תכנון בלוקים

* תהליך זרימת המידע בין תתי הבלוקים
* ממשקים של תתי הבלוקים מוגדרים היטב
* תפקיד כל תת בלוק וכל PORT שלו וכל GENERIC שלו
* מימוש כל בלוק
* צורות גלים אופייניות של ה-CORE וכל תת בלוק.

מידע וטריגר מגיעים מה-signal generator. המידע מגיע ל-WC והטריגר מגיע לרגיסטרים. GENERIC מגדיר את גודל המילה שתתקבל, בכל עליית שעון תיכנס ל-RAM דגימה של הסיגנלים המתאימים ליחידת זמן כלשהי ויישמרו בו כמילה (מתואר כשורה בטבלה למטה). האותות בעליית השעון הבאה יישמרו כמילה הבאה, וכך הלאה. המידע ייקרא בצורה טורית כאשר דגימות של כל אות מופיעות אחת אחרי השנייה בסדר כרונולוגי.

ציר הזמן

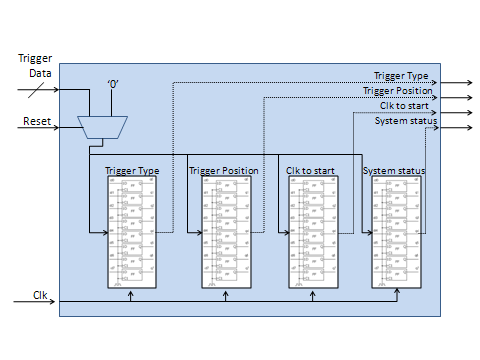
קריאה

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S4 | S3 | S2 | S1 | S0 | T=0 |
| S4 | S3 | S2 | S1 | S0 | T=1 |
| S4 | S3 | S2 | S1 | S0 | T=2 |
| S4 | S3 | S2 | S1 | S0 | T=3 |
| S4 | S3 | S2 | S1 | S0 | T=4 |

כתיבה

לאחר זיהוי עליית טריגר על ידי ה-WC מתבצע חישוב של לגבי המידע שנרצה לשלוח החוצה וכתובת תחילת המידע וסוף המידע מועברים ל-RC שאחראי על שליפת המידע מה-RAM בהתאם לכתובות שהתקבלו ומוציא אותו למשתמש.

סכימת רגיסטרים:



הכניסות הן 8 סיביות לכל רגיסטר, כלומר רוחב הכניסה- .

Read Controller

מקבל כתובת התחלה וכתובת סיום (מגיעות מ WC) של המידע שנמצא ב RAM אותו אנו רוצים לשלוח למשתמש. נבצע איטרציות עד אשר נוציא את כל המידע, בכל איטרציה נקבל כתובת התחלה ונקרא "מילה" מה RAM ונוציא אותה דרך WBM. גודל המידע שנקרא בכל איטרציה יהיה לכל היותר כגודל המילה שאנו יכולים לשלוח דרך ה WBM.

סכימת RC:



הערה:

כרגע ברכיב זה אנו מקבלים את כתובת ההתחלה וכתובת הסיום של המידע המוקלט ב RAM אותו אנו רוצים להוציא למשתמש, ובגלל שבכל מקרה אנו לא יכולים להוציא דרך ה WBM יותר מידע מאשר מילה יחידה (זהו בעצם צוואר הבקבוק שלנו ) אנו קוראים מה RAM כל פעם מילה אחת כזו, שולחים אותה החוצה ומחשבים ב ALU את הכתובת של המילה הבאה אותה אנו רוצים להוציא.

אפשרות שנייה תהיה לקבל בהתחלה את כתובת ההתחלה והסיום של המידע אותו אנו רוצים להוציא למשתמש, להעביר את כולו ל RC ובעצם לחלק אותו שם כאשר שוב בכל איטרציה אנו מוציאים מילה בודדת.

השתמשנו בשיטה הראשונה כיוון שנראה לנו שזוהי כפילות מידע להעביר את כל המידע הנחוץ על ההתחלה ל RC, השיטה השנייה אולי פשוטה יותר ליישום אך נראה לנו שבעזרת חישובים יחסית פשוטים, ניתן להשתמש בשיטה הראשונה ולא ליצור כפילויות במידע ולהקצות משאבים נוספים.

שאלות:

* הערה ראשונה, נשמח להתייחסות
* הגדרנו רוחב אותות wc\_to\_rc, Rc\_to\_RAM (כתובות ב RAM ) בתור Add\_width\_g, למרות שזהו בעצם רוחב כתובת של פרוטוקול ה WB, האם צריך לשנות או שכתובת ב RAM תהיה זהה לכתובת בפרוטוקול ה WB