Moshe's Code Review – 21.06.2012

# RAM\_300

1. ~~לכל PORT צריך להיות מלווה הערה (הסבר על תפקידו).~~
2. ~~שורות 63-65 בעייתיות מבחינת סינתזה: בסימולציה הכול תקין ורואים אפסים. אבל - הרי אתם רוצים שימומש RAM, ו- RAM הוא זיכרון שערך האתחול לכל תא לא קיים (רק באמצעים אחרים, כמו MEGAWIZARD ולא בקוד). מה שיקרה זה שהסינתזה תממש זאת בתור רגיסטרים ולא בתור זיכרון פנימי, וזה לא מה שרציתם. אם תשימו לב אז בגרסה הקודמת של הקובץ לא היה ערך אתחול וזה היה טוב. מה שמומלץ זה להפריד בין ה- PROCESS של הכתיבה ל- mem ובין ה- PROCESS של ההשמה ל- ram\_data\_out. בשניהם לא יהיה ערך אתחול ולא יהיה reset\_n ברשימת הרגישויות.~~

מחקתי את האתחול הראשוני של ה-RAM. הבעיה שנוצרת, היא שאנחנו לא כותבים לכל השורות ב-RAM, אלא רק לאלה שאנחנו רוצים שינוי בהם. ולכן נשארות שורות ב-RAM שלא ניגשנו אליהן, ומצד שני בלוק ה-manager יקרא מהן data שהוא לא ולידי (לא 'U').

איך כדאי לממש? אולי כן לעשות אתחול ראשוני של ה-RAM כשהמערכת רק עולה אחרי reset?

פתרון אפשרי: כאשר מעלים את כל המערכת, כחלק מהאתחול הראשוני נשלח חבילות מידע עבור כל הבלוקים בתצוגה ונקבע אותם להיות סמל שחור, וכך בעצם ניגשנו וכתבנו לכל שורה ב-RAM והערך שלה ולידי (לא 'U').

שתי אפשרויות שהוצגו מקובלות – או אחרי RESET לבצע 300 כתיבות ל- RAM כדי לאפסו ב- 300 מחזורי שעון רצופים, או להשתמש בתוכנה ולשלוח חבילת מידע של סמלים שחורים.

1. ~~שורה 76 - תמיד כדאי לציין את משמעות ה- LABEL של ה- END וכאן end architecture RAM\_300\_rtl.~~
2. חבל בכלל שמימשתם זאת ולא השתמשתם ב- ram\_simple מספריית MEMORY של בארי (קוד גנרי, נותר רק לשנות את ערכי הג'נריקס בקריאה ליח' מלמעלה).

# Opcode\_unite

1. ~~שורות 52-53: ערך אתחול ל- PORT-ים מותר רק עבור קוד המיועד לסימולציה התנהגותית ולא לקוד המיועד לסינתזה. יש להוריד את ערך האתחול מהצהרת ה- PORT-ים.~~
2. ~~ל- PORT בשם opu\_wr\_en חסר ערך אתחול בזמן RESET, ליד שורות 96-98.~~

# opcode\_store

1. ~~שורה 82 - יש ???~~
2. ~~שורות 88-90: מומלץ להוסיף הערות. גם בשורה 113, 120 ו- 123.~~
3. לגבי vsync\_active\_proc: דיברנו על דגימה פעמיים נקי במעבר שעונים. כאן יש שימוש ב- start\_trigger\_1 (דגימה רק פעם אחת) בשורה 144 וזה לא תקין (תהיה בעיית CDC). בכלל אפשר נראה לי לוותר על start\_trigger\_1 ו- start\_trigger\_4 בשורה 144 והכול יהיה תקין ללא שינוי נוסף (לוותר בכלל על השימוש ב- start\_trigger\_4).

שאלה למשה: כדי לא להשתמש בסיגנל op\_str\_rd\_start, אנחנו דוגמים אותו 4 פעמים שעון אחר שעון, ומחפשים את הרצף "0011". למה המימוש הזה לא נכון? הרעיון כאן היה לסנן ספייקים ולחפש את המעבר היציב של הסיגנל. האם התכוונת למשהו אחר?

במעבר שעונים ישנה הסתברות לבעיית מטאסטביליות. ה- FF הראשון במעבר (start\_trigger\_1) כנראה יכנס למטאסטביליות. אם לוקחים אותו (את start\_trigger\_1) ומשתמשים בו בלוגיקה נוספת (יש כאן לוגיקה התלויה ב- start\_trigger\_1 ו- start\_trigger\_2 ו- start\_trigger\_3 ו- start\_trigger\_4) ולא דגימה נקייה (start\_trigger\_1 נדגם נקי בלי לוגיקה נוספת) אז המוצא של כל התלות ב- 4 הסיגנלים גם יכול להכנס למטאסטביליות. כשדוגמים נקי פעם נוספת אז הסיכוי ש- start\_trigger\_2 יכנס למטאסטביליות הוא נמוך מאוד ואף אפסי. לכן מומלץ להשתמש רק ב- start\_trigger\_2 עם לוגיקה נוספת. במקרה שלכם יספיק start\_trigger\_2 ו- start\_trigger\_3. השימוש ב- start\_trigger\_4 אפשרי אך נראה לי מיותר.

1. ~~שורה 228 - האם ההערה מדוייקת (21)?~~
2. ~~שורה 244 - מומלץ לסיים עם ה- LABEL, כלומר architecture.~~

# mux2

1. אין לי הערות על הקוד.

2. המימוש הוא לא בדיוק MUX. הרי MUX זו יח' אסינכרונית, וגם ל- MUX יש SELECTOR וכאן יש מימוש של PRIORITY. כלומר, אם הראשון שווה ל- '1' אז קח את הכניסה שלו וכך הלאה, ואילו ב- MUX רגיל אין PRIORITY. זה לא בדיוק משנה - זה הקוד שאתם צריכים, רק רציתי לספר קצת על MUX. מימוש גנרי של MUX נמצא בספריית GATES של בארי. ראו mux\_generic.

3. ~~תמיד כדאי לכוון לגנריות ולא לתפור לפרויקט ספציפי, וגם כאן היה כדאי לתת שמות כלליים ל- PORTS ואת הרוחבים לשים ב- GENERIC. לא חובה לתקן.~~

# Manager

1. במסמך כתוב שהיח' קרויה בשם read manager בעוד שב- VHDL היח' קרויה בשם manager. יש ליצור תאימות.
2. ~~שורה 9 – איזה מידע מה- RAM שולחים ל- SDRAM? אולי יותר נכון לנסח טיפה שונה, כי אני חושב שאתם מבקשים מידע מה- SDRAM ולא כותבים אליו באמצעות היח' הזו~~ (בהתאם גם במסמך).
3. ~~שורה 11 – כאן כתוב read manager.~~
4. שורה 39 – רוחב של 13 ביט בעוד במסמך כתוב 14 ביט. יש לעדכן את המסמך (הערה כללית לגבי האי תאימות – יש לעדכן את כל המסמך).
5. ~~שורה 46 – תוסיפו הערה~~ (וגם הפין הזה לא מתואר כלל במסמך).
6. שורות 47-48 – אמורות להיות מוחלפות בהמשך עם ה- WBM? לא. הטיפול ב-WB יהיה בהיררכיה אחת מעל, כך שההיררכיה הזאת מקבלת סיגנלי data ו-valid.
7. ~~שורות 49-50, 52-53 – האם יש בהן צורך?~~
8. ~~שורה 74 - end entity manager~~
9. ~~שורות 82-86 – האם לא כדאי להכניס את המספרים שבגבול הימני ל- generics?~~
10. ~~שורה 87 – להוסיף הערה.~~
11. ~~שורות 94-98 – ראו הערה לגבי שורות 82-86 (אלו אותם generics מומלצים).~~
12. שורה 99 – האם במקום לספור לא יהיה חיווי ע"י סיגנל מתאים? או שזה זמני? הערה זהה לגבי sdram\_wait\_proc. זו ספירה פנימית שלנו מרגע שליחת בקשת קריאה ל-SDRAM עד שניתן לשלוח בקשה חדשה (סיימנו לשלוח את הבקשה, וקיבלנו את ה-data). מה שישתנה זה מספר ה- clk שסופרים.

נראה לי שספירה פנימית, בתוך הקומפוננטה הזו, זה לא דבר הכי טוב. מס' השעונים יכול להשתנות מפנייה לפנייה (כתלות ב- PAGE של ה- SDRAM וכו'). כפתרון זמני זה בסדר. מומלץ להשתמש בקו VALID כלשהו שתלוי מתי מתקבל DATA מה- SDRAM CONTROLLER.

1. ~~שורה 270 – במקום 19 להשתמש בקבוע או generic.~~
2. ~~לגבי המונים – אני לא בטוח שכדאי להשתמש בגבול שמאלי של -1 בגלל סינתזה. אני לא בטוח לגבי זה. אולי עדיף מ- 0 ובגבול הימני שהתכוונתם אז פלוס 1 ממה שהתכוונתם.~~ (עדכנתי את המונים שיתחילו מ-0 והוספתי/הורדתי 1, אבל אני חושבת שיש עכשיו בעיה איתם – צריך לבדוק בסימולציה!)
3. ~~שורה 314 – במקום 9 עדיף להשתמש ב- attribute בשם length. כלומר, ram\_addr\_rd'length.~~
4. ~~שורה 341 - end architecture manager\_rtl~~
5. אני משער שתוסיפו TB וסימולציה בהמשך כי לא ראיתי שקיים ב- SVN.

# הערות כלליות

1. txt\_util - לא קובץ ל- design ולכן כדאי להעבירו לספריית simulation.
2. ~~לא מצאתי קבצי DO (מקרו לקימפול ולקביעת ה- top tb - פקודות vcom ו- vsim).~~

בנוסף: אנחנו רוצים לעבור איתך על מעיין "תוכנית פעולה" לגבי התקדמות הפרוייקט:

התחלנו לעבוד על כתיבת קוד שישווה בין תוצאת פריים המתקבל מהgenereted symbol לבין הפריים שצריך להתקבל כדי לבדוק האם באמת הפריים מתקבל ללא טעויות.

אולי יש לך הצעות או כיווני חשיבה אחרים כך שלא נבזבז זמן יקר/ נעשה טעויות שאפשר להימנע מהן.

בנוסף, אנחנו רוצים לעבור שוב על הWB ולראות איך להתממשק לבארי.

הצעד הבא שהייתי ממליץ עליו הוא לבנות תוכנית בדיקות, ו- TB מתאים שיאפשר את הבדיקה, ברמת הקוד הנוכחי (שלכם, ללא ה- WISHBONE והתממשקות למערכת של בארי). הרי יש פה לא מעט מצבים לבדיקה, כמו:

* סימבולים שונים: צבעים שונים בתוך הסימבול, רק פיקסל אחרון בשורה האחרונה שונה, רק הפיקסל הראשון בשורה הראשונה שונה.
* מידע שמגיע מהתוכנה (ל- RAM) ב- TIMING שונה ביחס להוצאת ה- FRAME הנוכחי.
* הוספה / הורדה של סימבולים במיקומים שונים (אחרון, ראשון, רק אחד ב- FRAME, הרבה ב- FRAME, כולם באותו FRAME).
* הצגה של FRAME שכל קוביה מורכבת מ- SYMBOL שונה.
* אולי להשתמש ב- RANDOM ב- MATLAB כדי לייצר סצינות שקשה לחשוב עליהם.