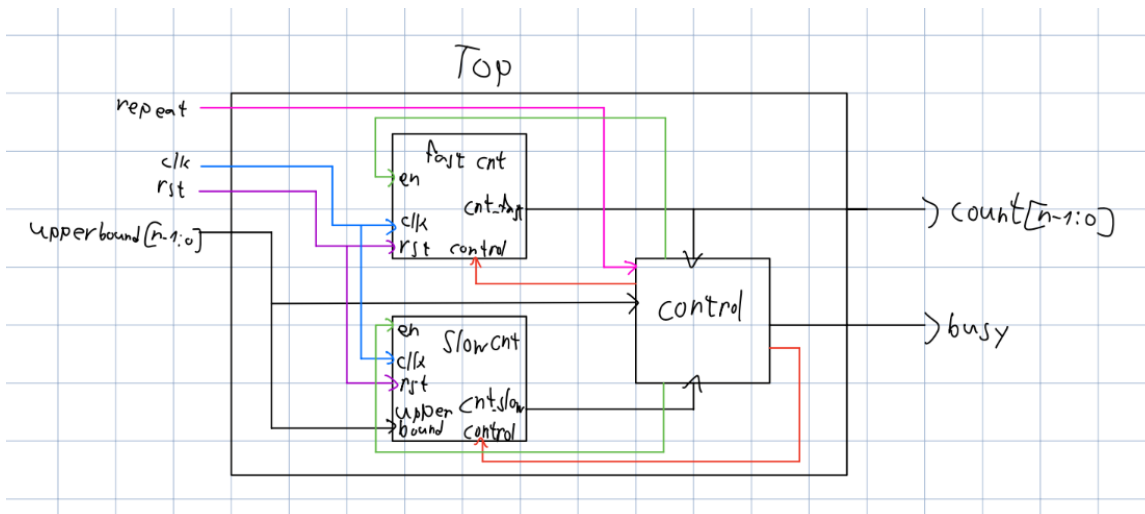


Preparation Report – Lab 2

Roy Shani (315995258), Yanai Maison (206701591) | 23.04.2025

Overall Design Layout



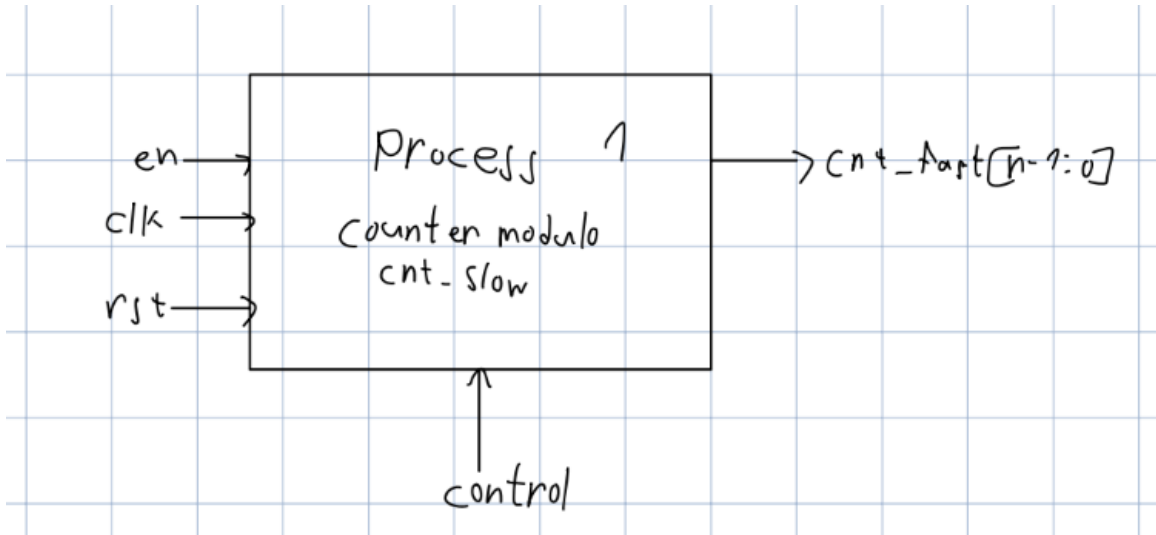
מצורף שרטוט של כלל המערכת ממבט על, כאשר נפרט על כל process וכיצד הוא עובד.

TOP.VHD

רכיב ה-top הוא הרכיב העליון של המערכת, ומכיל בתוכו 3 processes:

fast counter, slow counter, control,

PROCESS 1 – COUNTER MODULO SLOW COUNTER



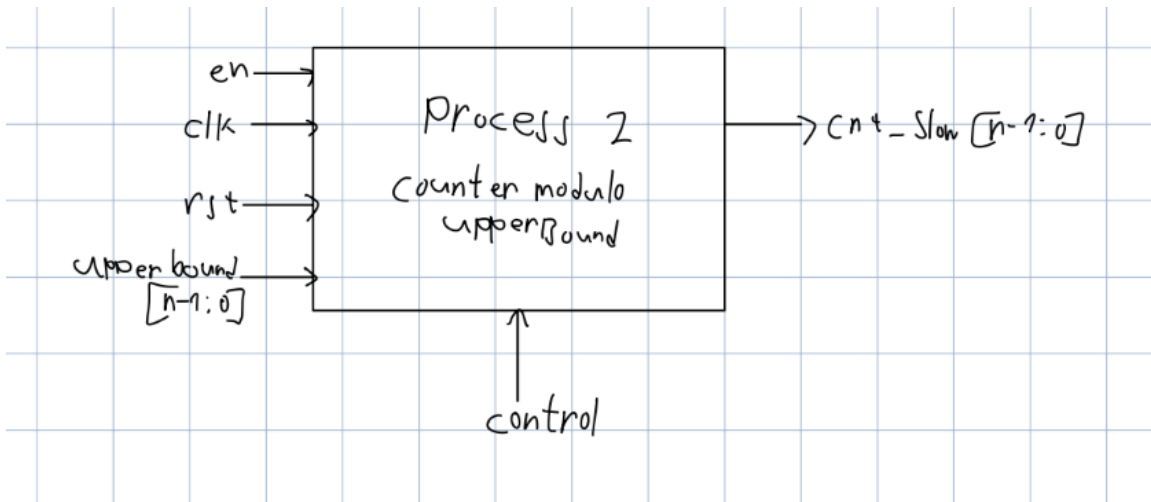
התהליך הראשון משמש לביצוע מנייה עולה לפי ערך עליון שמוגדר על ידי המונה האיטי.

התהליך מקבל 4 כניסות: en המציינת האם אנו במצב תקין למנייה לפי המצב של תהליך הבקרה, clk , כניסת שעון, rst כניסת איפוס חיצונית, כאשר היא מתקבלת המנייה מתחילה מחדש, $control$ זהו וקטור המכיל את המידע על הערך העליון למנייה. התהליך מוציא וקטור בודד באורך N שמכיל את ערך המנייה הנוכחי.

התהליך עובד באופן הבא:

לתהליך יש חלק אסינכרוני בו כאשר מתקבל אות rst המונה מתאפס, ותהליך סינכרוני שקורה בכל פעם שהתהליך דוגם עליית שעון. כאשר מתקבלת עליית שעון אם אנחנו במצב מניה כלומר אות en ב'1' לוגי, אנחנו בודקים האם המונה שווה לקונטרול (כלומר לערך המנייה העליון המתקבל מהמונה האיטי) ואם כן מאפסים את המונה, אם לא אז נעלה את המונה ב'1'. במצב בו אנחנו לא במצב מנייה כלומר en הוא '0' לוגי נשאר במצב המונה הנוכחי.

PROCESS 2 – COUNTER MODULO UPPERBOUND



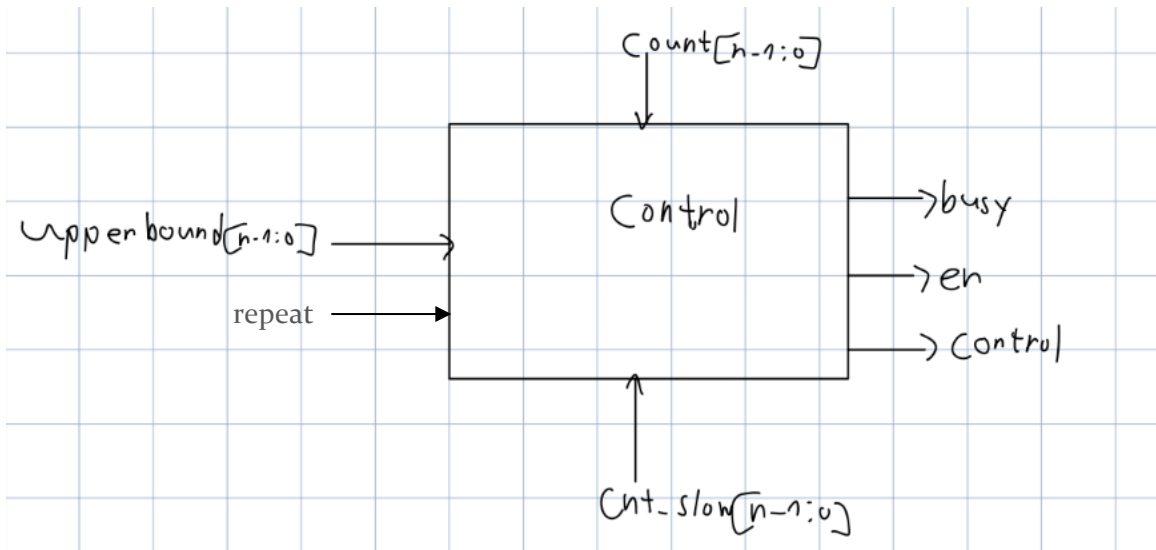
התהליך השני משמש לביצוע מנייה עולה לפי ערך עליון שמוגדר בכניסה לTOP.

התהליך מקבל 5 כניסות: en המציינת האם אנו במצב תקין למנייה לפי המצב של תהליך הבקרה, clk , כניסת שעון, rst כניסת איפוס חיצונית כאשר היא מתקבלת המנייה מתחילה מחדש, וקטור $upperbound$ מכיל את הערך העליון אליו המנייה יכולה להגיע לפני התחלה מחדש, $control$ הוא ערך בוליאני המכיל את המידע האם המונה המהיר הגיע לערך המנייה האיטי – נקרא בקוד $fast_done_r$ וגם מעביר את קו $repeat$ לתהליך. התהליך מוציא וקטור בודד באורך N שמכיל את ערך המנייה האיטי שהוא גם הגבול של המונה המהיר.

התהליך עובד באופן הבא:

לתהליך יש חלק אסינכרוני בו כאשר מתקבל אות rst המונה מתאפס, ותהליך סינכרוני שקורה בכל פעם שהתהליך דוגם עליית שעון. כאשר מתקבלת עליית שעון וגם המונה המהיר שווה למונה האיטי וגם אנחנו במצב מניה כלומר אות hch ב'1' לוגי וגם המונה האיטי קטן מהגבול העליון, נעלה את המונה האיטי ב1. אם אות הריפיט ב'1' לוגי והמונה האיטי שווה למונה החיצוני וגם נקבל מהקונטרול שהמונה המהיר שווה לאיטי, נאפס את המונה האיטי. אם הריפיט ב'0' לוגי המונה יגדל עד לגבול העליון ויעצור במצבו הסופי. אם אות hch ב'0' לוגי נעצור גם כן את המנייה.

PROCESS₃ – CONTROL UNIT



תהליך 3 הוא יחידת הבקרה, מטרתה לסנכרן בין שני התהליכים ולהכריע האם התהליך בפעולה או לא.

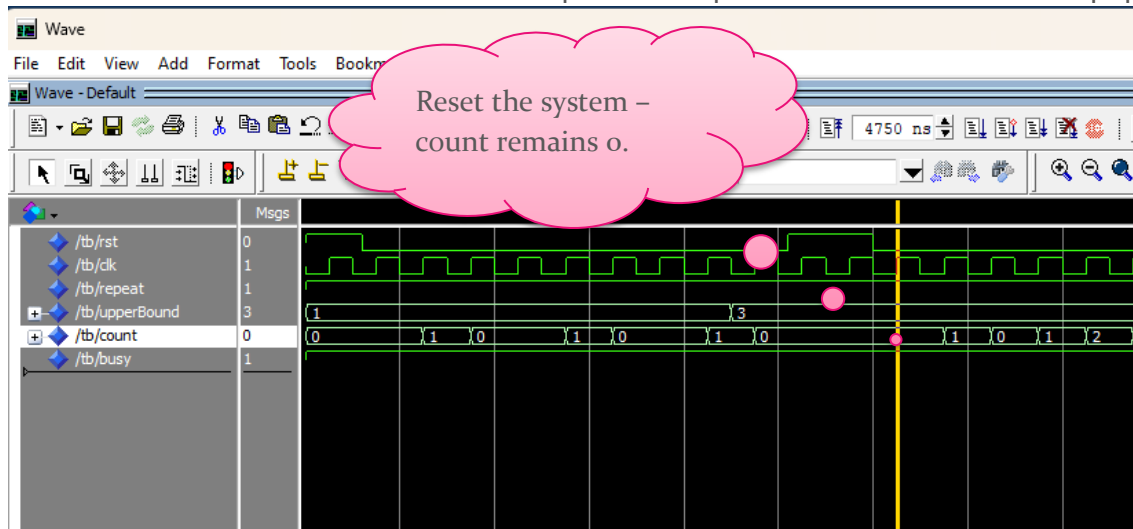
התהליך מקבל 4 אותות כניסה: `count` שהוא המונה המהיר, `Upperbound` שהוא החסם העליון למנייה, `cnt_slow` שמכיל את המניה האיטית ו`repeat` שמודיע האם לאחר סיום המנייה יש להמשיך למנייה נוספת מהתחלה.

לתהליך 3 אותות מוצא: `busy` שמודיע האם המערכת במנייה, `en` שמודיע לשני התהליכים האחרים שעליהם להמשיך במנייה, `control` שפירטנו בתהליכים הקודמים

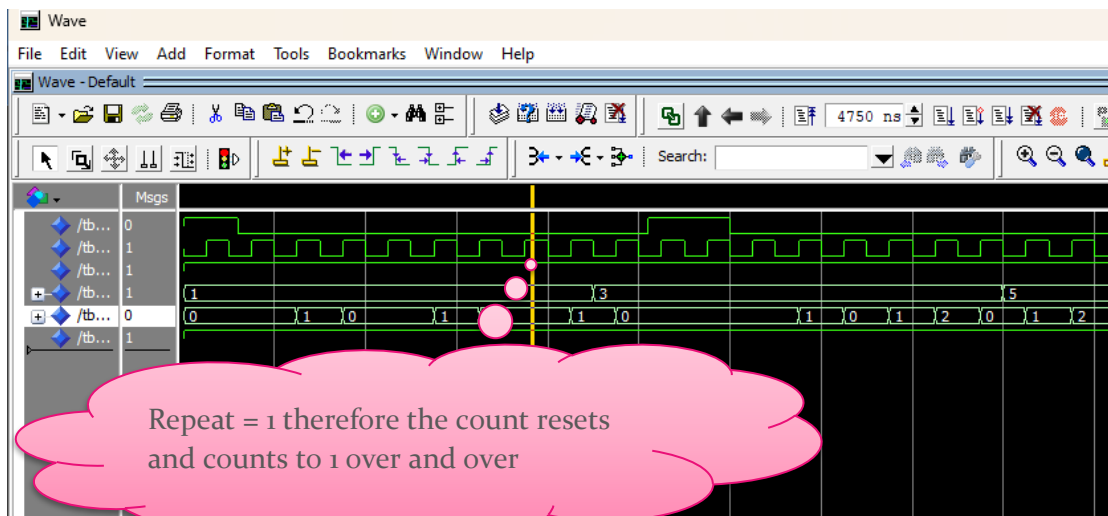
התהליך עובד בצורה בה הוא מחשב בצורה מקבילית מספר פרמטרים שמשמשים אותנו במנייה האיטית והמהירה: `fast_done_r` אות בקרה בוליאני פנימי בהתאם לשוויון בין המונה המהיר ולאיתי, `count_o` השמה של המונה המהיר החוצה, `busy` מתעדכן בהתאם לערך `en`, `en` אשר מתאפס כאשר ערך המונה האיטי גדול מחסם העליון וגם המונה המהיר שווה לאיטי וגם `repeat` התאפס, כל שאר הזמן הוא '1' לוגי כלומר אומר לתהליכים להמשיך למנות.

Simulation results

קובץ ה-TB שכתבנו מבצע מספר בדיקות לווידוא תקינות המערכת:



בבדיקה הראשונה נוודא שאכן reset מאפס את המערכת וגורם לספירה חוזרת מאפס.



בבדיקה השנייה נשאיר את repeat על ערך 1 ונצפה שהספירה תתאפס בכל פעם שתגיע לערך upperbound הרלוונטי.

