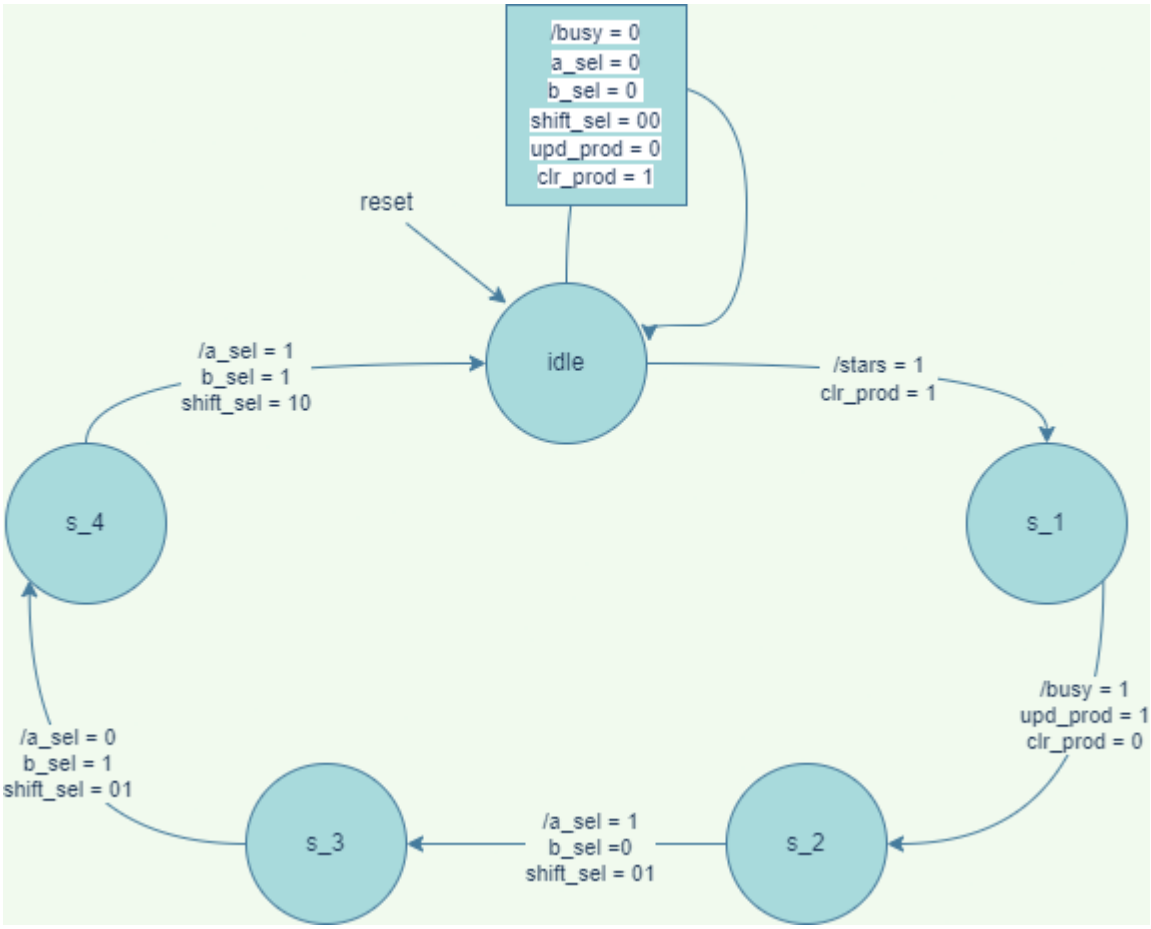
מגישים : אביאל כהן 316007988 , שירן דפט 208397414

1/15/2022

סימולציה 2 - מערכות ספרתיות

2.1

בהינתן התכן המתואר בחלק היבש, הכופל שני מספרים ברוחב 32 סיביות, והנחה כי כניסת ה-0 ב-$mux\ 2 \rightarrow 1$ בוחרת את הביט הגדול ביותר(msb) וכניסת ה-1 בוחרת בביט הקטן ביותר(lsb).
הנחה זו מתבססת על התרשים התכן וכן על פי $mux\ 4 \rightarrow 1$.

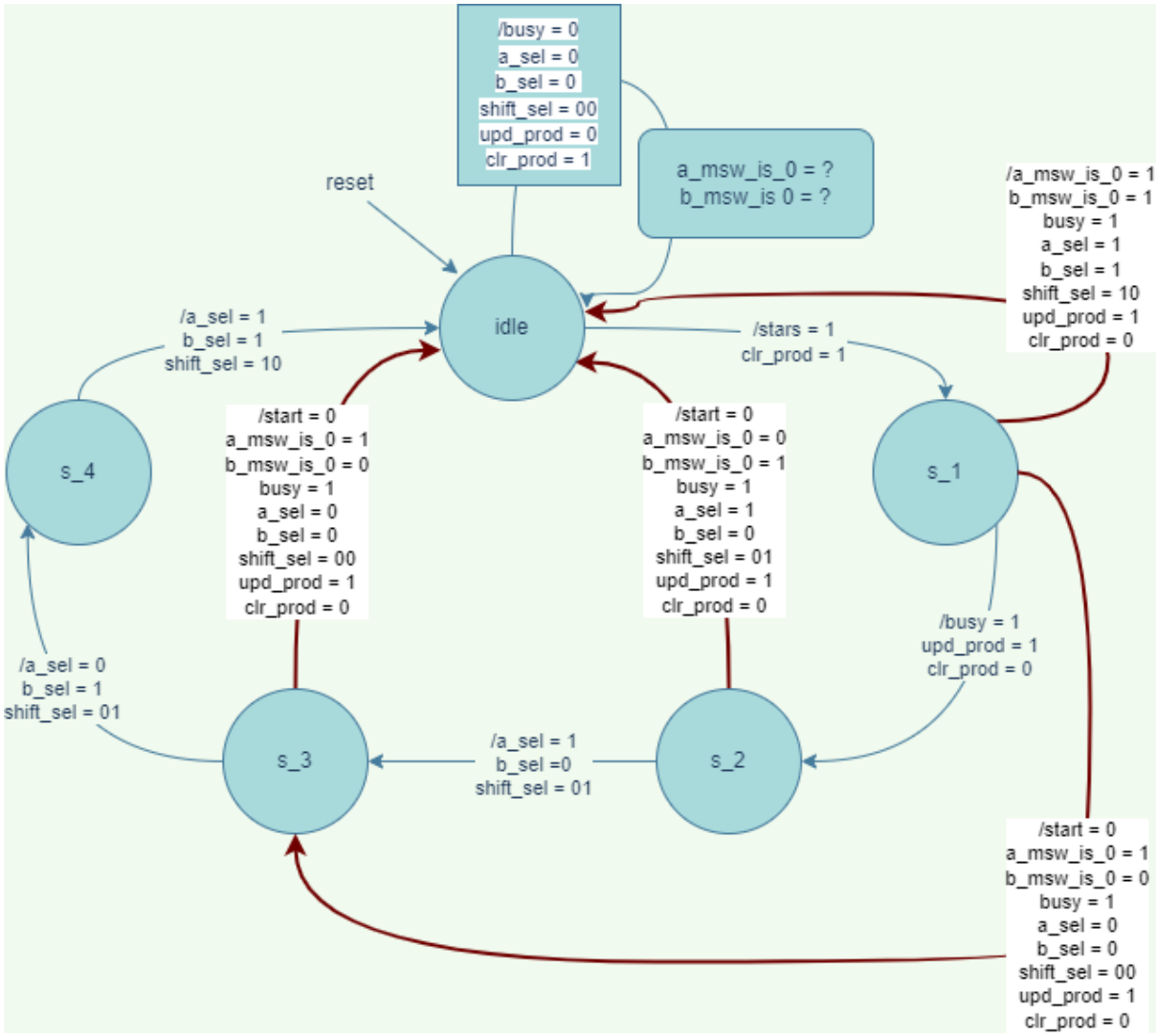


ניתן לראות כי רק לאחר ארבעה מצבים כלומר 4 מחזורי שעון יקבל busy את הערך 1.

2.2

נעזר בדיאגרמה מהסעיף הקודם אך נעזר בתוצאות הבאות,

- ראשית נכפול בין ה-lsw של a לבין ה-lsw של b, ואת תוצאת הכפל נשמור ברגיסטר.
- אם בצעד s_1 נקבל כי a_msw_is_0 וגם b_msw_is_0 הם 0, אז נחזור ל-s_0 ולא נבצע עוד חישובים, שכן כבר מצאנו.
- אם ה-msw של a הוא 0 xor זה של b: אז נבצע את החישוב בין ה-lsw של המילה עם הבה יש 0 ב-msw לבין המילה השניה.
- ובמידה ולא קיים 0 במילים אז נמשיך את המסלול כרגיל.



לכן בתאם לכך מספר מחזורי השעון משתנה בהתאם למציאת 0.

- במידה וה- msw של a ו-b הוא לא 0 אז יש לבצע מסלול שלם כלומר 4 מחזורי שעון.
- אם אחד מהערכים של msw_is_0_ הוא 1 אז נקבל בסה"כ 2 מחזורי שעון.
- אחרת שני הערכים של ה-msw_is_0_ הם 1, ונקבל 1 מחזורי שעון. פעולת כפל בין שני מילים באורך 8.

2.3

יהיו שני מספרים בגודל 8N כך שמתקיים כי $N = 2k, k \in \mathbb{N}$ ומעבר עם יכולת לכפול שני מספרים בגודל של 8 ו- 16 סיביות. לכן, נרצה כל פעם לחלק את המספרים שלנו לבלוקים של 8 סיביות ו-16 סיביות.
מהנתון על $N = 2k, k \in \mathbb{N}$ נקבל כי $\frac{N}{2} = k$ נתבונן במקטעים בני 16 סיביות במילה שאותה נחלק ל-k חלקים, ובמילה השנייה נחלקה ל-N מקטעים ובכל מקטע 8 סיביות.

<u>אלגוריתם החישוב של מכפלת המספרים</u>

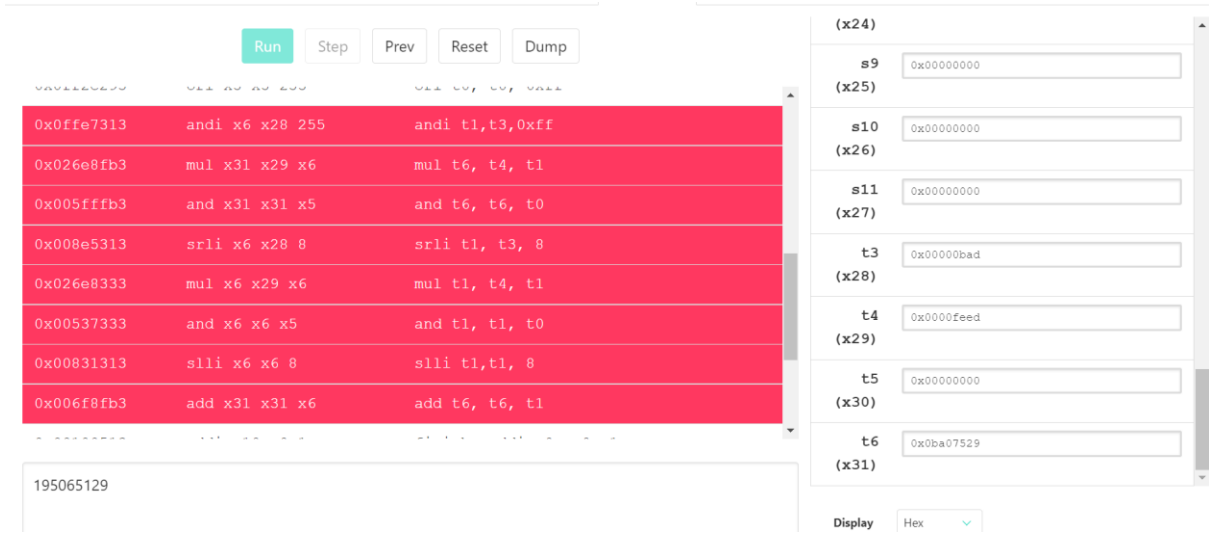
נסמן, $a \triangleq split\ the\ word\ into\ blocks\ of\ 8\ bit\ , b \triangleq split\ the\ word\ into\ blocks\ of\ 16\ bit$

נרוץ בלולאה פנימית וחיצונית, כך שנכפול 16 סיביות ב-8 סיביות לאורך כל המילה השניה תוך כדי מעבר על הבלוקים, בחיבור התוצאה יש לבצע הזזות לפי מיקום המכפלה ולחבר לתוצאה הסופית.
נשים לב כי מדובר באיטרציה פנימית בגודל N, וחיצונית בגודל $\frac{N}{2}$, ולכן נסיק כי הזמן הדרוש לביצוע המכפלה הינו $o(N^2)$.

זהו pseudo code המתאר את התהליך של האלגוריתם בעזרת קוד, ניתן לראות כי בהחלט מתקיים סיבוכיות זמן ריצה של $o(N^2)$.

```
unsigned int num_a[8N];
unsigned int num_b[8N];
unsigned int result = 0;

for(int i = 0; i < N ; i += 2){
    for(int j = 0; j < N ; j++){
        int temp = multiplier(num_a[i], num_b[j]);
        result += shiftLeft(temp, 8 * (j + 2i));
    }
}
return result;
```

2.4



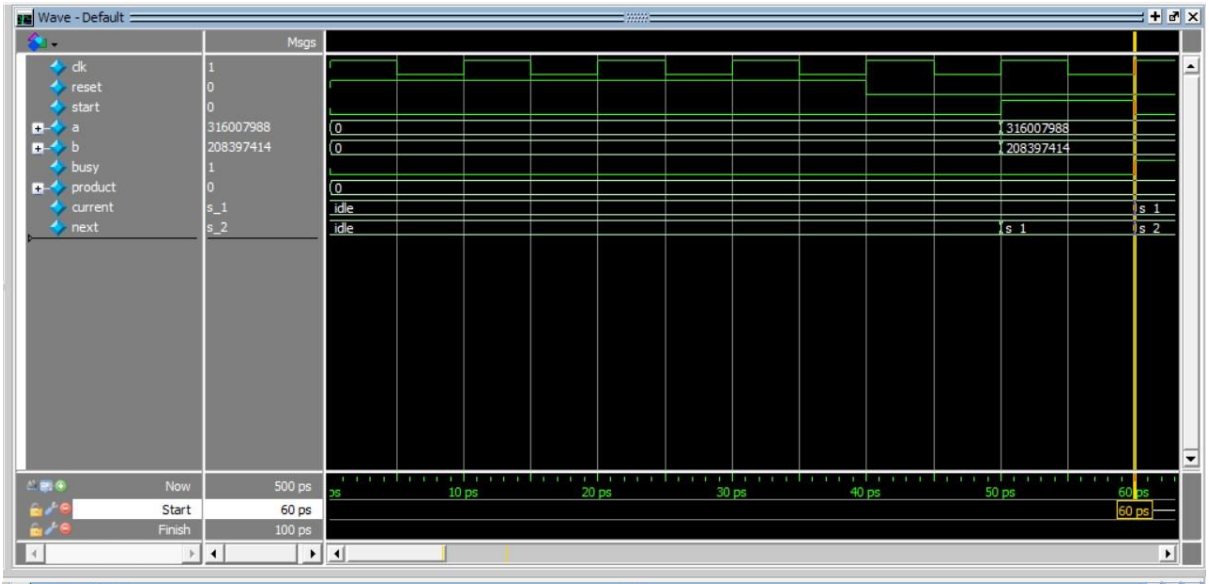
נתבונן, מספר הפקודות הוא זה המתאר את מחזורי השעון שהם במקרה זה 8.

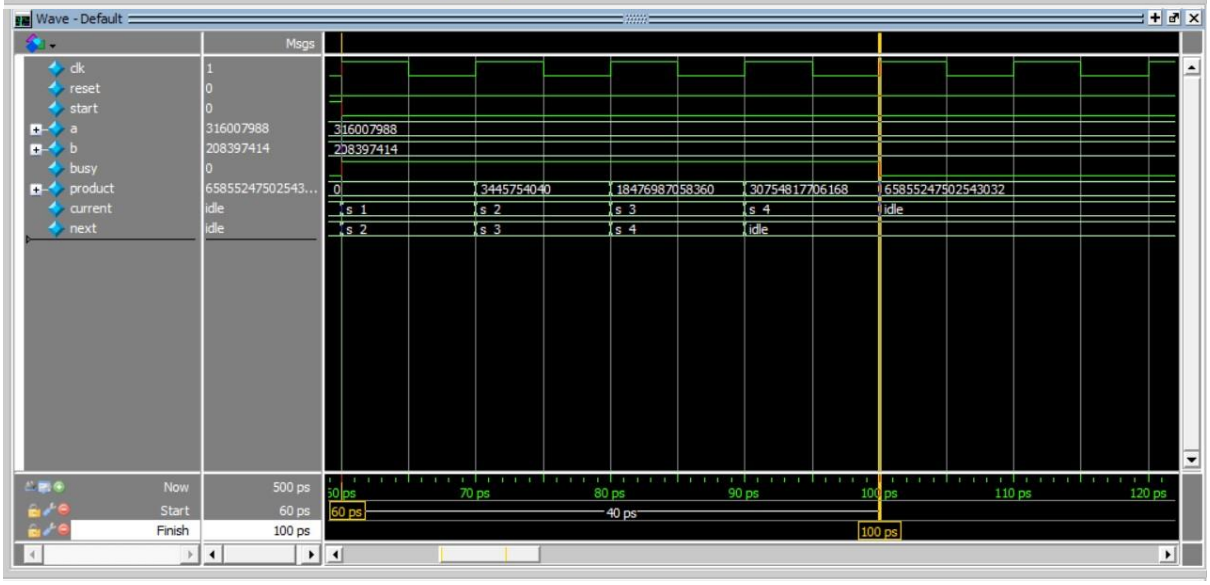
2.5

ניתן להוסיף את הפקודה beq לקוד, כך לאחר בדיקה של 8 הסיביות העליונות של המילים a או b אז נוכל לדלג על חישוב מהספר הבא.

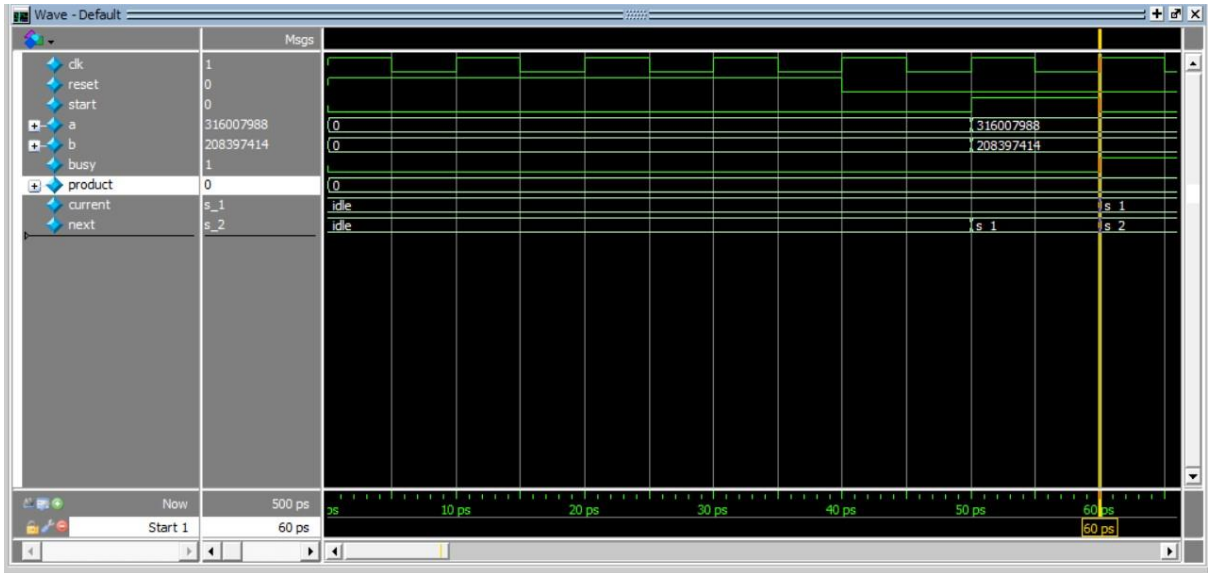
```
#####################
# Start of your code

        andi t1,t3,0xff
        mul t6, t4, t1
        and t6, t6, t0
        srli t1, t3, 8
        beq t1, x0, finish;
        mul t1, t4, t1
        and t1, t1, t0
        slli t1,t1, 8
        add t6, t6, t1
# End of your code
#####################
```

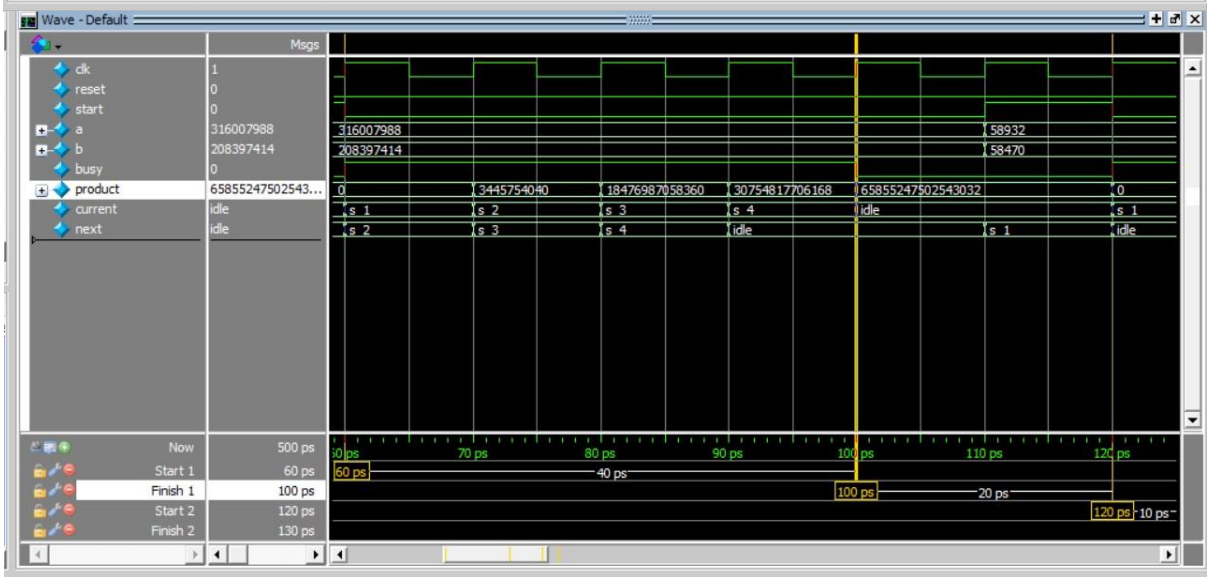
3.4

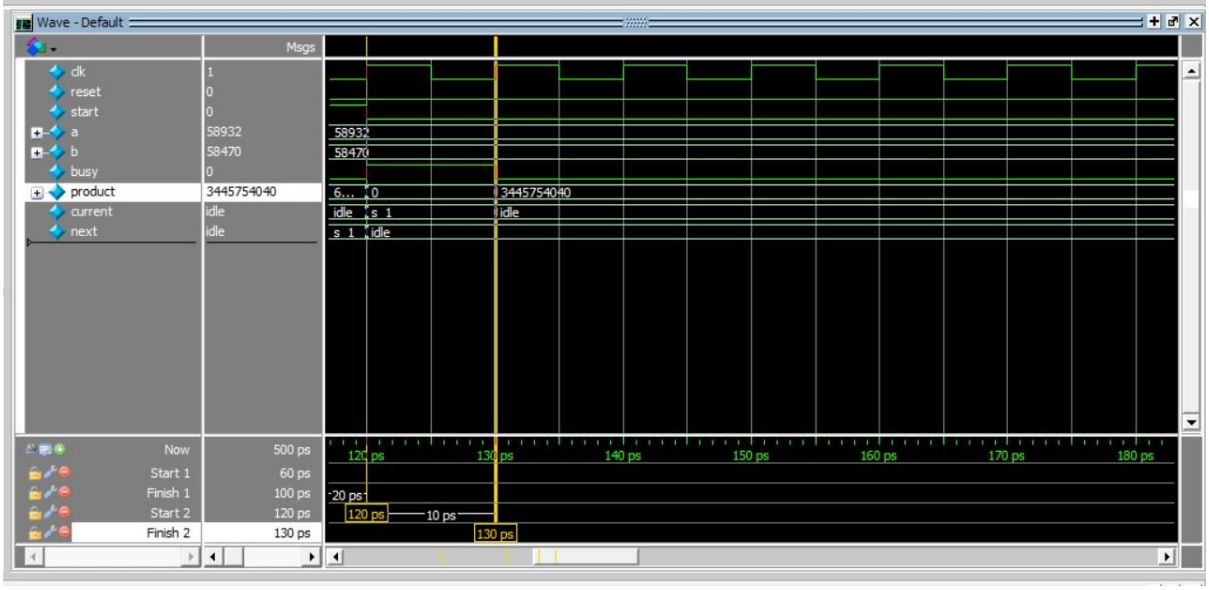




המערכת פועלת כפי שציפינו, המכפלה בין 2 תעודות הזהות הינה נכונה וניתן לראות כי לוקח למערכת 4 מחזורי שעון בכדי לבצע את הפעולה כצפוי.


Wave - Default
Msgs
clk 1
reset 0
start 0
a 316007988
b 208397414
busy 1
product 0
current s_1
next s_2
Now 500 ps
Start 1 60 ps


Wave - Default
Msgs
clk 1
reset 0
start 0
a 316007988
b 208397414
busy 0
product 65855247502543...
current idle
next idle
Now 500 ps
Start 1 60 ps
Finish 1 100 ps
Start 2 120 ps
Finish 2 130 ps


Wave - Default
Msgs
clk 1
reset 0
start 0
a 58932
b 58470
busy 0
product 3445754040
current idle
next idle
Now 500 ps
Start 1 60 ps
Finish 1 100 ps
Start 2 120 ps
Finish 2 130 ps

כמו בסעיף הקודם ניתן לראות כי המערכת אכן מוציאה תוצאות כפל נכונות, בנוסף לכך גם ניתן לראות שהפעולה מחולקת ל-2 מקרים כמצופה.

עבור המקרה הראשון המערכת מבצעת את המכפלה ואכן עוברת בכל המצבים.

עבור המקרה השני מכיוון שיש לנו מכפלה של אפסים ותכננו את המערכת בצורה מהירה המערכת עוברת רק במצב אחד ולכן פועלת בצורה הרבה יותר מהירה.