Core Data Flow

**Document purpose:**  describe the data flow between the differences parts in the core and characterize every sub-block.

Core top design:

Trigger

Configuration

Core

Clk

Data in WC

WBS

WBM

Read Controller

Reset

Trigger\_type

Trigger\_position

System\_status

wc\_to\_rc

Generics

Clk

Reset

rc\_to\_WBM

Clk\_to\_start

Addr\_in

clk

Reset

dout\_valid

aout\_valid

Data\_in\_RC

Clk

Reset

Write Controller

Generics

Generics

Trigger

Registers

Data in RAM

Trigger\_found

WBS\_to\_REG

Generics

RAM

Addr\_out

The core inputs are the trigger characteristic, data signals that came from the signal generator, and clk and Reset signals that came from Clock and Reset part.

The trigger characteristic are been save in the registers, and the data is stored in the RAM.

Registers

Responsible for saving data from the user according to the configuration of the trigger type. In advance, they will hold data that will allow us to check the validity of the system, and data needed for the regular behavior of the system.

The registers hold the data from the user and define the type &location of the trigger. When the system is ready, 'system status' signal will rise and the system start looking for trigger rise.

תפקיד הרגיסטרים לשמור מידע שנשלח מהמשתמש בהתאם לקונפיגורציה המתאימה של סוג הטריגר, בנוסף ישמר בתוכם מידע אשר יעזור לנו לבצע בדיקות תקינות במערכת ומידע אשר נחוץ לפעולה שוטפת של המערכת.



State Machine:

בתחילת ההפעלה ישלח מידע מהמשתמש וישמר ברגיסטרים המתאימים ויגדיר לנו את סוג ומיקום הטריגר. לאחר שהמערכת תהיה מוכנה, יעלה אות system status והמונה Clk to start יתחיל את פעולתו- מונה את מספר מילות הקלט שהתקבלו.



Write Controller

Responsible for checking if there was a trigger rise according to the system status(

תפקידי הרכיב הם לקבל את המידע ולאכסן אותו בRAM בכתובת המתאימה, ולקבל את אות הTRIGGER ולקבוע האם התרחשה עליית TRIGGER.



State Machine:

הרכיב מקבל מידע מהמשתמש, מחשב את הכתובת שבה המידע הנ"ל ישמר בRAM, ושולח את המידע והכתובת לRAM.

לאחר זיהוי עליית TRIGGER, הרכיב יחשב את כתובת ההתחלה וכתובת הסיום של המידע הרלוונטי אשר צריך לצאת למשתמש וישלח אותן לרכיב הRC. אם כל המידע הרלוונטי נשמר כבר ולא צריך לקבל מהמשתמש מידע חדש (נבדק לפי ערך TRIGGER POSOTION), המערכת תסיים את פעולתה. במידה ולא, המערכת תמשיך לקבל מידע מהמשתמש.



Read controller:

הרכיב מקבל את כתובת ההתחלה וכתובת הסיום של המידע הרלוונטי שצריך לצאת חזרה למשתמש, מוציא אותו מהRAM , ושולח אותו דרך הWBM חזרה למשתמש (GUI).



State Machine:

המערכת מקבלת את כתובת הTRIGGER, מחשבת את כתובת המידע הרלוונטי שצריך לצאת למשתמש, שולחת אותו לRAM, מקבלת חזרה את המידע המבוקש, שולחת אותו חזרה למשתמש דרך WBM, במידע ולא סיימנו לשלוח את כל המידע, נחשב את הכתובת הבאה של המידע שיקרא מהRAM ונבצע שוב את אותן הפעולות.



RAM

תפקיד הRAM הוא לאחסן את המידע המגיע בהתאם לכתובת המחושבת, ולהוציא את המידע המתאים לאחר מכן בהתאם לכתובת שנשלחה.



בלוקים פנימיים:



State Machine:

תחילה אנו מגדירים את גודל הRAM לפי הGENERICS הרלוונטים.

בהמשך אנו מקבלים מידע ("מילה" אחת בכל פעם), מקבלים את הכתובת שבה תישמר המילה הנוכחית, שומרים את המידע לפי הכתובת הרלוונטית.

לבסוף אנו מקבלים כתובת של מילה בRAM, מוציאים החוצה את המילה המתאימה לכתובת שקיבלנו. במידה ולא סיימנו להוציא את כל המידע, נקבל כתובת חדשה ונוציא את המילה המתאימה לה. במידה וסיימנו להוציא את כל המידע המבוקש עבור אות הTRIGGER הנוכחי, נחזור ונחכה לקבל מידע חדש לשמירה ברכיב עבור אות הTRIGGER הבא.

