

VLSI-ל-המעבדה

Introduction to OCEAN

יוסי יולס

הקדמה על השפה

ocean היא שפת סקריפטים אשר בעזרתה ניתן להריץ סימולציות spectre. ניתן לבצע מספר סימולציות זו אחר השנייה ולצייר מספר גרפים בו זמנית בכל הרצה. את פקודות השפה מריצים בחלון הפקודות של UNIX, ניתן להריץ פקודות ocean בצורה אינטראקטיבית או באמצעות קבצי סקריפט המכילים את הפקודות.

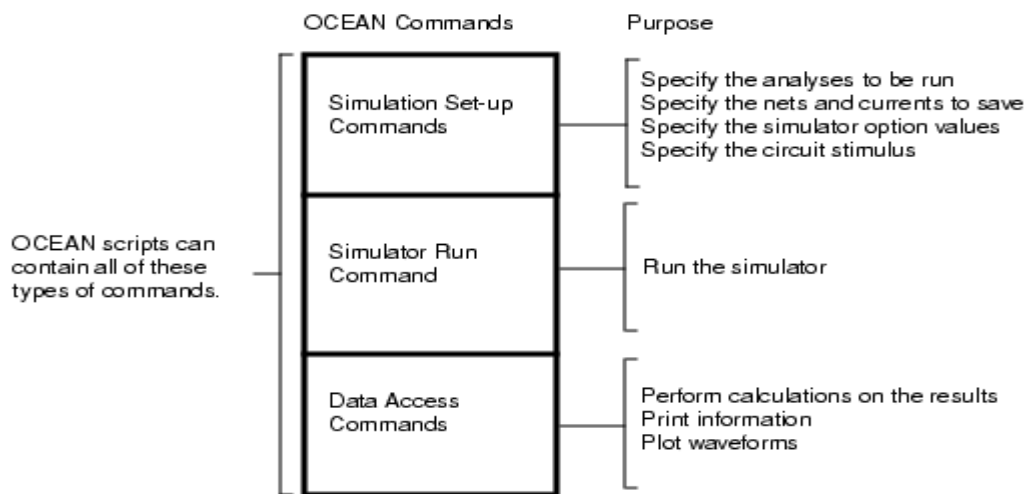
את המעגלים החשמליים מתכננים באמצעות cadence virtuoso ואחר כך מבצעים סימולציות באמצעות ocean.

באמצעות ocean ניתן:

- להריץ סימולציות ארוכות יותר כגון Parametric analysis ו-statistical analysis בצורה יעילה יותר.
- הרצת סימולציות ארוכות ללא צורך בהפעלת הממשק הגרפי של Virtuoso Analog Design Environment
- הרצת סימולציות מעמדות רחוקות (terminals) ללא צורך בממשק גרפי.

סוגי הפקודות ב-Ocean

ב-ocean ישנם שלושה סוגי פקודות:



השפה בה מתכנתים ב-Ocean היא SKILL של Virtuoso.

צורת כתיבת פקודות SKILL ב-Ocean

```
path("~/simulation1/schematic/psf" "~/simulation2/schematic/psf")
```

זוהי דוגמא לפקודת path, הפרמטרים של הפקודה מופיעים בסוגריים מיד אחרי הפקודה, הגרשיים משמשים לערכי string. כאשר נראה פקודה עם תחילת t_ נדע שזהו ארגומנט מסוג string, g_ ייצג ערך מספרי (לפי SKILL convention) לדוגמא:

```
desVar( t_desVar1 g_value1 t_desVar2 g_value2)
```

הפעלה ועבודה

יש לוודא כי `setenv DISPLAY` מוגדר כראוי. יש להקיש את הפקודה `ocean` (אין להריץ עם `&` מאחר וזוהי תוכנת scripting הרצה עם `&` עלולה לגרום לשגיאות), על המסך תופיע שורת הפקודה `ocean>` פקודה לדוגמא, הקש :

```
strcat("rain" "bow")
```

פקודה זו מאחדת שני `strings`, בפלט נראה את המילה `rainbow`. שימו לב- הקלדת הפקודה השגויה הבאה :

```
strcat("rain" "bow)
```

(ללא גרשיים לאחר ה-`bow`) תתקע את המערכת, ניתן לנסות להתאושש משגיאה זו ע"י הקלדה של `]` (גרשיים וסוגר מרובע אחריהם). אם המערכת לא מגיבה לפקודות לאחר ניסיון ההתאוששות יש להפעילה מחדש (`ctrl+D` יגרום לסגירתה) .

תפריט העזרה

הקשת `ocnHelp` תפתח את תפריט העזרה, הקשת `ocnHelp('ocean_command')` (גרש ושם הפקודה) תפתח את העזרה עבור הפקודה הספציפית. ניתן גם להפעיל את העזרה משורת הפקודות של חלון ה-`CIW` (החלון הראשי של התוכנה) של `icfb/virtuoso` באמצעות הפקודה `ocnHelp`.

שימוש בהיסטוריית הפקודות

הקשת `history` תציג את 20 הפקודות האחרונות, לכל פקודה מספר. חזרה על פקודה תעשה באמצעות הקלדת סימן קריאה ואחריה מספר הפקודה לדוגמא :

```
ocean > !2
```

תבצע את פקודה מספר 2 ברשימת ה-`history`. הקלדת `!des` למשל תריץ את הפקודה האחרונה המתחילה ב-`des`.

Symbols

גרש יחיד מסמל `symbol` והוא דומה ל-`enums` `constant` בשפת C. דוגמא :

```
analysis('tran ...)
```

ה-`symbol` `tran` מוגדר מראש עבור פקודה זו. ניתן לראות מהם ה-`symbols` המתאימים לפקודה על ידי בדיקה ב-`ocnHelp('analysis')`. לפי הקונבציה של `SKILL` כאשר נדרש פרמטר לפקודה עם תחילת `s_` הכוונה היא ל-`symbol`.

ארגומנטים (אופציונאליים) מסוג Keyword

לארגומנט אופציונאלי מסוג זה יש שני מרכיבים, הראשון הוא ה-`keyword` עצמו עם סימן שאלה לפניו. השני הוא הערך שמקבל אותו `keyword` ומגיע ישר אחריו. לדוגמא הפקודה הבאה :

```
analysis( 'tran ?start 0 ?stop 1u ?step 1n )
```

פקודה זו תריץ סימולציית `transient` שתתחיל בזמן 0 ותסתיים בזמן `1u` עם קפיצות של `1n`. לפי הקונבציה, פרמטרים מסוג זה יופיעו בתיעוד (עזרה) עם סוגריים מרובעים.

Data type used in OCEAN

בטבלה הבאה ניתן לראות השמות הפנימיים והתחיליות של סוגי ה-data types שמשמשים בהם ב-OCEAN:

Data Type	Internal Name	Prefix
floating-point number	flonum	f
any data type	general	g
linked list	list	l
integer, floating-point number, or complex number		n
user-defined type		o
I/O port	port	p
symbol	symbol	s
symbol or character string		S
character string (text)	string	t
window type		w
integer number	fixnum	x

Return Values

לרוב פקודות OCEAN יש ערכים חוזרים אשר ניתן להשתמש בהם כערכים לפקודות אחרות, בטבלה הבאה ניתן לראות מספר דוגמאות:

Assigning a Return Value to a Variable	Resulting Value for the Variable
<code>a=desVar("r1" 1k)</code>	<code>a=1k</code>
<code>a=desVar("r1" 1k "r2" 2k)</code>	<code>a=2k</code>
<code>a=desVar("r1")</code>	<code>a=1k</code> , assuming r1 was set in a desVar command
<code>a=desVar("r2")</code>	<code>a=2k</code> , assuming r2 was set in a desVar command

שימוש במשתנים

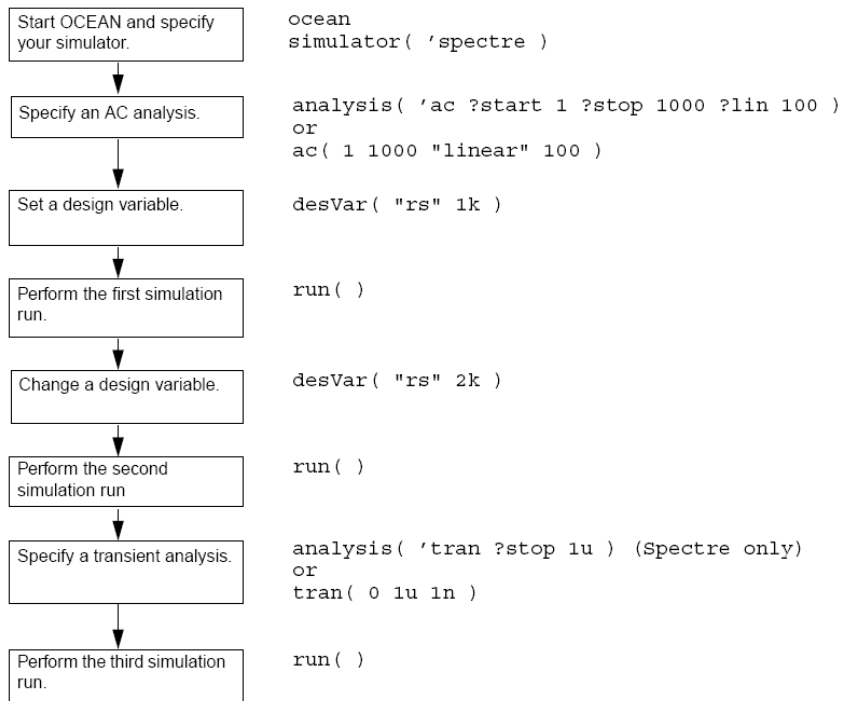
משתנים ב-OCEAN מתפקדים בצורה זהה למשתנים בסביבת Virtuoso. הגדרת משתנים אינה נעשית לפי הסדר בהם הם מוצהרים לדוגמא ההצהרה:

```
desVar("a" "b+1")
desVar("b" 1)
```

לא תגרום לשגיאה מאחר והמערכת תסדר אותם בצורה הבאה:

```
desVar("b" 1)
desVar("a" "b+1")
```

דוגמא ל- flow ריצה ע"י ocean :



הערה חשובה : ב-OCEAN זו טעות לשים רווח בין שם הפונקציה ובין הסוגר השמאלי.
 לדוגמא : הביטוי הבא הוא נכון : `ac(1 10000 2)`
 ואילו הביטוי הבא יגרום לשגיאת syntax : `ac (1 10000 2)`

ב. במידה והגדתם משתנים ב-schematic יש להגדיר אותם באופן ידני בקובץ ה-netlist, יש לפתוח את קובץ ה-netlist לעריכה ולהוסיף שורת פרמטרים לדוגמא:
parameters vdd=1.8 tt=100p cyc=10n

פקודת ה-modelFile מגדירה את קבצי המודלים, במידה וקובץ מודלים מכיל מספר מחלקות יש לפרט באיזה מחלקה אנו מעוניינים להשתמש, ניתן לראות כי עבור קובץ המודלים tsl018.scs נבחרה המחלקה tt_18 למשל. את פקודת ה-option יש לרשום במידה והינכם עובדים עם מודלים של tower18_6M.
הפקודה resultsDir מגדירה את מיקום ספריית הפלטים, אם הספרייה לא קיימת צרו אחת.

הרצת קבצי ocean

כדי להריץ את הקובץ, ב-ocean נקיש את הפקודה:

```
ocean> load("file_name.ocn")
```

טעינת קובץ זה מגדירה את הסביבה בה יעבוד ocean. בשלב הבא כבר נריך סימולציה ונצפה בתוצאותיה. ניתן גם להריץ את הפקודה מתוך חלון ה-CIW של virtuoso/icfb.

שמירת תוצאות :

באופן default הכלי לא ישמור נתונים על זרמים במעגל, אם מעוניינים לצפות בנתונים נוספים כגון זרם יש לפרט זאת באמצעות הפקודה save לפני הרצת הסימולציה:

```
save(s_saveType [t_saveName1] ... [t_saveNameN] )
```

שומר את מה שמוגדר ב-s_savetype עבור הרשתות שנתונות ב t_saveNameN

הערכים האפשריים לפרמטר s_savetype הם :

'v – שומר את מתח הרשת

'i – שומר את זרם הרשת

'all – שומר את כל הזרמים וכל המתחים עבור כל הרשתות

'allv – שומר את כל המתחים בכל הרשתות

'alli – שומר את כל הזרמים בכל הרשתות

דוגמא לשימוש :

שומר את הכול :

```
save( 'all )
```

שומר את הזרם של הרשתות R1 ו Q1/b :

```
save( 'i "R1" "/Q1/b" )
```

הרצת סימולציית DC וצפייה בתוצאות

לאחר טעינת הסביבה נריך סימולציה פשוטה מסוג DC, הקלד :

```
ocean> analysis('dc ?saveOppoint t ?oppoint "rawfile")
```

הארגומנט saveOppoint מציין לסימולטור לחשב נקודת עבודה dc, הארגומנט השני מציין למערכת לשמור את התוצאה.

כדי להריץ את הסימולציה הקלד :

```
ocean> run
```

בסיום הריצה ישמרו התוצאות לספריית ה-results כפי שהוגדר בקובץ הסקריפט.

הערות :

- גם את הסימולציות ניתן לתכנן מראש בקובץ הסקריפט כפי שנראה בדוגמא מורכבת בהמשך.

- ה-netlist שנבחר עבור סימולציה זו הוא קובץ top המכיל את ה-inverter symbol המחובר למקור מתח מתאים עם קבל ביציאה כפי שמודגם במסמך הדרכה cadence .virtuoso

צפייה בתוצאות הסימולציה

ניתן להריץ כמה סוגי סימולציות ולבסוף לבחור את הפלט שאנו רוצים לראות, את הפלטים הקיימים נוכל לראות באמצעות הפקודה results הקשת הפקודה תציג את כל הפלטים האפשריים עבור הסימולציות שביצענו, לדוגמא לאחר הרצת סימולציית DC ו-transient הקלד :

```
ocean> results
(dcOpInfo tranOp subckts instance dcOP
  Model variables output designParamVals tran
  Primitives
)
```

בכדי לבחור את הפלטים עבור מתחים בסימולציית ה-DC נקליד (dcOp).selectResults
כעת נוכל לצפות למשל במתחים על חוטי הכניסה והמוצא, נקליד :

```
ocean> v("/Vin")
0.0
ocean> v("/Vout")
1.8
```

נוכל גם לצפות באינפורמציה על ההתקנים במעגל על ידי הפקודה (dcOpInfo).selectResults
את ההתקנים עבורם ניתן לצפות באינפורמציה נמצא לפי הקלדת הפקודה :

```
ocean> outputs
("/CO" "/I6/M0" "/I6/M1" "/V0" "/V1")
```

נבחר :

```
ocean> report(?name "I6/M0")
```

למשל תציג את האינפורמציה עבור התקן NMOS M0 בנקודת העבודה. נסו גם את הפקודה :
report(?param "vgs")

סימולציית transient

הפקודה הבאה תריץ סימולציית transient :

```
ocean> analysis('tran ?start 0 ?stop 700n ?step 10n)
```

משמעות הארגומנטים היא טריוויאלית. יש להקליד run לביצוע הסימולציה.
כעת נבחר להציג את הפלטים עבור סימולציה זו :

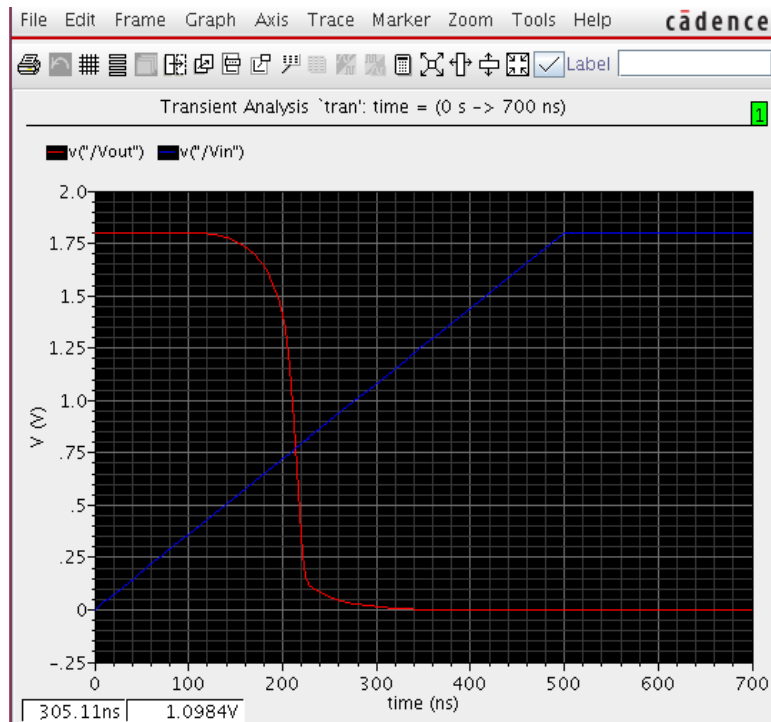
```
ocean> selectResults('tran)
```

הצגת מתח הכניסה והיציאה כתלות בזמן :

```
ocean> plot(v("/Vin") v("/Vout"))
```

ניתן להציג את הפלטים גם בצורה מספרית על ידי הפקודה :

```
ocean> ocnPrint(v("/Vin") v("/Vout"))
```



Vin & Vout [v] vs Time[ns]

Advanced OCEAN Scripting

עבודה עם קבצים

עבור כל הפלטים שעבדנו איתם עד עכשיו השתמשנו ב-console. בעבודה עם סקריפטים מורכבים זה לא ממש יעבוד מאחר והמידע הרלבנטי ילך לאיבוד בין כל שאר המידע שיוצף ב-console. הפתרון לכך הוא ליצור קובץ לפלט בו ישמר המידע הרלבנטי שנגדיר באמצעות הפקודה `fprintf` בכדי ליצור או לכתוב על קובץ ב-ocean script נשתמש בפקודה `outfile`:

```
of=outfile("./output_file")
```

בכדי להוסיף בסוף קובץ קיים:

```
of=outfile("./output_file" "a")
```

הדפסה לקובץ תעשה באמצעות הפקודות לדוגמא:

```
fprintf of "Here's some sample output."
```

```
fprintf of "Voltage at Vout: %f\n" v("/Vout")
```

בסוף הריצה יש לסגור את הקובץ באמצעות:

```
close(of)
```

פרוצדורות

עבור כל סקריפט ניתן להגדיר מספר פרוצדורות, לדוגמא:

```
(procedure run_test1( @optional (start_time stop_time))
```

```
  fprintf(of "Start time is: %d\n" start_time);
```

```
  fprintf(of "Stop time is: %d\n" stop_time);
```

```
)
```

```
(procedure run_test2( @optional (...))
```

```
  ...
```

```
  ...
```

```
)
```


לאחר ביצוע load וטעינת הסביבה הקשת run_test1(0 10) תריץ את הפרוצדורה run_test1 עם הפרמטרים 0 ו-10. @optional מגדיר כי הפרמטרים המסופקים לפרוצדורה הם אופציונאליים למרות שבדוגמא הרצת run_test1 ללא פרמטרים תגרום לשגיאה.

תנאי IF

מבנה הפקודה :

```
if( (argument1 <=> argument2)
then
    ....
else
    ....
)
```

לולאות foreach

בדוגמא הבאה לולאת foreach שעבור כל ריצה שלה מוגדר קובץ מודל אחר בהתאם לרשימה ModelList. ניתן להריץ לולאות מסוג זה גם על משתנים של המעגל וגם על פרמטרים של התוכנה כמו בדוגמא זו :

```
ModelList=list("path/Model1.scs" "path/Model2.scs" "path/Model3.scs")
foreach( model ModelList
    modelFile(model)
    ....
    ....
    analysis('tran ....)
    run()
    fprintf(....)
)
```

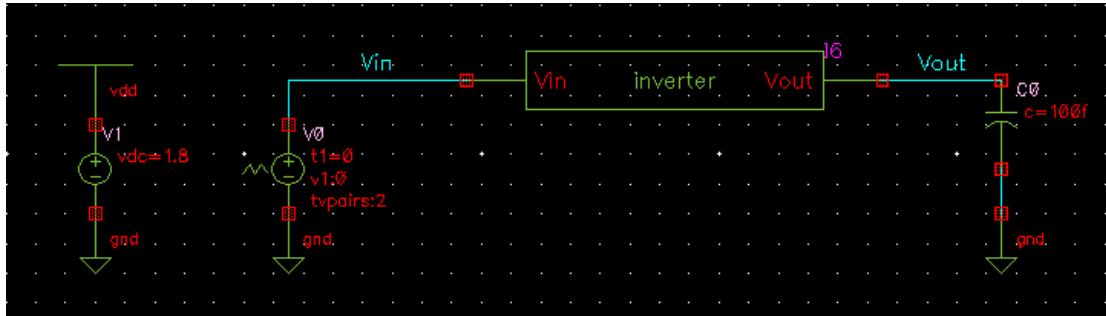
paramAnalysis

הצורה הנוחה להריץ סימולציה פרמטרית היא באמצעות הפקודה paramAnalysis. באמצעות פקודה זו ניתן להריץ מספר סימולציות פרמטריות מכוונות על משתנים במעגל. פקודת paramRun() תריץ את הסימולציה. בסיום הריצה ניתן יהיה להציג את התוצאות כאוסף של עקומות. דוגמא לסימולציה מכוונות :

```
paramAnalysis( "r1" ?start 200 ?stop 600 ?step 200
    paramAnalysis( 'rs ?start 300 ?stop 700 ?step 200
    )
)
paramRun()
```

הפקודה תרוץ בצורת לולאה בדומה לשפת C.

סקריפט לדוגמא עבור מעגל inverter



Top schematic for Inverter

הסקריפט הבא מגדיר את סביבת העבודה עבור מעגל ה-Top, מריץ סימולציות DC ו-transient ומדפיס תוצאות לקובץ וגרף. הסקריפט מאגד את רוב הפקודות שהוצגו במסמך זה. את קוד הסקריפט ניתן לראות בקובץ:

`/hm/iit/cadence/ocean/top_batch.ocn`

ניתן להעתיק את הקובץ לספרייה מקומית. יש לערוך את הקובץ ולשנות את מיקום הספרייה בחלק של:

`;Output file creation and constant definitions`

ולהריץ מ-ocean:

`ocean> load("top_batch.ocn")`

ולהריץ על ידי:

`ocean> run_all`

קובץ פלט לדוגמא עבור סקריפט זה ניתן למצוא -

`/hm/iit/cadence/ocean/top.out`

דוגמאות נוספות:

1. קבצי סקריפט נוספים ניתן למצוא ב-`/hm/iit/cadence/ocean`

2. - OCEAN reference

http://www.d.umn.edu/~htang/Cadence_doc/oceanref.pdf