The Andrew and Erna Viterbi Faculty of ELECTRICAL & COMPUTER ENGINEERING

Technion Israel Institute of Technology

המעבדה ל- VLSI Laboratory

מעבדה ב- VLSI ספרתי - 0450111

סינתזה ותכנון Layout ביצוע - (BackEnd)

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

הערות נא לשלוח ל-goel@ee

כל הערה תתקבל בברכה!

עדכון אחרון - 27/03/2025 12:16

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

תוכן עניינים

ביצוע הניסוי.....2
1. סינתזה של המעגל.....2
2. הוספת אמצעי בדיקתיות : Design For Testability (DFT).....2
3. LEC – Logical Equivalence Checking.....2
4. מימוש ה- Layout.....2
א. תכנון ה- Floorplan.....2
ב. הגדרת רשתות האספקה.....2
ג. מיקום התאים הסטנדרטיים ובנית עץ השעון (Clock Tree Synthesis CTS)..2
ד. חיווט: קווי האספקה והתכנון כולו.....2
ה. אנליזת תיזמון PostRoute והספק..... **Error! Bookmark not defined.**

ביצוע הניסוי

מטרת הסעיף הראשון היא ללמוד לסנתז את המעגל. במהלך הניסוי, נכיר לא רק את הכלים אלא גם קבצי הטכנולוגיה שכל כלי דורש על מנת שיוכל לעבוד.

1. סינתזה של המעגל

הערה : בסעיף זה נדגים כיצד שימוש של האופציות השונות מאפשרות ל- design vision לסנתז מימושים בעלי ביצועים משופרים.

סעיף זה יתבצע בספריה :

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

הפעל את כלי הסנתזה עם design_vision.

ראשית נוודא שהטכנולוגיה מוגדרת :

Application Variables@moon

target_library

Variable	Value
initial_target_library	
target_library	**tsl18fs120_typ.db**
upf_relax_target_library_subset_for_pm_cells	false

בשלב הראשון נריץ את script הסינתזה שהכנת בשם syn.tcl שהכנת בבית.

```
read_file -format sverilog {all RTL files}
current_design NeuralNet
change_selection [get_ports clk]
create_clock -name "clk" -period 5 -waveform { 0 2.5  }  { clk  }
set compile_seqmap_propagate_constants  false
set compile_seqmap_propa-gate_high_effort false
compile -exact_map
write -hierarchy -format verilog -output NeuralNet_syn.v
```

סינתזה עם compile ושעון של 5ns

- לחץ על File->Execute Script ובחר ב- syn.tcl.

הערה : פקודת ה- compile מבצעת את הסינתזה.

1.1 עבור על הפלט. האם נוצרו LATCHES? אילו יחידות זיכרון כן נוצרו ?

- לבדיקת התזמון, בצע : Timing->Report Timing Path.
- לבדיקת שטח בצע : Design->Report Area
- לבדיקת צריכת הספק, בצע : Design->Report Power

בצע את שלשת הבדיקות הנ"ל ורשום את התוצאות של התזמון, השטח ושל צריכת ההספק בטבלה שבהמשך.

סינתזה עם compile ושעון של 2ns

- הרץ את הפקודה :

```
create_clock -name "clk" -period 2 -waveform { 0 1 } { clk }
```

- בצע בחלון ה- DV : Design->Compile

1.2 בצע את שלשת הבדיקות הנ"ל ורשום את התוצאות של התזמון, השטח ושל צריכת ההספק.

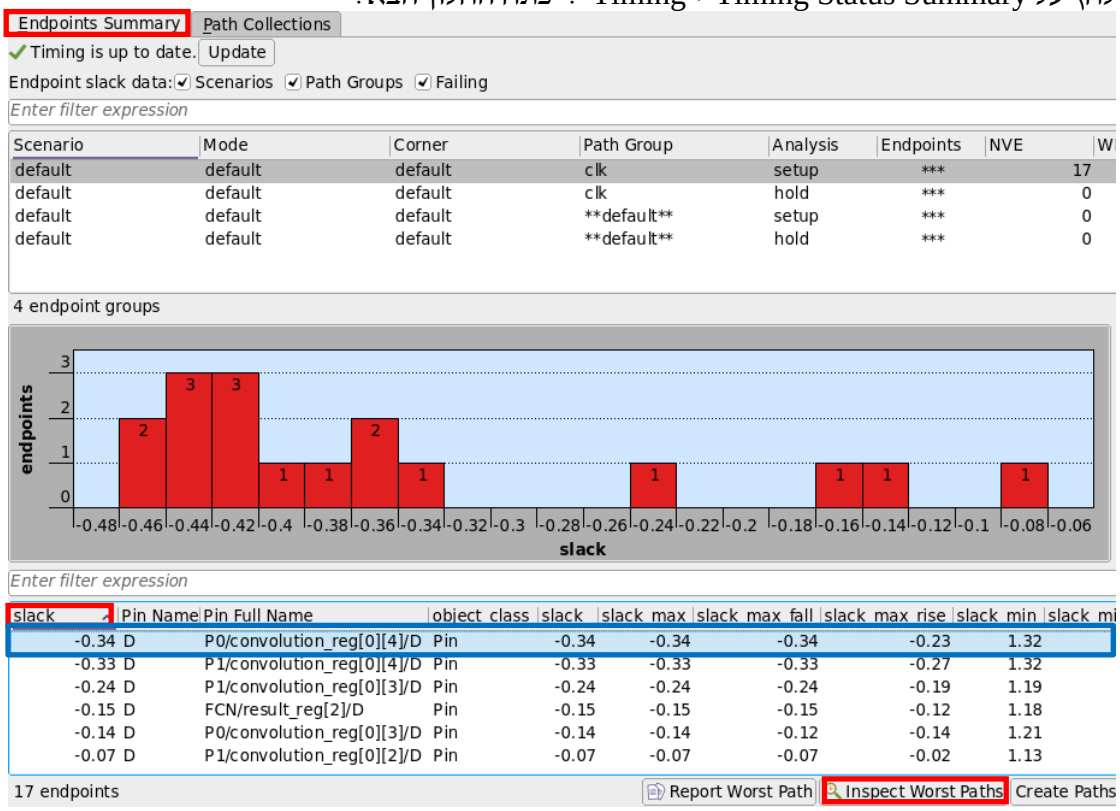
סינתזה עם compile_ultra ושעון של 2ns

- בצע בחלון ה- DV : Design->Compile Ultra

1.3 בצע את שלשת הבדיקות הנ"ל ורשום את התוצאות של התזמון, השטח ושל צריכת ההספק.

	Critical Path	Area	Power
Compile (5ns)			
Compile (2ns)			
Compile_ultra (2ns)			

- לחץ על Timing->Timing Status Summary . יפתח החלון הבא :



- יש לוודא שלשונית ה- Endpoints Summary לחוצה. בחר במסלול הראשון (מלבן כחול) ולחץ Inspect Worst Paths (מלבן ירוק).
- בחלון שנפתח, בחר ב- על Path Schematic. פעולה זאת מציגה הסכמה של המסלול הקריטי.

1.4 הוסף את הסכמה לדו"ח.

1.5 רשום בדו"ח את כל השערים שנמצאים על המסלול הקריטי. ניתן לראות אותם בחלון ה- Report Timing Path. מה ההשהיה של המסלול הקריטי?

- צא מהכלי.

1.6 הסבר במילים את התוצאות שקיבלת במדידות שבטבלה. מה הסיבות להבדלים שנרשמו בכל מצב בהספק, בשטח, ובמסלולים. מה התובנות מהניסוי?

חשוב : הסקריפט syn.tcl יצר את המעגל המסונתז לחלק ה- layout של הניסוי בקובץ בשם NeuralNet_syn.v.

2. הוספת אמצעי בדיקתיות : Design For Testability (DFT)

המטרה בסעיף זה היא להכיר את היכולת של ה- **design_compiler** לשפר באופן אוטומטי את הבדיקתיות של התכנון.

עבור לתיקיה **DFT**:

cd DFT

הפעל את ה- design_vision. נבצע כעת סינתזה של המעגל :

- בעזרת הפקודה File->Execute קרא את הקובץ syn.tcl ולחץ על Open בחלון שנפתח.

2.1 בצע את שלשת הבדיקות (תזמון, שטח והספק) כפי שהוסבר ורשום את התוצאות בטבלה שבהמשך.

2.2 בחר ביחידה **NeuralNet** בצד שמאל. לחץ על . כדי לראות את הסכמה לחץ שתי לחיצות מהירות על היחידת ה- **NeuralNet** שמופיעה בחלון. בצע שתי לחיצות מהירות על היחידת ה- **CN0**.

בצע zoom למרכז של הסכמה.

א. הוסף לדו"ח צילום מסך של הסכמה שמכיל כ- 4 רגיסטרים.

ב. האם קיים קשר בין היציאה של רגיסטר אחד לכניסה של רגיסטר אחר ? הסבר.

- חזור לרמה העליונה של הסכמה (לחיצה ימנית על העכבר ו- Back).

ג. מהן הכניסות והיציאות של NeuralNet?

בעזרת הפקודה File->Execute קרא את הקובץ dft.tcl ולחץ על Open בחלון שנפתח.

ב- **dft.tcl** רשומות שתי פקודות עם הגדרות עבור ה- **scan insertion**.

```
current_design NeuralNet
set_dft_signal -view existing_dft -type ScanClock -port clk -timing [list 45 55]
create_test_protocol -infer_asynch
```

- וגם הפקודה שמוסיפה את לוגיקת ה- DFT :

insert_dft

2.3 בצע את שלשת הבדיקות (תזמון, שטח והספק) כפי שהוסבר ורשום את התוצאות בטבלה.

	Critical Path	Area	Power
Before DFT			
After DFT			

2.4 רשום כיצד **insert_dft** השפיע על ערכים שבטבלה. הסבר מה הסיבות להבדלים לפני ואחרי.

2.5 בחר ביחידה **NeuralNet** בצד שמאל. לחץ על . כדי לראות את הסכמה לחץ שתי לחיצות מהירות על היחידת ה- NeuralNet שמופיעה בחלון. בצע שתי לחיצות מהירות על היחידת ה- **CN0**. בצע zoom למרכז של הסכמה

א. הוסף לדו"ח את החלק של הסכמה שמכיל כ- 4 רגיסטרים.

ב. האם קיים קשר בין היציאה של רגיסטר אחד לכניסה של רגיסטר אחר ? אם יש קשר ... מדוע יש עכשיו קשר?

ג. המן הכניסות והיציאות של NeuralNet?

ד. רשום את כל השינויים שהוכנסו למעגל כתוצאה של הפקודה **insert_dft**. התייחס גם לכניסות ויציאות החדשות.

- בצע את הפקודה :

write_scan_def -output scandef

- פקודה זאת יוצרת קובץ בשם **scandef** המכיל את כל הנתונים של ה- **scan chains**.

2.6 פתח את הקובץ ובחן את תוכנו ורשום בדו"ח את מספר השרשרות שנבנו, נקודת ההתחלה שלהם ונקודת הסיום. כמה **scan chains** נוספו למעגל ? מדוע ? איך אתה יודע ? הסבר את התפקיד של כל הכניסות והיציאות החדשות במעגל.

- אם ברצוננו לשנות את שיטת הפעולה של הפקודה **insert_dft** יש להשתמש בפקודה **set_scan_configuration**.

- ניתן לקבל הסבר מפורט על **configuration_scan_set** על ידי רישום :

- **man set_scan_configuration**

בחלון הפקודות.

2.7 בעזרת ההסבר רשום בדו"ח פקודה שיגרום ל- **dft_insert** להכניס 3 **chains_scan**

- אין צורך לבצע את הפקודה!
- סגור את ה- **dv** עם **File->Close**.

3. LEC – Logical Equivalence Checking

סעיף זה יתבצע בספריה :

/users/diglabN/student1_student2/DesignLib/LEC (N=1,2,3 or 4)

לאחר ביצוע סינתזה חובה לוודא שהמעגל המסונתז זהה מבחינה לוגית לתיאור ה- RTL. זאת המטרה של סעיף זה. קיימות סיבות שונות שיכולות לגרום לאי שוויון בין שני המימושים :

- בעיות הקוד
- פעולות שמבוצעות ע"י כלי הסינתזה כגון :
  - אופטימיזציה מתוחכמת
  - הוספת לוגיקה כמו DFT (ראה סעיף קודם) או Clock Gating במטרה לחסוך בהספק.

מפאת קוצר הזמן, בניסוי זה, אנו נתמקד רק באי התאמות שנובעות מהפעולות שמבוצעות ע"י כלי הסנתזה. בנוסף, אנו נספק את הקוד (שמממש מעגל טיפה שונה) שעליו נבצע את הבדיקה על מנת שכולם יעבדו עם קוד אחיד.

- חזור לחלון הפקודות של ה- LINUX
- עבור לתיקיית ה- LEC עם :

cd /users/diglabN/student1_student2/DesignLib/LEC

- ראשית נבצע שוב את הסינתזה. קובץ ה- script של הסינתזה מכיל :

```
read_file -format sverilog {./LEC/RTL/CNeuron.sv ./LEC/RTL/FCNeuron.sv
./LEC/RTL/NeuralNetCont.sv ./LEC/RTL/Pooling.sv ./LEC/RTL/Weights.sv
./LEC/RTL/NeuralNet.sv}
current_design NeuralNet
change_selection [get_ports clk]
create_clock -name "clk" -period 10 -waveform { 0 5 } { clk }
set compile_seqmap_propagate_constants false
set compile_seqmap_propa-gate_high_effort false
set_vsdc compile.vsdc
insert_clock_gating
```

compile_ultra
write -hierarchy -format verilog -output ./NeuralNet_syn_lec.v

הערות:

1. פקודת ה- set_vsdc compile.vsdc שומר מידע על סוג האופטימיזציה שבוצעה.
2. פקודת ה- insert_clock_gating מונע שינוי של השעון של flipflop כאשר שאין בו צורך.

- פתח את הכלי design_vision ובצע סינתזה של המעגל עם syn.tcl. הסינתזה יוצרת את המעגל מסונתז: NeuralNet_syn_lec.v.

הרצת LEC:

- קובץ ה- script בשם dofile של ה- LEC מבצע:
  - קריאת קבצי הטכנולוגית
  - קריאת ה- RTL והמעגל המסונתז
  - ביצוע השוואה בין שני המימושים.

- **<u>בחלון נפרד</u>** עבור לספרית ה- LEC:

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

- הפעל את הכלי עם: lec - ECOGXL
- בצע File->Do Dofile ובחר בקובץ בשם dofile. לחץ על Select.

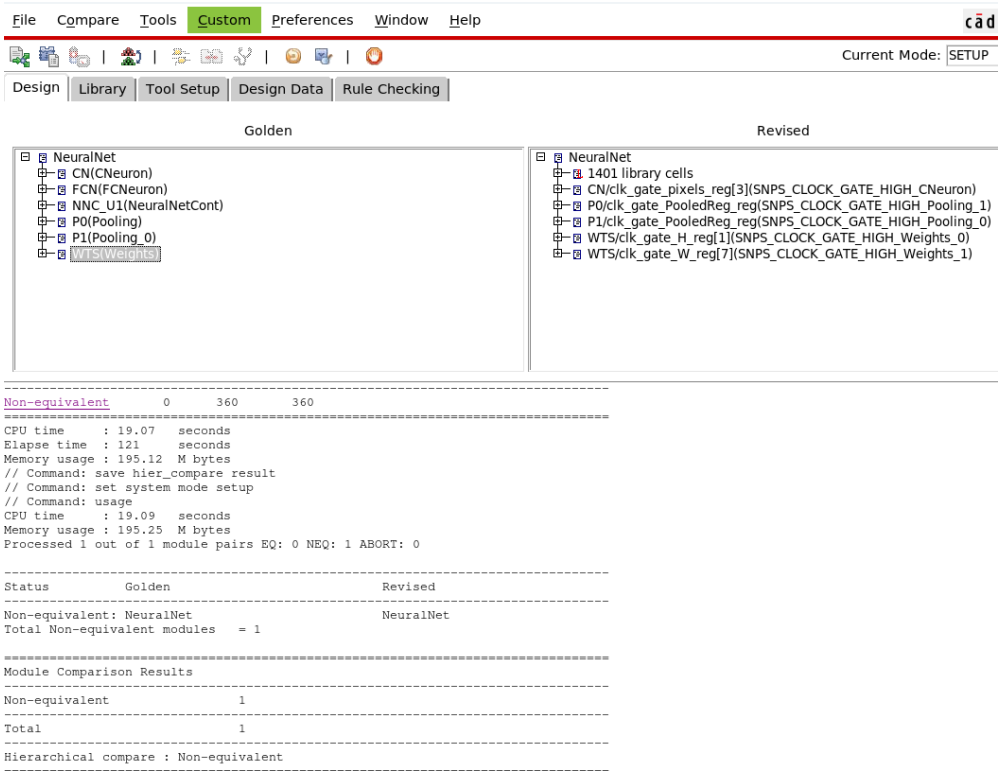
3.1: האם שני התכנונים שקולים ?

3.2: מדוע לפי דעתך קיימת אי התאמה כזאת?

- פתח את הקובץ dofile ולפני השורה: uniquify -all, הוסף את השורה:

read setup information compile.vsdc -type vsdc

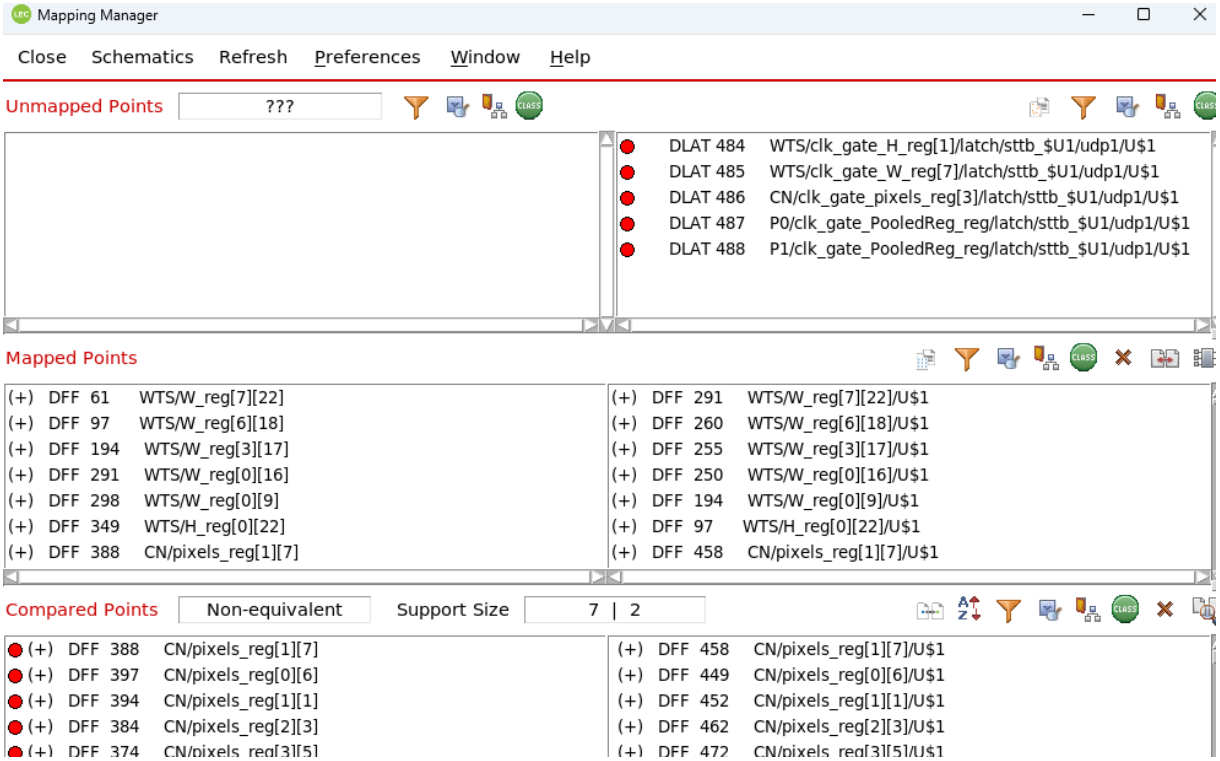
- כעת נבצע את ההשוואה מחדש. בחלון של ה- lec בצע File->Reset Design.
- בצע שוב File->Do Dofile ובחר בקובץ בשם dofile. לחץ על OK.
- כמו קודם, מתקבל החלון הבא:



- בחלון Conformal – Logical Equivalence Checking דפדף עד שמופיע:

- Non-equivalent

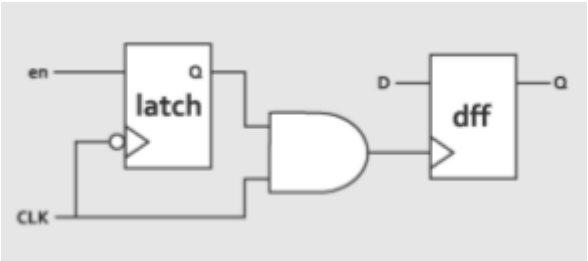
- ולחץ עליו. מופיע החלון הבא :



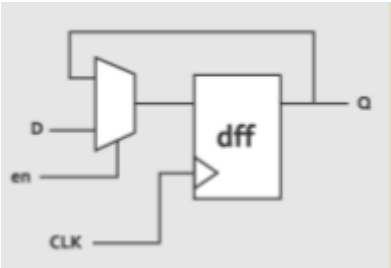
- ניתן לראות שקיימים LATCHES במעגל המסונתז שאינם קיימים בתכנון המקורי.

הסבר :

פקודת ה- insert_clock_gating בסינתזה מוסיפה latches כדי לנטרל את השעון עבור המחזורים שה- flipflop לא משנה את ערכו בהתאם לסכמה הבאה :



הפיכת המבנה הזה למבנה קומבינטורי טהור (ראה באיור הבא) יקל על תהליך ההשוואה.



ניתן לעשות זאת בעזרת הפקודה : set flatten model

פתח את ה- manual של LEC בעזרת הפקודה :

evince /eda/cadence/2022-23/RHELx86/CONFRML_22.10.200/doc/Conformal_User.pdf

עבור לעמוד 137 וחפש את בפסקה על Remodeling Latch-Based Clock Gating.

3.3 מה הפקודה שתאפשר מידול של המבנה עם ה- latch במבנה קומבינטורי טהור?
- הוסף את הפקודה הזאת לקובץ dofile אחרי שתי פקודות ה : read_design.
- כעת נבצע את ההשוואה מחדש. בחלון של ה- lec בצע File->Reset Design.
- בצע שוב File->Do Dofile ובחר בקובץ בשם dofile. לחץ על OK.

3.4 : האם שני התכנונים שקולים ?

סגור את ה- design_vision ואת ה- lec.

4. מימוש ה- Layout

סעיף זה יתבצע בספריה :

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

בחלק הזה של הניסוי אנו נבנה את ה- **layout** של התכנון בעזרת כלי אוטומטי של חברת **Cadence** בשם Innovus .

א. תכנון ה- Floorplan

כזכור, בחלק הראשון יצרנו קובץ מסונתז בשם **NeuralNet_syn.v**.

הקובץ המסונתז אינו מכיל pads הנחוצים לחיבור של השבב ל- pins של האריזה. לפני תחילת העבודה על ה- layout, יש להוסיף את ה- pads לקובץ. ניתן לעשות זאת באופן ידני או באמצעות סקריפט אשר יעדכן את הקובץ באופן אוטומטי. לשם בניית הקובץ יש להריץ את הסקריפט הבא :

לדוגמא :

./gentop.pl NeuralNet_syn.v NeuralNet

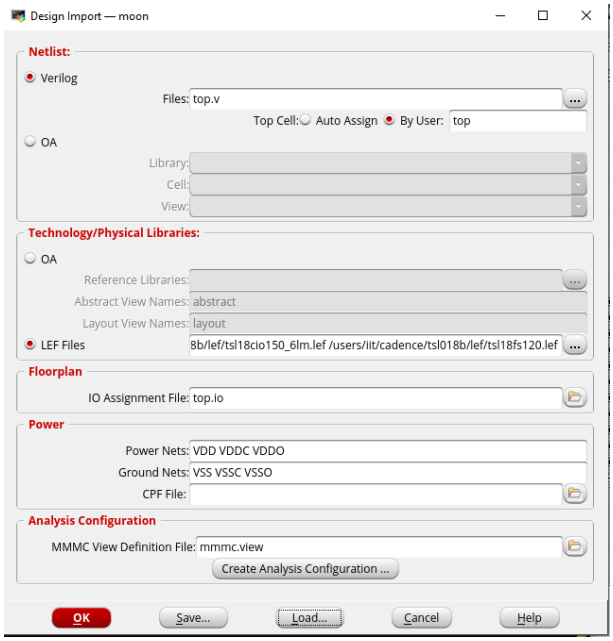
בסיום הריצה יתקבלו שני קבצים :

top.v
top.io

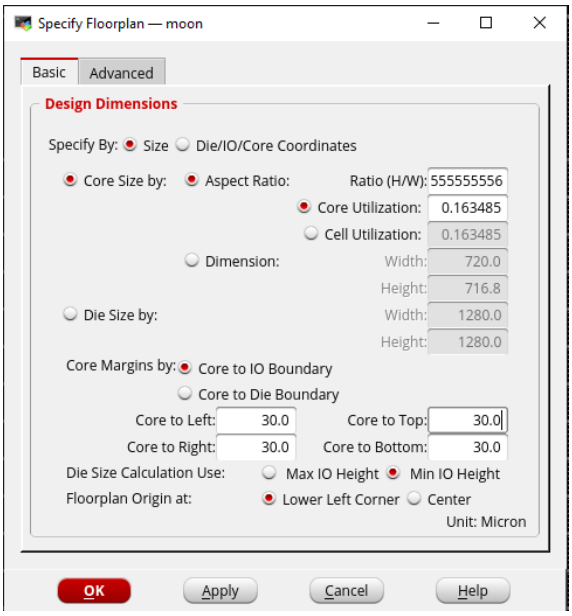
בקובץ ה-top.v נמצא התכנון המעודכן כולל ה-pads. בקובץ ה-top.io רשום מיקום הפינים.

בשלב ראשון, עלינו לקרוא את נתוני הטכנולוגיה והתכנון (קובץ ה- **Verilog** שהתקבל כפלט מהסינתזה. על מנת להקל על התהליך נבצע זאת באופן הבא :

- **קווי החיווט ב-** layout **מוסיפות קבלים ונגדים פרזיטיים ולכן יהיה בלתי אפשרי לעמוד במחזור שעון של** 5ns **כמו בסעיף הקודם.**
- **שנה את מחזור השעון ב-** NeuralNet.sdc **ל-** 6ns.
- הפעל את כלי ה- **layout** בעזרת הפקודה **innovus**.
- בצע **File->Import Design**. החלון הבא ייפתח :



- בשלב ראשון, כל השדות ריקים. לחץ על **Load** בחר בקובץ **NeuralNet.globals**.
- בדוק את שם קובץ ה- **Verilog** (בשדה **Files**) ואת שם המודול העליון.
- כאמור קובץ ה- **mmmc.view** קורא לקובץ **NeuralNet.sdc** (אילוצי התזמון)
- קבצי ה- **lef** מכילים את התיאור גיאומטרי (כולל מיקום מדויק של ה- **pins**) של התאים הסטנדרטיים.
- לחץ על **OK**. אם אין שגיאות הכלי יעלה את התכנון ואת כל קבצי הטכנולוגיה הדרושים.
- לחץ על כל אחת מהאפשריות מבאות (בצד ימין) : על מנת לראות ייצוגים שונים של המעגל.
- בצע **Floorplan->Specify**. השלם את הטופס באופן הבא, ולחץ על **OK** .



4.1 הסבר את המשמעות את השינויים שעשית בטופס.
4.2 הוסף את הסכמה לדו״ח.

ב. הגדרת רשתות האספקה

ראשית נגדיר אלו פינים יש לחבר לרשתות האספקה :

source glnets.src

בשלב זה נדרש להשתמש בחוטים רחבים עבור חוטים שמוליכים זרם גדול. לשם כך נבנה שתי טבעות (עבור **VDD** ו- **VSS**) של חוטים רחבים מסביב לבלוק שלנו. בנוסף נוסיף עוד כמה חוטים רחבים שחוצים את התכנון באופן אנכי. החיבור של כל התאים לרשתות האלו יתבצע באמצעות חוטים דקים.
את הטבעות ואת החוטים האנכיים ניתן לממש בעזרת שתי פקודות :

א. **Power->Power Planning->Add Ring**
ב. **Power->Power Planning->Add Stripes**

בעזרת **Power->Power Planning->Add Rings** נוסיף טבעות אספקה סביב לבלוק. בחלון שנפתח, הגדר איזה טבעות רצויות ובחר :
- ב- **Nets :** בחר **VDD** ו-**VSS** בלבד.
ב- **Ring Configuration** :
- הצלעות **top ,bottom** : **TOP_M**
- הצלעות **left ,right** : **M5**
- רוחב הפס : 6
- מרחק בין הפסים : 1.8
- ליד **offset** לחץ על – **Centre of Channel**.
בסוף לחץ על **OK**. בדוק שנוספו טבעות עבור **VDD** ו- **VSS**.

בעזרת **Power->Power Planning->Add Stripe** ניתן להוסיף רצועות נוספות של קווי האספקה. השלם את הטופס באופן שתואם את הטופס של הטבעות :
- בחר **Nets** בחר ב- **VDD** ו- **VSS** בלבד.
- הצלעות האנכיות : **M5**
- בדרך כלל נגדיר רק צלעות אנכיות או צלעות אופקיות
- רוחב הפס : 6
- מרחק בין הפסים : 1.8
- עבור **Set to set distance** רשום 100
- ב- **First/Last Strip** לחץ על **Absolute**. עבור **Start** רשום 330 ועבור **Stop** רשום 1000

4.3 הסבר את המשמעות של המספרים בסעיף הקודם.

לחץ על **OK**. בדוק שנוספו קוים אנכיים עבור **VDD** ו- **VSS**.

4.4 הוסף את הסכמה לדו״ח.

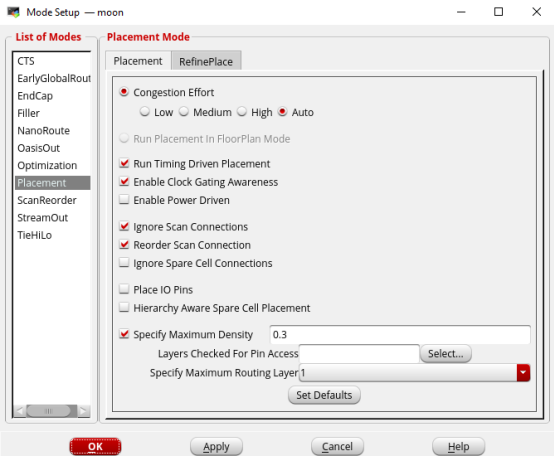
כלי ה- innovus יוצר קובץ ב- **directory** שבו הינכם עובדים בשם **innovus.cmdN** כאשר **N** הוא מספר. פתח את הקובץ עם ה **N** הגבוה ביותר. קובץ זה מכיל את רשימת כל הפקודות שבוצעו עד כה באמצעות הממשק הגרפי. עבור לסוף הקובץ.

4.5 העתק והוסף לדו״ח את הפקודה שהוסיפה את הטבעות של רשתות האספקה.
סגור את הקובץ. **שים לב שבעזרת קובץ זה ניתן לבנות** script files **בקלות רבה!**

ג. מיקום התאים הסטנדרטיים ובנית עץ השעון (Clock Tree Synthesis CTS)

שים לב שעדיין לא מיקמנו את התאים הסטנדרטיים. בשלב זה, נמקם את כל התאים הסטנדרטיים שמממשים את המעגל. על מנת למנוע מיקום של תאים מתחת לקווי **VDD** ו- **VSS** לחץ על **Place->Specify->Placement Blockage**. בחר ב- **M3** עד **M6**.

כעת לחץ על **Place ->Place Standard Cell**. בחלון שנפתח, לחץ על **Mode**. נרצה להגדיר את הצפיפות של התאים. נבקש צפיפות שלא עולה על **30%** על מנת להשאיר מספיק מקום לחיווט. בשדה של **Specify Maximum Density** רשום 0.3 .



- לחץ על **OK** ושוב על **OK**.

- לחץ על בצד ימין למעלה.

4.6 בצע זום ובחר בחוט כלשהו. לחץ על הכפתור הימני ובחר ב- **Attribute Editor**. מה ה- **wire status**? מה המשמעות של **status** זה?
<u>עליך להסביר למנחה את התשובה שלך!</u>

מכאן ניתן להסיק שהחיווט שמוצג אינו חיווט אמיתי/סופי אלה רק חיווט אפשרי. הכלי INNOVUS מאפשר לנו לבחור איזה שכבות נרצה להציג על המסך כדי שנוכל להתרכז רק בשכבות מסוימות. בצד ימין למטה ישנה רשימה של כל השכבות ומשבצת בחירה. נכבה את הצגת החוטים לחיצה על 1,2,3,4,5,6. בצע **zoom** והכנס לדו"ח תמונה של חלק קטן של ה- **layout**. שים לב שהיחס בין השטח הריק והמלא בתאים הוא בערך 1:1 בהתאם למה שביקשנו בשלב המיקום. החזר את כל חוטים שוב ע"י לחיצה על 1,2,3,4,5,6.

4.7 הוסף את הסכמה לדו"ח ללא המתכות.

4.8 הסבר את המושגים:
WNS (Worst Negative Slack) ו- **TNS (Total Negative Slack)**. רשום בדו"ח מדוע חשוב לדעת את שני מספרים? **<u>קרא למנחה על מנת להסביר לו את התשובה!</u>**

בצע **Timing->Report Timing (Pre-CTS)**. רשום את ה- **Setup WNS** ו- **Setup TNS** בדו"ח.
בצע את הבדיקה פעם עבור **Setup** ופעם נוספת עבור **Hold**. רשום את ה- **Hold WNS** ו- **Hold TNS** בדו"ח.

בצע:

optDesign -preCTS

יש לרשום את הפקודה בחלון ה- **console** כלומר החלון שממנו הופעל **Innovus**.
<u>הערה:</u> חשוב ביותר להפעיל את פקודות האופטימיזציה ובדיקות התזמון עם האופציות הנכונות אחרת תתקבלנה תוצאות לא נכונות!
פקודה זאת מבצעת אופטימיזציה של התכנון. שים לב שזה לוקח קצת זמן ☹.

בצע בדיקות ה- **Timing** ל- **Setup** ו- **Hold** ו רשום את התוצאות בטבלה? האם יש שיפור?

מכאן החשיבות להריץ את פקודת אופטימיזציה !

Clk = 12n	Setup WNS	Setup TNS	Hold WNS	Hold TNS
Pre-CTS before OptDesign				
Pre-CTS after OptDesign				
Post-CTS before OptDesign				
Post-CTS after OptDesign				

הערה : כעת יש נתונים רק לשתי שורות. את יתר השורות נמלא בהמשך.

<u>הגדרת עץ השעון</u>

תפקיד עץ השעון הוא להבטיח שהשעון מגיע לכל הפליפ' פלופים באותו זמן פחות או יותר.
עליך להריץ את הפקודות הבאות ב- **terminal** שבו הפעלת את **innovus**.
ראשית נגדיר את סיגנל השעון :

create_ccopt_clock_tree -name top -source I39/CIN

כעת נגדיר אלו תאים יכולים להשתתף בבניית עץ השעון :

set_ccopt_mode -cts_inverter_cells {invbd2 invbd4 invbd7 invbda invbdf invbdk}
set_ccopt_mode -cts_buffer_cells {bufbd1 bufbd2 bufbd3 bufbd4 bufbd7}

בשלב זה מגדירים את זמני עליה וירידה מכסימליים :

set_ccopt_property target_max_trans 250ps

וגם את ה- **skew** המכסימלי

set_ccopt_property target_skew 0.3

כעת בנה את עץ השעון עם :

source IITccopt.src

- פתח את הקובץ **top.v** .

4.9 מדוע מקור השעון מוגדר ב – **I39**?

- בדוק את מבנה השעון עם **Clock->CCOpt Clock Tree Debugger**

4.10 מה ההשהיה, כלומר הזמן בין שורש השעון ל- **FF** הראשון (בערך) ? מה ה- **skew** כלומר הפרש הזמן המכסימלי בין זמן ההגעת השעון ל- **FF** אחד כלשהו לאחרים (בערך)?

4.11 הוסף את הציור לדו"ח.

מעכשיו על הכלי לבצע ניתוח הזמנים עם עץ השעון ולא עם שעון אידיאלי. נגדיר זאת בעזרת הפקודות הבאת :

set_interactive_constraint_modes [all_constraint_modes -active]
set_propagated_clock I39/CIN

- בצע **Timing->Report Timing**. לחץ על האופציה **Post-CTS** בטופס. בצע את הבדיקה גם עבור **Setup** וגם עבור **Hold**.

4.12 רשום את התוצאות בטבלה. האם יש שיפור (ביחס ל- **Pre-CTS after OptDesign**)?

- בצע

optDesign -postCTS

4.13 בצע **Timing->Report Timing (Post-CTS)**. רשום את התוצאות בטבלה. האם יש שיפור?

ד. חיווט: קווי האספקה והתכנון כולו

- כעת נבצע חיווט של רשתות האספקה. בצע **Route->Special Route**. בשדה **nets** רשום **VDD VSS**. חבה את כפתור ה- **Pad Rings**. לחץ על **OK**. שים לב איך כל התאים מתחברים לרשתות האספקה.

בצע **Zoom-In** למעגל.

4.14 הסבר במילים שלך איזו פעולה מבצעת הפקודה **Route->Special Route**. כלומר, מה המשמעות של " חיווט של רשתות האספקה" ?

4.15 הוסף את ה- **layout** לדו"ח.

- שים לב שהחיווט ניסיוני נעלם.
- על מנת להשלים את כל החיבורים בצע **Route->Nanoroute->Route**

4.16 הסבר במילים שלך איזו פעולה מבצעת הפקודה **Route->Nanoroute->Route**.

4.17 בחר חוט כלשהו ובדוק עבורו ה- **Attribute Editor**. מה הסטטוס שלו ? מדוע הפעם ה- **status** שונה ?

4.18 הוסף את ה- **layout** לדו"ח.

סיום הניסוי !