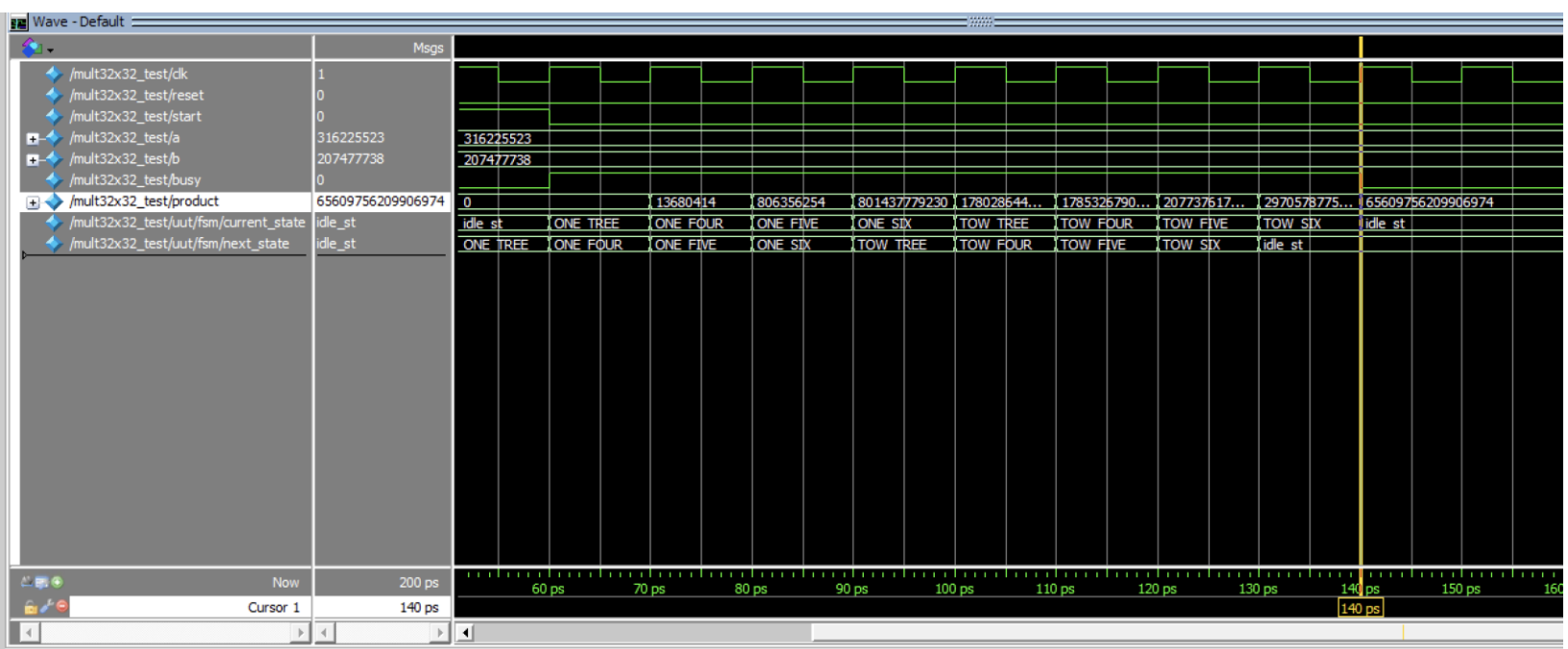
לכן, האם הדבר משתלם?
במידה ושמונת הביטים העליונים של אחד מהרגיסטרים שווה לאפס, אז הדבר משתלם שכן הפעולה תערך פחות מחזורי שעון.

אחרת, הפעולה לא תשתלם מכיוון שפעולה תערך יותר מחזורי שעון.

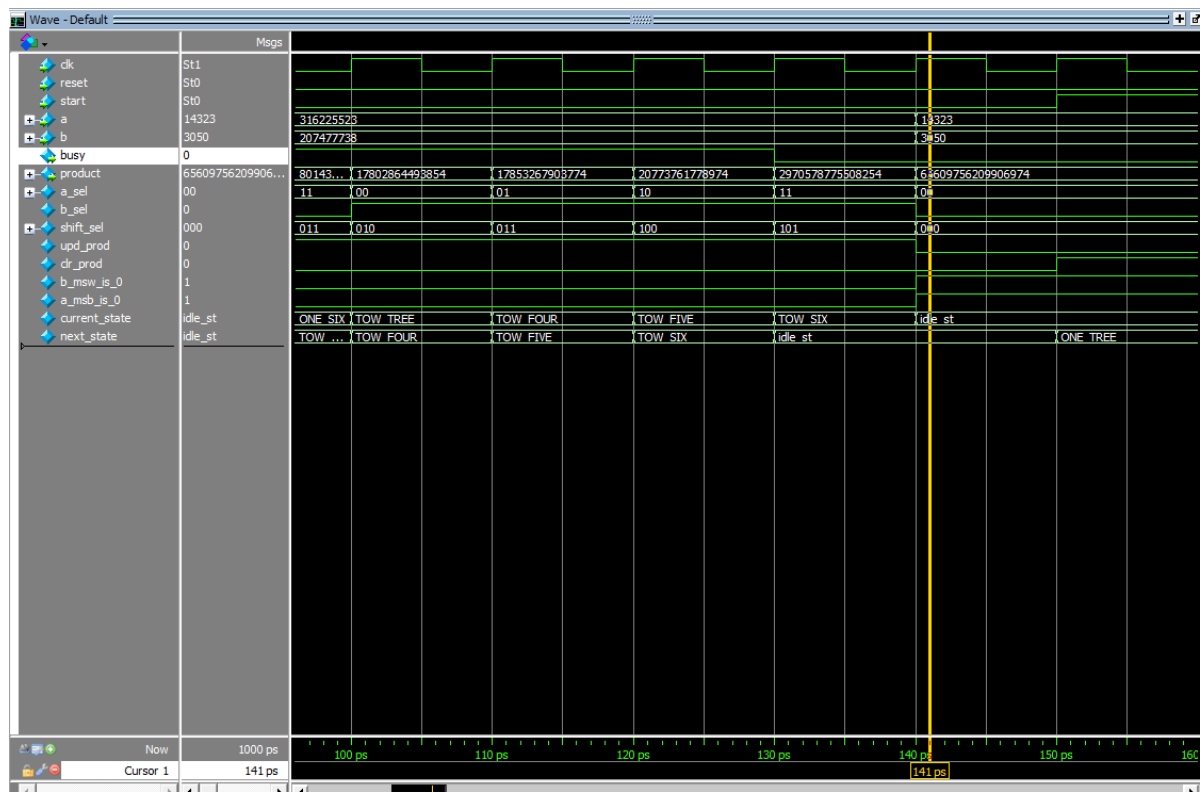
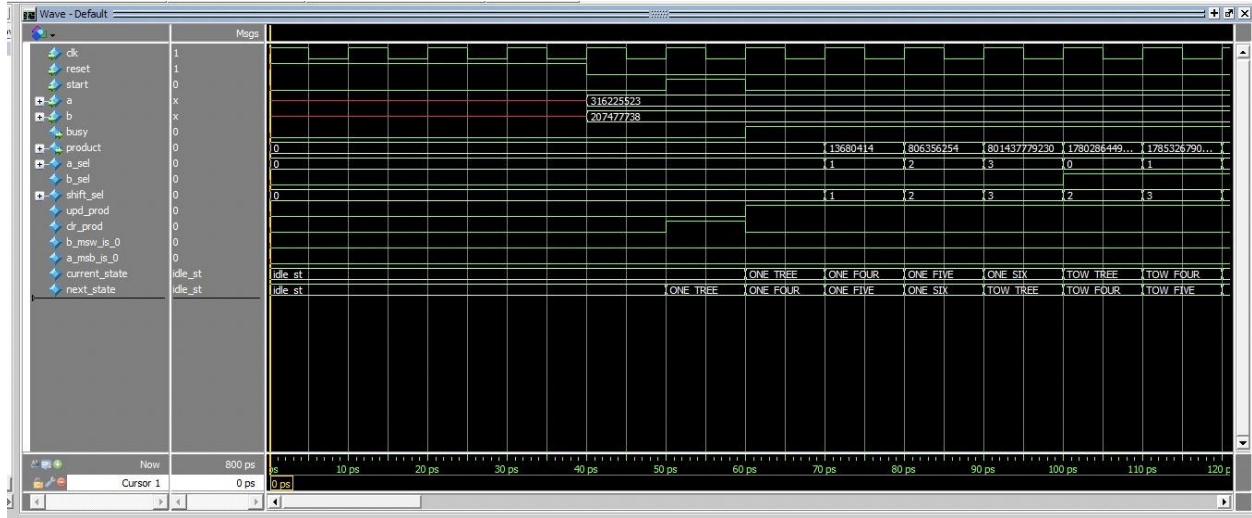
3.4



נצרך 2 צילומי מסך, בצילום הזה נסביר על ההתחלת הפעולה של המכפל.
נראה שכפי שנדרש רק לאחר 40 ננו שניות (אחרי עלית. השעון הרביעית) נכנסים הנתונים הרלוונטים (שני תעודות הזהות) והreset עובר מ1 ל0 כנדרש.
בכל הזמן הזה ניתן לראות שהחוסים שבהם עובר המידע של התז היו מנותקים, ובנוסף ה FSM היה במצב של השהייה שכן לא עבד, וכניסת הביזי הייתה על 0 כנדרש.
בעלית השעון הרביעית ניתן לראות שכניסת reset האסיכרוני יורד מאחד לאפס.
ברגע זה מכונת המצבים שלנו מתחילה את פעולתה.
לכן, מחזור שעון אחד לאחר מכן בעלית השעון החמישית, הסטארט עולה מ0 ל1 כפי שצוין בהדגמה שלכם ומחזור לאחר מכן יורד.
ברגע שהמצב הנוכחי של ה FSM משתנה מהשהייה, לאחר שש מחזורי שעון, אפשר לראות שברגע זה המכפל מתחיל לעבוד ולכן סיגנל busy עולה ל1.

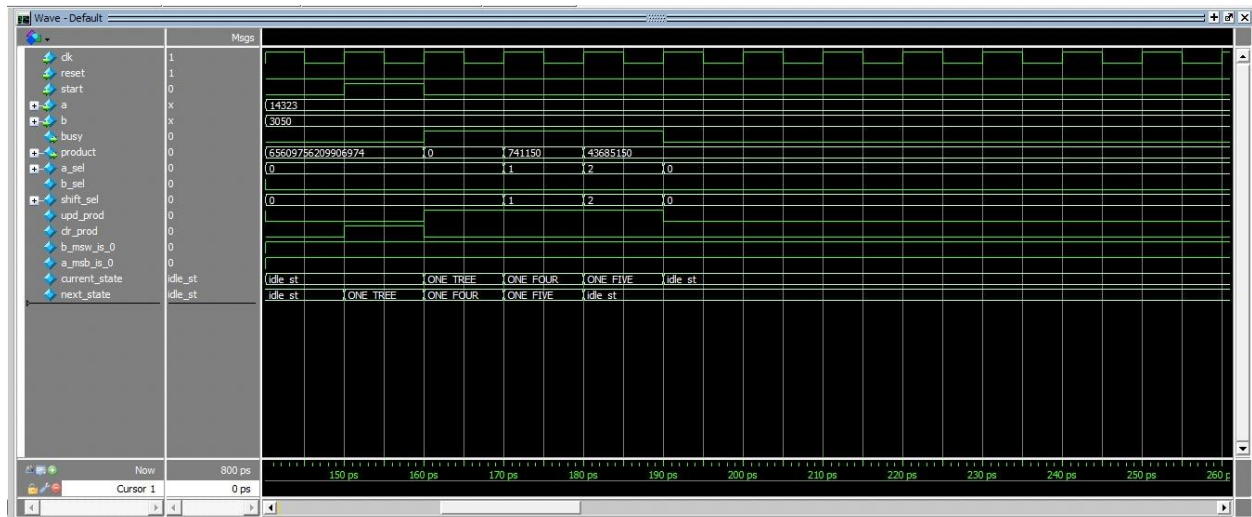


3.7



נשים לב, שפעולת הכפל הראשון של המכפל המהיר היא במדויק אותה הפעולה של המכפל הראשון, כצפוי.

(2 תמונות לתחילת ולסוף הכפל)
 בפרט, סיגנלי ה busy וה start מתנהגים בצורה זהה כצפוי.
 בנוסף למעברים הזחים בין המצבים של ה current, next state וכמובן שתוצאות המכפל זהה לסעיף הראשון.



כעת, נשים לב שבשונה מהפעמים הקודמות, בפעולה השניה של המכפל, מכיוון שהוא מזהה שגם הבית העליון של הרגיסטר הראשון וגם חצי המילה השניה של הרגיסטר השני הן 0, הפעולה נעשית בארבעה מחזורי שעון בלבד, ניתן לראות שבדומה לדיאגרמה ב2.2, המצב קופץ מ 15 למצב השהייה, בנוסף, כל סיגנלי ה busy וה start פועלים כהרגלם בדומה לסעיפים הקודמים. ניתן לראות שלבסוף תוצאות המכפל תקינה. לכן קפיצת המצבים עובדת בצורה תקינה - לפחות במקרה זה ולקפיצה הספציפית בה $a_msb_is_0==1, b_msw_is_0==1$.