1/15/2022

סימולציה 2 - מערכות ספרתיות

מגישים : אביאל כהן 316007988 , שירן דפט 208397414

# 2.1

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

Diagram

Description automatically generated

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.

Diagram, schematic

Description automatically generated

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 2מחזורי 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 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 נקבל כי נתבונן במקטעים בני 16 סיביות במילה שאותה נחלק ל-k חלקים, ובמילה השנייה נחלקה ל-N מקטעים ובכל מקטע 8 סיביות.

אלגוריתם החישוב של מכפלת המספרים

נסמן,

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 , וחיצונית בגודל , ולכן נסיק כי הזמן הדרוש לביצוע המכפלה הינו .

זהו pseudo code המתאר את התהליך של האלגוריתם בעזרת קוד, ניתן לראות כי בהחלט מתקיים סיבוכיות זמן ריצה של .

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

Graphical user interface, application, table

Description automatically generated

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

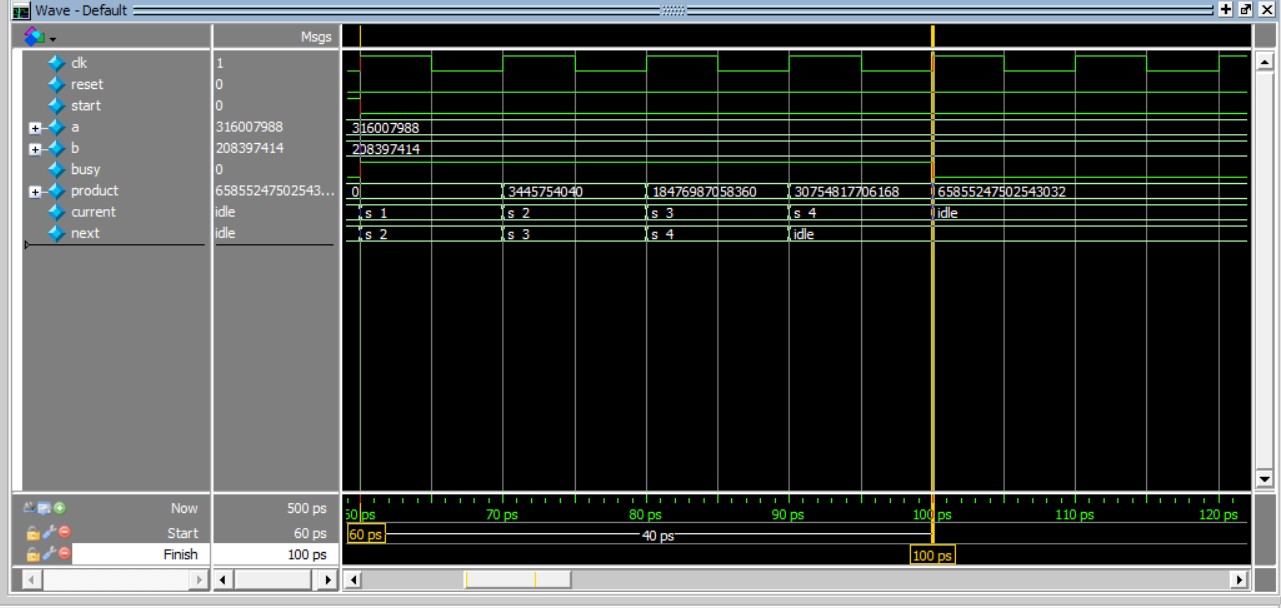
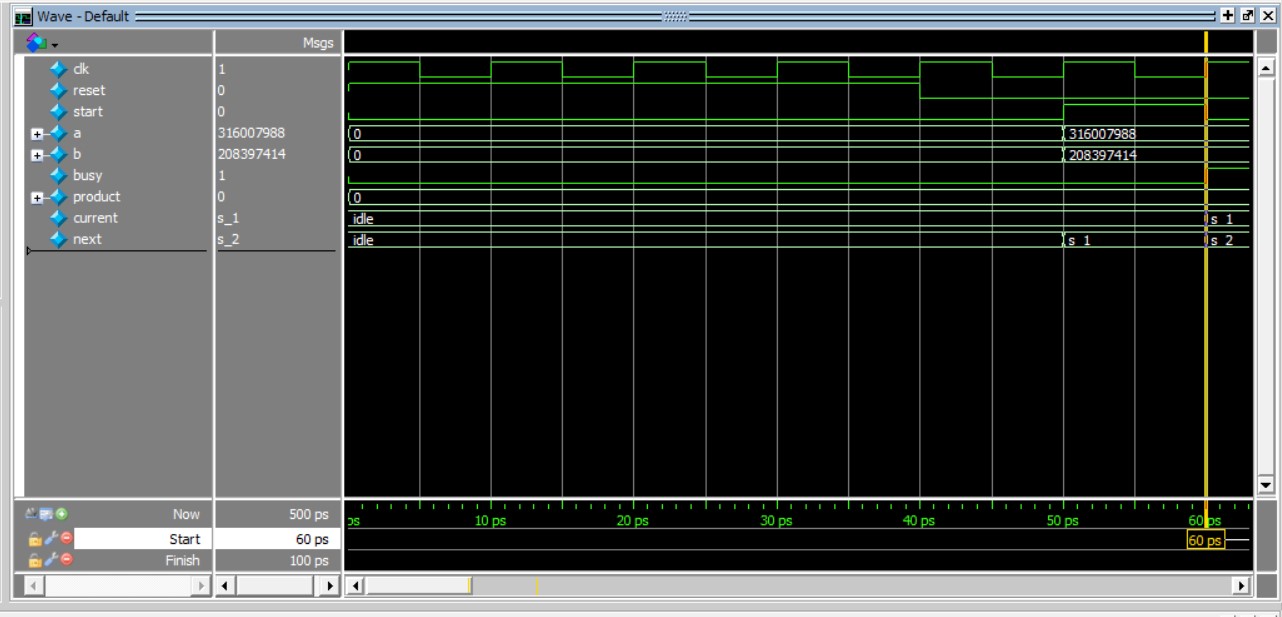
# 2.5

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.

A screenshot of a computer

Description automatically generated with medium confidence

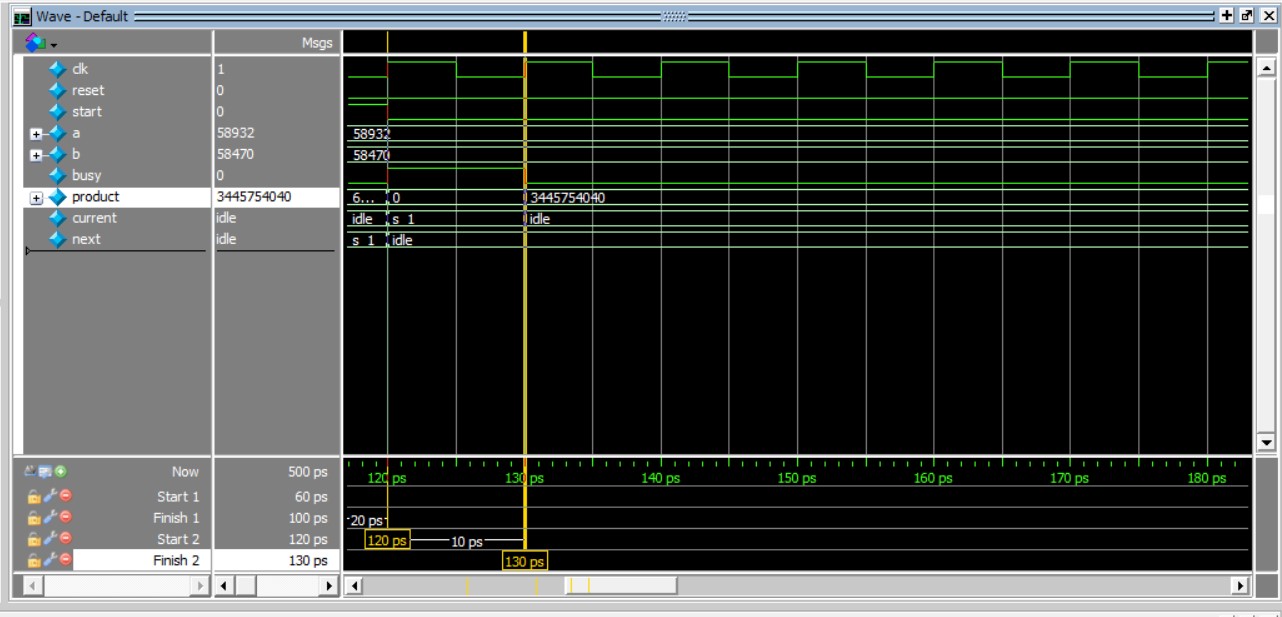
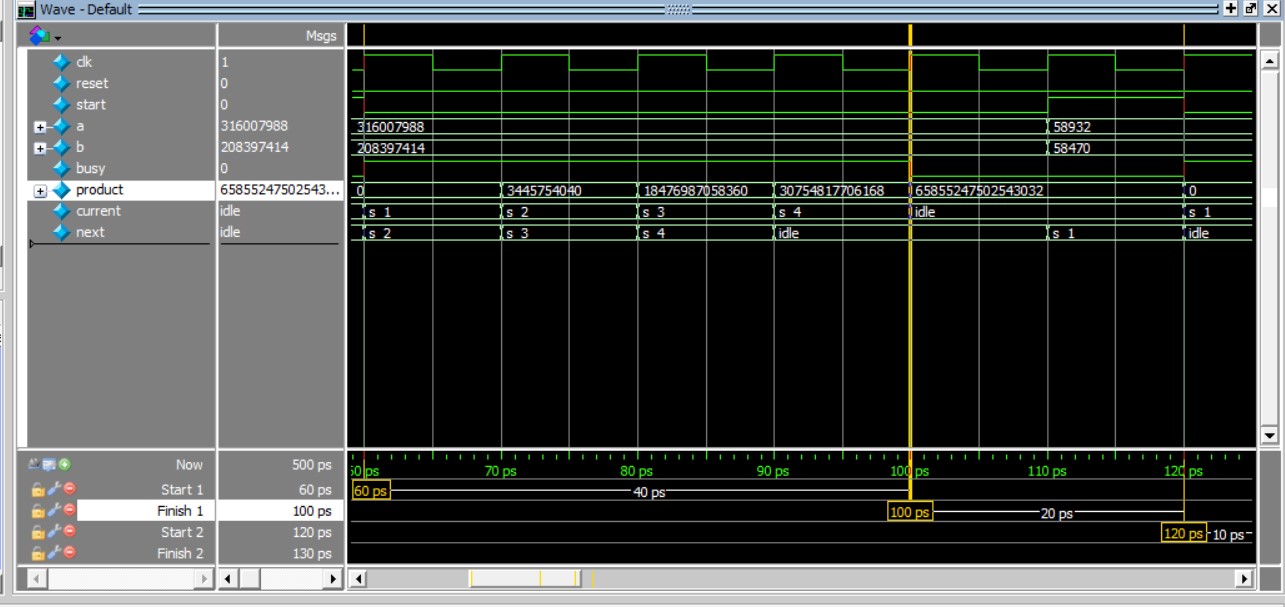
# 3.4



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

# 3.7

Diagram

Description automatically generated

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.