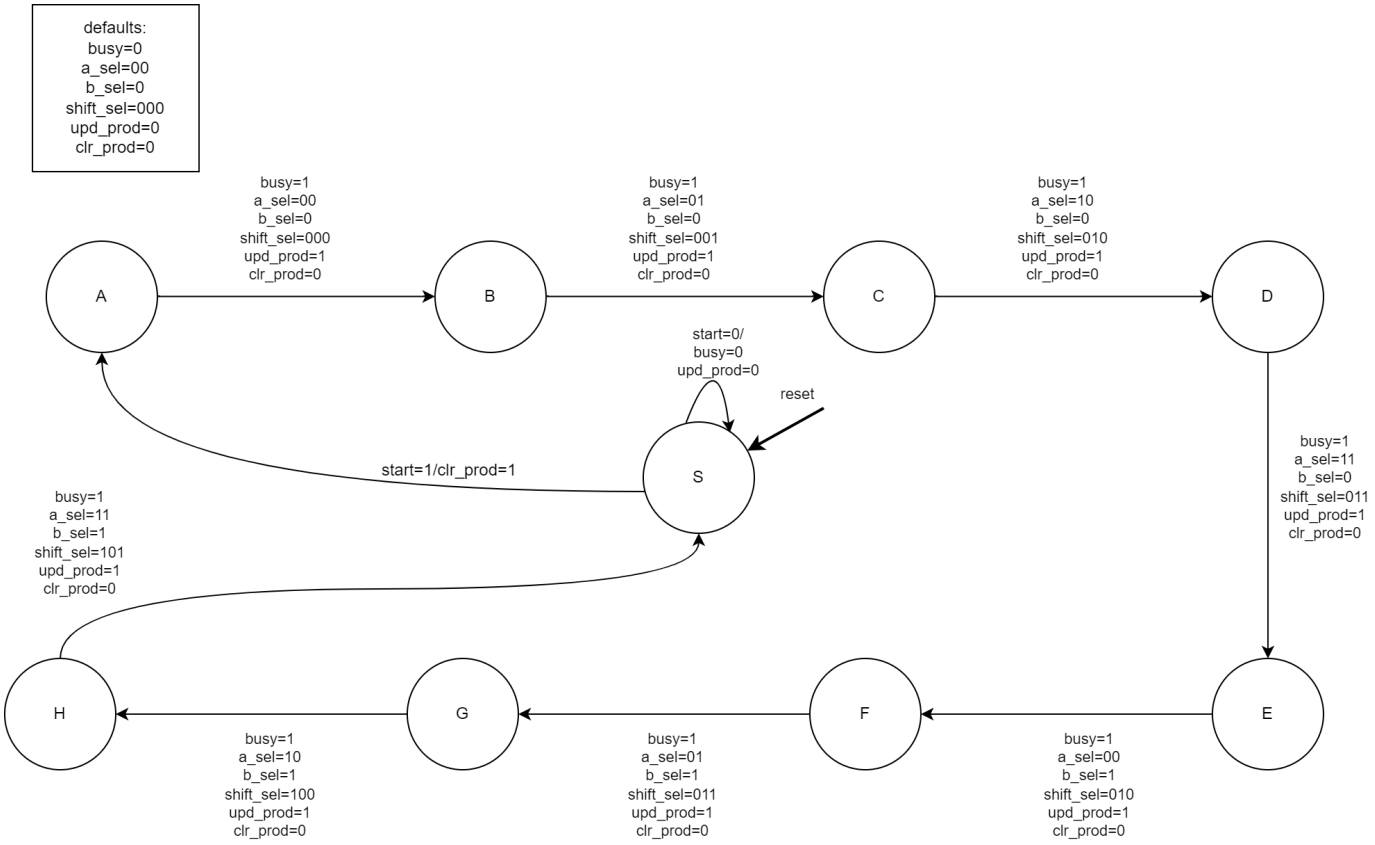
מערכות ספרתיות - סימולציה 2

מגישים:

עידו טאוסי	214008997
נועם ביטון	213745953

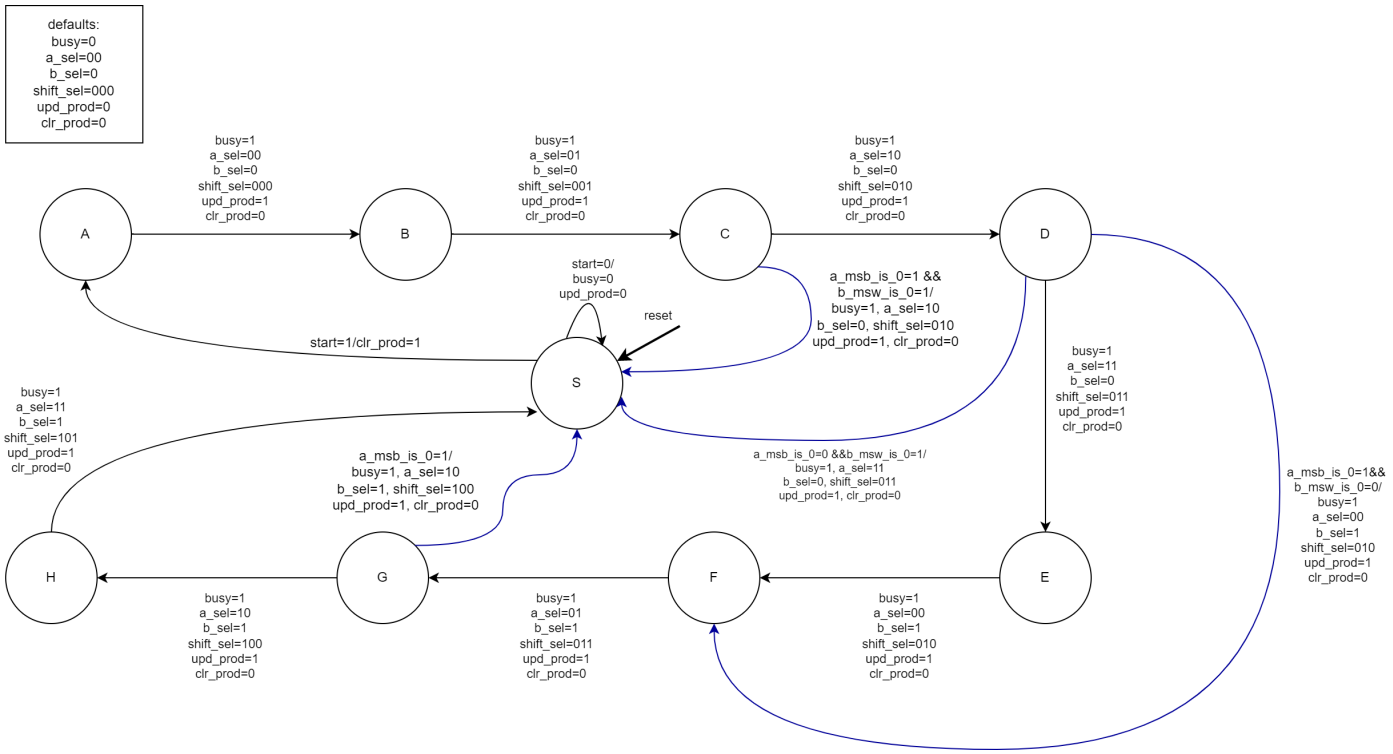
חלק יבש:

סעיף 2.1:



במכונת המצבים שיצרנו, פעולת הכפל הנדרשת שמתבצעת לוקחת 9 מחזורי שעון.

סעיף 2.2:



כאשר a_msb_is_0=0 && b_msw_is_0=0: ייקח לנו 9 מחזורי שעון, כמו בסעיף א'.

כאשר a_msb_is_0=0 && b_msw_is_0=1: ייקח לנו 5 מחזורי שעון.

כאשר a_msb_is_0=1 && b_msw_is_0=0: ייקח לנו 7 מחזורי שעון.

כאשר a_msb_is_0=1 && b_msw_is_0=1: ייקח לנו 4 מחזורי שעון.

<u>סעיף 2.3:</u>

האלגוריתם:

נסמן מספר אחד ב-a ואת המספר השני נסמן ב-b.

נתחיל בביצוע האלגוריתם, נאתחל משתנים $i = 0$, $sum = 0$.

בכל איטרציה כאשר $i = 0$ עד $i = \frac{N}{2} - 1$, ניקח את 16 הספרות של b החל מהמיקום ה-$16i$ עד המיקום ה-$16i + 15$.

כעת נאתחל משתנה $j = 0$ ובכל איטרציה כאשר $j = 0$ עד $j = N - 1$, ניקח את 8 הספרות של a החל מהמיקום ה-$8j$ עד המיקום ה-$8i + 7$. בכל איטרציה נבצע כפל של 16 הספרות הנוכחיות של b ב-8 הספרות הנוכחיות של a, נוכל לבצע זאת בעזרת המעבד שיודע לכפול מספרים בעלי 8 סיביות במספרים בעלי 16 סיביות. כעת, יש צורך לעשות shift left לתוצאה כך שנוכל לחבר אותה בצורה נכונה לסכום המכפלות עד כה, לכן נבצע shift left ב- $8j + 16i$. נכון לבצע הזזה זו מפני שיש לנו $8j + 16i$ ספרות לפני הספרות הנוכחיות שלנו שלא התייחסנו אליהן, לכן צריך לבצע הזזה שמאלה על מנת לקבל תשובה נכונה.

כעת, נוסיף את המכפלה הנוכחית שבוצע עליה shift left באיטרציה זו, לסכום כל המכפלות עד כה sum.

בסיום כל האיטרציות, החיצוניות והפנימיות, נקבל כי במשתנה sum נמצאת התוצאה של המכפלה הרצויה, לכן נחזיר אותה.

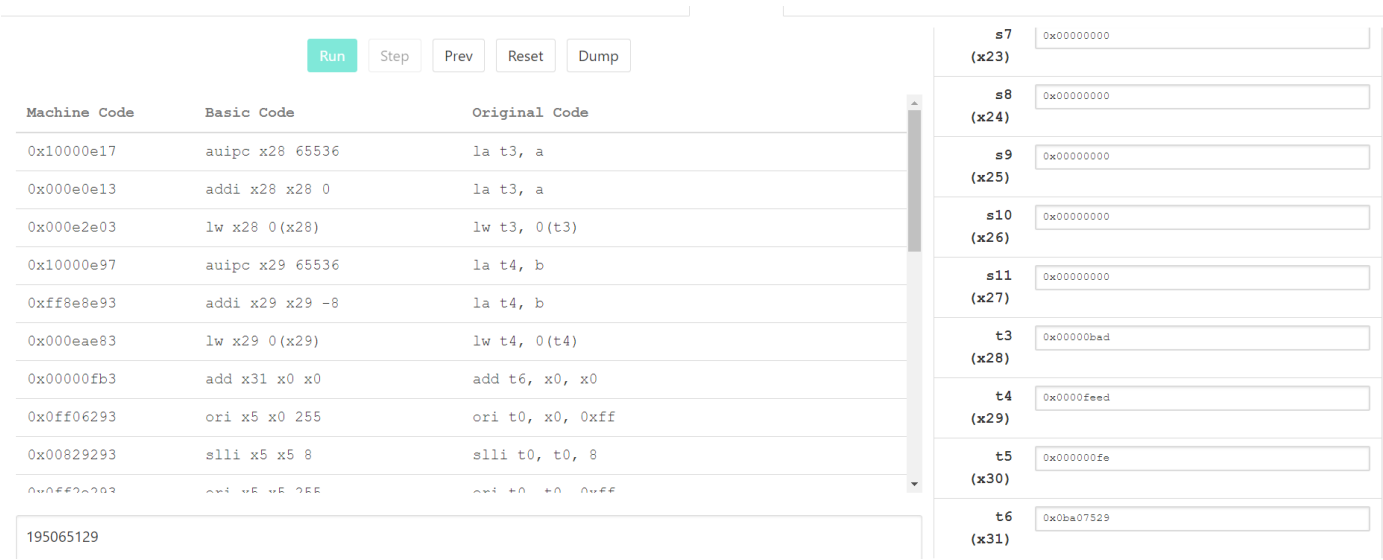
נשים לב כי ביצענו $\frac{N}{2}$ איטרציות, שבכל אחת מהן ביצענו N איטרציות, לכן בסך הכול ביצענו $\frac{N^2}{2}$ פעולות, לכן סיבוכיות זמן הריצה של האלגוריתם הוא $O(N^2)$.

ובפסודו קוד:

```
sum = 0;
current;
for(int i=0; i<N/2; i++)
{
    for(int j=0; j<N; j++)
    {
        current = a[8j:8j+7] * b[16i:16i+15];
        doShiftLeft(current, 8j+16i);
        sum+= current;
    }
}
```

סעיף 2.4:

צילום מסך של הסימולטור לאחר ההרצה:



ניתן לראות שאכן יצא לנו $t6 = 0x0BA07529$ כמצופה מחישוב הכפל: $0xBAD*0xFEED$

ביצענו את הפעולה ב-8 שורות כאשר כל שורה היא פקודה שזמן ביצועה הוא זמן מחזור אחד ולכן זמן ביצוע הפעולה הוא 8 מחזורי שעון.

סעיף 2.5:

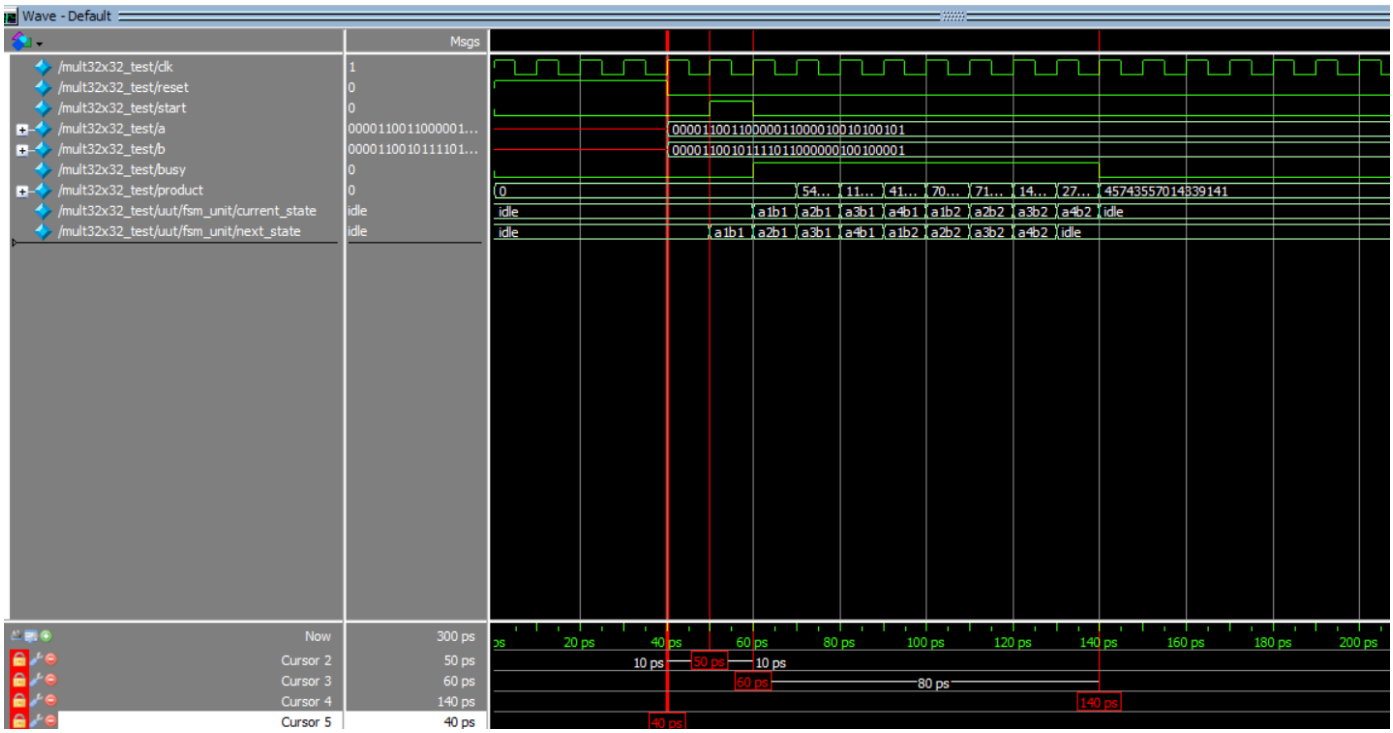
נוסיף בדיקה על הבית הראשון של a ועל הבית הראשון של b, נוכל לבצע זאת בעזרת בנייה של mask שיכיל $0xFF00$, ואז ביצוע AND bitwise, וכך נקבל את הבית העליון ב-a וב-b.
אם הבית העליון של אחד מהם שווה ל-0, נתייחס אליו בתור מספר של 8 ביטים, מפני ש-8 הביטים העליונים שלו הם 0, וכעת נוכל לחשב ישירות את המכפלה שלהם, מפני שאנו יודעים לבצע מכפלה של 8x16 באופן ישיר.

ניתן לבנות את ה-mask הרלוונטי ב-2 שורות, ואז נבצע AND עם a, ונבצע branch אם הוא שווה ל-0, לשורה בה יתבצע הכפל באופן ישיר של 8x16 כאשר a הוא המספר בעל 8 הסיביות, ואחריה נקפוץ ל-finish, סך הכול 5 שורות, לכן 5 מחזורי שעון. אם תוצאת ה-AND עם a לא שווה ל-0, נבצע AND של ה-mask עם b ונבצע branch אם התוצאה שווה ל-0, נקפוץ לשורה בה יתבצע הכפל באופן ישיר של 8x16 כאשר b הוא המספר בעל 8 הסיביות, ואחריה נקפוץ ל-finish, סך הכול 8 שורות, לכן 8 מחזורי שעון.

נשים לב כי בניית ה-mask וביצוע ה-AND ובדיקת השוואה ל-0, כולן יקרו לפני הקוד של הסעיף הקודם, שלוקח 8 מחזורי שעון, ולכן יאריכו את זמן הריצה שלו במקרה שבו הבית העליון של a וגם הבית העליון של b שונים מ-0.
שינוי זה יהיה משתלם רק במקרים בהם הבית העליון של a או של b הוא 0, במקרים אחרים הוא לא יהיה משתלם ואף יגרום להאטה, 8+8=16 מחזורי שעון בסעיף הזה לעומת 8 מחזורי שעון בסעיף הקודם.

חלק רטוב:

סעיף 3.4:



הגדרנו מחזור שעון להיות $10ps$.

ניתן לראות כי אות ה-$reset$ מאותחל ב-0 ודלוק 4 מחזורי שעון ($40ps$ שניות) ולאחר מכן יורד עד סוף הסימולציה.

עם ירידת אות ה$reset$ הוצבו בכניסות a ו-b ערכי תעודות הזהות שלנו.

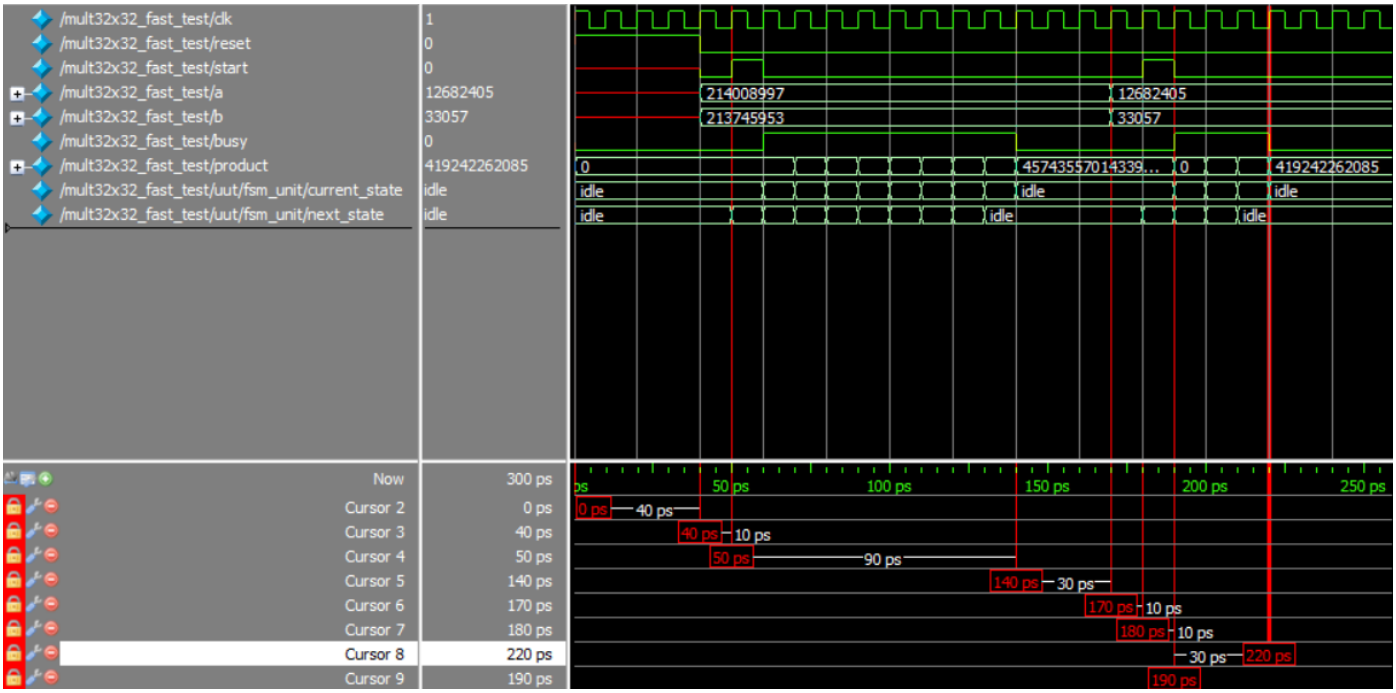
בנוסף, לאחר מחזור שעון נוסף ($50ps$ מתחילת הריצה) אות ה-$start$ עולה למשך מחזור שעון אחד. אות ה-$busy$ עולה מיד עד ירידת אות ה- $start$ והמכונה מתחילה את החישוב. לאחר 8 מחזורי שעון מעליית $busy$ נראה כי החישוב הסתיים והאות $busy$ יורד כאשר ב-$product$ מופיעה התוצאה הסופית של המכפלה.

ניתן לראות כי מכונת המצבים שלנו מתחילה במצב ה-$idle$, ולאחר עליית $start$ הוא עובר בכל שאר 8 המצבים, ולאחר מכן חוזר ל-$idle$.

תעודות הזהות שלנו הן: 214008997, 213745953.

ומכפלתן היא 45743557014339141, וזאת אכן התוצאה שקיבלנו ב-$product$ בסוף הריצה כאשר $busy$ ירד ל-0.

סעיף 3.7:



גם במקרה זה, הגדרנו את מחזור השעון להיות $10ps$.
ניתן לראות כי אות ה-$reset$ מאותחל ב-0 ודלוק 4 מחזורי שעון ($40ps$ שניות) ולאחר מכן יורד עד סוף הסימולציה.
עם ירידת אות ה$reset$ הוצבו בכניסות a ו-b ערכי תעודות הזהות שלנו.
בנוסף, לאחר מחזור שעון נוסף ($50ps$ מתחילת הריצה) אות ה-$start$ עולה למשך מחזור שעון אחד. אות ה-$busy$ עולה מיד עם ירידת אות ה- $start$ והמכונה מתחילה את החישוב. לאחר 8 מחזורי שעון מעליית $busy$ נראה כי החישוב הסתיים והאות $busy$ יורד כאשר ב-$product$ מופיעה התוצאה הסופית של המכפלה.
כמו בסעיף הקודם,
תעודות הזהות שלנו הן: 214008997, 213745953.
ומכפלתן היא 45743557014339141, וזאת אכן התוצאה שקיבלנו ב-$product$ בסוף הריצה כאשר $busy$ ירד ל-0.
ניתן לראות כי מכונת המצבים שלנו מתחילה במצב ה-$idle$, ולאחר עליית $start$ הוא עובר בכל שאר 8 המצבים, ולאחר מכן חוזר ל-$idle$.

לאחר מכן, ממתינים למשך זמן מחזור אחד ואז לאחר מחזור שעון נוסף אות ה-$start$ עולה למשך מחזור שעון אחד ואז ממתינים עוד 3 מחזורי שעון לסיום חישוב המכפלה הקודמת. כעת, מציבים את ערכי $a,\ b$ כאשר הבתים העליונים שלהם מאופסים, אות ה-$busy$ עולה מיד עם ירידת אות ה- $start$ והמכונה מתחילה את החישוב החדש. לאחר 3 מחזורי שעות מעליית $busy$ נראה כי החישוב הסתיים והאות $busy$ יורד כאשר ב-$product$ מופיעה התוצאה הסופית של המכפלה.
תעודות הזהות שלנו הן: 214008997, 213745953, ולאחר איפוס הבית האחרון של תעודת הזהות הראשונה והמילה האחרונה של תעודת הזהות השנייה נקבל : 12682405, 33057
ומכפלתן היא 419242262085 וזאת אכן התוצאה שקיבלנו ב-$product$ בסוף הריצה כאשר $busy$ ירד ל-0.
ניתן לראות כי מכונת המצבים שלנו מתחילה במצב ה-$idle$, ולאחר עליית $start$ הוא עובר בכל שאר 8 המצבים, ולאחר מכן חוזר ל-$idle$.