אפקה המכללה האקדמית להנדסה בתל-אביב
המחלקה להנדסת חשמל ואלקטרוניקה

פרויקט מסכם קורס - VHDL

מגישים:

איתי גורביץ'

עידן נודל

שם המרצה: ד"ר דורית מדינה

פרויקט מסכם VHDL - מכפל Radix 16

הקדמה

הפרויקט עוסק בתכנון ומימוש של מעגל חומרה ב-VHDL לביצוע פעולת כפל של שני מספרים שלמים עם סימן (signed integers) באמצעות אלגוריתם Booth. המערכת מורכבת משתי יחידות עיקריות: יחידת הבקרה (Control Unit - CU) ויחידת האופרציה (Operational Unit - OU), המאוגדות יחד ביחידת Top. המטרה היא לבצע כפל יעיל של שני מספרים באורך N סיביות, כאשר התוצאה מוחזרת כמספר באורך N2 סיביות. הפרויקט כולל גם Testbench לבדיקת תקינות המערכת עם מגוון ערכי קלט.

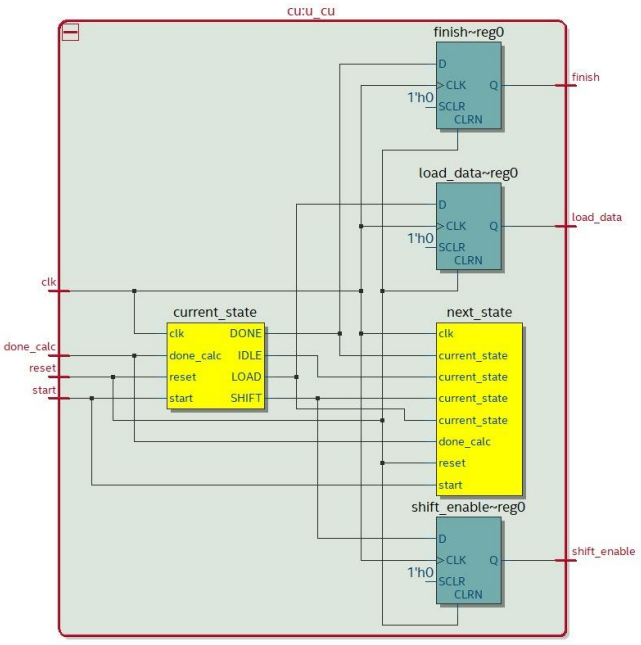
מבנה היחידה:

1. יחידת הבקרה:

יחידת הבקרה (cu) היא מכונת מצבים סופית (FSM) עם ארבעה מצבים: IDLE, LOAD, SHIFT ו-DONE. היא מקבלת קלטים: clk, reset, start ו-done_calc (מה-ou), ומפיקה שלושה אותות: load_data, shift_enable ו-finish. התהליך המרכזי פועל בעליית השעון (rising_edge(clk)) ומנהל את המצבים:

- **IDLE**: ממתין ל-start='1' כדי לעבור ל-LOAD.
- **LOAD**: מפעיל load_data='1' להעמסת קלטים ומעבירה למצב SHIFT.
- **SHIFT**: מפעיל shift_enable='1' לביצוע חישובים עד ש done_calc='1'- ואז עובר למצב DONE.
- **DONE:** מפעיל finish='1' וחוזר למצב IDLE.

סינתזה של יחידת הבקרה:



2. יחידת האופרציה:

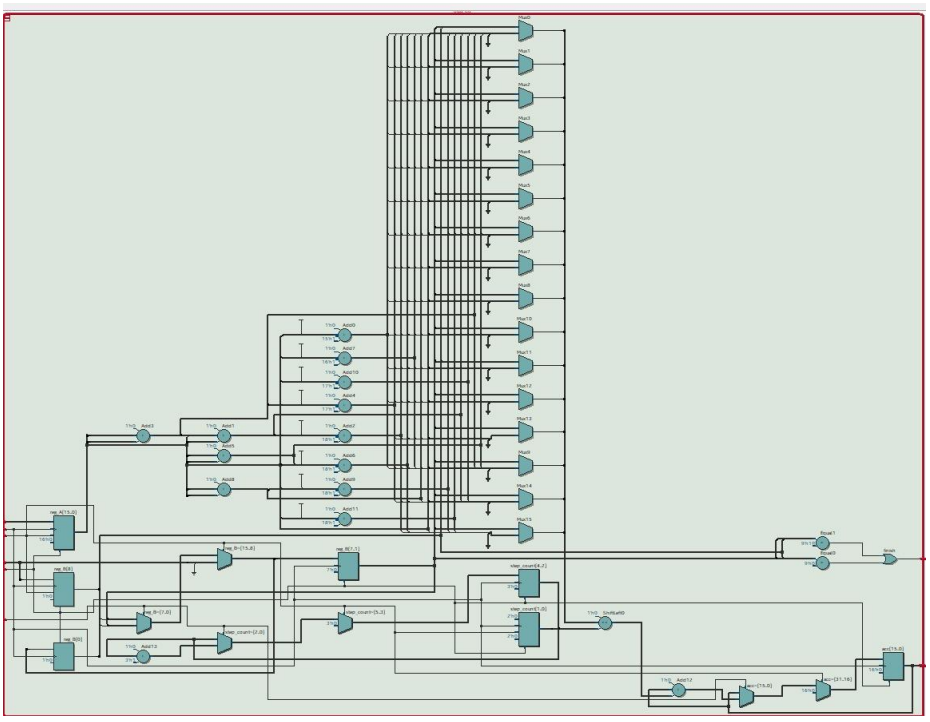
יחידת האופרציה (ou) מממשת את אלגוריתם Booth ברדיקס 16 לביצוע הכפל. היא מקבלת קלטים A ו-B (N סיביות), אותות load ו-shift_en, ומפיקה תוצאה (product) באורך 2N סיביות ואות finish. היחידה משתמשת ברגיסטרים:

- **reg_A** : באורך 2N סיביות, מכיל את A יחד עם הקצאת סימן (sign extension).
- **reg_B**: באורך N+1 סיביות, מחזיק את B עם '0' ב-LSB.
- **acc:** צובר התוצאה, באורך 2N סיביות.
- **step_count:** מונה הסטות (Shift), גדל ב-4 בכל איטרציה.

התהליך מתבצע בעליית השעון:

- **איפוס ('reset='1')**: מאפס את acc, reg_A ו-step_count, ומגדיר reg_B עם '1' ב-LSB למניעת סיום מוקדם.
- **העמסה ('load='1')**: מעמיס את A לתוך reg_A עם sign extension, את B לתוך reg_B עם '0' ב-LSB ומאפס את acc ואת step_count.
- **חישוב ('shift='1')**: לוקח חלון (window) של 5 סיביות מ-reg_B, ממיר אותו לערך שלם (digit , ערך בין 16- עד 15), וקובע את התוצאה החלקית (partial) באמצעות case על פי ערכי Booth (למשל, A_ext, A_ext sll 1, וכו'). התוצאה מוסטת שמאלה לפי step_count, מוסיפה ל-acc, ו-reg_B מוסט ימינה ב-4 סיביות.
  התהליך נמשך עד ש-reg_B הופך ל-0 או -1, ואז 'finish='1' והתוצאה מועברת ל-product. השימוש ברדיקס 16 מפחית את מספר האיטרציות תוך שמירה על דיוק.

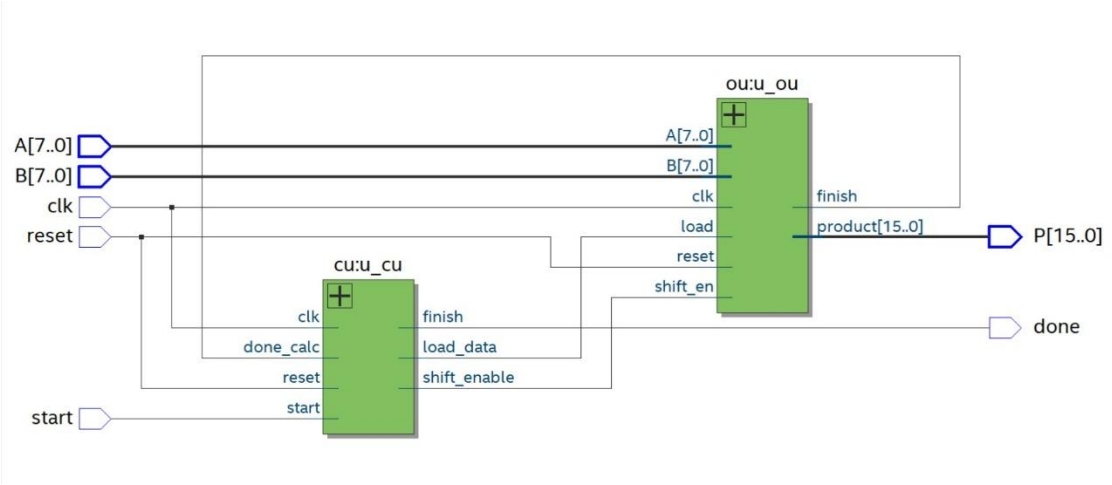
סינתזה של יחידת האופרציה:



3. יחידת ה-TOP:

יחידת ה-Top מחברת את ה-cu וה-ou לכדי מערכת שלמה. היא מגדירה משתנה גנרי N (ברירת מחדל 8, מתוך נוחות לכפולות של 4) ומספקת ממשק חיצוני: done, P, clk, start, A (N bits) ו-B (2N bit). ה-cu שולטת ב-ou באמצעות אותות load, shift_en ו-finish, וה-ou מחזירה את התוצאה ל-P. ה-Top מבטיח תיאום בין היחידות, תוך שמירה על גמישות באורך הקלטים באמצעות N.

סינתזה של יחידת ה-TOP:



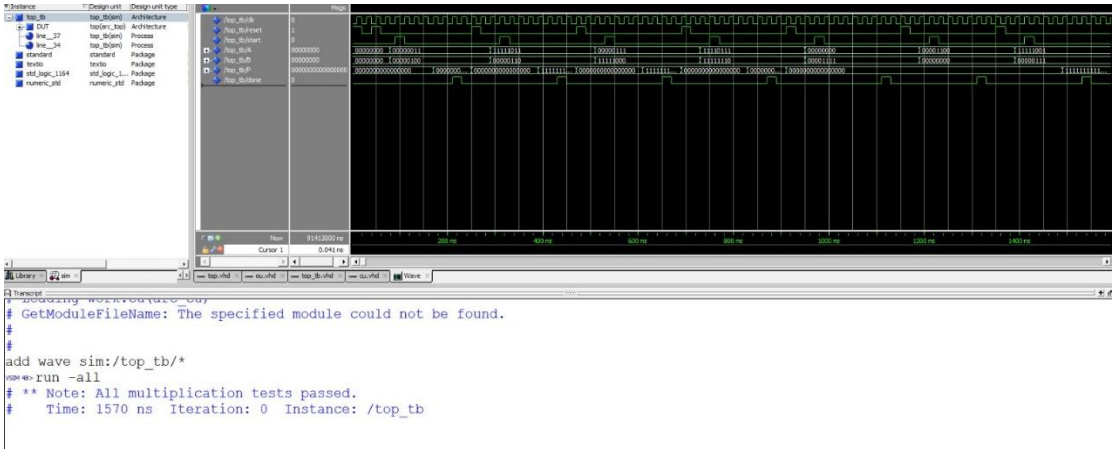
4. Testbench:

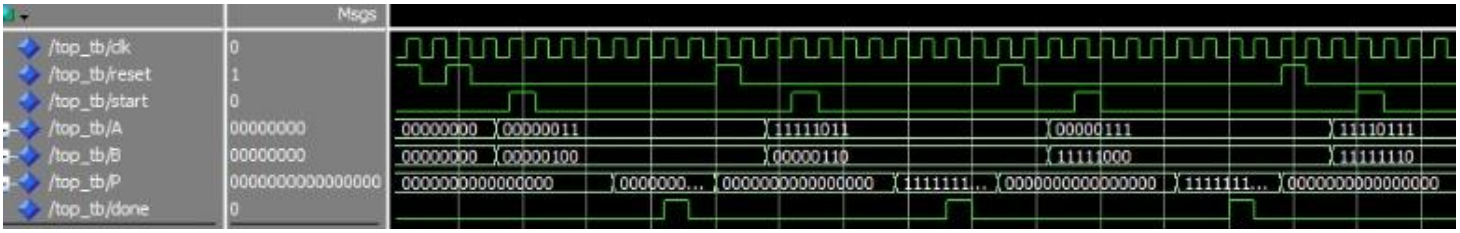
ה-Testbench (top_tb) בודק את תקינות המערכת על ידי סימולציה של שבעה זוגות קלטים (A, B) הכוללים מספרים חיוביים, שליליים ואפס. הוא מייצר שעון (clk) עם מחזור של 20 ננו-שניות ומבצע:

- **איפוס:** מפעיל reset='1' למשך 20 ננו-שניות לפני כל מקרה.
- **הזנת קלטים:** ממיר זוגות קלטים (מוגדרים ב-int_pair) ל-std_logic_vector ומזין אותם ל-A ו-B.
- **בדיקת תוצאה:** מפעיל start='1' למחזור אחד, ממתין ל-done='1', ומשווה את P לתוצאה הצפויה (A × B) באמצעות assert.
- **דיווח:** מדפיס הודעה על הצלחת כל הבדיקות או כשלון עם פרטי המקרה.

ה-Testbench מאמת את תפקוד המערכת תחת תנאים שונים, ומבטיח שהכפל מתבצע כראוי.

סימולציה:


Instance
Design unit
Design unit type
top_tb
top_tb(sim)
Architecture
DUT
top(arc_top)
Architecture
line__27
top_tb(sim)
Process
line__34
top_tb(sim)
Process
standard
standard
Package
textio
textio
Package
std_logic_1164
std_logic_1...
Package
numeric_std
numeric_std
Package
Msgs
/top_tb/clk
/top_tb/reset
/top_tb/start
/top_tb/A
/top_tb/B
/top_tb/P
/top_tb/done
Now
91413000 ns
Cursor 1
0.041 ns
200 ns
400 ns
600 ns
800 ns
1000 ns
1200 ns
1400 ns
Library
sim
top.vhd
cu.vhd
top_tb.vhd
cu.vhd
Wave
Transcript
GetModuleFileName: The specified module could not be found.
#
#
add wave sim:/top_tb/*
run -all
** Note: All multiplication tests passed.
Time: 1570 ns Iteration: 0 Instance: /top_tb


Msgs
/top_tb/clk
/top_tb/reset
/top_tb/start
/top_tb/A
/top_tb/B
/top_tb/P
/top_tb/done
0
1
0
00000000
00000000
0000000000000000
0
00000000
00000011
11111011
00000111
11110111
00000000
00000100
00000110
11111000
11111110
0000000000000000
0000000...
0000000000000000
1111111...
0000000000000000
1111111...
0000000000000000