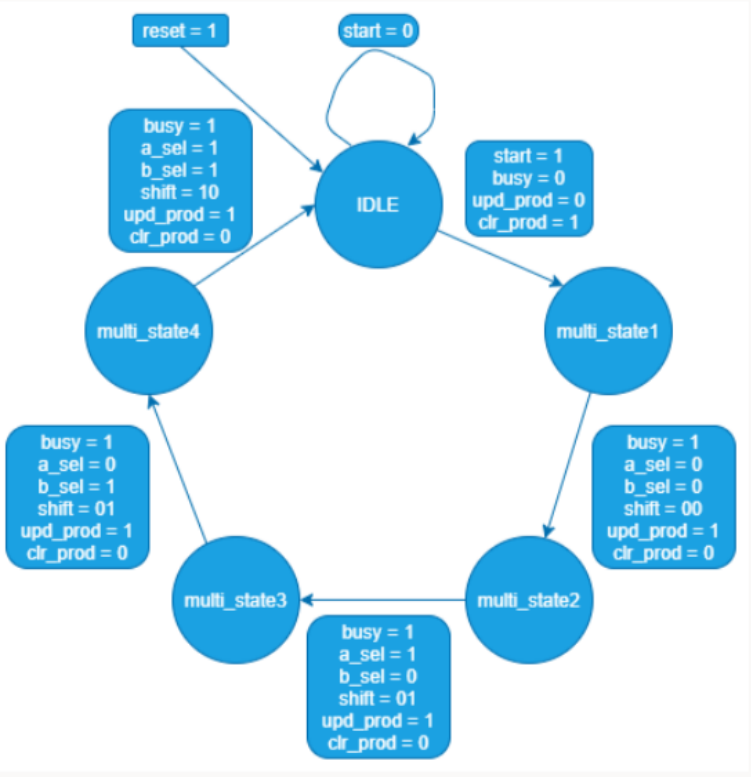
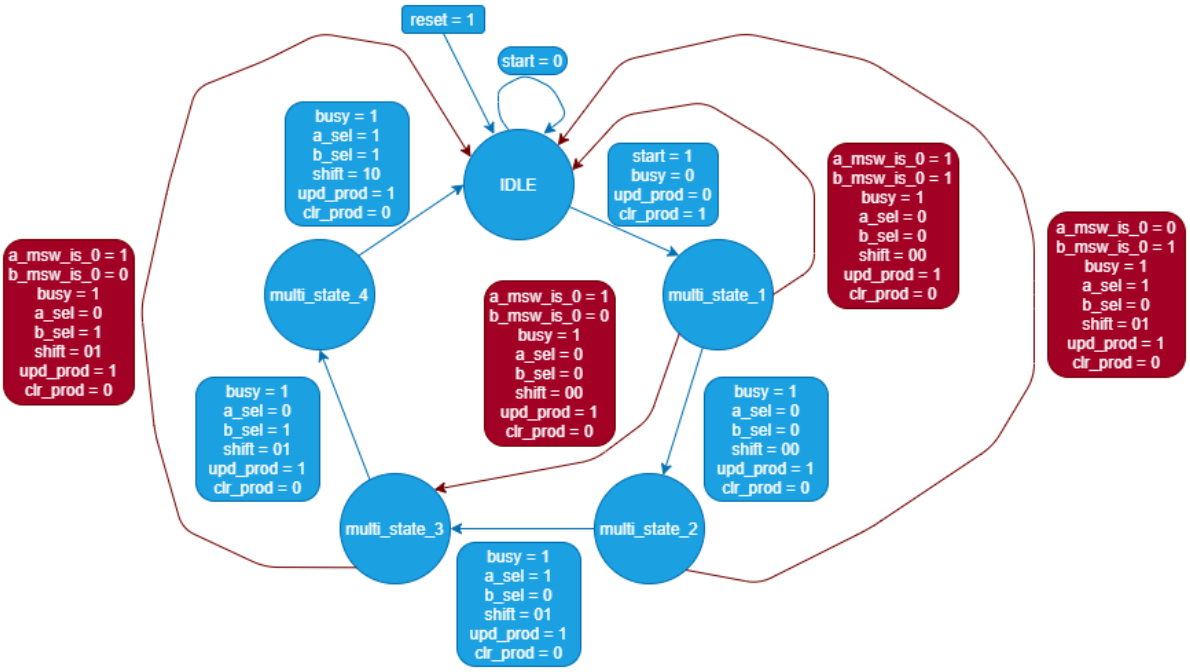
עופר ניסים	312367576
עידן גבאי	315073163

2.1 תיאור מכונת המצבים:



פעולת הכפל לוקחת 4 מחזורי שעון, בהתאם למספר המצבים שבהם מקבל המוצא busy את הערך 1.

2.2 תיאור מכונת המצבים עם זיהוי גורמי 0 במכפלה:



a_msw_is_0	b_msw_is_0	מספר מחזורי שעון
0	0	4
0	1	2
1	0	2
1	1	1

כעת פעולת הכפל לוקחת מחזור שעון אחד (אם a_msw_is_0 = 1 וגם b_msw_is_0 = 1) או 2 מחזורי שעון (אם a_msw_is_0 = 1 וגם b_msw_is_0 = 0 או להפך) או 4 מחזורי שעון (אם a_msw_is_0 = 0 וגם b_msw_is_0 = 0).

המכונה תעבוד הכי מהר במצב הראשון, בו יתקבלו הערכים a_msw_is_0 = 1 וגם b_msw_is_0 = 1.

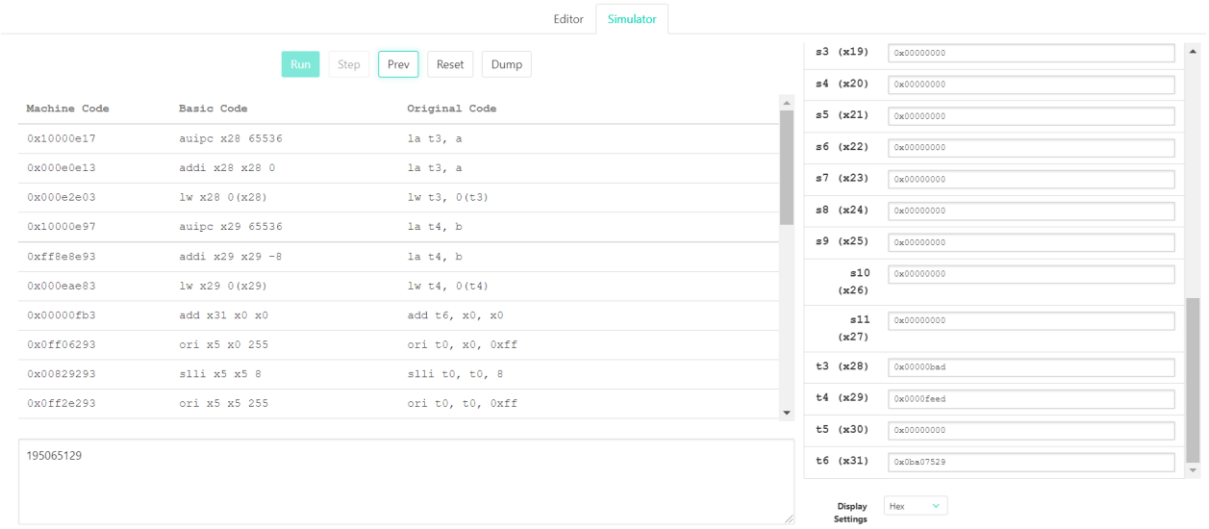
2.3 תיאור האלגוריתם:

ניעזר בפעולת הכפל של המעבד הנתון (נסמנה **(*)**), שמסוגל לכפול מספר בגודל 8 סיביות במספר בגודל 16 סיביות ולהוציא תוצאה בגודל 24 סיביות. נסמן ב-a וב-b את שני המספרים הנכפלים באורך 8N (כאשר N מספר טבעי זוגי). נחלק את המספר a לחלקים בני 8 סיביות ואת b לחלקים בני 16 סיביות. האלגוריתם שנציע יכלול שתי לולאות – חיצונית ופנימית. הלולאה החיצונית עוברת על כל חלק של b החל מה-LSB (ישנן N/2 איטרציות – לפי כמות החלקים של b) ומבצעת בלולאה הפנימית פעולת את פעולת הכפל **(*)** של חלק זה עם כל חלק של a החל מה-LSB (ישנן N איטרציות – לפי כמות החלקים של a). בנוסף נחזיק משתנה סכימה, כך לאחר כל פעולת כפל בלולאה הפנימית, נסכום אליו את תוצאת הפעולה. כדי לכפול בכל איטרציה של הלולאות את החלקים הבאים של המספרים – תתבצע בכל איטרציה של הלולאה הפנימית הזזה של 8 סיביות ובחיצונית הזזה של 16 סיביות. התהליך המתואר ימשיך עד החלקים האחרונים של a ו-b, כך שנקבל לבסוף את תוצאת המכפלה המבוקשת.

סיבוכיות האלגוריתם תלויה במספר האיטרציות בלולאה הפנימית (N) ובחיצונית (N/2), כך שהפעולות המתוארות דורשות בסך הכל:

$$.\mathrm{N} \cdot \frac{N}{2} = \frac{N^2}{2} = O(N^2)$$

2.4 הרצת קוד הכפל בסימולטור:



פעולת הכפל לוקחת 9 מחזורי שעון (בהנחה שכל פקודה אורכת מחזור שעון אחד), כמספר הפעולות בקטע קוד האסמבלי.

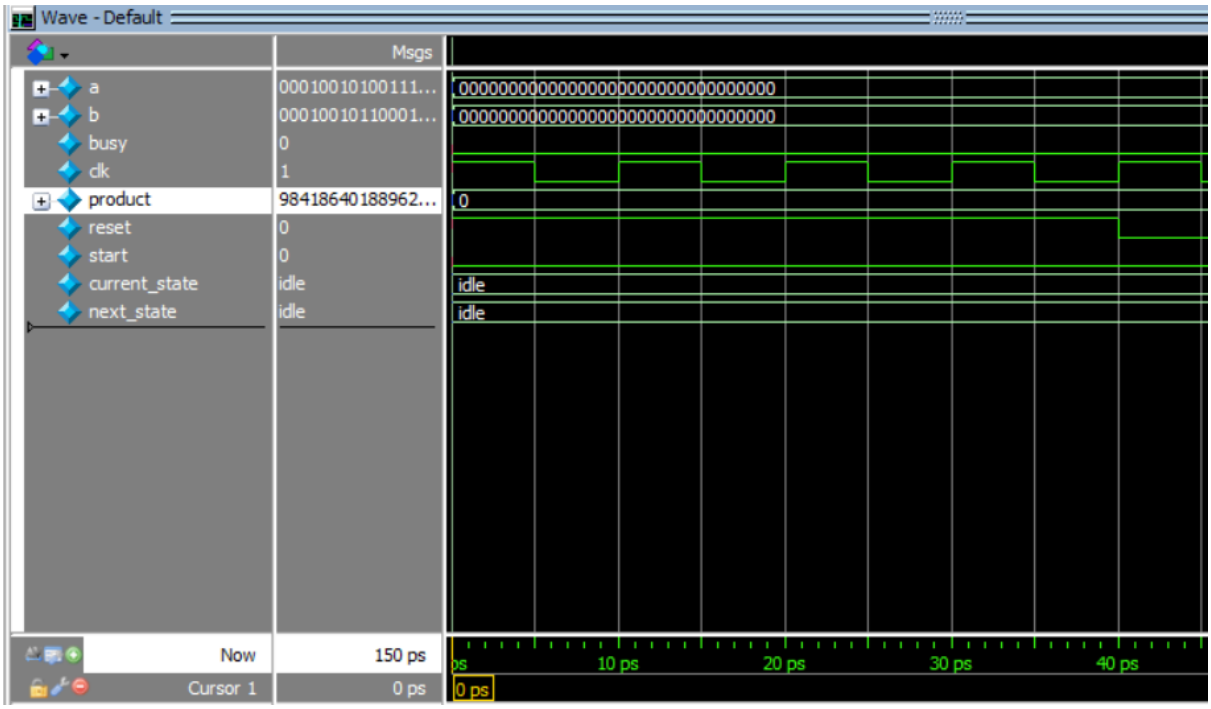
2.5 שינוי הקוד עבור דילוגים על אפסים בבתים העליונים:

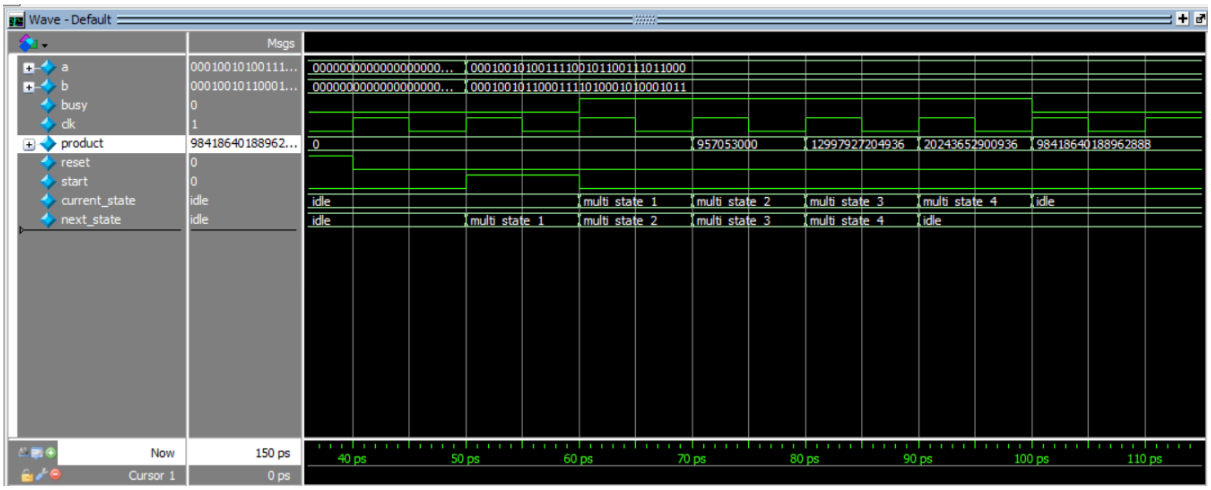
```
####################
# Start of your code

andi t2, t3, 0xff   # t2 = 8 lower bits of a
mul t6, t4, t2  # t6 = b * t2
and t6, t6, t0  # using the given mask
srli t2, t3, 8  # shift right by 8 all bits of a
andi t2, t2, 0xff   # t2 = 8 next bits of a
beq t2, x0, finish  # if t2 is 0 then finish
mul t2, t4, t2  # t2 = b * t2
and t2, t2, t0  # using the given mask
slli t2, t2, 8  # shift left by 8 all bits of a
add t6, t6, t2  # t6 = t6 + t2
```

השינוי הנדרש בקוד מסעיף 2.4 כדי לממש דילוגים על אפסים בבית העליון של a ו/או b הוא בדיקה של הבית העליון של המספרים באמצעות הוספת פקודת beq. אם הבית העליון אכן שווה ל-0, תתבצע קפיצה ל-label הנתון finish (היות שאין צורך בהוספת מכפלה שערכה 0). נשים לב שאין צורך לבדוק את הבית העליון של המספר השני במכפלה – b, שכן פעולת הכפל היא בגודל 16x8 ובכל מקרה יש לבצע כפל על כל 16 הספרות של מספר זה (אם פעולת הכפל הייתה בגודל 8x8 למשל, הייתה נדרשת התייחסות לבדיקה זו).

זמן הריצה החדש יהיה 10 מחזורי שעון אם הבית העליון של a שונה מ-0, ו-6 מחזורי שעון אם הבית העליון של a שווה ל-0. שינוי זה אכן משתלם, משום שהוא מוסיף מחזור שעון נוסף אחד לקוד במקרה בו הבית העליון של a שונה מ-0, אך חוסך 3 מחזורים אם הבית העליון של a שווה ל-0.

3.4





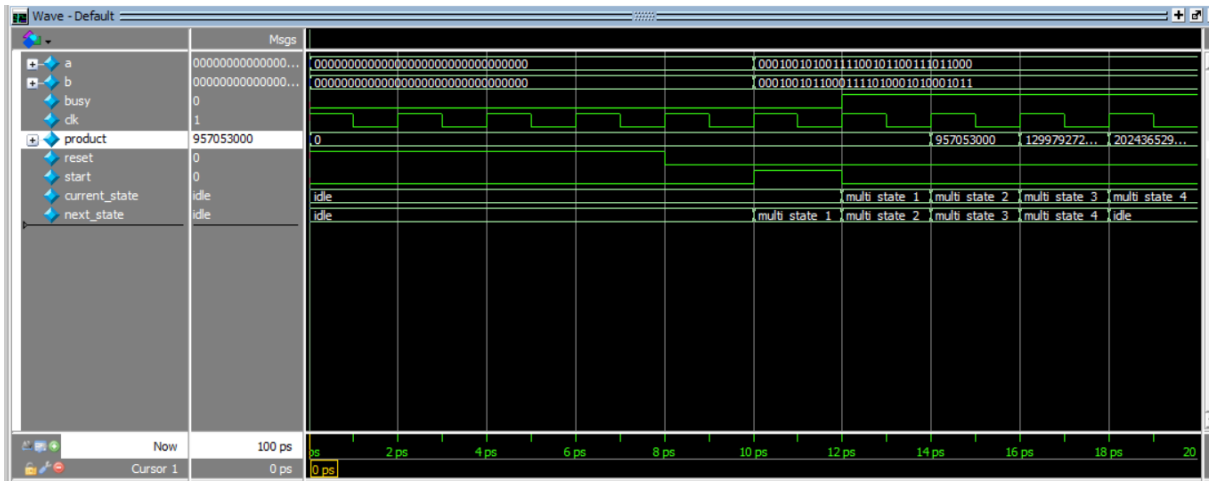
אנחנו בחרנו מחזור שעון כ ps10.

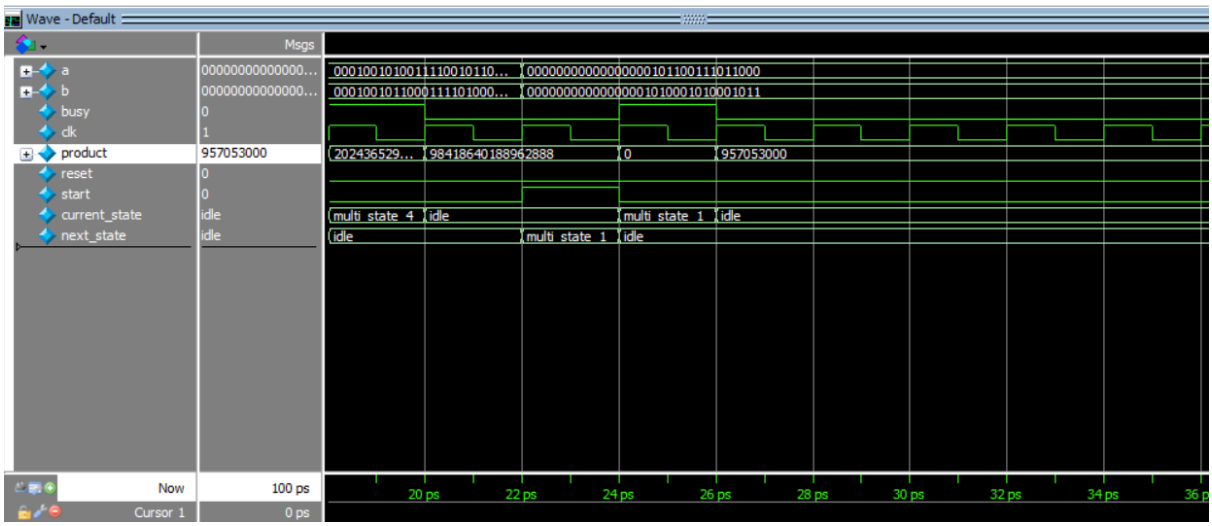
ניתן לראות שהסיגנל reset עבר מ 1 ל 0 לאחר ps40 – שזה 4 מחזורי שעון, כמבוקש.

כמו כן מחזור שעון אחד לאחר מכן ניתן לראות שגם עודכנו הערכים של a,b לערכים של תעודות הזהות וגם start עודכן לערך 1.

בארבעת מחזורי השעון הבאים ניתן לראות מצד אחד עדכון בערכים של product ובנוסף ניתן לראות את השינוי במצב current_state כאשר הוא למעשה מתקדם בדיוק כמו שהוא אמור, ובכך סוכם כל איבר בכפל הארוך אחד אחרי השני, ולבסוף מגיע למצב ההתחלתי ללא דגימות נוספות של הרגיסטר ובכך הערך נשמר ביציאה. כמו כן ניתן לראות שbusy עלה בדיוק ל 4 מחזורי שעון מיד לאחר שstart זוהה כ 1, שגם זה בדיוק מה שאמור לקרות.

3.7





(שימו לב שהתמונה השנייה מתחילה מ ps18 ולא מ ps20 (ps20 זה הסוף של התמונה הראשונה)).

בחרנו מחזור שעון שהוא 2.

ניתן לראות כי בהתחלה השעון מתחיל בערך -1 , a,b באפסים, reset בערך 1 וstart בערך 0.

לאחר ps8 – 4 מחזורי שעון, ניתן לראות שהורדנו את הערך של reset ל 0 ומחזור שעון אחד לאחר מכן הצבנו את הערכים הרלוונטים ב a,b ועדכנו את start ל -1 למשך מחזור שעון אחד כך שלאחר מחזור שעון אחד ערכו שוב עודכן ל – 0.

לאחר 4 מחזורי שעון ניתן לראות שהערך של busy חזר שוב ל 0 בדיוק כאשר הערכים ב,a,b התייצבו לערך של התוצאה הרצויה.

לאחר המתנה של מחזור שעון נוסף עדכנו את הערכים של a,b מחדש כך שיש להם אפסים בהתחלה ועדכנו את start ל 1 למשך מחזור שעון אחד.

כעת ניתן לראות שbusy עלה רק למחזור שעון אחד, וזה תקין כי החלקים העליונים של a ו b היו אפסים מה שגרם למכונת המצבים לדלג על כל החיבורים חוץ מהראשון שעדיין חיוני לטובת החישוב.