**שאלות לקראת פגישה 18.1.2012**

* לקראת מצגת אמצע- דרישות מיוחדות למצגת אמצע: מה אנו נדרשים להראות במצגת אמצע? איזו רזולוציה?
  + ב- 09/12/2011 שלחתם מסמך (וגם ב- 13/12/2011 עם סכימת בלוקים כללית), ובו אחד הסעיפים כולל תכנון בלוקים. יש לחזור למסמך הזה, ולהמשיך לעבות את תכנון הבלוקים – ליצור סכימת בלוקים מפורטת, והסבר מפורט על תפקיד ואופן הפעולה של כל בלוק.
  + מצגת אמצע – הייתי רוצה שמצגת האמצע תתקיים לפני תקופת המבחנים (סוף ינואר, תחילת פברואר. נא העריכו האם נוכל לקיים את המצגת במועד). היא צריכה לכלול:
    - את תכנון הבלוקים שלכם: סכמות בלוקים, הסברים בע"פ, אנימציות.
    - השינויים שבוצעו בקוד של הפרויקט של בארי ואלון. ב- BULLETS עם סכימת ה- TOP. ניוון המערכת, ביטול הדחיסה ב- matlab, ביטול הפריסה ב- vhdl, יצירת top design + top tb עם סימולציה הכוללת: כתיבת image לזיכרון החיצוני דרך ה- rx path, הצגת ה- image לצג דרך ה- vesa generator, כתיבה לרגיסטר, בקשת קריאה מרגיסטר וקבלת התשובה מה- tx path (משער שהאחרון עדיין לא ניתן כי המערכת של בארי עדיין לא תומכת בכך). קשיים שהתגלו. הורדת רזולוציה בהתחלה. מה עדיין פתוח.
    - הייתי מאוד רוצה להראות דמו במצגת אמצע של משהו חי ועובד. כנראה, שלא נספיק.
* אנחנו מבינים שנצטרך לממש בקר (מכונת מצבים) שמנהל את כל תהליך הכתיבה והקריאה לזכרון: סדר הפעולות הנכון (כתיבה התחלתית לזכרון, סיבוב כתיבה מחדש, קריאה ע"י ה-VESA). היכן ימוקם? האם ב-TOP של בלוק ה-MEMORY ?

ממליץ (לא חובה, ניתן לשלב ב- memory management כפי שהצעתם במסמך מה- 13.12.2011) על הוספת בלוק TOP חדש של image\_manipulation ולא לשבץ את כל הבלוקים שלכם ב- memory management (ואז ה- arbiter ישוב להיות רק ל- 2 ולא ל- 4, וארביטציה נוספת תהיה ברמת הווישבון ב- interconn. סך-הכול, עדיין תהיה ארביטציה בין 4 פעולות – פעמיים קריאה, ופעמיים כתיבה). יהיו 6 בלוקים עיקריים ב- DESIGN: RX PATH, מערכת ה- Interconns, TX PATH, MEMORY MANAGMENT, DISPLAY TOP, וה- IMAGE MANIPULATION TOP.

* בלוק ה- rx path: אין שינוי מהפרויקט של בארי.
* מערכת ה- Interconns: בארי אמור לשנות את הקוד, ובנוסף, יתכן שתצטרכו לשנות בקטנה (ברמת חיווטים), כי נוסף עוד בלוק עיקרי גדול (image manipulation).
* בלוק ה- tx path: כרגע, הוא לא מוכן מהפרויקט של בארי. ברגע שיהיה מוכן, לא צופה שינויים אצלכם.
* בלוק ה- memory management: בנוסף, לשינוי הקטן שביצעתם, תצטרכו לבצע עוד כמה שינויים קטנים, בעיקר בנושא ה- bank switch. ראו בהמשך הסבר על 4 אזורים שונים: שניים לקריאה, ושניים לכתיבה, וזה אומר שצריך לנהל את הבנקים הללו, שעד עתה נוהלו (רק שני banks) ב- memory management. זה אומר, שכנראה, כן תצטרכו לגעת, שוב, בקוד הזה, כדי לאפשר בחירה / ניהול גנרי יותר של כל ה- bank switch, כלומר, ניהול של מאיזה בנק קוראים, ומאיזה בנק כותבים (בנוכחי יש רק שני בנקים ואצלכם יש 4).
* בלוק ה- display top: כנראה, שלאחר הניוון שלכם, ותיקון ה- pixel manager לא תצטרכו לגעת יותר בקוד הזה.
* בלוק ה- image manipulation: בלוק חדש שיכיל את כל הפונקציונאליות שלכם.
* לגבי התחלת התכן והקידוד: להבנתנו אנחנו צריכים להתחיל את המימוש של המודולים הקטנים, להריץ עליהם test bench ולעלות לרמה הגבוהה יותר. האם זה תהליך נכון? נשמח לטיפים ורעיונות, שכן עד עתה לא יצא לנו לממש רכיבים חדשים.

כן, זהו תהליך נכון – bottom to up, לאחר ש- up to bottom כבר קיים. טיפ מרכזי ל- test bench ברמת כל הבלוק של ה- image manipulation: ברגע שיהיה לכם מודל סימולציה ל- wishbone master ול- wishbone slave, החיים יהיו יותר קלים, כיוון שלא תצטרכו להריץ ברמת top של כל ה- design. יתכן וקיים כאלו מודלים לבארי.

* בשלב עיבוד התמונה מתבצעות קריאות וכתיבות בקצב שונה, עבור כל 4 פיקסלים שאנחנו קוראים נכתב רק אחד לזכרון. בהקשר של בעיות תזמון, איך לתכנן מראש על מנת להימנע מהן מבעוד מועד?

הקריאה של תמונה מסובבת קיימת, שיוצאת לצג, תקבל עדיפות, ולכן, לא יהיה מצב שתמונה לא תוצג כנדרש. הכתיבה של התמונה המסובבת אינה בעלת עדיפות. היא יכולה להמתין (ובינתיים קריאה של תמונה מקורית כדי לחשב את ערכי התמונה המסובבת תיעצר עד אשר יכתבו ערכי הפיקסלים של התמונה המסובבת). זה לא משנה אם כל החישוב והכתיבה של תמונה מסובבת חדשה תהיה שלושה frames (או יותר – סתם מספר), כיוון שבינתיים מוצגת תמונה קודמת מסובבת.

* התייחסות לשיטת התכנות כך שהתכן יהיה סינתזבילי?

היתרון שדיבגתם מערכת עד עתה תוך קריאת לא מעט קוד קיים, ולא צללתם לכתיבת קוד מ- scratch אמור לבוא לידי ביטוי כעת, בכתיבת קוד חדש משלכם. כלומר, ניסיון ודוגמאות מקוד קיים, שימוש ב- coding guidelines ששלחתי (ואם לא שלחתי אז תאמרו לי, ואשלח), ו- code review שאבצע לכם פר בקשה.

* שחרור תמונה לVESA לאחר סיום מניפולציה: עד עתה ה-pixel manager ביצע דרישות קריאה מהזכרון. האם יש צורך להתערב בקצב הדרישה שלו לקריאה מהזכרון, או שיהיה חיווי מהבלוק זכרון כאשר תמונה מוכנה?

ה- pixel manager ימשיך בשלו, בנוגע לבקשות קריאה, כיוון שכל 16.66msec יש להציג תמונה בצג. בתוספת קוד שלכם, צריך לבוא לידי ביטוי, מהיכן קוראים בזיכרון החיצוני. הרי צריך לקרוא מהמקום בו מאוחסנת התמונה המסובבת. מקום זה יכול להשתנות עם הזמן - חלוקה פונקציונאלית ל- 4 אזורים בזיכרון: כתיבת תמונה חדשה שמגיעה מה- host, תמונה מקורית מלאה מאוחסנת, תמונה שכרגע עוברת תהליך של סיבוב ונכתבת, תמונה מלאה מסובבת שנקראת מהזיכרון ומוצגת לצג. האזורים מתחלפים (מין toggle).

* סוגיית הסינוס/קוסינוס – הסוגייה הועלתה במצגת אפיון, עדיין לא הוחלט איך לטפל בה.

אם בחומרה אז טבלאות ROM קבועות. אם בתוכנה – אז התוכנה מחשבת, ושולחת לחומרה את שני הערכים הללו. לי לא משנה מה שתבחרו. ברור שעדיף להתנסות בחומרה, אך מבחינה מערכתית זה לא משנה, כי הזווית משתנה פעם בהרבה זמן, ע"י המשתמש בלבד, אז במקום לשלוח את הזווית, שהיא תשלח כבר את החישוב.

נושאים נוספים:

* תיעוד בקוד – יש לרשום את מהות השינויים שבוצעו, בטבלת השינויים ב- Header של הקובץ.
* ב- SVN לא ראיתי תיעוד של הסימולציות.
* מסמך הפרויקט צריך להתקדם עם הפרויקט. כל שבוע יש עדכון ומסמך שאלות ותשובות, וזה טוב, אך אם המסמך היה מתעדכן בהתאם, אז היה הרבה יותר קל לכולם להבין את המצב הנוכחי.
* ממליץ לנהל טבלת אקסל עם משימות.
* שאלתי אתכם לא אחת על ערך ה- pixels\_req שיוצא מה- vesa\_gen\_cntrl ועדיין לא השבתם בנושא (וגם הקובץ עדיין לא מעודכן לאחר התיקון של בארי). לדעתי, יש שם באג.
* יתכן ולפגישה ביום ד' לא יהיה ערך מוסף, אם לא תבואו עם שאלות חדשות, בעיקר בנוגע לתכנון הבלוקים המפורט.