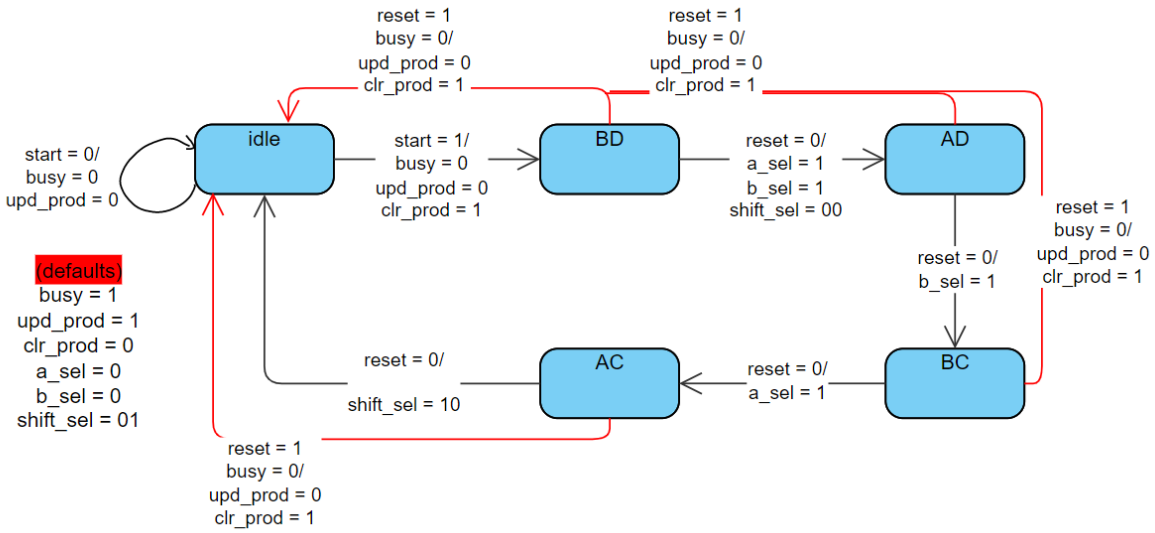
209146471	אלון הרטמן
207923566	תומר אברשקין

חלק יבש

2.1

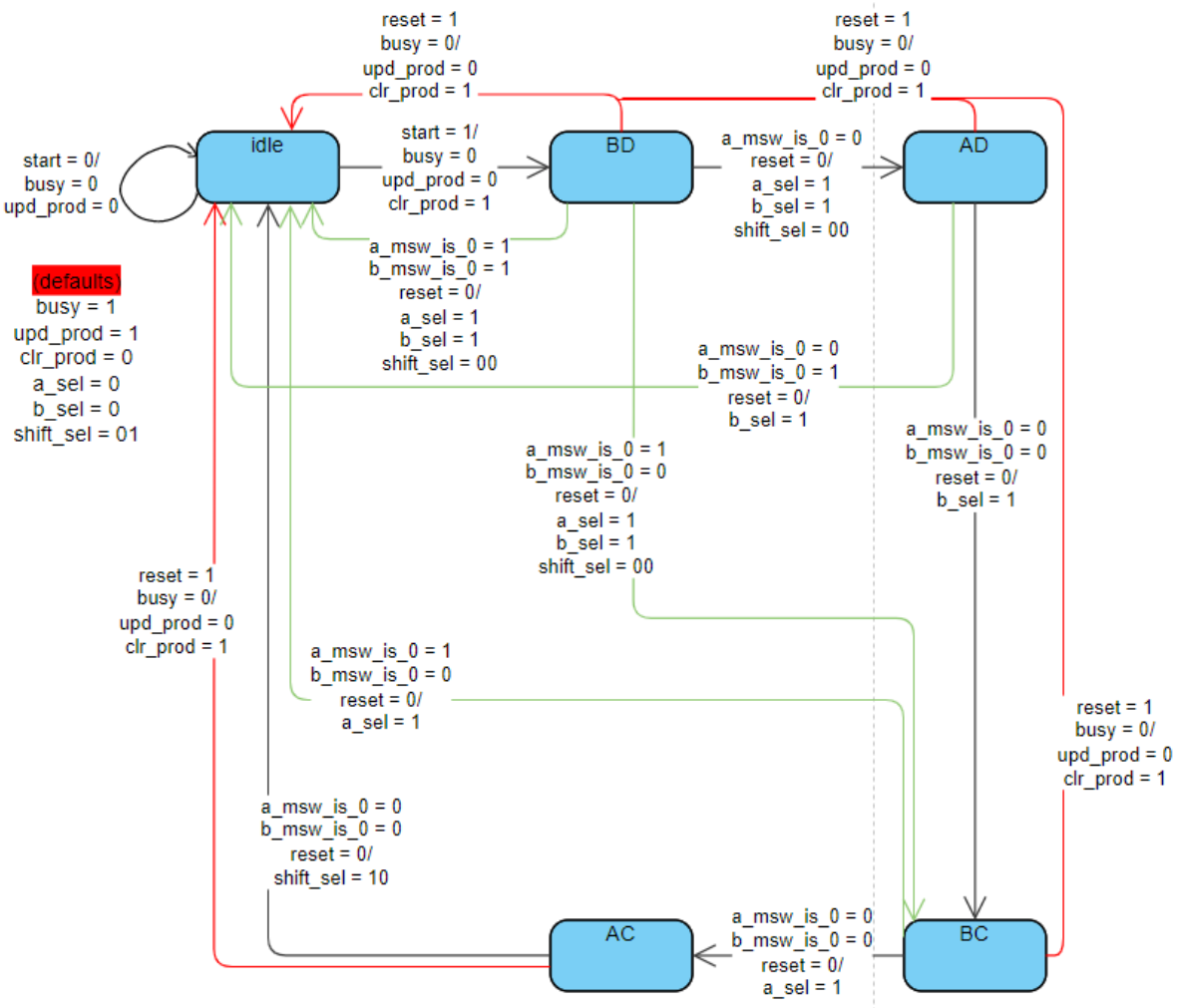
דיאגרמת המצבים של מכונת המצבים:



החישוב יקח 4 מחזורי שעון, כלומר במשך 4 מחזורי שעון היציאה busy תהיה על ערך '1'.

2.2

דיאגרמת המצבים של מכונת המצבים החדשה:



מספר מחזורי השעון בהם busy=1 הוא:

4 אם $a\_msw\_is\_0 = b\_msw\_is\_0 = 0$

2 אם $a\_msw\_is\_0 \neq b\_msw\_is\_0$

1 אם $a\_msw\_is\_0 = b\_msw\_is\_0 = 1$

2.3

נתאר את האלגוריתם באמצעות pseudo code:

נסמן את הקלטים בתור $a = a_{N-1}a_{N-2} \dots a_1a_0$, $b = b_{N-1}b_{N-2} \dots b_1b_0$, כאשר גודל a, b הוא $8N$, וגודל כל a_i, b_i הוא 8 סיביות.

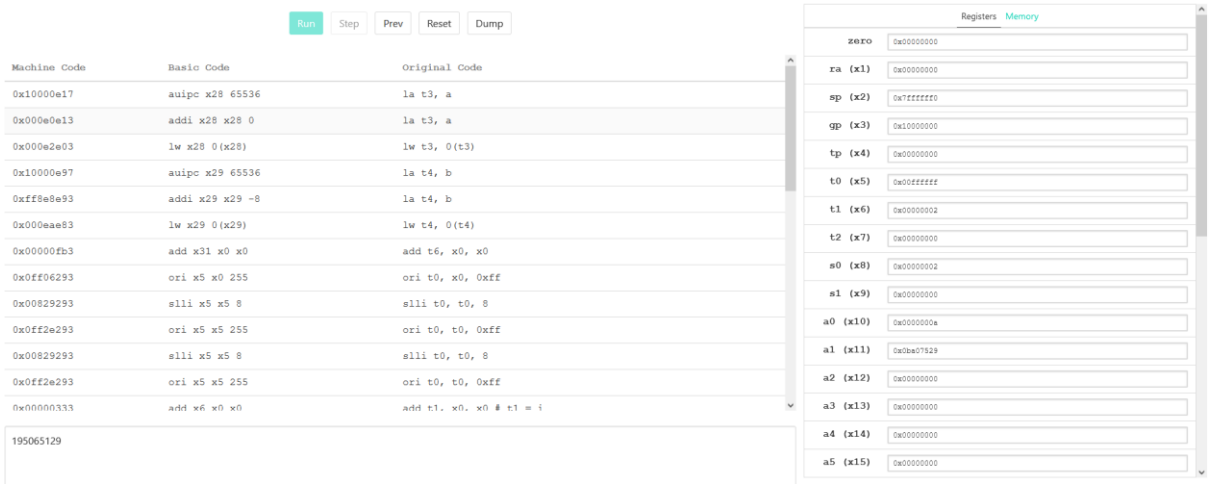
כמו כן נסמן $shift\_left(input, bits)$ בתור פונקציה המקבלת קלט $input$ ועושה לו הזזה $bits$ סיביות שמאלה.

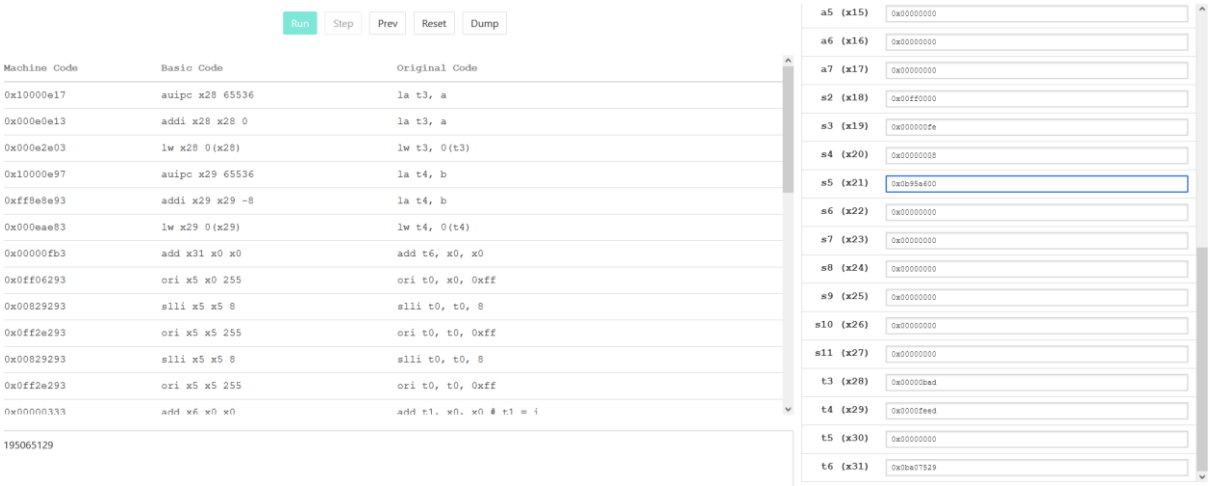
לבסוף נסמן $a_{2j+1}a_{2j}$ בתור המספר הבינארי המיוצג על ידי שרשורם.

```
int result, i, j;
for(i = 0; i < N; ++i) {
    for(j = 0; j < N/2; ++j) {
        int temp = shift_left(b_i · (a_{2j+1}a_{2j}), 8i + 8 · 2j);
        result += temp;
    }
}
```

הסיבוכיות של האלגוריתם היא $O(N^2)$ כיוון שרצים על לולאה שמתבצעת $O(N)$ פעמים ובתוכה רצים על לולאה שמתבצעת $O(N)$ פעמים גם היא. בסך הכל - $O(N^2)$.

2.4





פעולת הכפל לוקחת 25 מחזורי שעון, כלומר 25 פקודות מתבצעות.

2.5

נסמן – a=xy כאשר x אלו 8 הסיביות השמאליות של a ו-y אלו הימניות ובאופן דומה b=wz.

נפריד למקרים:

- אם x=0, z=0: נקבל ש-a,b הם באורך 8 ביטים. נחשב את תוצאת mul על a ו-z, נכניס ל-t_6 ונסיים.
- אם x=0, z≠ 0: נקבל ש-a באורך 8 ו-b באורך 16. נחשב את תוצאת mul על b ו-y, נכניס את התוצאה ל-t_6 ונסיים.
- אם $x \neq 0, z = 0$, נקבל ש-a באורך 16 ו-b באורך 8. נעשה כמו במקרה הראשון.

אחרת, נעשה את התוכנית המקורית מהסעיף הקודם.

קיבלנו שצריך להוסיף 3 בדיקות ולטפל בהם.

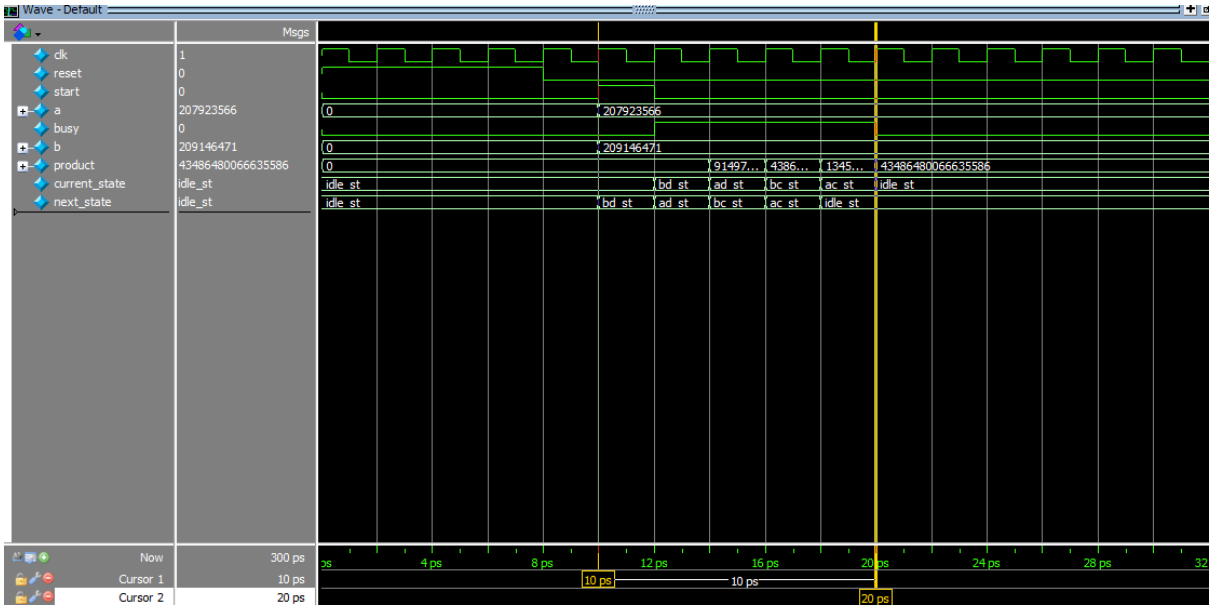
בכל אחד מהמקרים יהיה חישוב יותר פשוט ומהיר מאשר החישוב מהסעיף הקודם, אך אם אף אחד מהמקרים לא מתקיים, הוספנו בדיקות "מיותרות" לחישוב המקורי.

יחסית לכל המקרים האפשריים (יותר מ-2 מיליארד תוצאות אפשריות של קומבינציות של שני מספרים בינאריים באורך 16), המקרים לעיל, שחוסכים לנו זמן, הם בערך 33 מיליון מקרים. כלומר ברוב המקרים אנו ניתקל בחישוב ארוך יותר ולא נחסוך זמן. לכן השינוי הזה לא בהכרח משתלם.

חלק רטוב

3.4

תוצאות הסימולציה:



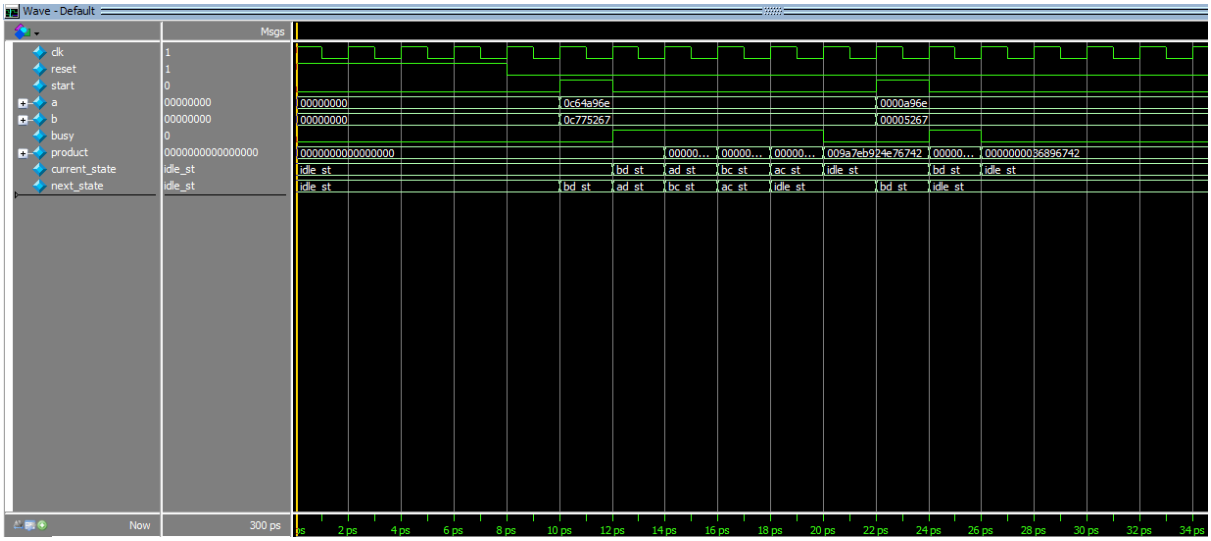
הסבר התוצאה:

כפי שניתן לראות, בתחילת הסימולציה הערכים מאותחל כמתבקש, לאחר 4 מחזורי שעון הסיגנל reset יורד ל-0 ולאחר מחזור שעון נוסף הסיגנל start עולה, הכניסות a ו-b מקבלות את ערכי תעודות הזהות שלנו והחישוב מתחיל.

ניתן לראות שהמכונה עוברת בכל המצבים כפי שתיארנו בסעיף 2.1 ובסוף החישוב, כלומר לאחר 4 מחזורי שעון מעליית busy ל-1, מתקבלת התוצאה הרצויה ונשמרת עד לעליית start או reset.

3.7

תוצאות הרצת הסימולציה:



הסבר התוצאה:

כפי שניתן לראות, בתחילת הסימולציה הערכים מאותחל כמתבקש, לאחר 4 מחזורי שעון הסיגנל reset יורד ל-0 ולאחר מחזור שעון נוסף הסיגנל start עולה, הכניסות a ו-b מקבלות את ערכי תעודות הזהות שלנו והחישוב מתחיל.

לאחר כ-5 מחזורי שעון המכונה מתייצבת על הפלט הרצוי, ולאחר מחזור שעון אחד מכניסים לה חישוב חדש בו שני חצאי המילה העליונות של תעודות הזהות מאופסים ולאחר מחזור שעון אחד התוצאה הרצויה מתקבלת בפלט. זה תואם את ניתוח הזמן שעשינו למערכת בסעיף 2.2.

בסוף כל חישוב, לאחר ש-busy יורד ל-0, מתקבלת התוצאה הרצויה ונשמרת עד לעליית start או reset.