

# היכרות עם הסימולטור ModelSim


אחראי התרגיל: נמרוד אדמוני [nimrodadmoni@campus.technion.ac.il](mailto:nimrodadmoni@campus.technion.ac.il)


שאלות בקשר לדרישות התרגיל יש להפנות לאחראי התרגיל דרך הפורום. בקשות מיוחדות יש לשלוח במייל

בקשות לדחייה ללא סיבה מוצדקת ידחו על הסף (ראו נוהל להגשה באיחור)

אין להגיש שום חלק מודפס – את החלק היבש יש לכתוב במעבד תמלילים (למשל: Word) ולצרף לחלק הרטוב. יש להגיש כקובץ pdf.

## 1. הנחיות כלליות

מסמן שאלות שיש לענות עליהן במסמך (החלק היבש). ניתן לענות גם באנגלית. 

מסמן חלק שיש לבצע בסימולטור. את תוצאת הסימולציה (waveform) יש לצרף לחלק היבש ולהסביר את התוצאות בצורה איכותית. Waveform אפשר לקבל בעזרת צילום מסך. 

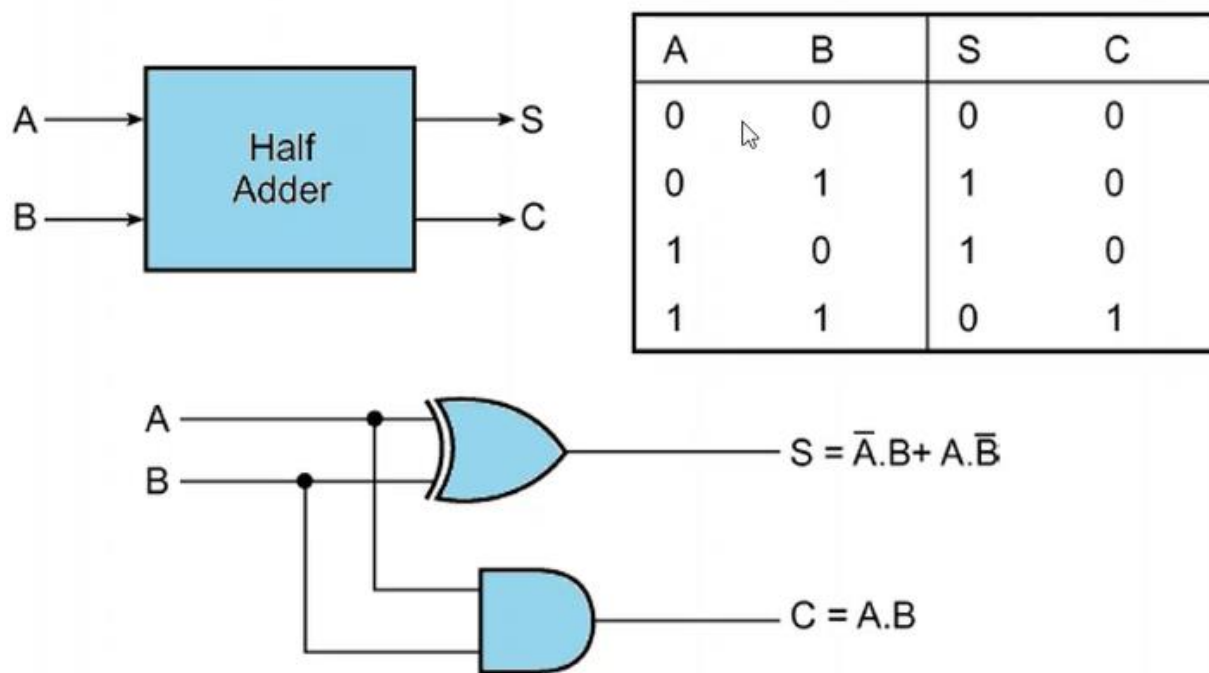
במקרה של קושי בפתרון הסעיפים הרטובים, יש לפנות למתרגלי הסדנאות בפורום SystemVerilog וסימולציות ב-Moodle, או בשעות הקבלה שלהם.

בכל שאלה על דרישות התרגיל, כלומר קשיים בהבנת דרישות הסעיפים השונים בתרגיל, יש לפנות אל אחראי התרגיל בפורום SystemVerilog וסימולציות ב-Moodle, או בשעת הקבלה.

שימו לב! למרות שהגשת התרגיל היא בזוגות, מומלץ לפתור לבד.

רקע

בתרגיל זה נדבג את פעולתו של רכיב ה-Half Adder. לרכיב ה-Half Adder שתי כניסות ושתי יציאות:



רכיב ה-Half Adder מממש את הפונקציות הלוגיות הבאות:

$$sum = a XOR b$$

$$carry = a AND b$$



ModelSim

2. התקנת הסימולטור

באתר המודל מופיע קובץ "Modelsim Installation Guide" עם הוראות התקנה מפורטות.

אם אתם נתקלים בבעיות בהתקנה, היעזרו במסמך בעיות ב-ModelSim וב-SystemVerilog הנמצא באתר הקורס (ModelSim\_and\_SystemVerilog\_FAQ.pdf). תכתבו פנייה בפורום רק אחרי שעברתם על קובץ זה.

במידה ואינכם מעוניינים להתקין את התוכנה על מחשבכם, ניתן להשתמש במחשבים הנמצאים בחוות המחשבים בפקולטה להנדסת חשמל שעליהם כבר מותקנת התוכנה. הסבר על העבודה בחוות בפקולטה להנדסת חשמל נמצא כאן:

<https://ece.technion.ac.il/services/computing-unit/undergraduate-students/?lang=he>

שימו לב כי התוכנה מיועדת למערכת הפעלה Windows. לא קיים installer למערכות הפעלה אחרות, וצוות הקורס לא יתמוך בהתקנת התוכנה על מערכות הפעלה אחרות.

לצורך פתרון הסעיפים הבאים, ניתן להיעזר בוידאו הנמצא בקישור הבא לפתרון התרגיל:

<https://www.youtube.com/watch?v=745QaT2XVB0&t=661s>

הצפייה בוידאו אינה חובה אך מומלצת.

### 3. הסבר על הסימולטור ModelSim

ModelSim הוא כלי וריפיקציה וסימולציה לשפות תיאור חומרה שונות, כגון SystemVerilog ו-VHDL. בקורס זה נבין מהי שפת תיאור חומרה ונלמד את שפת SystemVerilog.

השלבים בהרצת סימולציה בעזרת התוכנה הינם:



במדריך זה נעבור על כל אחד מהשלבים ונסביר כיצד לבצע אותו באמצעות דוגמה פשוטה.

## יצירת ספריית עבודה

צרו תיקייה חדשה בשם ModelSimTutorial. מומלץ למקם את התיקייה בנתיב שאינו כולל רווחים או אותיות בעברית. למשל, לא ב:

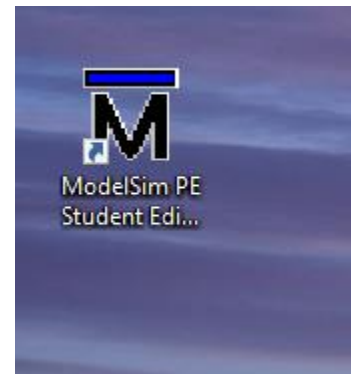
C:\ModelSimTutorial\2\טכניון\סמסטר\

אלא, למשל, ב:

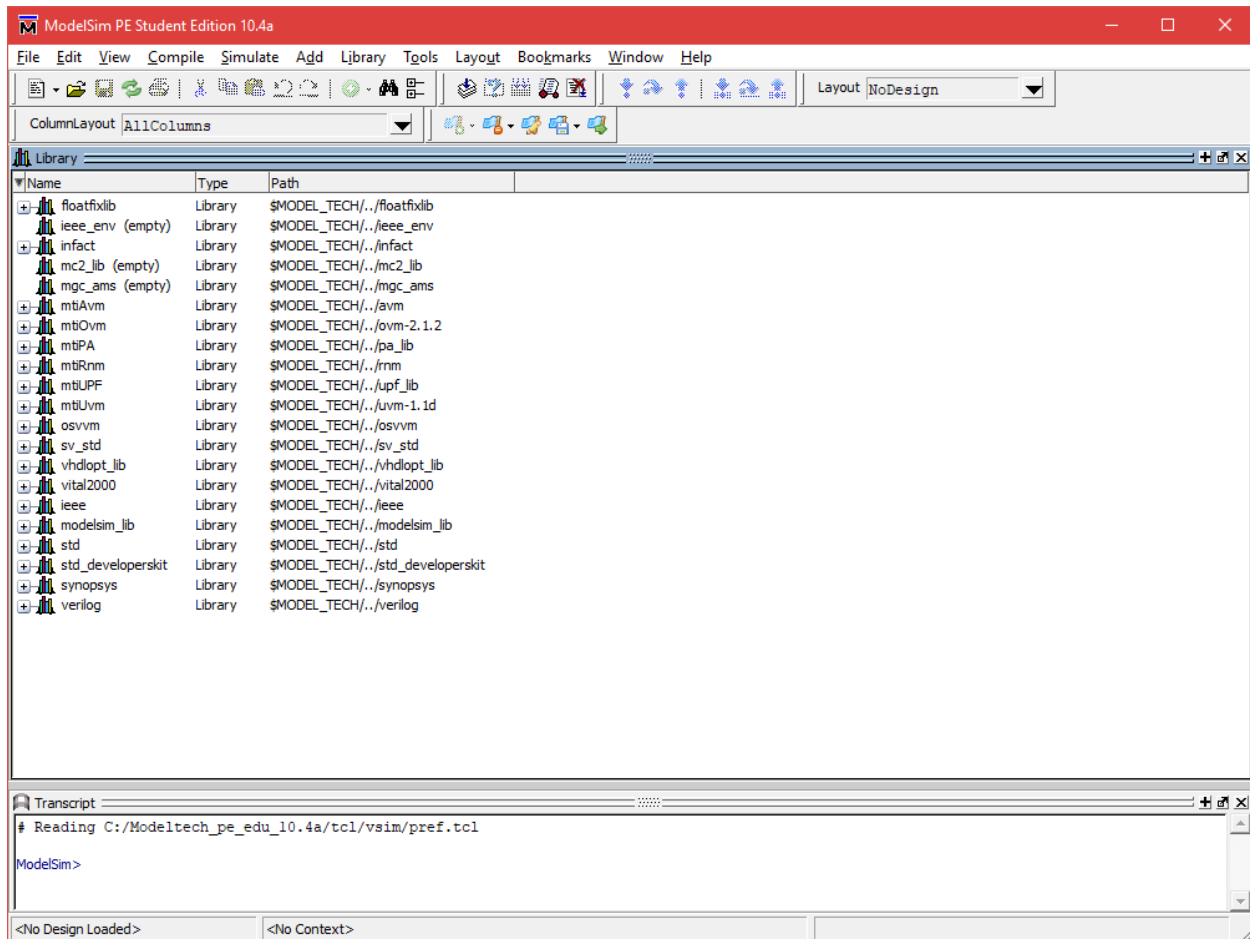
C:\technion\semester\_2\ModelSimTutorial

הורידו את הקבצים HA.tb ו-HA.tb.sv המופיעים ב-Moodle ושמרו אותם בתיקייה ModelSimTutorial הנ"ל. הקובץ HA.tb.sv מכיל מימוש של Half Adder הנלמד בהרצאות. הקובץ HA.tb.sv מכיל סביבת בדיקה למימוש זה הנקראת testbench.

פתחו את תוכנת ModelSim:



לאחר שהתוכנה נפתחת, מופיע החלון הבא:



בחלק העליון של החלון (Library) מופיעות ספריות מקומפלות שניתן להשתמש בהן בסימולציה. אנו ניצור ספריה חדשה שלתוכה נקמפל את התכן שלנו. בחלק התחתון של החלון (Transcript) מופיע Console בו ניתן להקליד פקודות ולקבל פלט מהסימולטור.

תחילה נרצה לשנות את ה-Working Directory לתיקייה שבה נעבוד – התיקייה שיצרתם קודם. הפקודה שנריץ היא:

```
cd "C:/technion/semester_2/ModelSimTutorial"
```

כאשר C:\technion\semester\_2\ModelSimTutorial היא התיקייה בדוגמה שלנו. שנו את נתיב התיקייה בפקודה בהתאם למיקום שבחרתם. שימו לב שאתם משאירים את המרכאות מסביב לנתיב, וכן שיש להשתמש בלוחסנים (slash-ים) רגילים ולא הפוכים (כלומר: / ולא: \).

על-מנת לקמפל קבצי תכן, תחילה יש ליצור ספריה חדשה שלתוכה התכן יקומפל. כבירת מחדל, סימולטורים מקמפלים מודולים ומחפשים אותם בתוך ספריה בשם work, לכן ניצור ספריה חדשה בשם הזה בעזרת הפקודה הבאה:

vlib work

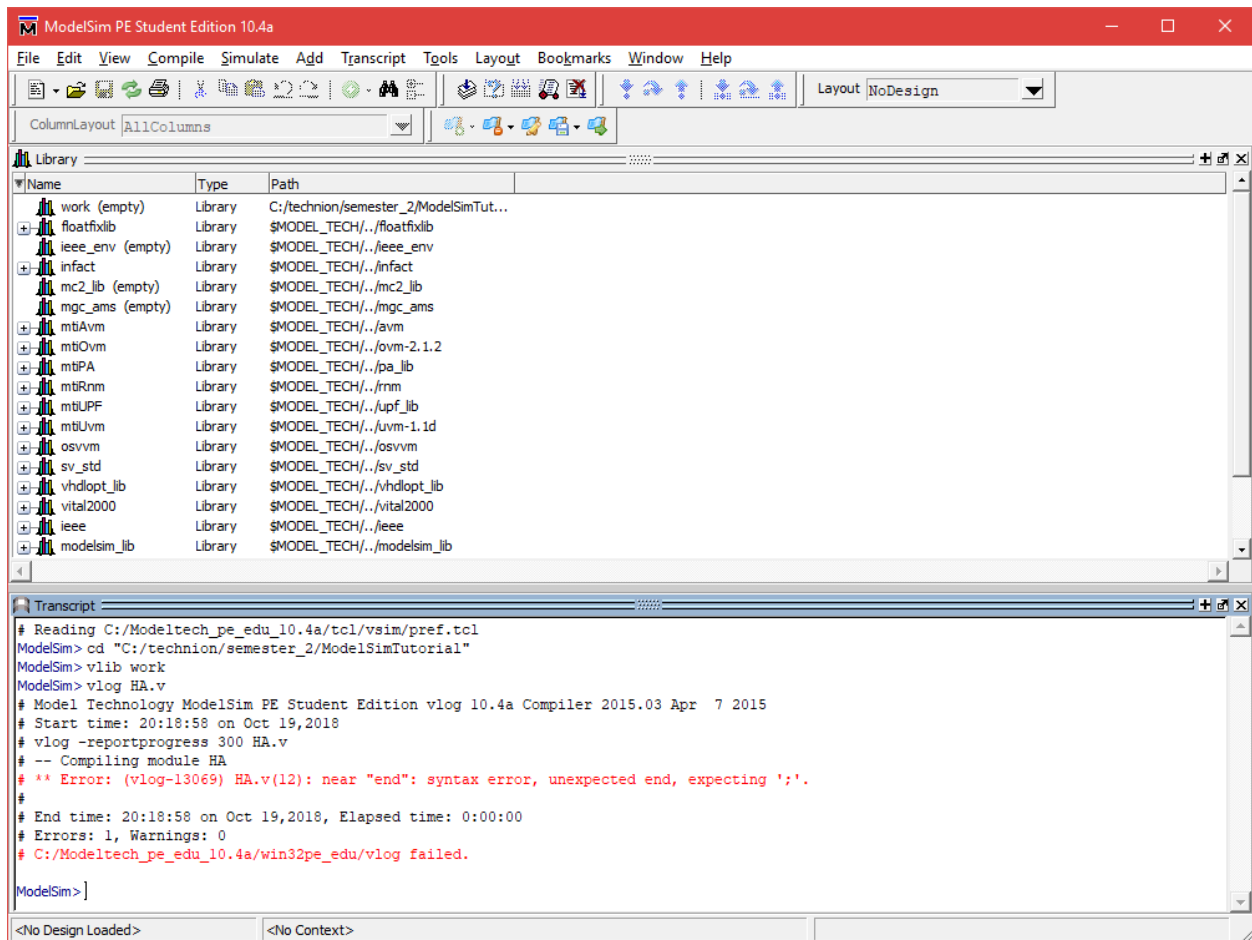
אין לשנות את שם הספרייה **work** לשם אחר.

## קומפילציה של קבצי התכן

כעת נקמפל את קובץ התכן בעזרת הפקודה הבאה:

vlog HA.sv

התבוננו בפלט המתקבל:



The screenshot shows the ModelSim PE Student Edition 10.4a interface. The Library pane on the left lists various libraries including 'work (empty)'. The Transcript pane at the bottom shows the following output:

```
# Reading C:/Modeltech_pe_edu_10.4a/tcl/vsim/pref.tcl
ModelSim> cd "C:/technion/semester_2/ModelSimTutorial"
ModelSim> vlib work
ModelSim> vlog HA.sv
# Model Technology ModelSim PE Student Edition vlog 10.4a Compiler 2015.03 Apr 7 2015
# Start time: 20:18:58 on Oct 19, 2018
# vlog -reportprogress 300 HA.sv
# -- Compiling module HA
# ** Error: (vlog-13069) HA.v(12): near "end": syntax error, unexpected end, expecting ';\'.
#
# End time: 20:18:58 on Oct 19, 2018, Elapsed time: 0:00:00
# Errors: 1, Warnings: 0
# C:/Modeltech_pe_edu_10.4a/win32pe_edu/vlog failed.
ModelSim>
```

כפי שניתן לראות, הקומפילציה לא עברה מכיוון שקיימות שגיאות syntax בקובץ התכן שלנו.

הדרך הנכונה לטפל בשגיאות syntax היא להתחיל תמיד מהשגיאה הראשונה שמופיעה בפלט הפקודה, לתקן אותה, לקמפל שוב ולחזור על התהליך עד שהקוד מתקמפל בהצלחה. הסיבה לכך היא ששגיאת syntax אחת יכולה לגרום לקומפיילר לפרש את המשך הקוד באופן לא נכון ולכן שגיאות עוקבות עלולות להיות לא רלוונטיות.

במקרה שלנו יש רק שגיאה אחת. ניתן לראות בפלט באיזה קובץ השגיאה ובאיזו שורה (12). כמו-כן, ניתן לראות את תיאור השגיאה.

על-מנת לתקן את הקוד פתחו אותו בעורך טקסט (מומלץ Notepad++), תקנו את הקוד (הוסיפו את הסימן החסר) וקמפלו שוב ע"י הרצת אותה פקודה.

כעת ניתן לראות שהקוד התקמפל (שימו לב שבפלט מספר ה-Errors הוא 0. מומלץ לטפל גם בכל ה-Warnings שלא מכשילים את הקמפול אבל כן מתריעים על בעיה או bad practice).

כעת נקמפל את ה-testbench. זהו קובץ שמטרתו לבדוק בסימולציה את התכן הממומש ב-HA.sv (מימוש של Half Adder) ע"י הכנסת קלטים שונים לתכן. ניתן להוסיף גם בדיקות אוטומטיות על הפלטים. ב-testbench המצורף לא קיימות בדיקות כאלה ולכן נבדוק את הפלטים באופן ידני.

ב-testbench הקלטים השונים מצוינים עם תחילית ב'1 לפני הערכים, למשל 1'b0 עבור ערך 0 ו-1'b1 עבור ערך 1. תחילית זו מציינת שמדובר בערך בעל ביט אחד בבסיס הבינארי. נלמד על הפורמט הנ"ל בסדנה השנייה, ובינתיים, ניתן להניח כי מדובר ב-0 ו-1 (ללא תחילית כלשהי) כפי שנלמד בסדנה הראשונה.

קמפלו את קובץ ה-testbench:

```
vlog HA_tb.sv
```

קומפילציה זו צריכה להסתיים בהצלחה.

## הרצת הסימולציה

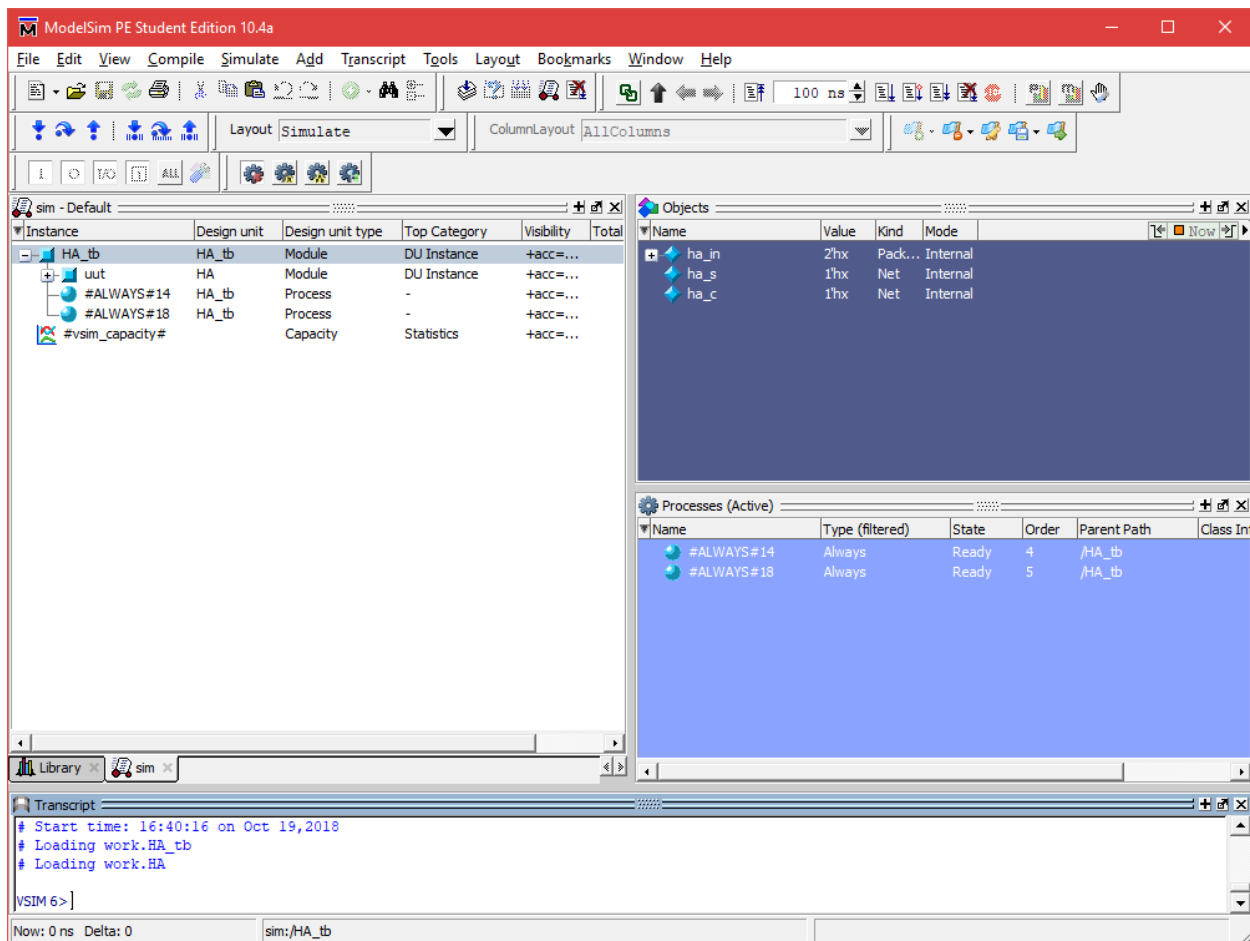
לאחר שהתכן וה-testbench מקומפלים, ניתן להריץ את הסימולציה. על-מנת להתחיל את הסימולציה נריץ את הפקודה הבאה:

```
vsim HA_tb
```

שימו לב שיש לכתוב HA\_tb ולא HA\_tb.sv (ללא הסימט sv)!

שימו לב שככלל, לעתים שגיאות או אזהרות קומפילציה יתקבלו גם בשלב זה.

לאחר הרצת הפקודה, במידה ואין שגיאות קומפילציה, הסימולטור יעבור לתצוגה הבאה:



## במידה וחלון התצוגה הנ"ל לא נפתח, המשיכו לסעיף הבא ונסו להקליד את הפקודה הבאה שבמדריך.

בצד שמאל (sim – Default) ניתן לראות את כל המודולים שקומפלו כאשר הם מוצגים בהיררכיה המתאימה בעץ. במקרה שלנו ניתן לראות שהרמה העליונה היא ה-`testbench` עצמו, `HA_tb`, ותחתיו אפשר לראות את התכן הנבדק `HA` שניתן לו השם `ut` (unit under test).

מצד ימין למעלה (Objects) ניתן לראות עבור כל מודול את האובייקטים המוכלים בו (כניסות, יציאות וסיגנלים פנימיים).

מצד ימין למטה (Processes (Active)) ניתן לראות את התהליכים המוגדרים בכל מודול.



## דיבוג

סימולציה של חומרה בד"כ כוללת בחינה של התנהגות המעגל לאורך זמן. כדי לבדוק את נכונות המימוש, ה-testbench מציב לכניסות התכן ערכים שמשתנים לאחר פרקי זמן מוגדרים לאורך ריצת הסימולציה. באופן זה ניתן לבחון בסימולציה אחת מספר קלטים שונים ולבדוק את תקינות המעגל.

בשונה מדיבוג תוכנה, דיבוג של חומרה נעשה באמצעות דיאגרמת גלים. זוהי דיאגרמה שבה מופיעים כל הסיגנלים (קלטים, פלטים וסיגנלים פנימיים) במעגל וניתן לראות את מצבם הלוגי על ציר זמן. ניתן לראות על דיאגרמת הגלים את כל הכניסות שהוצבו ולבחון את תקינות היציאות עבור כל נקודת זמן.

שימו לב שמטרת הסימולציה היא לבדוק תקינות לוגית ולא עמידה במשטר זמנים, לכן ההנחה היא שכל השערים אידיאליים ( $t_{pd}, t_{cd}, t_{cq} = 0$ ).

ModelSim אינו שומר כברירת מחדל את ההיסטוריה של כל הסיגנלים לאורך הסימולציה. לכן אם נרצה לבחון את המצב של סיגנל מסוים בדיאגרמת הגלים לאורך זמן הסימולציה נצטרך להוסיף אותו מראש לדיאגרמה.

הוסיפו את כל הכניסות והיציאות של ה-Half Adder ע"י הרצת הפקודה הבאה:

```
add wave sim:/HA_tb/uut/*
```

אם לא היה חלון של דיאגרמת גלים פתוח בתוכנה, לאחר הרצת פקודה זו הוא יפתח.

הריצו את הסימולציה למשך 80 ns ע"י הרצת הפקודה הבאה:

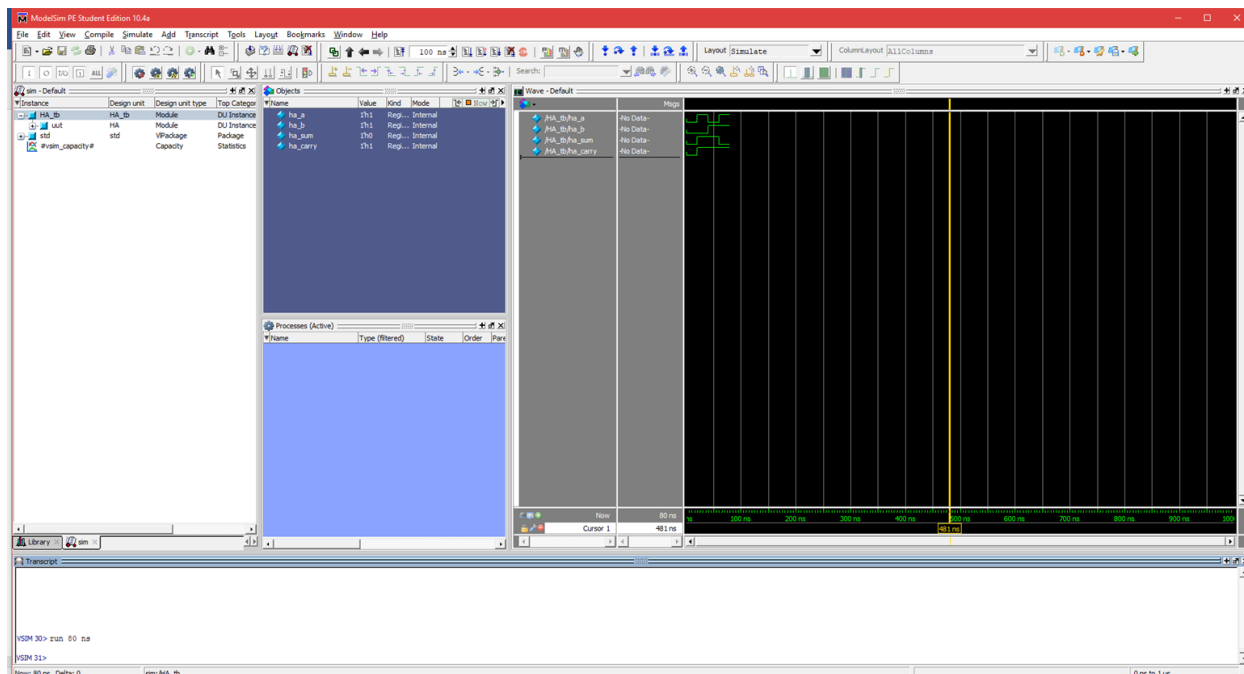
```
run 80
```

ניתן גם להריץ את הסימולציה מהתחלה עד סופה אוטומטית ללא הגדרת זמן ספציפי ע"י הפקודה:

```
run -all
```

(שימו לב שהתו "-" בפקודה הוא מקף, ולעתים כאשר הוא מועתק ישירות מה-pdf של התרגיל הוא מוחלף בתו אחר. לכן עדיף להקליד אותו בעצמכם).

לאחר הרצת הפקודה דיאגרמת הגלים תיראה כך:



הסרגל הירוק מתחת לדיאגרמה מציין את ציר הזמן. כל שורה בדיאגרמה מציינת סיגנל אחר ואת אופן שינויו בזמן הסימולציה, כאשר קו גבוה משמע הערך הלוגי 1, וקו נמוך משמע הערך הלוגי 0.

על מנת לבחון את דיאגרמת הגלים בצורה נוחה, נכיר מספר פעולות ופקודות שניתן לבצע בדיאגרמה. ראשית, לחצו עם העכבר על חלון הדיאגרמה כדי להפוך את החלון לפעיל.

הסמן (Cursor) הוא קו אנכי צהוב שממוקם בנקודת זמן מסוימת על דיאגרמת הגלים. ניתן להזיז אותו לכל נקודת זמן שנרצה ולראות בצד שמאל של הדיאגרמה (תחת העמודה Msgs) את הערך המספרי של כל הסיגנלים באותה נקודת זמן. ניתן לשנות את פורמט הייצוג של הערכים המספריים (עשרוני, הקסה-דצימלי, בינארי וכו') ע"י לחיצה עם הכפתור הימני בעכבר על הסיגנל הרצוי ובחירת הייצוג המספרי הרצוי תחת תפריט Radix.

ניתן לבצע Zoom על הדיאגרמה בעזרת הכפתורים שבסרגל הכלים למעלה (

או בעזרת קיצורי המקלדת: zoom in – c במיקום הסמן, – o zoom out, f – zoom fit (פרישת הדיאגרמה על כל החלון).

בצד שמאל של הדיאגרמה מופיעים שמות הסיגנלים הכוללים את ההיררכיה המלאה שלהם. ניתן להחליף לתצוגה של שמות הסיגנלים ללא היררכיה באמצעות קיצור המקלדת h.

## 1.1 צרפו את דיאגרמת הגלים שקיבלתם.



כפי שניתן לראות, ה-testbench שלנו בחן את כל המצבים האפשריים של כניסות a ו-b של ה-Half Adder. שימו לב למצב היציאה carry בזמן שבין ns 20 ל-ns 40. בזמן זה:

$$a = 1, b = 0$$

לכן היינו מצפים שתוצאת ה-carry תהיה 0. עם זאת, ניתן לראות בדיאגרמת הגלים שבזמן זה היציאה היא 1! זאת משום שקיימת טעות לוגית במימוש ה-Half Adder.

כדי לתקן את השגיאה, פתחו שוב את הקובץ HA.sv בעורך טקסט. התבוננו שוב בפירוט על רכיב ה-HA (סעיף 1) ובדקו היכן נפלה הטעות. תקנו את הקוד על מנת שיבצע את הנדרש.

מכיוון שקובץ הקוד נערך יש לקמפל אותו מחדש. הריצו שוב את פקודת הקימפול:

```
vlog HA.sv
```

אין צורך לקמפל מחדש את ה-testbench משום שלא ביצענו בו שינויים.

לאחר קימפול מחדש, יש להתחיל את הסימולציה מחדש ולהריץ אותה שנית. כשהסימולציה כבר פעילה ניתן להתחיל את הסימולציה מחדש בעזרת הפקודה:

```
restart -f
```

ModelSim שומר על הסיגנלים שנטענו לדיאגרמת הגלים ולכן אין צורך להוסיף אותם מחדש, אלא רק להריץ את הסימולציה שוב ע"י הפקודה:

```
run 80
```

## 1.2 בדקו שדיאגרמת הגלים שקיבלתם היא כצפוי וצרפו אותה.



1.3 הסבירו מה שיניתם בקוד של הקובץ HA.sv, ומה השתנה בדיאגרמת הגלים בעקבות התיקון.



### טיפ לסיום

ModelSim תומך בהרצת סקריפטים המכילים את הפקודות שראינו במדריך זה וכן הרבה פקודות אחרות. אפשרות זו שימושית מאוד במקרים בהם נרצה להריץ כמה פקודות ברצף מספר רב של פעמים (לדוגמה: לאחר כל תיקון של שגיאה בקוד נרצה תמיד לקמפל את הקוד, להתחיל סימולציה ולהריץ אותה).

ניתן לעשות זאת ע"י יצירת סקריפט בקובץ חדש בתיקיית העבודה שלנו עם סיומת .tcl. בתוכו יופיעו כל הפקודות הרצויות בשורות נפרדות. הרצת הסקריפט נעשית ע"י פקודת source מה-console, לדוגמה:

```
source compile_and_sim.tcl
```

### הכנה לסימולציה 1

בסימולציה 1 נשתמש ב"תכן היררכי", כלומר נתכנן מודולים פשוטים ונשתמש בהם כדי לממש מודולים מסובכים יותר. להלן קישור לוידאו המומלץ לצפייה כהכנה לסימולציה הבאה (הצפייה אינה חובה):

<https://youtu.be/Um-30GWDnuQ>

## חלוקת הציון

חלק יבש: 50%

חלק רטוב: 50%

## הוראות הגשה

- ההגשה בזוגות בלבד. ניתן לחפש בני זוג דרך פורום חיפוש שותפים. הגשה ללא בני זוג ללא אישור מראש תגרור הורדה בציון של 10 נקודות.
- יש להגיש את הקבצים הבאים (ואותם בלבד):
  - HA.sv (לאחר השינויים)
  - Dry.pdf
- יש לארוז את כל הקבצים הנ"ל (קבצי הקוד וקובץ התשובות של החלק היבש בפורמט pdf בלבד) בקובץ zip אחד, בשם **<id>.zip**, כאשר <id> זהו מס' ת.ז. מלא של אחד מבני הזוג.
- בתחילת קובץ התשובות לחלק היבש יש לכתוב שמות ות.ז. של כל אחד מהסטודנטים בטבלה כמו בדוגמה הבאה:

שם 1	123456789
שם 2	987654321

- שימו לב להגיש קובץ בפורמט zip בלבד! (לא rar, לא z7 ולא שום תוכנת כיווץ אחרת)
- אין להדפיס אף חלק בתרגיל. קובץ zip שיגיע ללא חלק יבש (קובץ pdf) יגרור ציון 0 על התרגיל כולו.
- הסימולציה תעבור בדיקה אוטומטית. אנו נריץ סימולציה על הקבצים שתספקו ולכן חשוב להשתמש באותם module-ים (שמות ו-port-ים) המופיעים בתרגיל.
- עליכם לעקוב אחרי הודעות אשר מתפרסמות באתר הקורס, הודעות אלו מחייבות.
- כל שאלה על התרגיל אשר איננה בקשה אישית צריכה להישאל דרך הפורום באתר הקורס.
- אנא בדקו היטב את הקבצים לפני ההגשה. טענות מסוג "אבל בבית זה עבד נכון" לא תתקבלנה. מומלץ לבדוק את תקינות הקוד על סביבה "נקייה" (למחוק את הספרייה work, ליצור אותה מחדש ולקמפל לתוכה מחדש את כל קבצי הקוד).
- [קוד שלא מתקמפל יגרור הורדת נקודות מלאה של הסעיף.](#)
- שימו לב ל-Warnings שמתקבלים כפלט של הסימולטור. נקודות יורדו על Warnings חמורים.
- הגשה באיחור ללא אישור: על כל יום איחור יורדו 5 נקודות. הגשות באיחור מותרות עד שבוע מתאריך ההגשה.