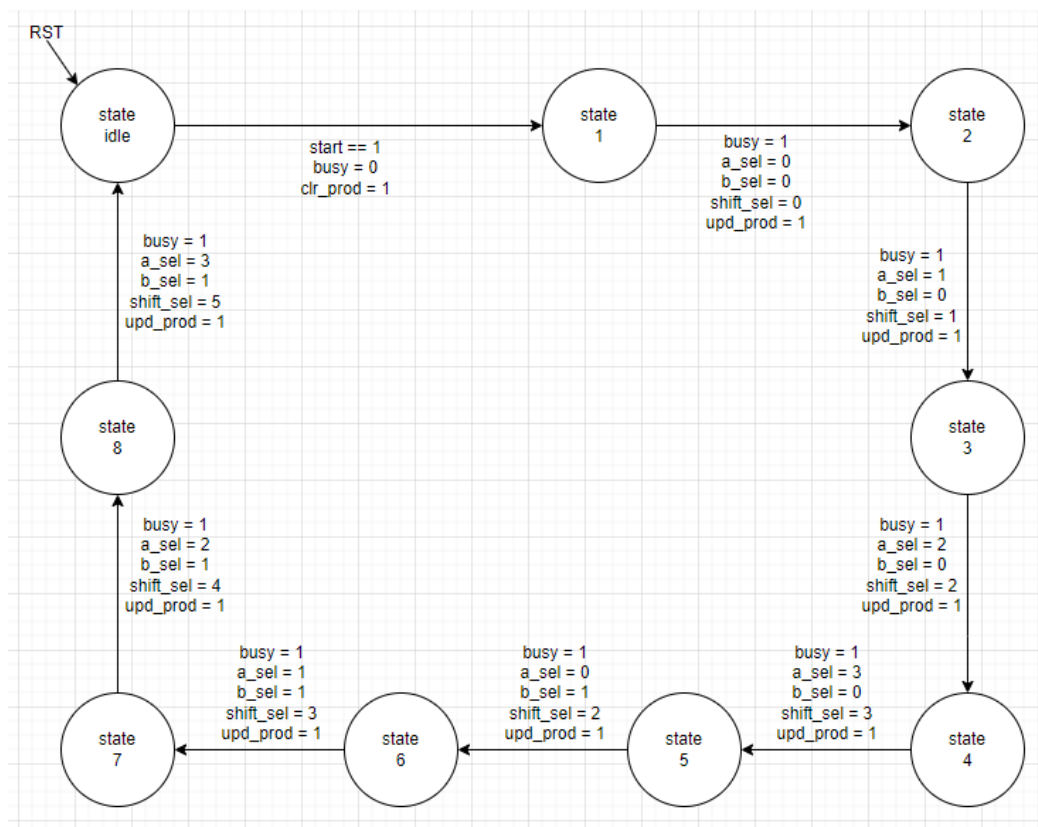


תרגיל בית 2

315157909	עדי סויסה
208459867	דר ניסן

2.1 מכונת מצבים



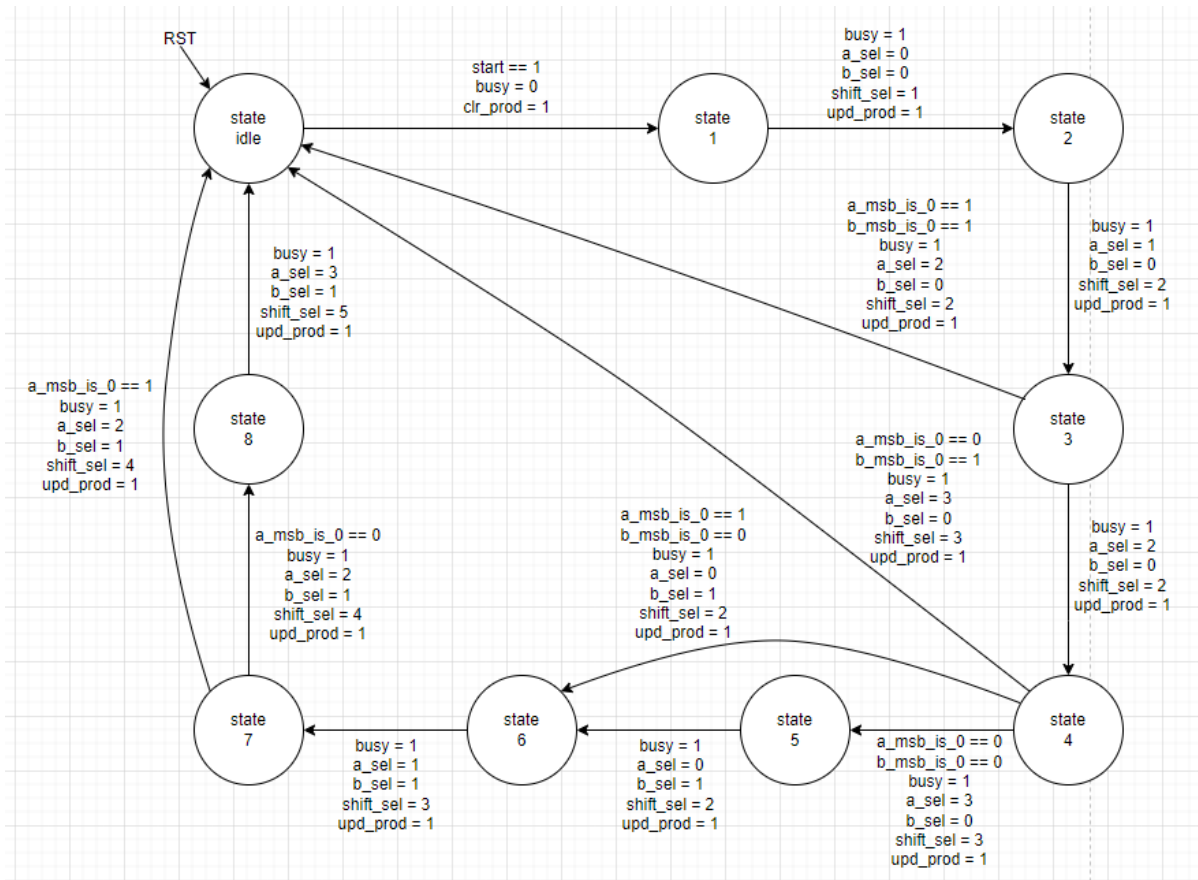
פעולת הכפל עצמה לוקחת 8 מחזורי שעון, מעליית הסיגנל busy ועד הירידה שלו בסוף החישוב.

2.2

נשפר את היעילות של המכונה בכך שנסתכל על הסיביות העליונות שבכניסות A ו-B, אם ה- most significant byte בכניסה A או אם ה- most significant word בכניסה B הן אפס, אז נסיים את החישוב ונדלג על ההכפלה המיותרת באפס מכיוון שהיא לא תשנה את תוצאת החישוב.

עם השינויים שתיארנו המכונה תעבוד הכי מהר כאשר גם ה-MSB של A וגם ה-MSW של B שווים אפס, תוך 3 מחזורי שעון, ובאופן כללי פעולת הכפל לוקחת לכל היותר 8 מחזורי שעון, מעליית הסיגנל busy ועד הירידה שלו בסוף החישוב.

להלן דיאגרמת המצבים החדשה בהתאים לשינויים:



Result = 0

For (i = 0; i < 8N; i += 16)

{

For (j = 0; j < 8N; j += 8)

{

Temp = Multiply8x16(A[i],B[j])

shift (temp, 2i + j)

result += temp

}

}

נשים לב שהלולאה הראשונה רצה $5N \cdot 0$ איטרציות, והלולאה הפנימית רצה N איטרציות ולכן סיבוכיות הזמן הינה $O(N^2)$.

The screenshot displays a MIPS simulator interface with two main panels: 'Editor' and 'Simulator'. The 'Editor' panel shows a table of assembly code with columns for Machine Code, Basic Code, and Original Code. The 'Simulator' panel shows the state of registers and memory.

Machine Code	Basic Code	Original Code
0x10000e17	auipc x28 65536	la t3, a
0x000e0e13	addi x28 x28 0	la t3, a
0x000e2e03	lw x28 0(x28)	lw t3, 0(t3)
0x10000e97	auipc x29 65536	la t4, b
0xff0e0e93	addi x29 x29 -8	la t4, b
0x000eae83	lw x29 0(x29)	lw t4, 0(t4)
0x00000fb3	add x31 x0 x0	add t6, x0, x0
0x0ff06293	ori x5 x0 255	ori t0, x0, 0xff
0x00829293	slli x5 x5 8	slli t0, t0, 8
0x0ff2e293	ori x5 x5 255	ori t0, t0, 0xff
0x00829293	slli x5 x5 8	slli t0, t0, 8
0x0ff2e293	ori x5 x5 255	ori t0, t0, 0xff
0x018e9393	slli x7 x29 24	slli t2 t4 24

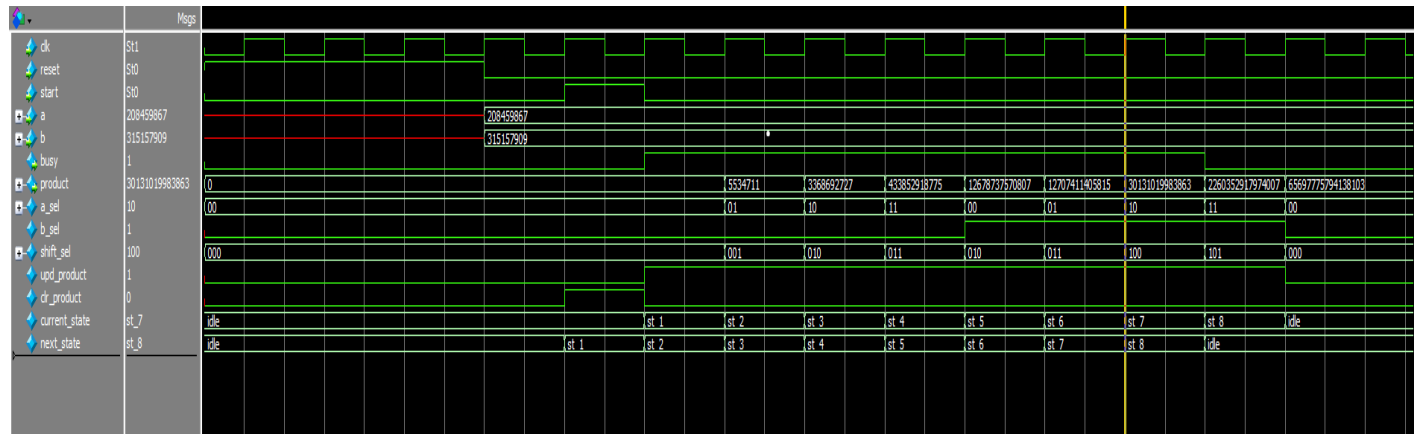
The 'Simulator' panel shows the state of registers and memory. A blue arrow points to the 't6 (x31)' register, which contains the value 0x18e07529.

במימוש שלנו ישנו 11 פקודות בחלק הדרוש ולכן לפי הנתון כי כל פקודה תרוץ במהלך מחזור שעון אחד, ניתן לומר כי עלות הפעולה כולה היא 11 מחזורי שעון.

2.5 כדי ליצור קוד שממש דילוגים על אפסים בבית העליון של a ו/או b נצטרך לבצע בדיקה של הבית העליונים של a ושל b. השינוי שנכניס בקוד הוא בדיקה של הבית העליונים של b ושל a באמצעות השוואה שלהם לרגיסטר אפסים x0. נשים לב שבשביל לבצע השוואה רק של הבית העליון נצטרך לבצע השמה רק של הבית העליון לרגיסטר עזר ובעצם להשוות את רגיסטר העזר הזה לרגיסטר אפסים. ועל פי הנתון כי כל פקודה מתבצעת

במהלך מחזור שעון אחד נוכל להגיד כי לא מתבצע חסכון לעומת הקוד השמה של . שכן גם אם מתבצע דילוג מירבי – קרי הבתים העליונים גם של a וגם של b הם אפסים עדיין נרצה לכפול את הבתים התחתונים במכפלת 8x8 ביטים. לכן רק עלות ההשמה וההשוואה לרגיסטר אפסים עולה לנוה כפי שהיינו מבצעים כפל רגיל. מכאן שהשינוי המוצע , לא משתלם לפי ההנחות הנתונות.

3.4

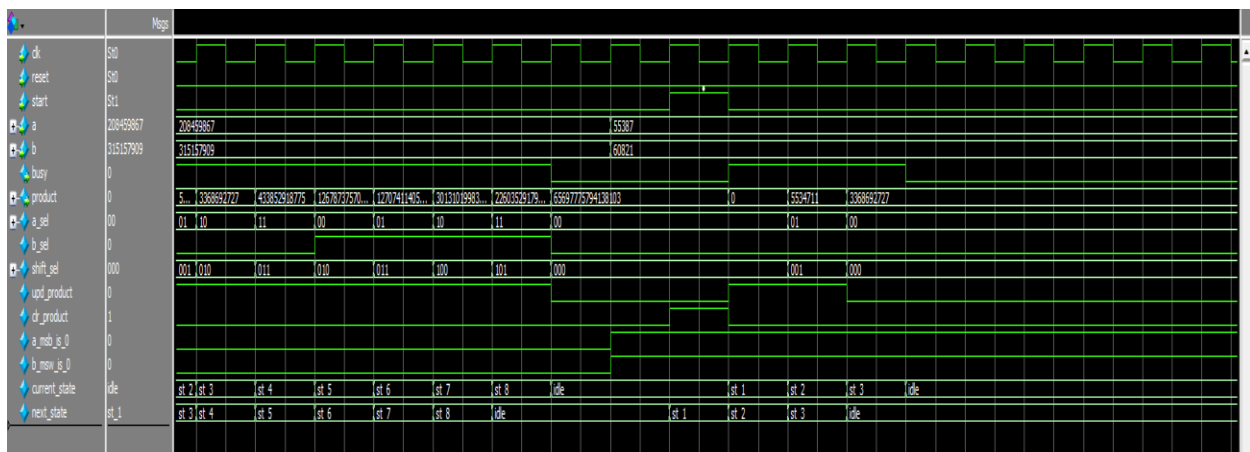
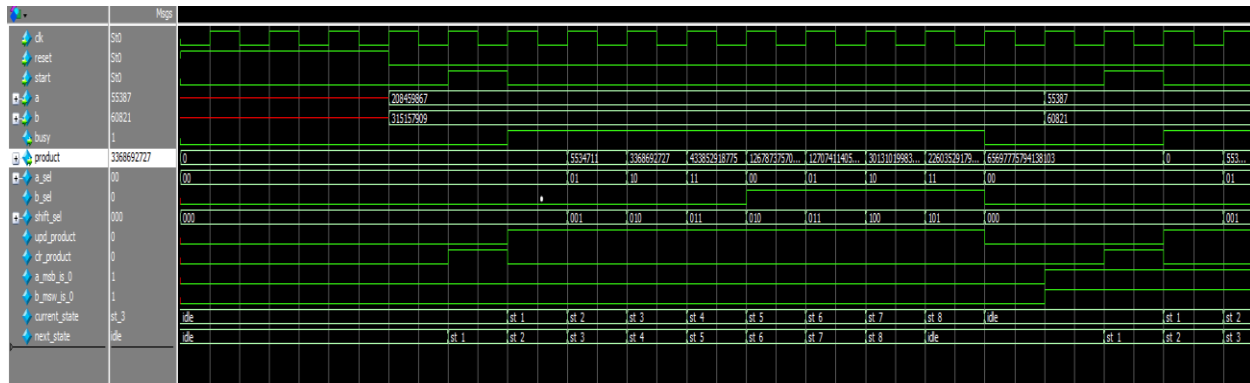


נשים לב שלפי דיאגרמת הגלים שמצורפת כאן הטסט מפעיל את המכונה שמבצעת את הפעולות הבאות לפי ההנחיות שניתנו:

- השמה של ערך 1 באות reset למשך 4 מחזורי שעון
- אתול start בערך 0
- יציאה מאות reset והשמת ערכי תעודות הזהות ברגיסטרים a ו b
- המתנה של מחזור שעון אחד
- הצבת 1 ב start למשך מחזור שעון אחד – עם עליית start המכונה כבר מעדכנת את next state ל st1
- עם ירידת start עולה גם אות busy
- ובעליית busy בעצם המכונה מתחילה לעבור בין המצבים st
- נשים לב שלאחר כל סיום של st_i כלשהו מתבצע חישוב מתאים ומושם בתוך רגיסטר product
- לאחר סיום של החישוב האחרון st8 נוכל לראות כי המכונה חוזרת למצב idle , אות busy יורד, וכן התוצאה נשמרת ברגיסטר product כנדרש.

כמו כן נשים לב כי תוצאת החישוב נכונה

$$208459867 \cdot 315157909 = 65,697,775,794,138,103$$



נסביר את שלבי הטסט הנ"ל

- אנו מאתחיל אות *reset* שפעיל בתחילת הסימולציה ל4 מחזורי שעון ולאחר מכן יורד עד סוף הסימולציה
- עם ירידת *reset* מושמים תעודות הזהות לרגיסטרים *a, b*
- ניתן לראות אתחול של *start* לערך 0
- מתרחשת המתנה של מחזור שעון אחד
- הצבה של 1 ב *start* למשך שעון אחד
- עם ירידת *start* עולה *busy* ולאחר מכן מתבצע חישוב של תעודות הזהות. החישוב הראשון זהה לחלוטין לחישוב שהתבצע בסעיף הקודם.
- עם ירידת *busy* אנו ממתינים מחזור שעון אחד ולאחר מכן משימים את ערכי תעודות הזהות עם שני הבתים העליונים מאופסים
- לאחר מכן ממתינים מחזור שעון אחד
- מציבים 1 ב *start* למשך מחזור שעון אחד
- וכעת מתבצע החישוב

- נשים לב ש $st3$ מתבצעת הבדיקה לגבי ערך הבתים העליונים. לאחר הבדיקה המכונה לא ממשיכה לשאר המצבים שכן מדובר בהכפלות בביתים בהם ישנם רק אפסים
 - לכן המכונה עוברת למצב $idle$ ומעדכנת את האות $busy$ לרדת
 - נשים לב גם כאן לנכונות התוצאה
- $$55387 \cdot 60821 = 3,368,692,727$$
- נשים לב שהריצה במכונת ה $fast$ עבור קלטים שבהם הבתים העליונים מאופסים אכן לוקחת פחות זמן שכן לאחר החישוב המתאים חוזרים למצב ה $idle$ ולא מבצעים חישוב מיותר.