מטלה מסכמת - קורס VHDL: תכנון מכפל סידרתי סנכרוני

מבוא:

מטלה זו נועדה לבחון את הבנתכם בתכנון מערכות ספרתיות באמצעות VHDL, תוך התמקדות בתכנון מכפל סידרתי סנכרוני. המכפל יבצע כפל של שני ערכים בני 4 ביט כל אחד (נכפל A וכופל B).
המטלה כוללת תכנון מודול הבקרה (CU) ומודול הביצוע (OU) של המערכת, והטמעתם ב-VHDL.

תיאור המערכת:

המכפל הסידרתי הסנכרוני מבצע את הפעולות הבאות:

1. שמירת הנכפל והכופל באוגרים.
2. חישוב תוצאת ביניים (PP - partial products) על ידי הכפלת הנכפל בביט התורני (משמאל לימין) של הכופל.
3. צבירת תוצאת הביניים בצובר (accumulator). ערך הצובר בסיום התהליך הוא התוצאה הסופית P.
4. הזזה של הנכפל ביט אחד שמאלה, והזזה של הכופל ביט אחד ימינה.
5. בדיקת תנאי סיום: אם הכופל שווה ל-0, התהליך מסתיים; אחרת, חוזרים לשלב 2.

שיטה זו דורשת ביצוע N מחזורי חישוב/צבירה/הזזה עבור אופרנדים בני N ביט, או לחלופין, בדיקה מתמדת של הכופל, וכאשר הכופל שווה ל-0 לאחר הזזה, זהו הסיום.

דוגמת ביצוע:

נדגים את פעולת המכפל עבור נכפל $A = 11_d = 1011_b$ והכופל $B = 11_d = 1011_b$

1. **אתחול:**
   - טעינת הנכפל $A = 1011_b$ לאוגר RA בגודל 8 ביט (גודלה של התוצאה הסופית הצפויה).
   - טעינת הכופל $B = 1011_b$ לאוגר RB בגודל 4 ביט.
   - מצב האוגרים: RA = 00001011, RB = 1011.
2. **איטרציה 1:**
   - הביט הימני ביותר של RB הוא '1'.
   - חישוב PP: PP = 00001011
   - צבירת PP לצובר P: P = 0001011
   - הזזה שמאלה של RA: RA = 00010110
   - הזזה ימינה של RB: RB = 0101

- תנאי סיום: RB אינו 0.

3. **איטרציה 2:**
   - הביט הימני ביותר של RB הוא '1'.
   - חישוב PP: PP = 00010110
   - צבירת PP לצובר P: P = .00100001
   - הזזה שמאלה של RA: RA = 00101100
   - הזזה ימינה של RB: RB = 0010
   - תנאי סיום: RB אינו 0.
4. **איטרציה 3:**
   - הביט הימני ביותר של RB הוא '0'.
   - חישוב PP: PP = 00000000
   - צבירת PP לצובר P: P = 00100001P
   - הזזה שמאלה של RA: RA = 01011000
   - הזזה ימינה של RB: RB = 0001
   - תנאי סיום: RB אינו 0.
5. **איטרציה 4:**
   - הביט הימני ביותר של RB הוא '1'.
   - חישוב PP: PP=$.01011000^{31}$ = PP
   - צבירת PP לצובר P: P = 01111001
   - הזזה שמאלה של RA: RA = 10110000
   - הזזה ימינה של RB: RB = 0000
   - תנאי סיום: RB שווה ל-0. הפעולה מסתיימת.

התוצאה הסופית: $P=01111001_b$ (שהוא 121_d, ו-11×11=121).

לשם המחשה, ניתן לראות את סיכום תוצאות הביניים והסכום הסופי:

00001011_b $(A=11_d)$
*
1011_b $(B=11_d)$
−−−−−−−−−−−−−−−−−−
00001011_b $(PP=RA=A)$ 11_d
+
00010110_b $(PP=RA=A*2)$ 22_d
+
00000000_b $(PP=0)$
+
01011000_b $(PP=RA=A*8)$ 88_d
−−−−−−−−−−−−−−−−−−
01111001_b $=121_d$

מבנה המערכת:

מהתיאור המילולי, המשמש אותנו כדרישת המערכת, נעבור לבניית מודל המערכת ברמת הפשטה RTL.
נעשה זאת על פי הגישה שכבר הוצגה, חלוקת המערכת לשתי יחידות - בקרה והביצוע.

המערכת מחולקת לשתי יחידות עיקריות:

- **יחידת בקרה (CU - Control Unit):** אחראית על ניהול רצף הפעולות.
- **יחידת ביצוע (OU - Operational Unit):** מבצעת את הפעולות האריתמטיות והלוגיות.

הנתונים (A ו-B) נטענים ליחידת הביצוע, והתוצאה (P) מתקבלת ממנה.
הבקשה לתחילת עבודה (start) והודעת סיום (done) קשורות ליחידת הבקרה.

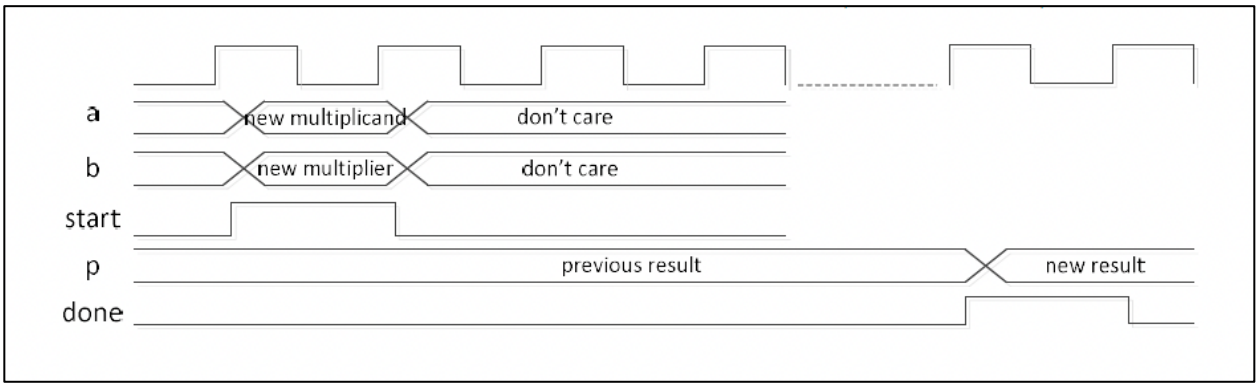
תזמון אותות חיצוניים:

נתחיל מהגדרת מילות הבקרה והסטטוס.
מילת הבקרה של המערכת כוללת 4 ביט ומורכבת מהאותות:

- load :המשמש לטעינת האוגרים בערכי הנכפל והכופל.
- shift_enable המשמש להזזת הכופל והנכפל
- clear המשמש לאיפוס הצובר בתחילת העבודה
- done המודיע על סיום

מילת הסטטוס תכלול שני ביט: start- המתקבל מהסביבה החיצונית, ו- end המודיע ליחידת הבקרה על סיום.

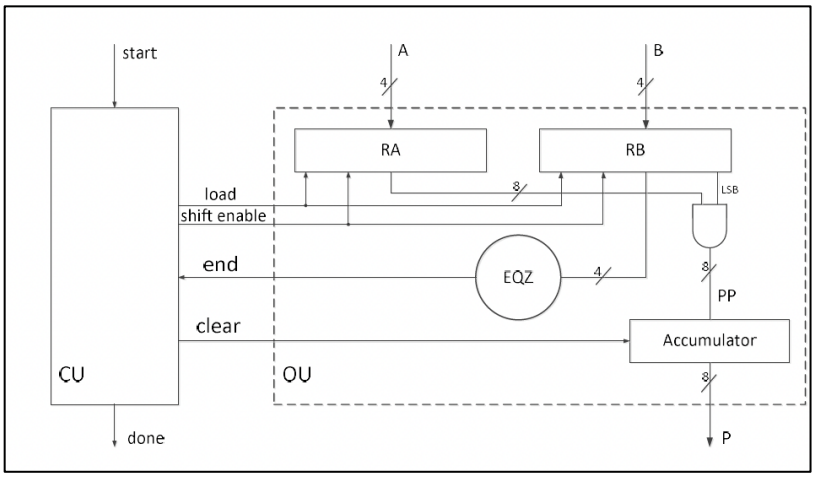
תזמון האותות החיצוניים מוצג באיור 5:



איור 5: תזמון אותות חיצוניים

מבנה המערכת הכולל:

מודל המערכת לפי גישה זו מוצג באיור 6.
במסגרת המקווקוות נמצאת יחידת הביצוע ובצד שמאל יחידת הבקרה.



איור 6: מודל המערכת הכולל (OU ו-CU)

רכיבי יחידת הביצוע (OU):

ביחידת ביצוע מתבצעות הפעולות הבסיסיות: חישוב תוצאת ביניים, צבירה של תוצאת ביניים, הזזה של הנכפל שמאלה ושל הכופל ימינה, בדיקת סיום.
מפעולות אלו נגזרים מבני החומרה הבסיסיים הנדרשים.

מבנים אלה הם:

- **אוגר RA:** 8 ביט, אוגר הזזה עם טעינה מקבילית, להזזה שמאלה (לנכפל).
- **אוגר RB:** 4 ביט, אוגר הזזה עם טעינה מקבילית, להזזה ימינה (לכופל).
- **צובר (Accumulator):** 8 ביט, עם יכולת אתחול.
- **משווה (EQZ):** 4 ביט, בודק שוויון ל-0 (עבור RB).

אותות בקרה וסטטוס:

נתחיל מהגדרת מילות הבקרה והסטטוס.

- מילות בקרה (מיקרו-פקודות - Y_i):

מילת הבקרה של מערכת זו כוללת 4 ביט. המיקרו-פעולות לפי שמם הפורמלי, השמות הנמצאים בשימוש במערכת ומשמעותם המערכתית מרוכזים בטבלה.

מיקרו-פקודה	שם במעגל	תכלית
y_1	Load	טעינת אוגרי הזהה RA ו-RB
y_2	shift_enable	אפשור הזזה של אוגרים RA ו-RB
y_3	Clear	איפוס הצובר P
y_4	Done	הודעת סיום הפעולה

טבלה 1: מילות בקרה (מיקרו-פקודות)

רשימת כל המיקרו-פקודות והרכבן ממיקרו-פעולות מופיעה בטבלה 2.
טבלה זו כולל גם את המיקרו-פקודה Y_0 המזוהה עם אי עשייה.

מיקרו-פקודה	מיקרו-פעולות			
	y_1	y_2	y_3	y_4
Y_0	0	0	0	0
Y_1	1	0	1	0
Y_2	0	1	0	0
Y_3	0	-	0	1

טבלה 2: הרכב המיקרו-פקודות

ניתן לראות חפיפה מלאה בין שתי מיקרו-פעולות Y_1 ו-Y_2 . על כן, הוחלט להחליפן במיקרו-פעולה אחת, Y_1, המקבלת את השם load_clear ומשמשת לטעינת האוגרים ואיפוס הצובר בו זמנית. כלומר, במקום Y_1 (load) ו-Y_2 (Clear) בטבלה 2, הפונקציונליות שלהם תאוחד למיקרו-פקודה אחת. (שימו לב: בטבלה 2, "0 1 0 1" עבור Y_1 מצביע על Y_1=1 ו-Y_2 =1, וזה מייצג את הפעולה המאוחדת).

- מילות סטטוס (X_i):

מילת הסטטוס תכלול שני ביט. מילת סטטוס ומרכביה מופיעים בטבלה 3.

ביט הסטטוס	שם במעגל	תכלית
x_1	start	תחילת הפעולה
x_2	end	זיהוי הסיום

טבלה 3: מילות סטטוס

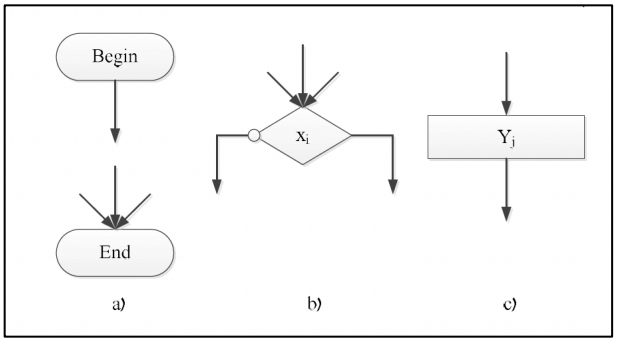
יחידת הבקרה היא זו שמתקשרת עם הסביבה החיצונית. היא אחראית על ביצוע הפעולה עם קבלת אות start, טעינת האוגרים על ידי האות load (שילוב של load_clear), שליטה על הזזה על ידי האות shift_enable, אתחול הצובר על ידי האות clear (שילוב של load_clear), והודעה כלפי חוץ על סיום הפעולה באמצעות האות done העולה ל-'1' למחזור שעון אחד. אות end (מילת סטטוס 2x) מתקבל מיחידת הביצוע ליחידת הבקרה.

תרשים ASM של יחידת הבקרה (CU):

רצף המיקרו-פקודות ומעבר ביניהן כתלות בביטי מילת הסטטוס מוצג בתרשים זרימה פורמלי המכונה ASM (Algorithmic State Machine).

בבניית תרשים זרימה כ-ASM יש שימוש ב-3 צורות:

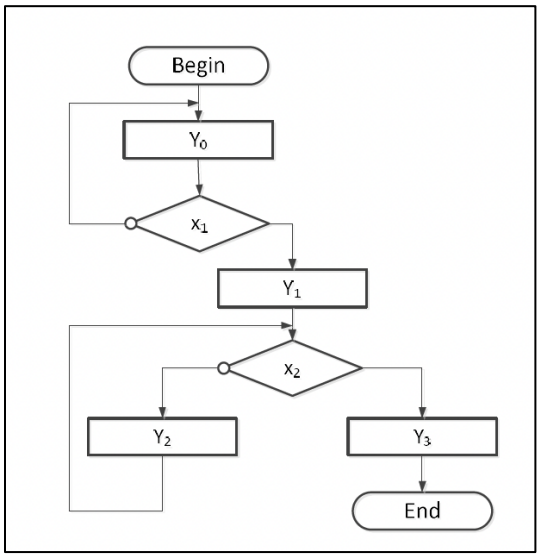
- **אליפסה (Begin/End):** מייצג קודקודי התחלה והסוף (איור a7).
- **מעוין (X_i):** מייצג קודקוד בדיקה (איור b7). בקודקוד זה נבדק משתנה אחד ממילת הסטטוס וממנו שתי יציאות: אחת כאשר המשתנה הנבדק שווה ל-1 (אמת), השנייה (עם סימן שלילה) מזוהה עם משתנה הנבדק השווה ל-0 (שקר).
- **מלבן (Y_j):** מייצג קודקוד ביצוע (איור c7). קודקוד הביצוע מייצג מיקרו-פקודה אותה יש לבצע כאשר מתקיימים התנאים המובילים אליו.



איור 7: צורות בסיסיות של תרשים ASM (a קודקודי התחלה וסיום, (b קודקוד בדיקה, (c קודקוד ביצוע

תרשים זרימה חייב לכלול לפחות מסלול אחד המוביל מקדקוד Begin לקדקוד End. מוסכם, שבהגעה לקודקוד End מתבצע מעבר אוטומטי לקודקוד Begin.

תרשים הזרימה של יחידת הבקרה של המכפל מוצג באיור 8.



איור 8: תרשים זרימה (ASM) של יחידת הבקרה של המכפל

המטלה:

1. **מודול יחידת ביצוע (OU) ב-VHDL:**
   - תכננו את מודול יחידת הביצוע (OU) ב-VHDL.
   - המודול יכלול:
     - אוגר RA בגודל 8 ביט.
     - אוגר RB בגודל 4 ביט.
     - צובר (Accumulator) בגודל 8 ביט.
     - משווה לבדיקת EQZ (RB = 0).
     - לוגיקה לחישוב (PP (partial product בהתבסס על הביט הימני ביותר של RB.
   - וודאו כי המודול מגיב כראוי לאותות הבקרה (אותות load_clear ו-shift_enable המגיעים מה-CU) ומפיק את אות הסטטוס (end) ואת התוצאה הסופית (P).
   - המודול צריך לקבל כקלט את A (נכפל, 4 ביט) ואת B (כופל, 4 ביט).
2. **מודול יחידת בקרה (CU) ב-VHDL:**

- תכננו את מודול יחידת הבקרה (CU) ב-VHDL, תוך שימוש בתרשים ה-ASM (איור 8) כמכונת מצבים (FSM).
- המודול יכלול את המצבים המתוארים בתרשים ה-ASM (המיוצגים על ידי המיקרו-פקודות Y_0, Y_1, Y_2, Y_3).
- ודאו כי המודול מגיב כראוי לאותות הסטטוס החיצוניים (start) והפנימיים (end) ומפיק את אותות הבקרה (load_clear, shift_enable, done) בהתאם לרצף הפעולות.
- האות done צריך לעלות ל-'1' למחזור שעון אחד בלבד בסיום הפעולה.

3. **אינטגרציה ובדיקה:**
   - שלבו את שני המודולים (CU ו-OU) למערכת אחת שלמה.
   - כתבו Test Bench מתאים לבדיקת המערכת.
   - הריצו סימולציה עבור 5 זוגות קלטים, לדוגמה $A = 1011_b$ ו- $B = 1011_b$ (כפי שהוצג בדוגמת הביצוע), והראו את התוצאה הסופית P ואת רצף אותות הבקרה והסטטוס לאורך הסימולציה.
   - הסבירו בקצרה את פעולת המערכת והציגו את התוצאות של הסימולציה.

דרישות הגשה:

- קובץ VHDL עבור מודול יחידת הביצוע (OU).
- קובץ VHDL עבור מודול יחידת הבקרה (CU).
- קובץ VHDL עבור המערכת המשולבת (Top Level).
- קובץ VHDL עבור ה-Test Bench.
- מסמך קצר (עד 2 עמודים) שיכלול:
  - הסבר על התכנון של כל אחד מהמודולים.
  - צילומי מסך מתוצאות הסימולציה המציגים את פעולת המערכת ואת התוצאה הסופית.
  - ניתוח קצר של התוצאות והסבר על תקינות הפעולה.
- סרטון של עד 15 דקות המסביר בפירוט את הקוד שכתבתם ואת תוצאות הסימולציה.

מבנה הציון:

המטלה תדורג במספר רמות, המאפשרות לכם להרחיב את הידע והמימוש מעבר לדרישות הבסיסיות:

- **ציון בסיס (עד 80 נקודות)**

  עבור מימוש מלא ותקין של כל הדרישות שהוגדרו בסעיפים 1-3 במטלה (מודול יחידת ביצוע (OU) , מודול יחידת בקרה (CU) , אינטגרציה ובדיקה), כפי שתוארו עד כה, תהיה זכאות לציון מקסימלי של 80 נקודות.

- **הרחבה אופציונלית א' (עד 90 נקודות)**
  - **למידה עצמית ומימוש של מכפל:Radix-4 (Booth's Algorithm)**
    - סטודנטים שיבחרו להרחיב את המטלה ולממש מכפל בשיטת Radix-4 (לדוגמה, אלגוריתם בות') יוכלו לקבל ציון מקסימלי של 90 נקודות.
    - ההרחבה דורשת למידה עצמית של עקרונות Radix-4 וכיצד הוא שונה מהגישה הסידרתית הפשוטה.
    - יש להציג את התכנון (מודול CU ו OU-מעודכנים או חדשים, במידת הצורך) ולהדגים ב Test Bench-את תקינות הפעולה עבור מספר דוגמאות.

- **הרחבה אופציונלית ב' (עד 100 נקודות)**
  - **למידה עצמית ומימוש של מכפל:Radix-16**
    - סטודנטים שיבחרו לאתגר את עצמם ולממש מכפל בשיטת Radix-16 יוכלו לקבל ציון מקסימלי של 100 נקודות.
    - הדבר מצריך הבנה מעמיקה עוד יותר של עקרונות Radix-16 ויישומם במעגל.
    - יש להציג את התכנון ולהדגים ב Test Bench את תקינות הפעולה עבור מספר דוגמאות.

הערה חשובה :בחירה בהרחבות האופציונליות אינה מבטלת את דרישות הבסיס, אלא מוסיפה עליהן. תכנון בסיסי חלקי/לא תקין, גם אם מלווה בניסיון למימוש הרחבות, לא יקנה ציון גבוה. ההרחבות מיועדות לסטודנטים המעוניינים להעמיק ולהצטיין מעבר לדרישות המינימום.

עבודה נעימה ובהצלחה!