16.02.2013 - Integration Status

# התמקדות ב- Display Controller

בהתחלה, חיברנו את הבלוק שלנו Symbol\_Generator\_Top בצורה כזאת בתוך disp\_ctrl\_top:

disp\_ctrl\_top

Symbol\_Generator\_Top

VESA\_gen\_ctrl

DC\_FIFO

VSYNC , req\_ln\_trig

Opcode\_Parser

SDRAM\_symbol\_model

Flush

נדרש debug בתוך Symbol\_Generator\_Top, בבלוקים: Manager, mux2

שינוי נוסף שנדרש בבלוק disp\_ctrl\_top הוא הוספת סיגנל שמבצע flush ל-DC\_FIFO לפני תחילת שורה, כדי לנקות את הפיקסלים הנותרים מהשורה הקודמת.

התוצאה הייתה תמונות bmp תקינות במוצא הסימולציה.

בשלב הבא הוספנו רגיסטר ב- disp\_ctrl\_top, שיתממשק ל-Symbol\_Generator\_Top:

disp\_ctrl\_top

Symbol\_Generator\_Top

VESA\_gen\_ctrl

DC\_FIFO

VSYNC , req\_ln\_trig

Opcode\_Parser

SDRAM\_symbol\_model

SG register

WBS

כעת הגישה ל- Symbol\_Generator\_Top היא באמצעות הרגיסטר שמקבל את ה-opcodes, במקום מודל הסימולציה של opcode\_parser.

הכתיבה לרגיסטר היא ב-burst באורך מקסימלי של 900 בתים = 300 בלוקים על המסך X 3 בתים לכל opcode. לכן נדרש שאורך הכתובת של הרגיסטר תהיה 10 ביטים, שתהיה תמיכה במרחב כתובות מעל 900 (כולל את הרגיסטרים הקיימים כבר ב-disp\_ctrl\_top).

בבלוק gen\_reg הוספנו generic שנותן תמיכה במרחב כתובות לרגיסטר (addr\_space\_g), כלומר טווח הכתובות של הרגיסטר הוא מכתובת הבסיס addr\_val\_g ועד כתובת addr\_val\_g+addr\_space\_g-1.

גם ברמות האחרות (זיכרון וה-top) שינינו את ה-generic של אורך כתובת רגיסטר ל-10 ביטים.

מכיוון שהסתבכנו קצת בשינוי הנדרש ב-tx\_path, החלטנו לא לגעת בו, במקום זה ביטלנו באופן זמני את הגישה אליו (ע"י שינוי ערך קבוע ששולט בכתיבות לרגיסטר ה-type בתחילת טרנזקציית UART).

כדי לנווט נכון את הגישה לגיסטר שלנו ב- disp\_ctrl\_top, הוספנו את מרחב הכתובות הנדרש ב- intercon\_pkg.

כעת, עלינו להמיר את קובץ ה-test איתו עבדנו ב-opcode\_parser לקובץ UART כזה שמודל ה- uart\_tx\_gen\_model מצפה לקבל.

הבסיס ליצירת קובץ ה-UART נכתב בפונקציית מטלב uart\_tx\_simulation.m שמייצרת בדיוק את קובץ ה-test הזה בצורת קובץ UART.

שידרוג ה-GUI שלנו לייצור קובץ UART (על בסיס פונקציית המטלב לעיל), כולל address של הרגיסטר שלנו, ה-opcodes בתור payload וחישוב CRC בהתאם לפרוטוקול.

GUI משודרג נמצא ב: Matlab\GUI\SG\_GUI

כמו כן, בוצע debug של שליחת קובץ ה-UART והגעתו ל- Symbol\_Generator\_Top, וקבלת תמונות bmp זהות לאלה שקיבלנו בעזרת ה-opcode\_parser.

השלבים הבאים:

1. לדאוג להפרדה בין chunk ל chunk בקובץ ה UART ע"י שליטה בתזמונים, במודל ה- UART\_TX . כדאי לבצע זאת ע"י שימוש בפענוח הפקודה -t כמו ב opcode parser
2. ממשק זכרון:
   1. להבין איך להגיע מה display controller אל ממשק הזיכרון.
   2. טעינת הסמלים בזיכרון דרך UART.
   3. לנתק את sdram symbol model ולעבוד עם ה sdram של בארי.