

**מעבדה ב-VLSI ספרתי - 0450111**

**ווריקציה - ביצוע**

<http://www.ee.technion.ac.il/vlsi/>

[הערות נא לשלוח ל-goel@ee](mailto:goel@ee)

כל הערה תתקבל בברכה!

עדכון אחרון - 11:16 19/09/2024

מסמך זה כתוב בלשון זכר ע"מ להקל על הכתיבה אך מתייחס לגשים ולגברים כאחד. עמכם הסליחה.

## תוכן עניינים

3	ביצוע ניסוי .....
4	1. הזנת מידע ל- DUT .....
6	2. יצירת Sequences דטרמיניסטיים והעברתם ל- DUT .....
7	3. יצירת Sequences אקראיות וקביעת אילוצים על ה- sequences .....
8	4. כיסוי הבדיקות Coverage .....

כאמור, רכיב ה-DUT עבור חלק זה של הניסוי הוא מימוש מאיץ של מערכת לומדת (בשם NeuralNet) הממומש בחלק הראשון בשפת Systemverilog. בחלק זה של הניסוי יסופק כל הקוד של ה-DUT ושל סביבת האימות. **לא תמיד יסופק הקוד המלא**. בכל סעיף יהיה צורך לזהות מה הבעיה ולהשלים את החסר בעזרת ההסברים שמופעים במסמך זה וכמובן במידת הצורך גם בעזרת המדריך. בצורה זאת ניתן יהיה להתמקד בסוגיות שונות של סביבת האימות. שים לב שמסופק גם reference model אבל **חשוב להבין שה-reference model מספק תוצאה נכונה רק עבור כניסות ולידיות**.

**חשוב:** הרץ פקודת **make** מריצה את כל הפקודות שבקובץ בשם **Makefile**. קובץ זה מכיל הוראות קומפילציה והוראות להרצת ה-**test**. במילים פשוטות, **make** למעשה מריצה את ה-**test**. פקודת **make** יוצרת פלט ארוך. פקודה מסוג :

`make | grep InputImage`

מאפשרת לנו להציג רק השורות שמעניינות אותנו, במקרה זה, שורות המכילות את המילה **InputImage**.

עבור לספריית VER/EX1. ספרייה זאת מכילה את קבצי ה-DUT :

- CNeuron.sv
- FCNeuron.sv
- Pooling.sv
- NeuralNet\_cont.sv
- NeuralNet.sv - top level module

מימוש ה-reference model :

- NeuralNet\_Ref.sv

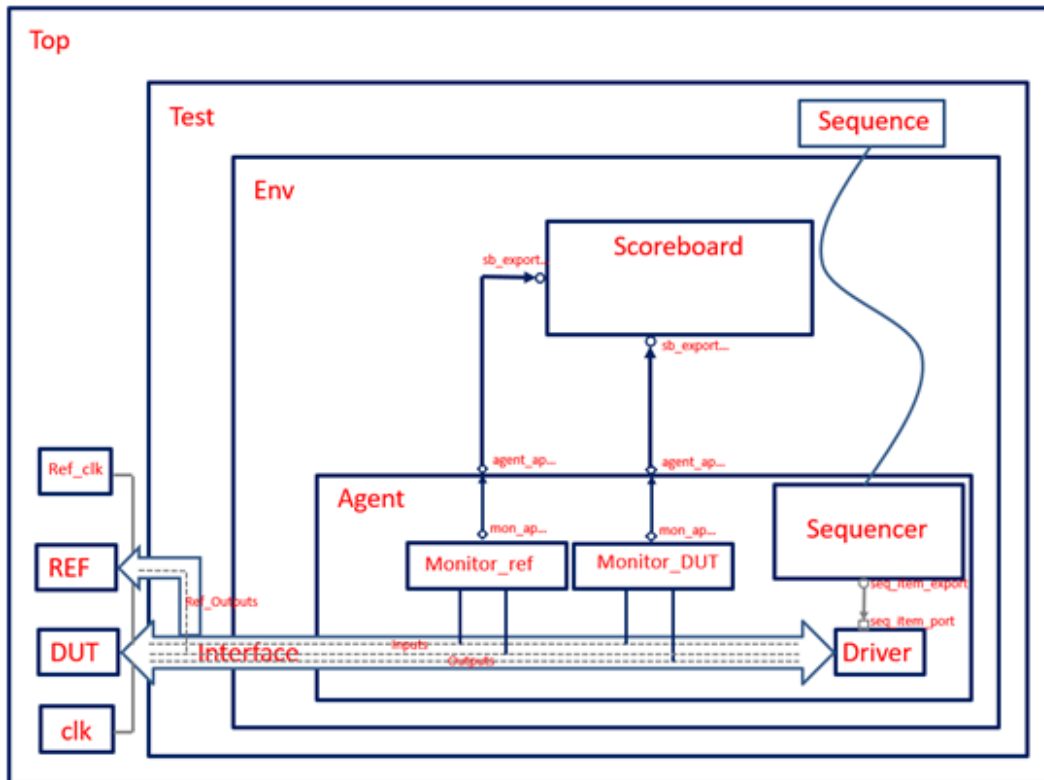
מימוש הממשקים ל-DUT ול-reference model :

- NeuralNet\_if.sv
- NeuralNet\_Ref\_if.sv

מימוש כל קיבצי סביבת האימות :

- NeuralNet\_sequencer.sv
- NeuralNet\_driver.sv
- NeuralNet\_monitor.sv
- NeuralNet\_agent.sv
- NeuralNet\_scoreboard.sv
- NeuralNet\_env.sv
- NeuralNet\_pkg.sv
- NeuralNet\_test.sv
- NeuralNet\_test\_tb.sv

כפי שהוסבר ברקע, הקבצים הנ"ל ממשים סביבה וורטיפיקציה בעלת ההבנה הבא:



### 1. הזנת מידע ל- DUT

המטרה של סעיף זה היא להתעמק בשיטות השונות להזנת המידע ל- DUT. עבור לספריית EX1 בעזרת:

`cd /users/diglabN/student1_student2/DesignLib/VER/EX1` (N=1,2,3 or 4)

- הרץ את הסימולציה באמצעות הפקודה:

`make | grep "Test|Result"`

זה מציג את תוצאות ה- DUT וה- reference model ותוצאת ההשווה ביניהן.

- הרץ את הסימולציה שוב באמצעות הפקודה:

`make | grep InputImage`

כאן רואים את ערכי ה- InputImage שמוזנים ל- DUT.

**1.1** האם תוצאות ה- DUT וה- reference model זהות? מה יכולה להיות הסיבה (בהנחה שהמימוש נכון)? **אין להתחשב בתוצאה שהיא 0.**

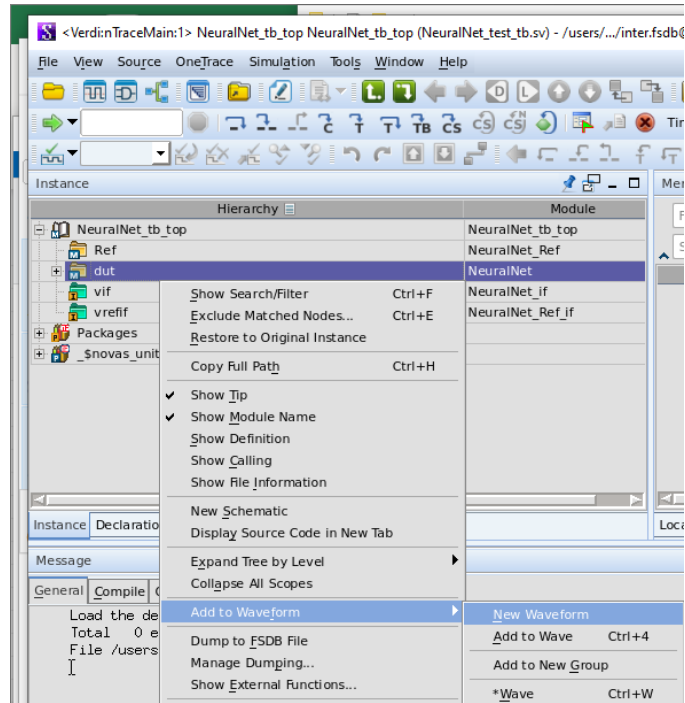
- עיין בקבצים `NeuralNet_test_tb.sv` ו- `NeuralNet_driver.sv`.

**1.2**: אלו כניסות של ה- DUT מסופקים ע"י כל אחד מהקבצים הנ"ל?

- פתח את הממשק הגרפי עם:

`./simv -gui`

בחר ב- DUT ובחר ב- `Add to Waveform-> New Waveform`



להרצת הסימולציה, לציץ על החץ הירוק בצד שמאל למעלה :  
 שים לב לזמני השינוי של Learn, rst ו-Classify.  
**1.3**: שמור את צורת הגלים בדו"ח.  
 סגור את GUI עם File-Exit.

כעת נראה כיצד ניתן להזין סיגנלים ממקומות שונים:  
 פתח את NeuralNet\_test\_tb.sv ושים בהערה (//) את כל משפט ה- initial שמספק את הערכים ל- Learn, rst ו-  
 Classify. פתח את NeuralNet\_driver.sv והוסף קוד שמזין ערכים ל- Learn, rst ו- Classify זהות לערכים  
 שקיבלו ע"י ה- NeuralNet\_test\_tb.sv לפני מחיקת משפט ה- Initial. שים לב שעליך להוסיף קוד רק למצבים  
 0, 1 ו- 2. הרץ את הסימולציה ופתח את הממשק הגרפי עם :

```
make  
./simv -gui
```

וודא שהערכים שהזנת נכונים. אם לא תקן את הקוד.

**1.4**: הוסף את צורת הגל ואת הקוד של שלשת המצבים הנ"ל לדו"ח.

- הרץ את הסימולציה שוב באמצעות הפקודה :

```
make | grep InputImage
```

**1.5**: כמה ערכים שונים מתקבלים עבור InputImage? איך קובעים מספר ה- InputImage שנוצרים?  
 - **אם אתה לא יודע את התשובה, עליך לפנות למנחה.** שנה את המספר ל 5, 10, 20 ועבור כל מקרה בדוק שמספר  
 הכניסות המבוקש אכן מתקבל בעזרת :

```
make | grep InputImage
```

**1.6**: מה עוד יכול להשפיע על מספר ה- InputImage שמתקבלים? תקן את הקוד עד שמתקבל מספר הכניסות הרצוי  
 ורשום את השינוי שביצעת בדו"ח.  
 - **אם אתה לא מצליח, עליך לפנות למנחה.**

- 1.7:** פתח את הקובץ `NeuralNet_driver.sv`: הסבר בדו"ח:
- מה הם `vif` ו-`vrefif`?
  - מתי מעבירים ל-`DUT` את התמונה?
  - מתי מעבירים ל-`Reference Model` את התמונה?
  - את הקוד במצבים 2,3,4 ו-5.

## 2. יצירת Sequences דטרמיניסטיים והעברתם ל-DUT

המטרה של סעיף זה היא להתמקד ביצירת sequences דטרמיניסטיים והסדר שבו הם מועברים ל-DUT. בקובץ `NeuralNet_sequencer.sv` יוצרים סידרה של כניסות עם ערכים אי זוגיים 1,3,5,.... עבור לספריית EX2 בעזרת:

```
cd /users/diglabN/student1_student2/DesignLib/VER/EX2 (N=1,2,3 or 4)
```

- עיין בקובץ `NeuralNet_driver.sv`
- שים לב לפקודת ההדפסה `$display` שתאפשר לדעת איזו יחידה הדפיסה את ההודעה.
- עיין בקובץ `NeuralNet_sequencer.sv`
- שים לב לפקודת ההדפסה ולדרך יצירת ה-`sequence`.
- הרץ את הסימולציה באמצעות הפקודה:

```
make | grep InputImage
```

**2.1:** הוסף את הפלט לדו"ח. אילו יחידות הדפיסו את השורות?

- פתח את קובץ `NeuralNet_sequencer.sv`
- הוסף // לשורות:

```
//start_item(ml_pkt_odd);
//finish_item(ml_pkt_odd);
```

- הרץ את הסימולציה באמצעות הפקודה:

```
make | grep InputImage
```

**2.2:** הוסף את הפלט לדו"ח. אילו יחידות הדפיסו את השורות?

- פתח את הממשק הגרפי עם:

```
./simv -gui
```

בחר ב-DUT ובחר ב-`Add to Waveform-> New Waveform` כפי שעשית בסעיף הקודם.



- להרצת הסימולציה, לחץ על החץ הירוק בצד שמאל למעלה:

**2.3:** האם מתקבלות צורת גל? מה החשיבות של השורות הנ"ל?

**2.4:** מה החשיבות של השורות `start_item` ו-`finish_item`?

- הסר את ה-`// שהוספת`.

- הוסף לקובץ `NeuralNet_sequencer.sv` את כל שורות הקוד הדרושות ליצירה **סידרה נוספת** של ערכים זוגיים כך שתהינה הדפסות של הסדרות גם מה-`sequencer` וגם מה-`driver`.
- הרץ את הסימולציה באמצעות הפקודה:

```
make | grep InputImage
```

- בדוק שאכן מתקבלים הערכים שנכונים. אם לא, תקן את הקוד.

**2.5:** הוסף את הפלט לדו"ח וצרף את הקוד החדש של `NeuralNet_sequencer.sv` לדו"ח.

- פתח את הממשק הגרפי עם:

```
./simv -gui
```

בחר ב-DUT ובחר ב-`Add to Waveform-> New Waveform`.



להרצת הסימולציה, לצץ על החץ הירוק בצד שמאל למעלה :

**2.6:** הוסף אותם וגם את צורות הגל לדו"ח. הסבר את הערכים של הסיגנל InData.  
 - שנה את הקובץ NeuralNet\_sequencer.s כך שהסיגנל **InData** של ה-DUT יקבל את הערכים :  
 1, 2, 3, 4, 5 .....32

**חשוב :** עבור סדרות ארוכות, חייבים למנוע את המצב שפקודות בקובץ NeuralNet\_test\_tb.sv יפסיקו את הסימולציה לפני הזמן. פתח את הקובץ NeuralNet\_test\_tb.sv ושנה את השורה:

```
#400 $finish;
```

ל:

```
#1200 $finish;
```

- הרץ את הסימולציה באמצעות הפקודה :

```
make | grep InputImage
```

- בדוק שאכן מתקבלים הערכים שנכונים. אם לא, תקן את הקוד.

- פתח את הממשק הגרפי עם :

```
./simv -gui
```

**2.7:** הוסף את הקוד, הפלט של make ואת צורות הגל לדו"ח.

### 3. יצירת Sequences אקראיות וקביעת אילוצים על ה-sequences

עבור לספריית EX3 בעזרת :

```
cd /users/diglabN/student1_student2/DesignLib/VER/EX3 (N=1,2,3 or 4)
```

- פתח את הקובץ :

```
NeuralNet_sequencer.sv
```

- שים לב לשורה :

```
if (!ml_pkt.randomize()) .....
```

**3.1:** הסבר את ההבדל בתהליך יצירת ה-sequence בסעיף הזה לעומת הסעיף הקודם.

- הרץ את הסימולציה באמצעות הפקודה :

```
make | grep InputImage
```

**3.2:** האם הערכים של InputImage הם ולידיות ?

- כפי שהכנת בבית, הוסף constraint בשם c\_InputImage למחלקה NeuralNet\_transaction על מנת להגביל את הערכים המוגרלים ל-

```
'72h01ff01ff01ff01ff01,72'hff01ff01ff01ff01ff,72'h01ffffff01ffffff01,72'hffff01ff01ff01ffff
```

- הרץ את הסימולציה שוב באמצעות הפקודה :

```
make | grep InputImage
```

**3.3:** הוסף לדו"ח את הקוד שרשמת. הוסף לדו"ח את כל הערכים שהוגרלו עבור InputImage.

- פתח את הקובץ NeuralNet\_sequencer.sv והוסף את השורה הבא:

```
ml_pkt.c_InputImage.constraint_mode(0);
```

לפני השורה:

```
start_item(ml_pkt);
```

- הרץ את הסימולציה שוב באמצעות הפקודה :

```
make | grep InputImage
```

**3.4:** הוסף את הפלט לדו"ח והסבר את ההשפעה של השורה שהוספת.

- שנה את ה- mode ל 1 והרץ את הסימולציה שוב.

**3.5:** הוסף את הפלט לדו"ח והסבר את ההשפעה של השינוי.

- הוסף ל- transaction אילוף שמגביל את ערך הכניסות ל :

- 72'h01ff01ff01ff01ff01,72'h01ffffff01ffffff01

**3.6:** הוסף לדו"ח את הקוד שרשמת. הוסף לדו"ח את כל הערכים שהוגרלו עבור InputImage.

- הוסף ל- sequencer משפט שמבטל את האילוף. בדוק שזה עובד.

**3.7:** הוסף לדו"ח את הקוד שרשמת לדו"ח.

- בטל את כל האילוצים ב- sequencer וב- transaction.

- פתח את קובץ בשם constraint. כאן ניתן לראות שיטה שונה לקביעת האילוצים. השתמש בשיטה זאת על מנת להגביל את ההגרלה לשני ערכים בלבד.

- הרץ את הסימולציה שוב באמצעות הפקודה :

make | grep InputImage

**3.8:** איפה הוספת את הקוד? הוסף את הפלט לדו"ח והסבר את ההשפעה של השורה שהוספת.

- הרץ את הסימולציה באמצעות הפקודה :

make | grep Test

**3.9:** האם ה- test עובר בהצלחה, כלומר האם תוצאות ה- DUT וה- reference model זהות?

- פתח את הממשק הגרפי עם :

./simv -gui

בחר ב- Dut ובחר ב- Add to Waveform-> New Waveform.

בחר ב- Ref ובחר ב- Add to Waveform-> Add to Wave.

בדוק את הערכים של InputImage שה- reference model מקבל.

**3.10:** הסבר מדוע תוצאות ה- DUT וה- reference model אינן זהות? כיצד ניתן לתקן את הבעיה ?

- בצע את התיקון והראה אותו למנחה. אם לא מצאת את הפתרון, העזר במנחה.

- הרץ את הסימולציה באמצעות הפקודה :

make | grep Test

**3.11:** האם ה- test עובר בהצלחה, כלומר האם תוצאות ה- DUT וה- reference model זהות? הסבר את התוצאות.

הערה: שים לב לנוחיות בשימוש של reference model.

#### 4. כיסוי הבדיקות Coverage

כאמור, coverage מהווה מדד של יכולת הבדיקות לעבור על כל הפונקציונליות של התכנון. בסעיף זה נכיר סוגים שונים של coverage. עבור לספרית EX4 בעזרת :

cd /users/diglabN/student1\_student2/DesignLib/VER/EX4 (N=1,2,3 or 4)

כמו שבסעיף הקודם, מופיעים כאן כל קבצי המימוש וכל קבצי סביבת האימות.

#### כיסוי בעזרת הוראות קומפילציה

כדי לצור דו"ח על הכיסויים מסוימים, יש להוסיף את האופציות המתאימות לפקודות הקומפילציה.

- פתח את ה- Makefile והוסף את האופציות הבאות לפקודות הקומפילציה. יש לעשות זאת לקטע של ה- VCS ולמקם את השורות לפני השורה שמתחילה ב- \$(UVM\_HOME) :

-cm\_cond allops+anywidth+event -cm\_noconst\



-cm line+cond+fsm+branch+tgl -cm\_dir ./coverage.vdb\  
- וגם לקטע של ה- SIMV יש להוסיף את השורה :  
-cm line+cond+fsm \  
- מיד אחרי השורה +UVM\_TR\_RECORD +UVM\_LOG\_RECORD :

**4.1:** הרץ את הפקודה vcs -help. הסבר כל אחת מהאופציות בשתי השורות הנ"ל.  
- הרץ את הסימולציה שוב באמצעות הפקודה :

make

- על מנת לראות את תוצאות הכיסוי בצע :

```
urg -dir coverage.vdb  
cd urgReport  
firefox dashboard.html
```

- לחץ על modlist.

**4.2:** הוסף לדו"ח את הטבלה שמסכמת את ה- coverage.

לחץ על NeuralNetCont (מכונת המצבים).

**4.3:** עבור מכונת המצבים :

א. רשום את אחוז הכיסוי של השורות. רשום דוגמא של שורה שלא כוסתה אם קיימת כזאת.

ב. רשום את אחוז הכיסוי של התנאים. רשום דוגמא של תנאי שלא כוסה אם קיים כזה.

ג. רשום את אחוז הכיסוי של המצבים. רשום דוגמא של מצב שלא כוסה אם קיים כזה.

ד. רשום את אחוז הכיסוי של המעברים. רשום דוגמא של מעבר שלא כוסה אם קיים כזה.

- סגור chrome.

## סיום הניסוי !