

מעבדה ב-VLSI ספרתי - 0450111

ווריקציה - הכנה

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

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

כל הערה תתקבל בברכה!

עדכון אחרון - 11:15 19/09/2024

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

תוכן עניינים

3	הכנה
3	שאלות רקע
4	כניסות ויציאות לסביבת ה-UVM
4	קביעת אילוצים על ערכים מוגרלים

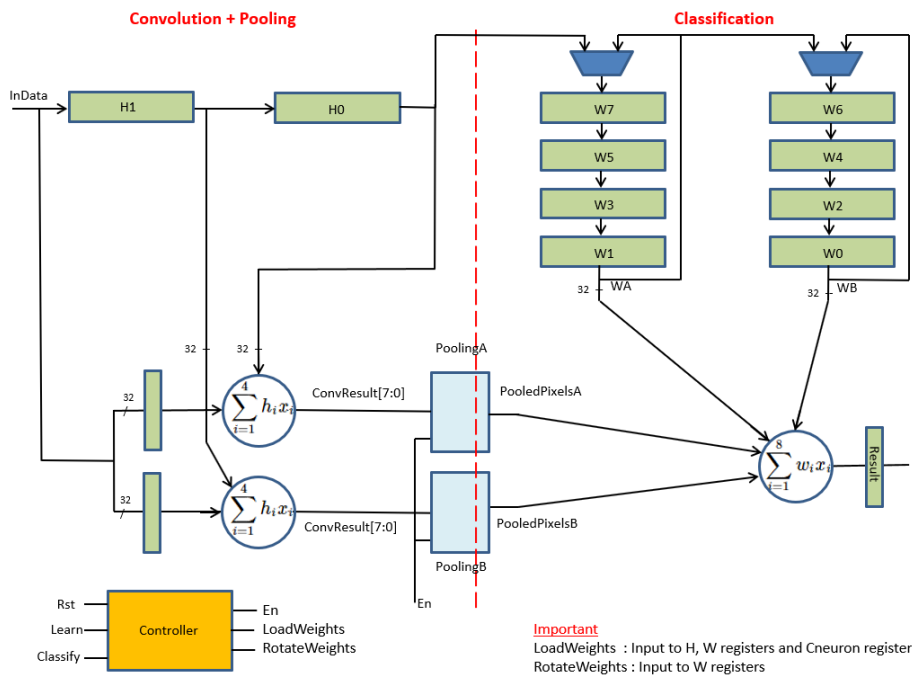
חשוב : לפני הכנת דו"ח המכין חובה לקרוא לפי הסדר:

- חוברת הדרכה על הוראות הבטיחות: **VerificationInformation.pdf**
- חוברת הסיכום של כל הניסוי : **UVMBBackground.pdf**

שאלות רקע

1. מה זה reference model ?
2. מה הפאזות העיקריות של המחלקות בסביבת ה-UVM ? הסבר בקיצור את התפקיד של כל הפאזה.
3. אלו מודולים מופיעים בבלוק העליון ?
4. באיזו מחלקה מגדירים את מבנה הכניסות ל-DUT ?
5. איזו מחלקה יוצרת סידרה של הכניסות שמוזנות ל-DUT ?
6. איזו מחלקה דוחפת כניסות ל-DUT ?
7. איזו מחלקה דוגמת את יציאות ה-DUT ?
8. איזו מתודה גורמת להפעלת ה-test ?
9. הסבר בקיצור את סוגי הכיסוי שמוזכרים במסמך זה.
10. מה חשוב לכסות במישוש של מכונת מצבים ?
11. מה המשמעות של ציון 100% עבור בדיקת כיסוי ?

רכיב ה-DUT הוא מימוש מאיץ של מערכת לומדת שממומשת בחלק הראשון של המעבדה. חשוב להיזכר בממשק שלו ואת האופן שבו מוזנות כניסות למעגל. המעגל פועל בשני שלבים, לאחר ה-reset מתבצע השלב הלימוד (עשרה מחזורים) ולאחר מכן (כל יתר הזמן) מתבצע שלב הסיווג. הכניסה למעגל היא למעשה תמונה בגודל 3x3 פיקסלים ששמורה במערך בשם InputImage. מערך זה אינו מופיע באיור מס' 1. להלן סכמת המלבנים של המערכת :



איור מס' 1 – הארכיטקטורה של מאיץ של מערכת לומדת

כפי שניתן לראות, המערכת מורכבת מרגיסטרים, שני Convolution Neurons, שתי יחידות Pooling, Fully Connected Neuron אחד והבקר.

כניסות ויציאות לסביבת ה-UVM

להלן חלק של מימוש ה-transaction עבור רשת שלנו:

```
class NeuralNet_transaction extends uvm_sequence_item;
    rand logic [71:0] InputImage;
    logic [7:0] result;
    .
    .
    .
endclass: NeuralNet_transaction
```

שים לב לנתונים שה-transaction מספק ומקבל מ-DUT. יתר הכניסות מוזנות בדרכים אחרות.

12. כיצד מזינים data ל-DUT?

13. כזכור, ה-InputImage מגיע לרכיב דרך InData. כמה מחזורים דרושים להזין מידע זה ל-DUT?

סביבת ה-UVM מכילה Reference Model שמספק תוצאות נכונות (עבור כניסות ולידיות בלבד) לטובת ההשוואה עם תוצאות ה-DUT.

```
virtual function void compare;()
if (transaction_dut.result == transaction_ref.result) begin
`uvm_info("compare", {"Test: OK!"}, UVM_LOW)
end else begin
`uvm_info("compare", {"Test: Fail!"}, UVM_LOW)
end
endfunction: compare
```

14. כמה השוואות תתבצענה עבור כל ערך של InputImage? מדוע?

קביעת אילוצים על ערכים מוגרלים

קיימות שיטות רבות לגרום לערכים מוגרלים לקיים אילוצים המוגדרים. נכיר מספר דרכים.

דוגמא 1:

```
`define MAX_D 100
`define MIN_D 50
class Base;
    rand integer Var;
    constraint range { Var < MAX_D ; Var > MIN_D ;}
endclass
```

דוגמא 2:

```
class set_mem;
    rand bit [0:2] Var;
    constraint range { !(Var inside {0,1,5,6});}
endclass
```

- 15. הסבר את הקוד של דוגמא 1.
- 16. הסבר את הקוד של דוגמא 2.
- 17. בעזרת ההסבר ב:

http://www.testbench.in/CR_15_CONSTRAINT_EXPRESSION.html

הגדר משתנה Var מסוג integer אשר מקיים את הפילוג הבא כאשר מגרילים אותו :

var = 1 – 10% , var = 2 – 20%, var = 3 – 30% , var = 4 – 40%

18. רשום עבורו משפט להוספה של constraint בשם c_InputImage למחלקה NeuralNet_transaction על מנת להגביל את הערכים המוגרלים ל-

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