Description: technion-logoטכניון – מכון טכנולוגי לישראל

**תכן לוגי  
תרגיל רטוב 1#**

הגשה עד : 07.05.18 בשעה 23:55.

המתרגל האחראי על התרגיל: משה ליכטנשטיין.

שאלותיכם במייל (smosesli@campus **ולא** @cs), (כולל עניינים מנהלתיים) יופנו רק אליו.

**כתבו בתיבת subject: רטוב תכן לוגי.**

שאלות בעל-פה ייענו על ידי כל מתרגל.

הוראות הגשה:

* את החלק הרטוב מגישים דרך אתר הקורס, את החלק היבש (שכולל גם דף שער, ציור מכונת המצבים ואת הקוד מודפס) מגישים בתא הקורס.
* ההגשה בזוגות.
* **לחלק היבש יש לצרף דף שער הכולל ברקודים של ת"ז המגישים.**
* תמונת ברקוד אפשר לייצר לדוגמה בקישור :  
  <https://barcode.tec-it.com/en/Code128>

**חלק יבש**

**שאלה 1**

הסבירו את ההבדל בין X ו Z

מלאו את טבלאות האמת הבאות

שער and

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| z | x | 1 | 0 |  |
|  |  |  |  | 0 |
|  |  |  |  | 1 |
|  |  |  |  | x |
|  |  |  |  | z |

שער or

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| z | x | 1 | 0 |  |
|  |  |  |  | 0 |
|  |  |  |  | 1 |
|  |  |  |  | x |
|  |  |  |  | z |

**שאלה 2**

נניח כי a = reg[3:0], אילו ערכי a יחשבו כערך אמת (if (a)) ?

הסבירו את ההבדל בין האופרטור == לבין האופרטור ===

הסבירו את ההבדל בין האופרטור reduction OR לבין האופרטור bit-wise OR

הסבירו את ההבדל בין האופרטור הבינארי & לבין האופרטור הבינארי &&

**שאלה 3**

בכל אחת מהשורות הבאות מלאו בהתאמה reg / wire / both

|  |  |
| --- | --- |
|  | left-hand side of an assign statement |
|  | right-hand side of an assign statement |
|  | left-hand side of an = or <= sign in an always@ block |
|  | right-hand side of an = or <= sign in an always@ block |
|  | can be connected to the input port of a module instantiation |
|  | can be connected to the output port of a module instantiation |
|  | can be used as outputs within an actual module declaration |
|  | can be used as inputs within an actual module declaration |

**שאלה 4**

הסבירו את ההבדל ביןBlocking לבין Nonblocking assignmentותנו דוגמה בה שימוש בכל אחת יביא לתוצאה שונה

Blocking assignment – פעולת השמה חוסמת , כלומר, מבצעת פעולת השמה לתוך האופרטור ברגע שהפעולה הקודמת הסתיימה. לדוגמא: במידה וישנם כמה פקודות ברצף של השמה, קודם הפקודה הראשונה מתבצעת ואז השניה וכו'...

Test\_1 =1'b1

Test\_2=Test\_1

בדוגמא זו, לtest\_2 יכנס הערך 1.

Nonblocking assignment – פעולת השמה לתוך אופרטור, מבצעת השמה לתוך האופרטור באותו רגע ולא מחכה לסיום שאר הפעולות.

בהמשך לדוגמא הקודמת –

Test\_1<=1'b1

Test\_2<=Test\_1

יכנס לtest\_1 הערך 1 ולtest\_2 הערך הישן של test\_1.

**חלק רטוב**

**בתרגיל זה נממש את גרסת מור של מכונת המצבים מתרגול #1 (שקפים 28-32)**

* המערכת מקבלת מספר טבעי X ומשדרת מספר טבעי Y כך ש  
* הקלט והפלט בינאריים, בכל מחזור שעון נקלט ונפלט ביט אחד, החל מהMSB.
* הLSB של Y משודר מחזור שעון אחד לאחר המחזור בו נקלט הLSB של X (מכונת מור).
* למערכת ישנה כניסת reset, כאשר reset==1 מכונת המצבים צריכה לחזור למצבה ההתחלתי.

**הוראות נוספות:**

* אין לשנות את הכניסות או היציאות למודול.
* אין לשנות את שם המודול או שם הקובץ.
* בהגשה האלקטרונית יש להגיש רק את הקובץ my\_fsm.
* **יש לצרף לחלק היבש ציור של מכונת המצבים וכן את הקוד שכתבתם מודפס.**
* מדריך להורדתModelSim נמצא באתר הקורס של סמסטר אביב 17.
* עבור העבודה עם ModelSim מומלץ לראות את הוידאו בקישור   
  <https://www.youtube.com/watch?v=Z8whdGa7RtY>
* שימו לב שמספיק להוריד רק את ModelSim, לא צריך את Quartus או device support.
* בכתיבת הקוד ניתן ומומלץ להיעזר בtemplate הבא :  
  <https://www.altera.com/support/support-resources/design-examples/design-software/verilog/ver_statem.html>
* לשם בדיקת המודול שכתבתם קיבלתם את הקובץ test\_fsm.v עם טסט בסיסי בלבד, מומלץ לכתוב טסטים מלאים שיוודאו נכונות עבור כל קלט.