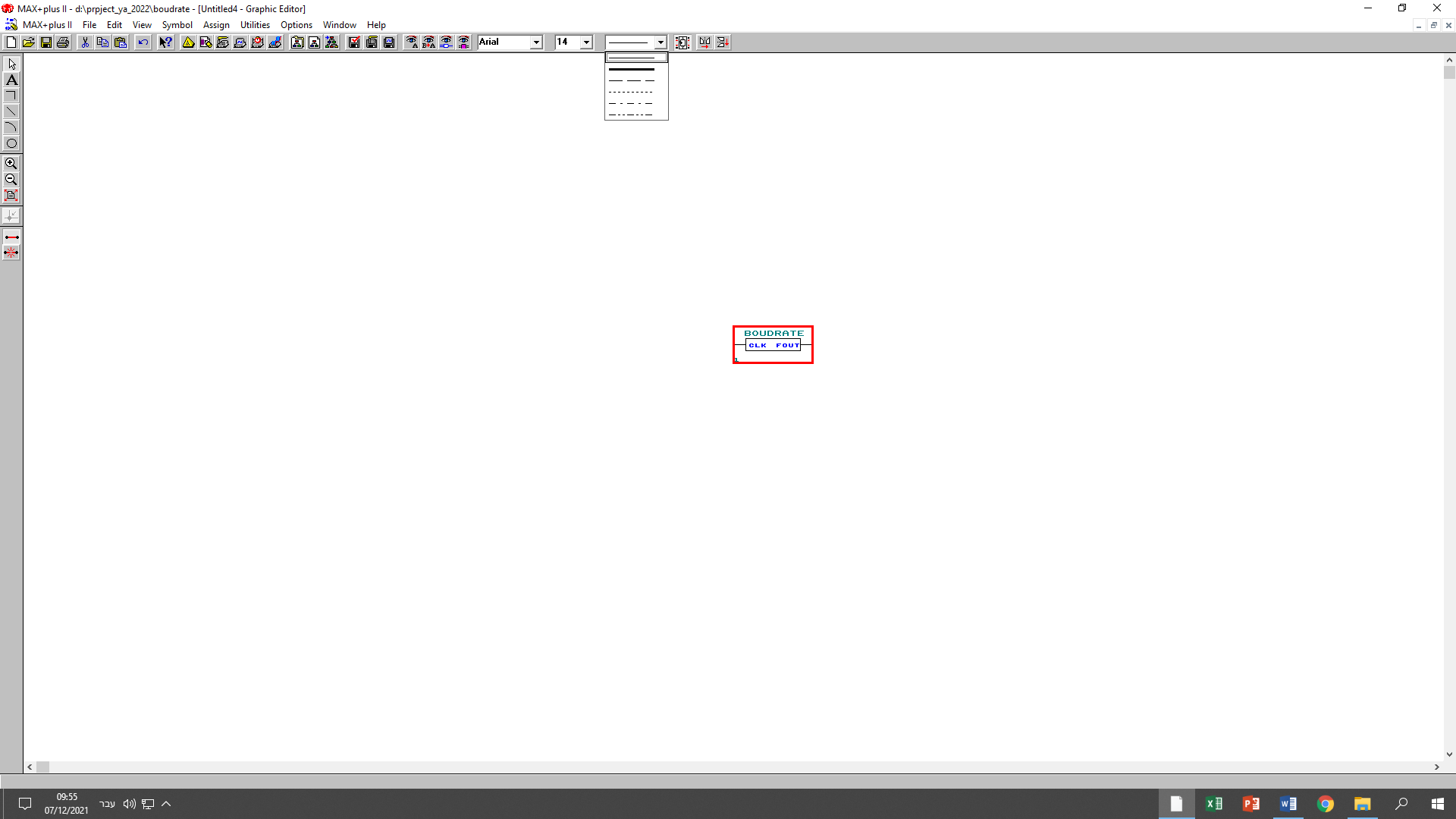
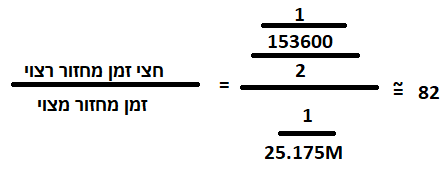
**מחלק תדר boudrate**



**תפקיד התוכנית:**

לקבל תדר של MHz125.175 מ-main\_clk ולהוציא תדר של Hz153600.

**עקרון כתיבת התוכנית:**



אנחנו נרצה תדר גבוה פי 16 מקצב העבודה: 16 \*9600 = 153600

* עקרון לכל כתיבת מחלק תדר: **2/ f רצוי / f מצוי**

**הסבר כניסות ויציאות:**

כניסת clk – תגיע מהMain\_clk- בתדר של 21.175MHz.

מוצא Fout – תוציא תדר של 153600Hz ל-My\_TX.

**עקרון כתיבת התוכנית מחלק תדר:**

1. נגדיר משתמה עזר cnt שיכיל מספר עשרוני מ-0 עד 82.
2. בכל עליית שעון מצוי של תדר 25.175kHz נוסיף אחד לcnt (cnt++).
3. כש-cnt= 82 נאפס אותו ונהפוך מצב קודם במוצא.

**קוד התוכנית:**

library ieee;

use ieee.std\_logic\_1164.all;

entity BOUDRATE is

port ( clk: in bit; -- 25175000hz

Fout : buffer bit);

end;

architecture behave of BOUDRATE is

signal cnt : integer range 0 to 82; -- 25175000/153600/2=82

begin

process (clk)

begin

if clk'event and clk='1' then

if cnt<82 then

cnt<=cnt+1;

else

cnt<=0;

Fout<=not Fout;

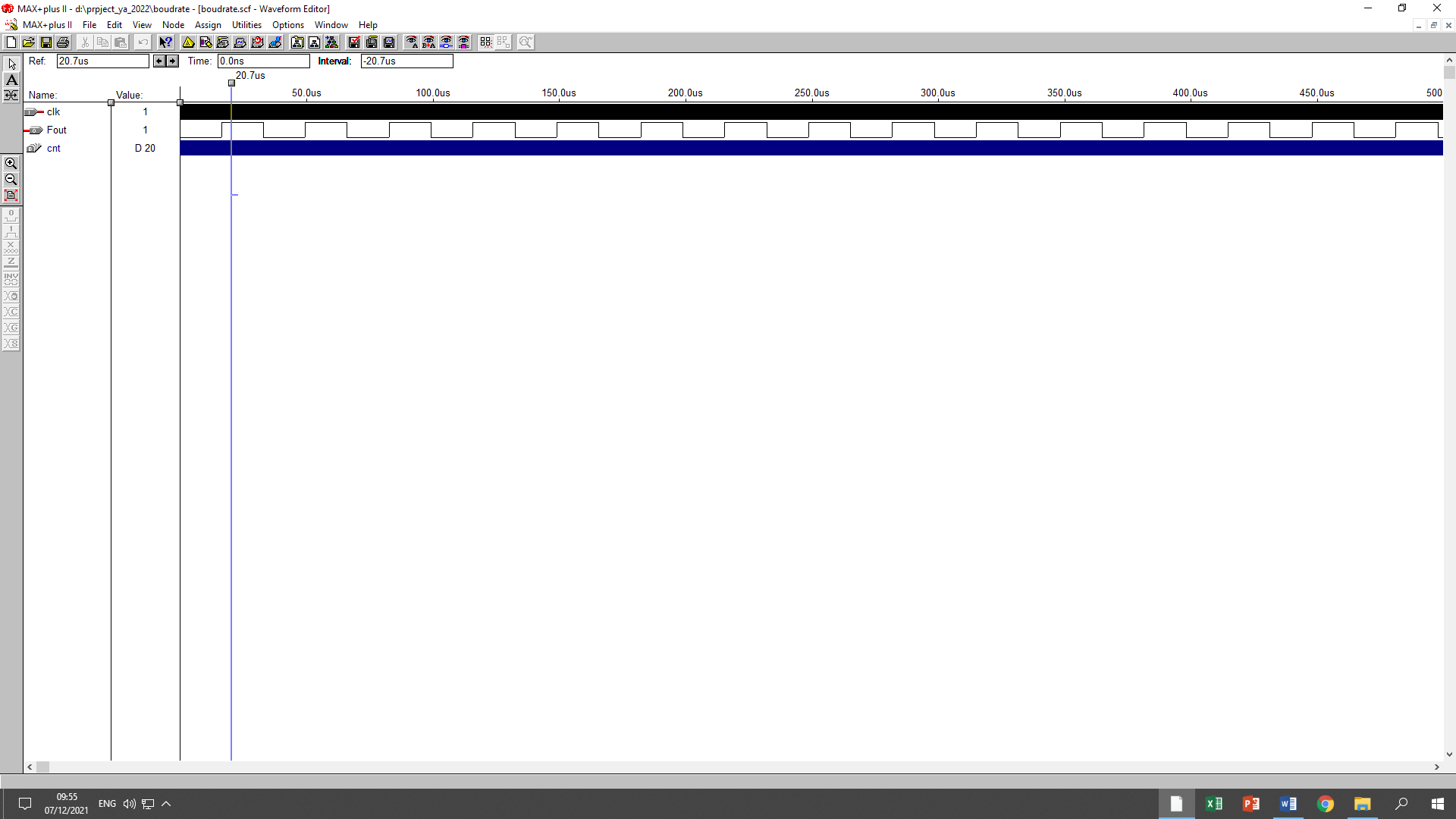
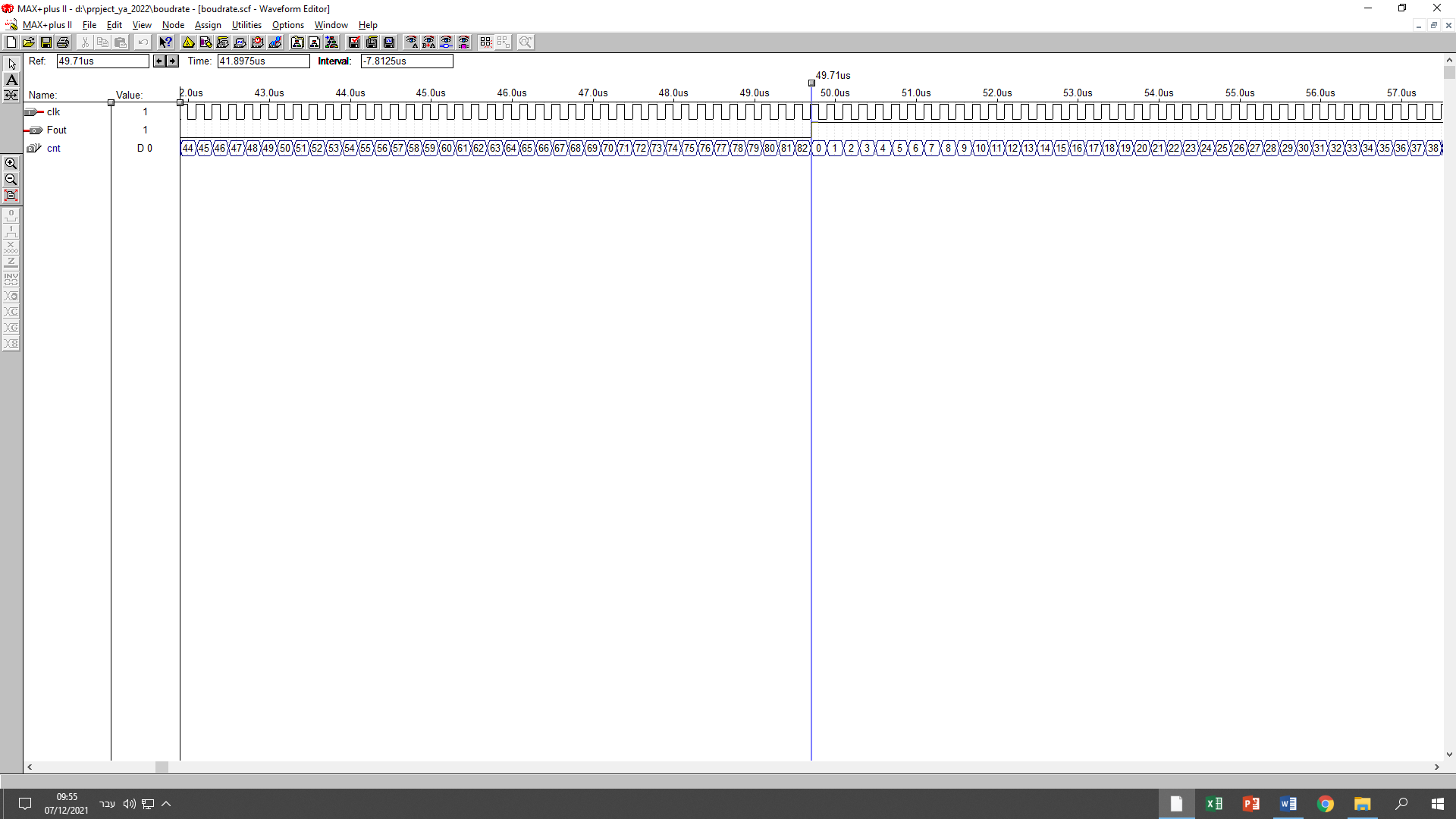
end if;

end if;

end process;

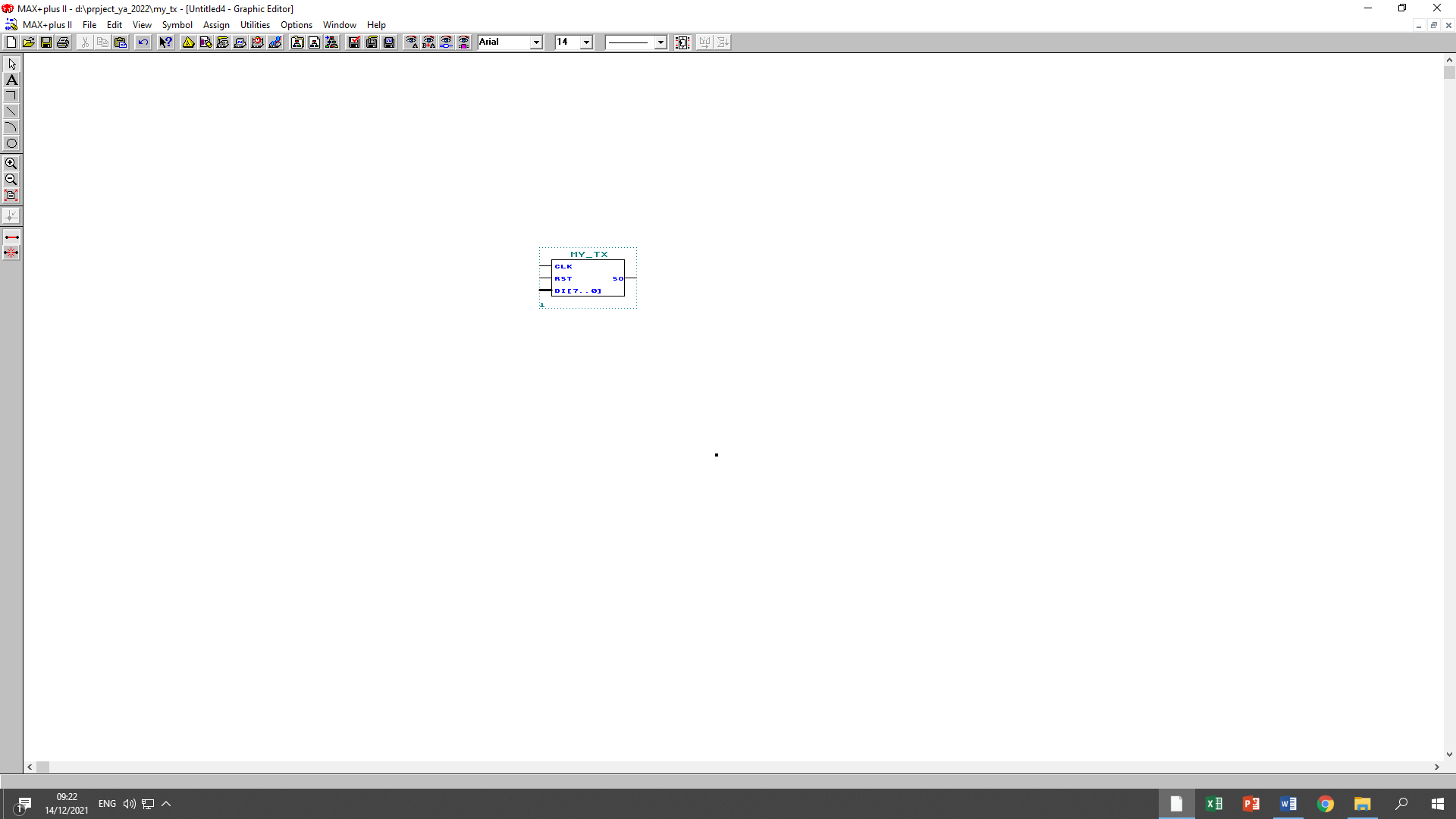
end behave;

**דיאגרמות זמן המוכיחות את פעולת התוכנית:**



בתמונה השנייה ניתן לראות את דיאגרמת הזמן הראשונה בזום. ניתן לראות שאיפה שנמצא הסמן cnt בדיוק הגיע ל-82 מיד אחרי הסמן ניתן לראות שהמוצא התהפך ו-cnt התאפס.

**My Tx**



תדר של 153600

clk

TX

So

8

מידע טורי בתקן RS232

DI

מידע מקבילי

**תפקיד התוכנית:**

לקבל מידע מקבילי מה-freq meter ולהוציא מידע טורי בתקן RS232 (מה-LSB ל-MSB , start bit וstop bit- , ומתיחת זמן השליחה ל16 זמני מחזור)

**עקרון כתיבת התוכנית:**

* נגדיר משתנה עזר פנימי cnt שיספור עד 160 במחזוריות(מכיוון שיש 10 סיביות בסך הכל כולל start bit ו-stop bit כאשר כל אחת נמשכת 16 זמני מחזור שעון).
* נגדיר משתנה עזר פנימי clr לאיפוס בסוף שליחת מידע.

**הסבר כניסות ויציאות:**

כניסת clk – תגיע מהמוצע של מחלק תדר Boud Rate שיוציא תדר של 153600Hz.

כניסת DI – תגיע מהמוצא של freq meter כמידע מקבילי בעל 8 סיביות

מוצא SO – תוציא את המידע שהתקבל כמידע טורי בתקן RS232.

**קוד התוכנית:**

library ieee;

use ieee.std\_logic\_1164.all;

entity my\_tx is

נגדיר את di כ-bit\_vector מכיוון שנרצה לגשת לחלקים ספציפיים במידע ואת זה לא נוכל לעשות עם integer

port (clk,rst: in bit;

di : in bit\_vector(7 downto 0);

so : out bit);

end;

architecture behave of my\_tx is

signal cnt : integer range 0 to 160;

נגדיר את cnt כ-integer מכיוון שנרצה לבצע עליו פעולה מתמטית של הוספה

signal clr : bit;

begin

process (clk,rst)

begin

if rst='1' or clr='1' then cnt<=0; so<='1';

במשך 16 זמני מחזור ראשונים נקבל start bit

elsif clk'event and clk='1' then

if cnt<160 then cnt<=cnt+1; else cnt<=0; end if;

if cnt<16 then so<='0'; -- start bit

elsif cnt<32 then so<=di(0); -- lsb

elsif cnt<48 then so<=di(1);

elsif cnt<64 then so<=di(2);

שליחת סיביות המידע

elsif cnt<80 then so<=di(3);

elsif cnt<96 then so<=di(4);

elsif cnt<112 then so<=di(5);

elsif cnt<128 then so<=di(6);

elsif cnt<144 then so<=di(7); -- msb

במשך 16 זמני מחזור אחרונים נקבל stop bit

elsif cnt<160 then so<='1'; -- stop bit

end if;

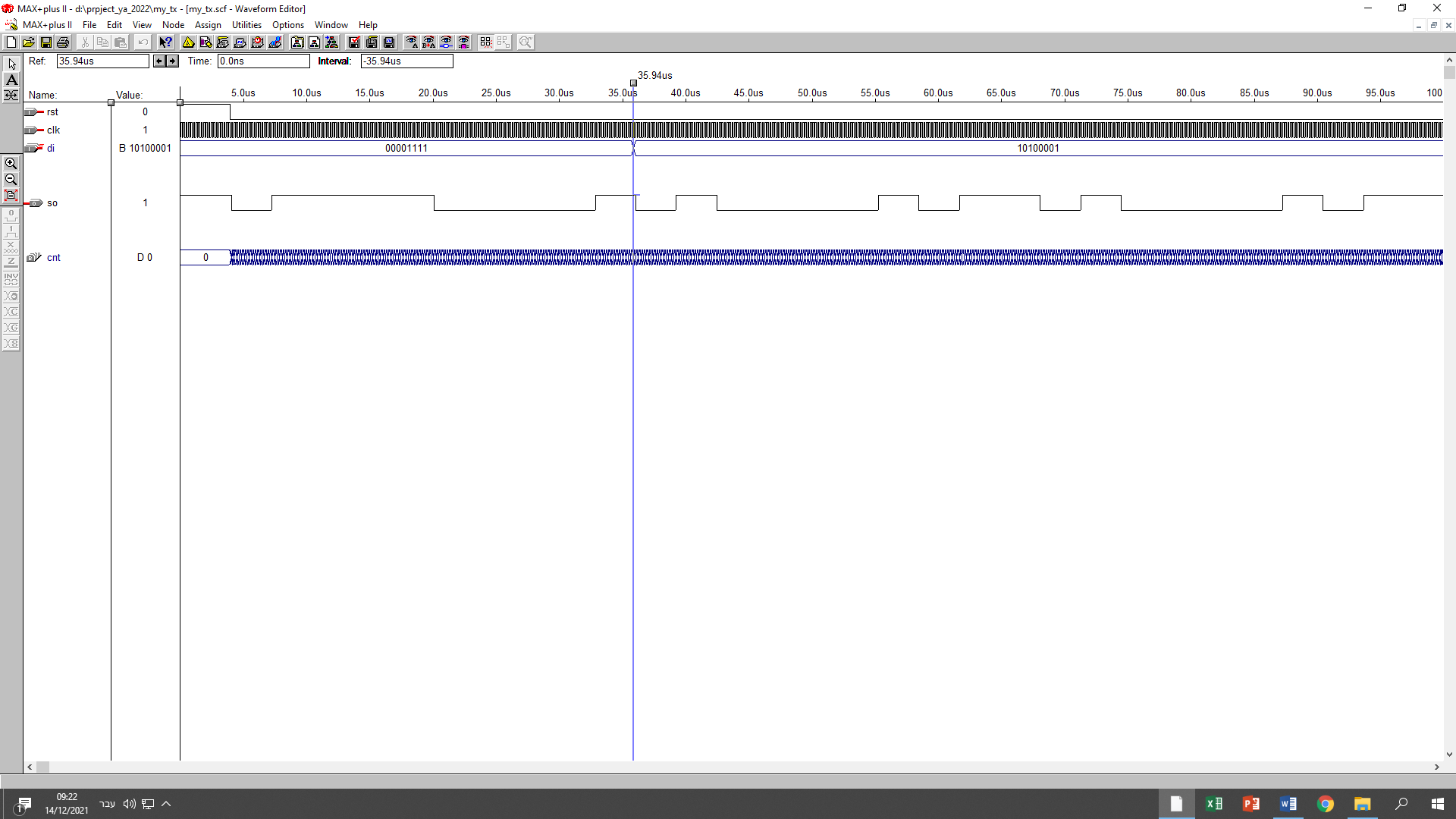
end if;

end process;

clr <='1' when cnt=160 else '0';

end behave;

**דיאגרמות זמן המוכיחות את פעולת התוכנית:**



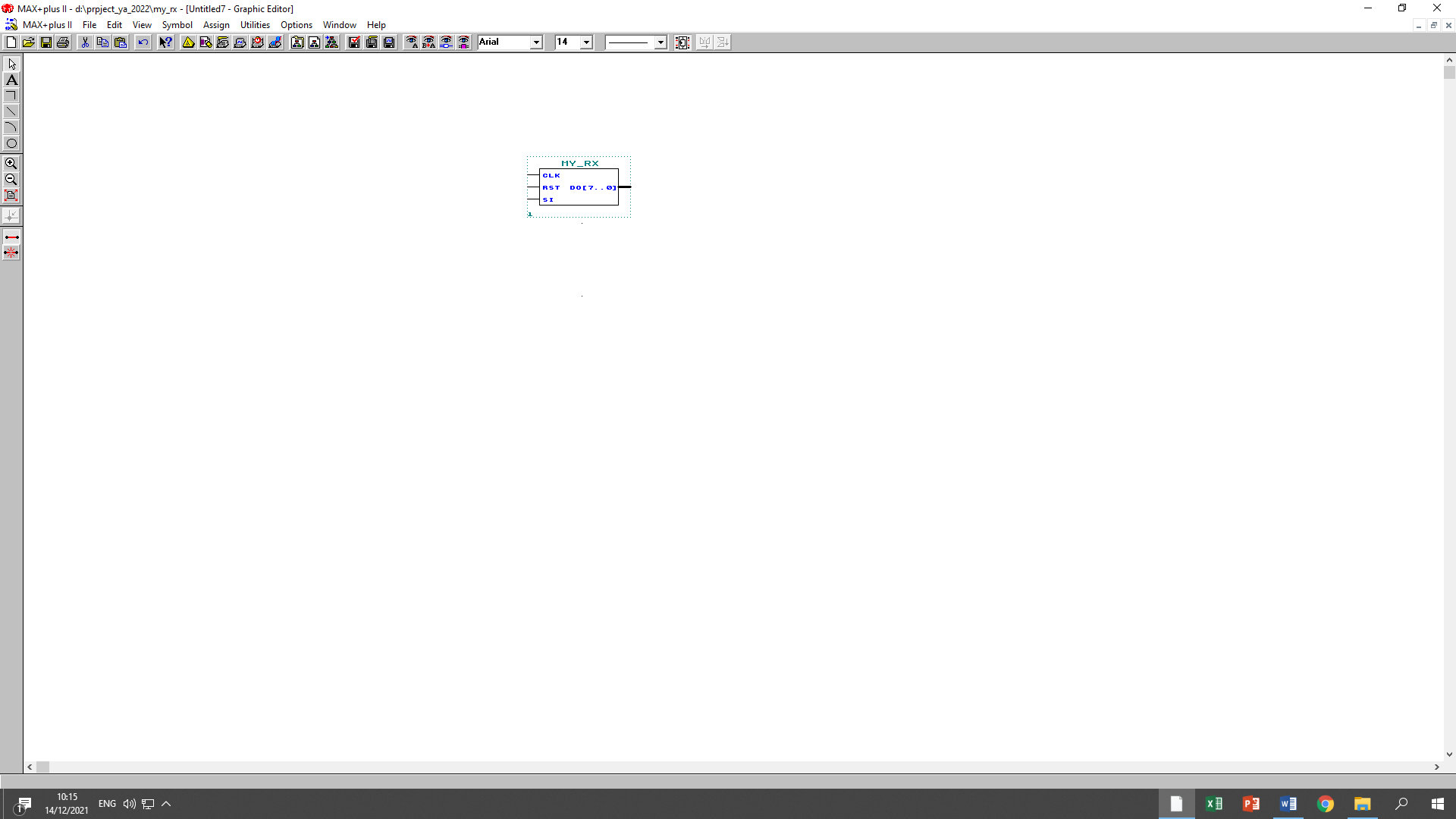
Start bit

ניתן לראות שבין stop bit לבין start bit יש מידע בעל 8 סיביות שמתחיל מה-LSB ונגמר ב-MSB

Stop bit

ארבעה אפסים נגמר ב-stop bit

ארבעה אחדות מתחיל מה-LSB

**My RX**

תדר של 153600

clk

rst

8

RX

Si

Do

מידע טורי בתקן RS232

**תפקיד התוכנית:**

תפקידו לקבל מידע טורי בתקן RS232 ולהוציאו בתנאי שהוא תקין.

**הסבר כניסות ויציאות:**

כניסת clk –תדר של 153600Hz שיגיע ממוצא ה-Boud Rate.

כניסת Si – מידע טורי שיגיע מה-Pc בתקן RS232 שמייצג את ערך האור הרצוי.

מוצא Do – תוציא את המידע שהתקבל כמידע מקבילי בעל 8 סיביות.

**עקרון כתיבת התוכנית:**

1. נגדיר Process ראשון לזיהוי ירידת '1' ל-'0' בכניסת Si שמשמעותו מעבר מ-stop bit ל-start bit ונעלה דגל start ל-'1'.

2. ב-process שני נספור בכל עליית שעון עד 160 מחזורים ונדגום באמצע זמן שליחת כל סיבית את המידע.

3. לקראת סוף קבלת המידע לתוך משתנה פנימי Temp נבדוק תקינות ורק אם המידע תקין נעביר את Temp למוצא Do.

* שלבים 2,3 יקרו רק כאשר דגל start='1'.

**למה צריך process נפרד לבדיקת מעבר מ-stop bit ל start bitב-Si?**

מכיוון שאנחנו צריכים לבדוק אם הייתה ירידת אות ב-Si (צריך לבדוק event) ואי אפשר לבדוק 2 event שונים ב-process אחד (ב-process השני נצטרך לבדוק עליית שעון שלזה נצטרך לבדוק event).

המקשר בין 2 ה-process הוא הדגל start.

**קוד התוכנית:**

library ieee;

use ieee.std\_logic\_1164.all;

entity my\_rx is

לתוך start\_ bit ו-stop\_bit נדגום את סיביות ההתחלה והסיום, משתנה clr יהווה איפוס פנימי, משתנה temp יאגור את סיביות המידע ויעביר אותו למוצא רק אם הוא תקין, START מחבר בין שני ה-process.

port (clk,rst,si : in bit;

do : out bit\_vector(7 downto 0));

end;

architecture behave of my\_rx is

signal clr,start\_bit,stop\_bit:bit;

signal temp : bit\_vector( 7 downto 0);

signal cnt: integer range 0 to 160;

signal START : bit; -- flag

begin

-------------- waiting for srart bit process ----------------

process (clr,rst,si)

begin

if rst='1' or clr='1' then START<='0';

elsif si'event and si='0' then -- waiting for srart bit

START<='1';

הכול יקרה בתנאי ש-START שווה '1'

end if;

end process;

------------------- get the bits ------------

process (clr,rst,clk)

begin

if rst='1' or clr='1' then cnt<=0;

elsif clk'event and clk='1' then

if START='1' then

if cnt<160 then cnt<=cnt+1; else cnt<=0; end if;

if cnt=8 then start\_bit<=si;

elsif cnt=24 then temp(0)<=si; -- lsb

elsif cnt=40 then temp(1)<=si;

elsif cnt=56 then temp(2)<=si;

elsif cnt=72 then temp(3)<=si;

דגימת סיביות המידע

elsif cnt=88 then temp(4)<=si;

elsif cnt=104 then temp(5)<=si;

elsif cnt=120 then temp(6)<=si;

דגימת סיביות המידע

בדיקה שהמידע תקין

elsif cnt=136 then temp(7)<=si;

elsif cnt=152 then stop\_bit<=si;

elsif cnt=157 then

if start\_bit='0' and stop\_bit='1' then do<=temp; end if;

end if;

end if;

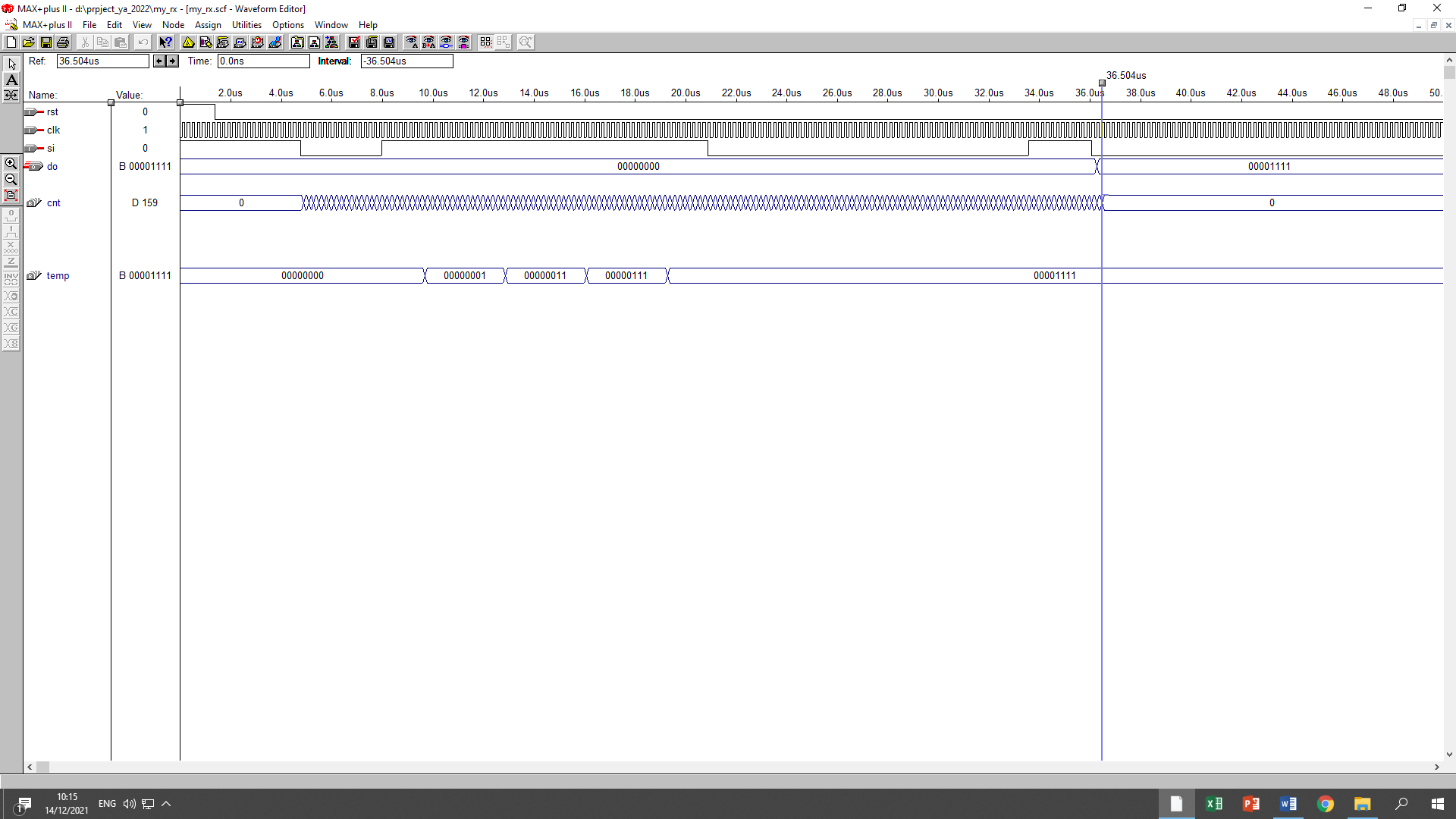
end if;

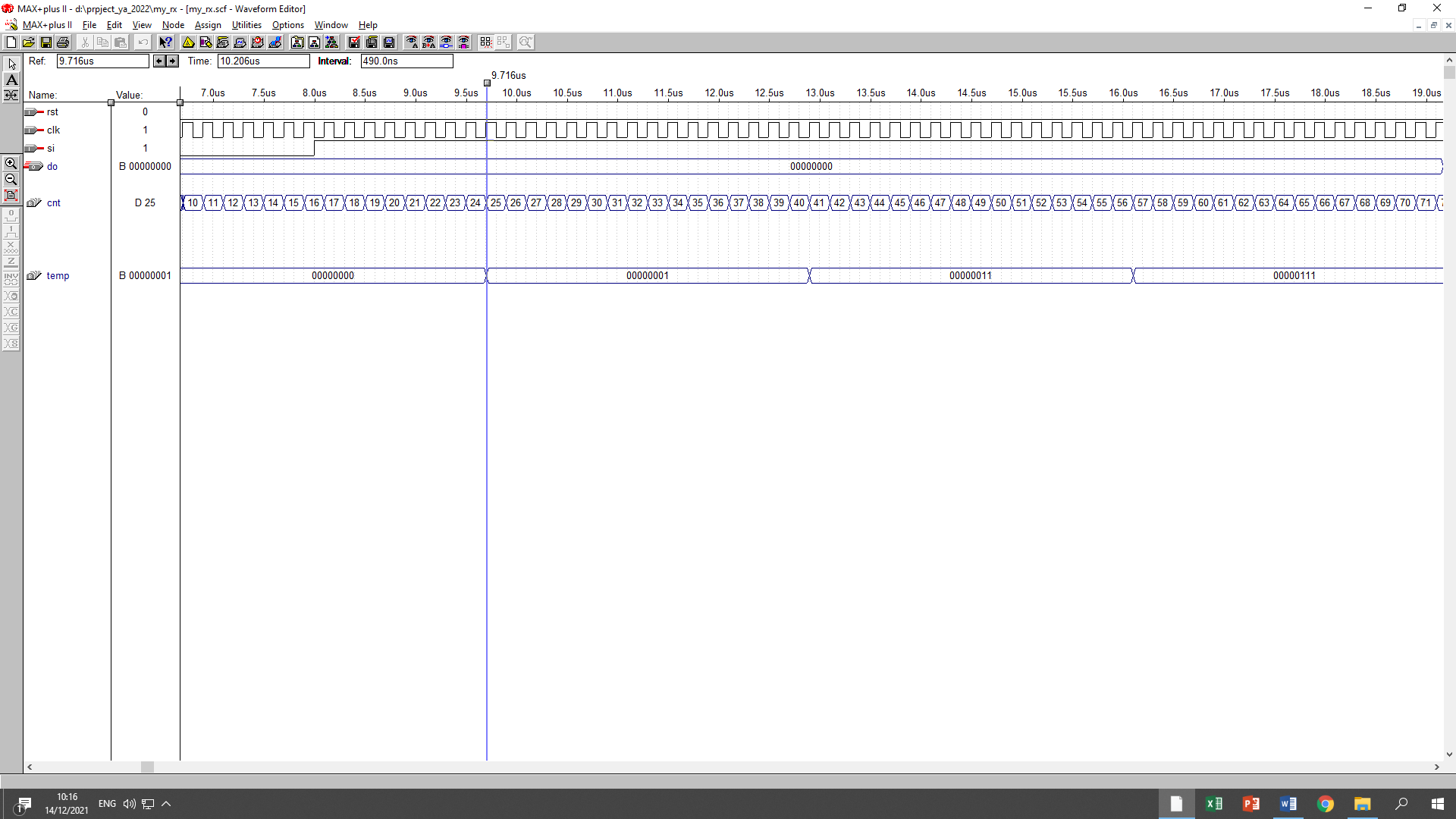
הפעלת ה-clr תמיד מכיוון שזה מחוץ ל-process

end process;

clr<='1' when cnt=159 else '0';

end behave;





ארבעה אפסים שהם ה-LSB

נכנס מידע 0F מה-LSB ל-MSB ובסיום 160 זמני מחזור שעון המידע עובר מ-temp ל-Do

ארבעה אחדות שהם ה-MSB

סיבית נכנסה אחרי 16

זמני מחזור

סיבית נכנסה אחרי 16

זמני מחזור

סיבית נכנסה 16

זמני מחזור אחרי האמצע של start , כלומר האמצע של הסיבית הראשונה

**splitter**

library ieee;

use ieee.std\_logic\_1164.all;

entity spliter is

port ( din : in bit\_vector(7 downto 0);

high,low : out bit\_vector(3 downto 0));

end;

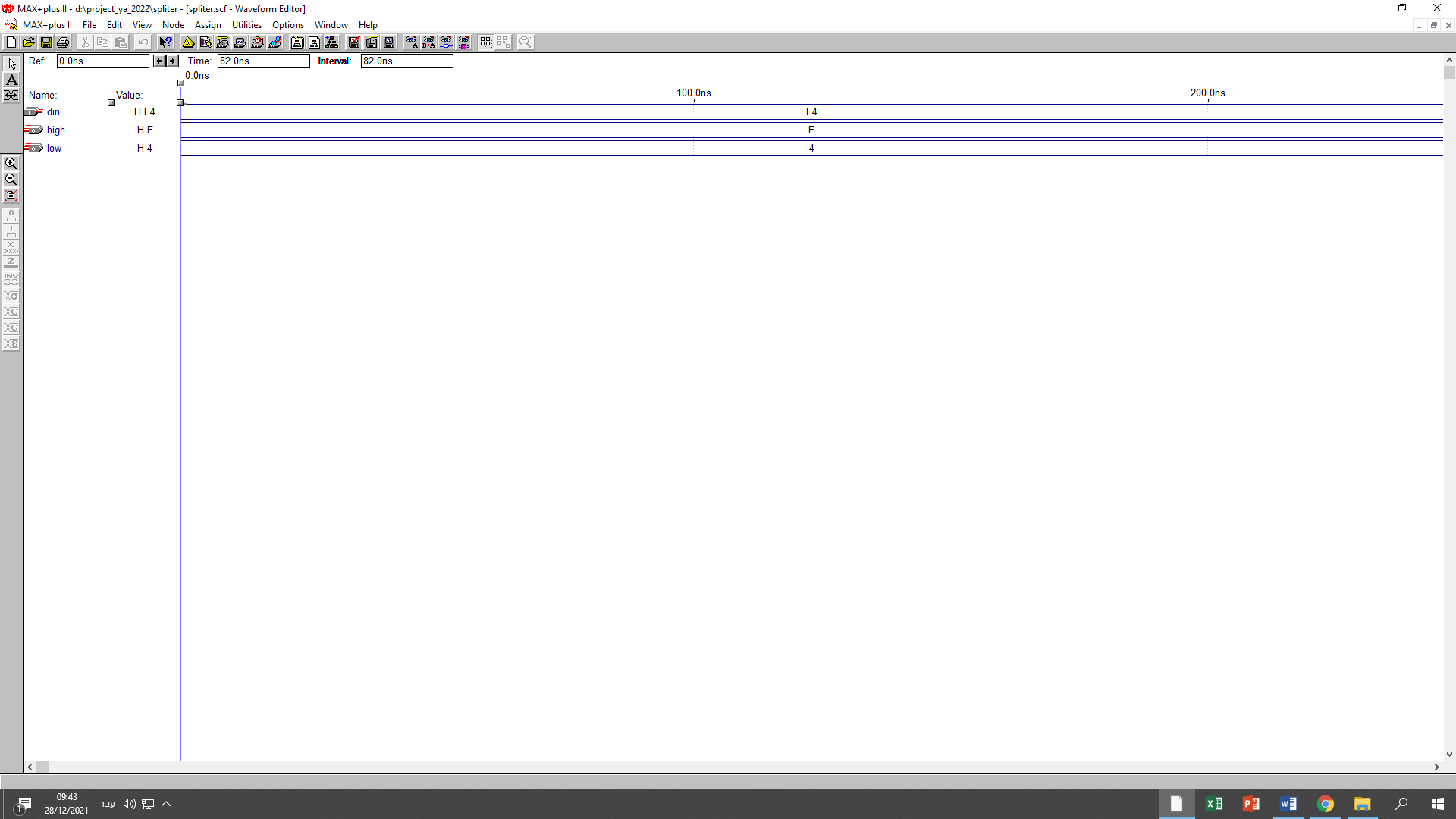
architecture behave of spliter is

begin

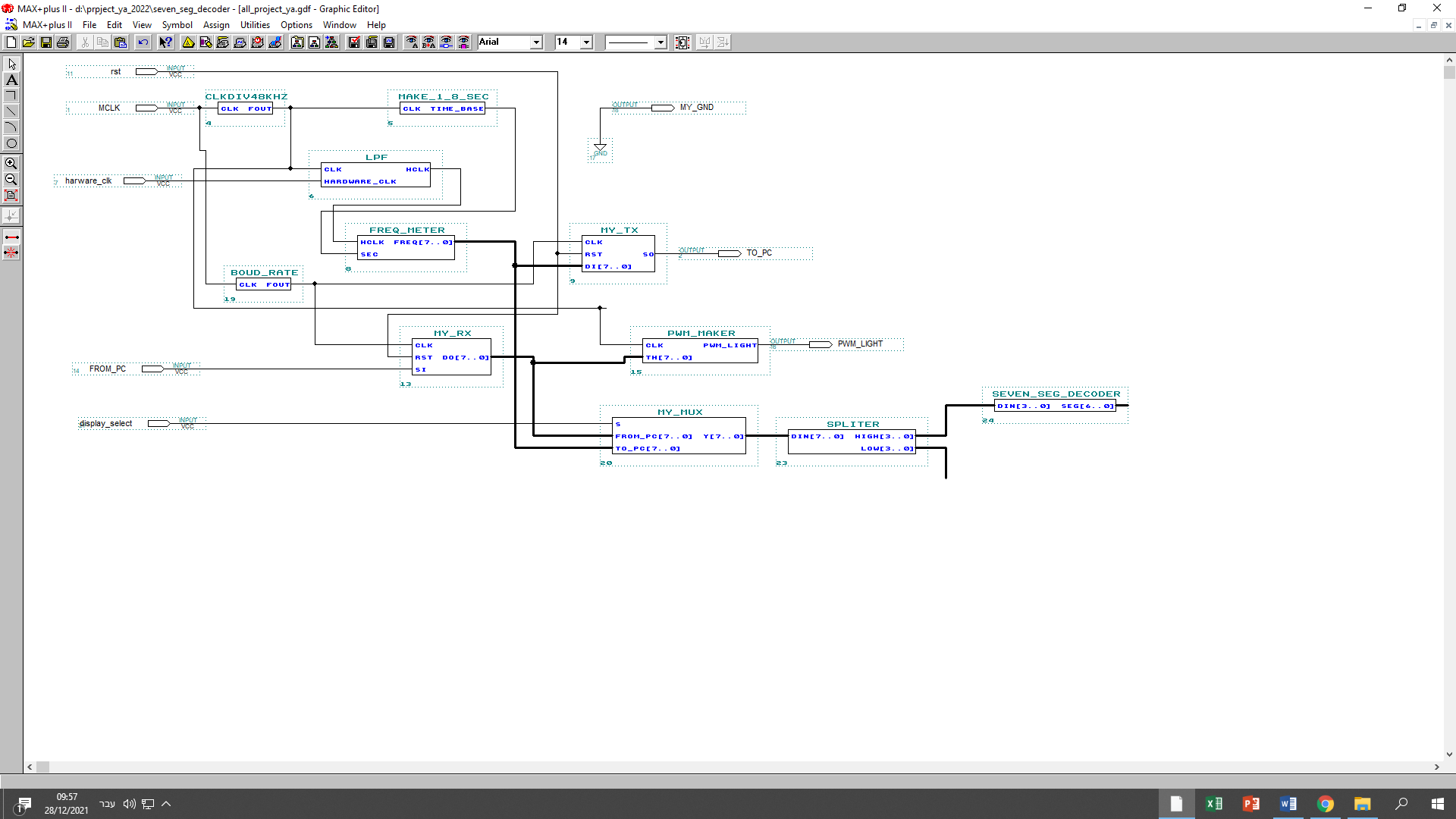
high<=din(7 downto 4);

low<=din(3 downto 0);

end behave;



**Seven segment decoder**



library ieee;

use ieee.std\_logic\_1164.all;

entity seven\_seg\_decoder is

port ( din : in integer range 0 to 15;

seg: out bit\_vector(6 downto 0));

end;

architecture behave of seven\_seg\_decoder is

begin

-- 6543210

-- abcdefg

with din select

seg<="0000001" when 0,

"1001111" when 1,

"0010010" when 2,

"0000110" when 3,

"1001100" when 4,

"0100100" when 5,

"0100000" when 6,

"0001111" when 7,

"0000000" when 8,

"0000100" when 9,

"0001000" when 10,

"1100000" when 11,

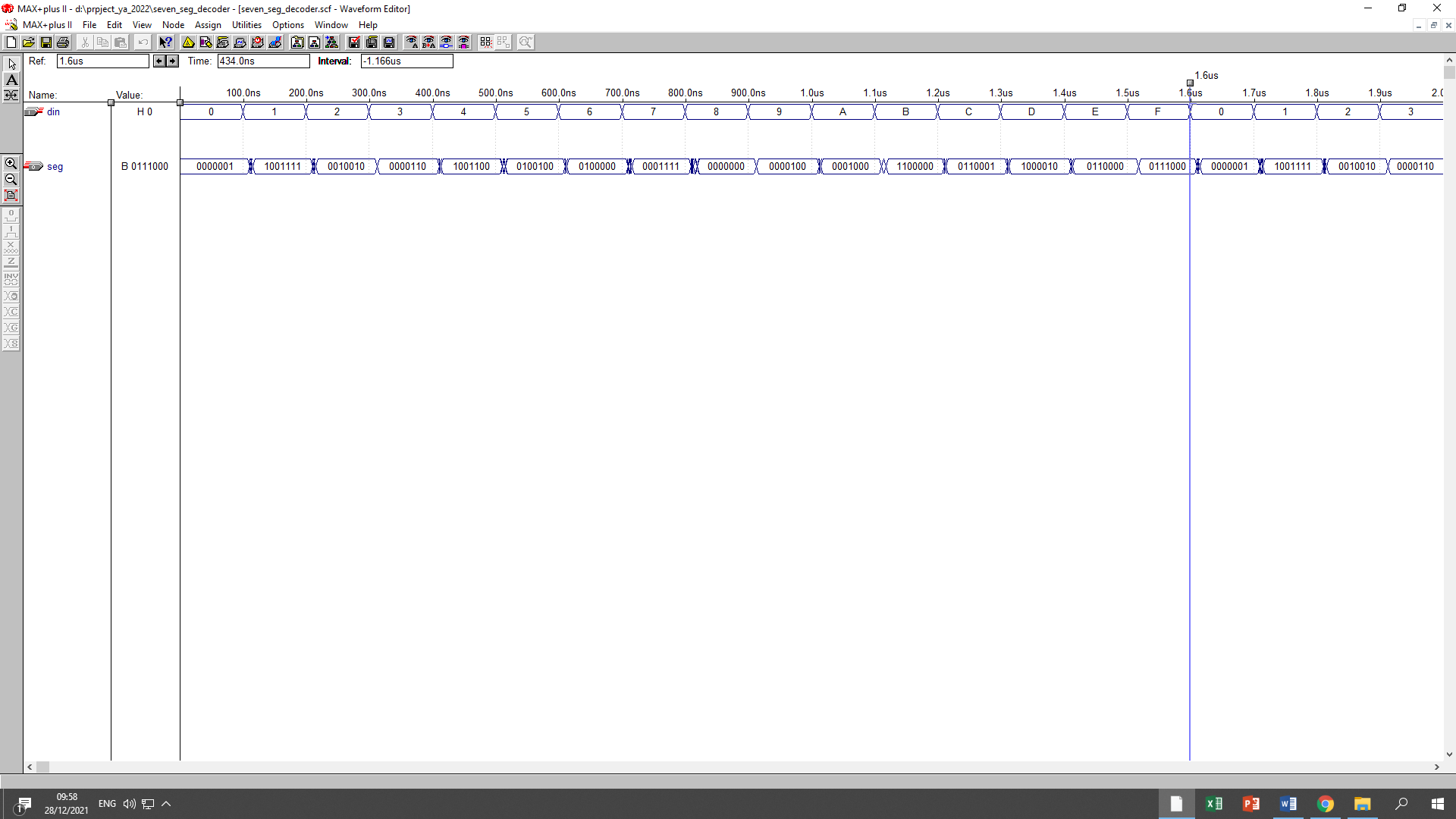
"0110001" when 12,

"1000010" when 13,

"0110000" when 14,

"0111000" when 15;

end behave;



**קוד esp מוסבר**

#include <Arduino.h>// מוסיף לפרויקט את ספריית ארדואינו

#include <WiFi.h>//מוסיף לפרויקט את ספריית וייפיי

#include <FirebaseESP32.h>// מוסיף לפרויקט את ספריית פיירבייס

#define FIREBASE\_HOST "alteratutorial-default-rtdb.europe-west1.firebasedatabase.app"//לינק למסד הנתונים

#define FIREBASE\_AUTH "9U5cGGnuLfcHWckDeVCA5hvMSf1Ki55yUNlVU6BQ"//קוד סודי

#define WIFI\_SSID ""//שם הוויפי

#define WIFI\_PASSWORD ""//קוד סודי לוויפי

#define RXD2 16 //הגדרת פין המקלט של האי-אס-פי

#define TXD2 17//הגדרת פין המשדר של האי-אס-פי

#define ledpin 2//הגדרת פין לד

FirebaseData fbdo;//משתנה פיירבייס

int FromFB, FromAlt;//הגדרת משתנים שיאחסנו את הנתונים מהפיירבייס ומהאלטרה

bool a = false;//משתנה בוליאני להבהוב הנורה

void blink();//הצהרת פונקציית הבהוב הנורה

void setup()

{

Serial.begin(115200);//אתחול ערוץ העבודה הראשון עם קצב עבודה

Serial2.begin(9600, SERIAL\_8N1, RXD2, TXD2);//אתחול ערוץ העבודה של האלטרה

pinMode(ledpin, OUTPUT);//הגדרת הלד כמוצא

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); //אתחול וויפי

Serial.print("Connecting to Wi-Fi");//מראה בקונסול שהתחבר לוויפי

while (WiFi.status() != WL\_CONNECTED)//תנאי הפועל כל עוד הוויפי לא מחובר

{

Serial.print(".");//הופעת נקודה של טעינה בקונסול

delay(300);//עיכוב של 300 מילישניות

}

Serial.println("Connected with IP: ");//מראה בקונסול שהתחברת עם איפי:

Serial.print(WiFi.localIP());//מראה את האיפי

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);//אתחול הפיירבייס

Serial.println("Firebase Connected");//מראה בקונסול התחברות לפיירבייס

Firebase.reconnectWiFi(true);//חיבור מחדש לפיירבייס עם וויפי במקרה של כשל

}

void loop()

{

if (Firebase.getInt(fbdo, "to\_altera"))//תנאי הקורה כאשר הפיירבייס מקבל נתון מענף "לאלטרה"

{

FromFB = fbdo.intData();//מציב במשתנה "לאלטרה" את המידע מהענף

}

delay(20);//עיכוב של 20 מילישניות

if (Serial2.available())//תנאי כאשר ערוץ העבודה של האלטרה פועל

{

Serial2.write(FromFB);//מעביר לאלטרה את המידע מהמשתנה

}

delay(20);//עיכוב של 20 מיילשניות

while (Serial2.available())//תנאי הפועל כל עוד ערוץ העבודה של האלטרה עובד

{

FromAlt = Serial2.read();// מציב במשתנה "מאלטרה" את המידע הנקרא מערוץ העבודה

delay(2);//דיליי של 2 מילישניות

Firebase.setInt(fbdo, "From\_altera", FromAlt);//מציב בענף "מאלטרה" בפיירבייס את המידע מהמשתנה "מאלטרה"

}

blink();//מפעיל את פונקציית הבהוב הלד

delay(50);//עיכוב של 50 מילישניות

}

void blink()

{

a = not a;//הופך מצב קודם במשתנה הבולאני

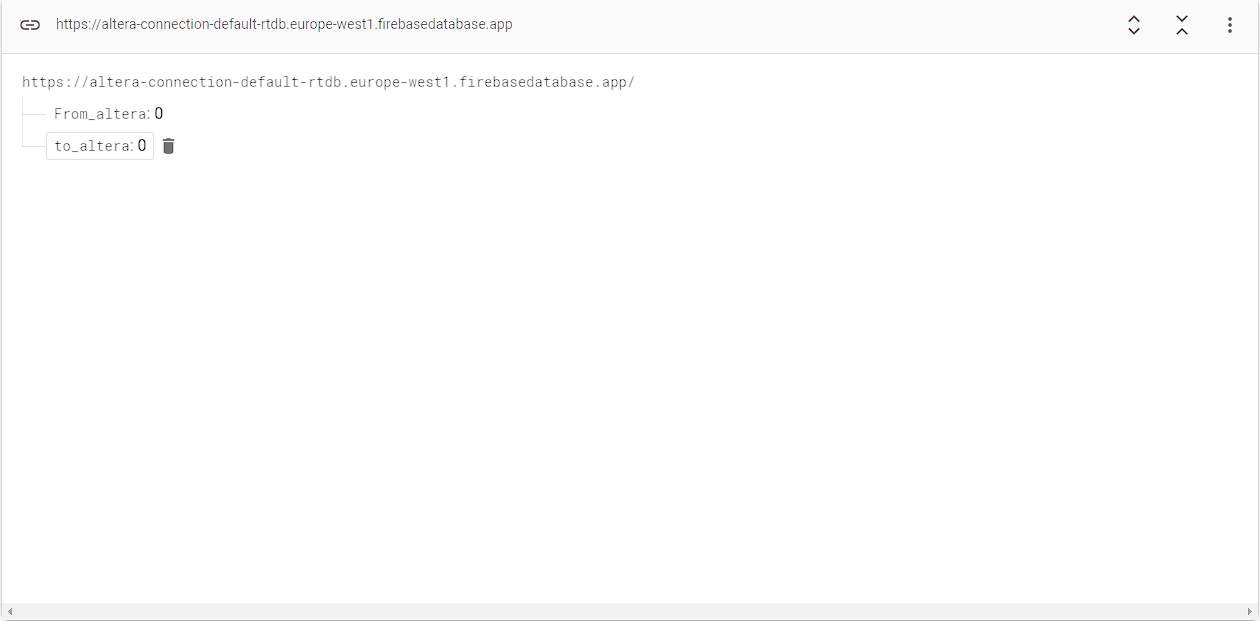
if (a)//כאשר המשתנה במצב'1'

digitalWrite(ledpin, HIGH);//מדליק את הלד

else//כאשר המשתנה במצב'0'

digitalWrite(ledpin, LOW);//מכבה את הלד

}

**מסד הנתונים בפיירבייס**