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

Input Data

WBS

WBM

Read Controller

Reset

Trigger\_type

Trigger\_position

System\_status

ram\_to\_wc

wc\_to\_rc

WBS\_to\_wc

ram\_to\_rc

Clk

Reset

rc\_to\_WBM

Clk\_to\_start

System\_status

wc\_to\_ram

clk

Reset

Out\_valid

Enable\_out\_valid

Data\_out

Clk

Reset

Write Controller

RAM

Generics

Generics

Trigger

Registers

Input Data

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.

בתחילת ההפעלה ישלח מידע מהמשתמש וישמר ברגיסטרים המתאימים ויגדיר לנו את סוג ומיקום הטריגר. לאחר שהמערכת תהיה מוכנה, יעלה אות system status ואח"כ תתחיל המערכת לחפש עליית טריגר.



Write Controller

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

תפקיד ה WC הוא לבדוק לפי נתוני המערכת האם התרחשה עליית טריגר (כמובן בהתאם לקונפיגורציה המתאימה).

מגיע אות טריגר ונשמרות N הדגימות האחרונות שלו (בהתאם להגדרת סוג הטריגר) ברכיב shift register, לאחר מכן מתבצעת השוואה בין דגימות אלו לבין סוג הטריגר (נקרא מרגיסטרים). בהתאם לתוצאת ההשוואה, למצב המערכת (אות system status) ולכתובת שנשלחה מה RAM (RAM to WC) נקבעת כתובת הקריאה מהRAM. (wc to RAM). במקביל ממציאת עליית טריגר מתחילה ספירת counter (clk to start) לצורכי דיבוג המערכת.



RAM

תפקיד הRAM הוא לאחסן את המידע המגיע בהתאם לכתובת המחושבת, ולהוציא את המידע המתאים לאחר מכן ע"י כתובת שנשלחת מרכיב WC.

מידע מגיע מsignal generator (Data in) לכל רכיבי simple RAM הנמצאים במקביל, ובהתאם לכתובת המתאימה (WBS\WC) נשמרת ברכיב המתאים ובמקום המתאים בו. כאשר נרצה לשלוף מידע, שוב יגיע אות (wc to RAM) ויאפשר את רכיב ה simple RAM המתאים והוא יוציא את המידע הרלוונטי (Data out)



Read controller

רכיב זה מקבל את המידע הרלוונטי מהRAM ומוציא אותו חזרה למשתמש דרך WBM.

המידע מגיע דרך הRAM (RAM to RC) ויוצא למשתמש דרך WBM (rc toWBM).

שאלות:

* כרגע יש קצת בלאגן לגבי מי, איפה ואיך מחושבת כתובת הכתיבה לRAM (כאשר מגיעים האותות, הכתובת צריכה להתאפס לתא הראשון ואז פשוט להעלות את הכתובת בהתאם למספר האותות המוקלטים). כמובן שנתקן ונסנכרן הכל לאחר תשובתך, כרגע אצלנו יש גם קצת חוסר התאמה רצינו רק את חוות דעתך לפני שאנו משנים, חישוב הכתובת של המידע **הנכנס** כרגע מבוצע בWRITE CONTROLLER לפי השיטה שהצענו למעלה.
* האם לrdcntr תפקידים נוספים? כרגע הוא רק מקבל את המידע היוצא מהRAM ופשוט מעביר אותו החוצה דרך WBM (נראה כאילו אפשר להחליפו ברגיסטר פשוט) או שהוא בעצם מכין את המידע היוצא לפורמט ש WBM יכול לקרוא? אם נדרש שינוי בצורת המידע, כיצד אנו מבצעים זאת?