

Digital computer structure – final lab 2 report

במטלת זמן אמת של מעבדה זו התבקשנו להוסיף מצב נוסף למערכת שלנו, state4, באופן דומה למצבים הקודמים בכך שאנו נעשה שימוש בטיימרים על מנת לתזמן במדויק אירועים בבקר ולממש את הלוגיקה שלנו בהסתמך על כך.

דרישת המטלה הינה לחבר את הpush button הנותר ל2.0p ובכך לתת טריגר למעבר לstate4, דבר זה יגרור הוצאת אות PWM מרגל 2.2p בתדר קבוע של 1khz, כאשר את הduty cycle של האות אנו משנים כל שנייה בין שלושת הערכים 25%, 50%, 75%, כעת נסביר כיצד מימשנו את הלוגיקה של המצב באמצעות הטיימרים שלמדנו במטלת הדוח המכין למעבדה זו.

השתמשנו בtimer A0 בסט הקונפיגורציות הבאות:

- $Smclk - 1Mhz = 2^{20}$
- Up mode
- Compare mode – compare to $0x100 = 2^8$

בסיטואציה זו אנו מקבלים אינטרפט $2^{20} \cdot 2^8 = 2^{28} = 4k$ פעמים בשנייה, כלומר כל רבע מחזור של האות, יש לנו משתנה בשם second_counter אשר אנו בכל כניסה לרוטינה מעדכנים אותו ומוסיפים לו 1, עד להגעה לערך של 4k ואז אנחנו מאפסים אותו, כלומר, תחת ההגדרות הללו המשתנה הנ"ל מתאפס כל שנייה בדיוק, כאשר הוא מגיע לערך המקסימלי אנו מעדכנים את ערך המשתנה duty cycle באופן הבא:

$$duty_cycle = (duty_cycle) \% 3 + 1$$

וכך למעשה מממשים שינוי נכון של הערך שלו כל שנייה, ועל מנת באמת להוציא את האות הריבועי אז אנו בכל כניסה לרוטינה מסתכלים על הגודל של $second_counter \% 4$, אשר כאמור הוא משתנה כל $1/4k$ שניות ולכן למעשה משלים מחזור (0,1,2,3) כל 1ms, שזה בדיוק זמן המחזור של הגל שלנו, כל פעם אנו נסתכל על הערך שלו בהשוואה לduty_cycle ואנו נוציא 1 במוצא רק אם הערך של המשתנה נמוך משל duty_cycle ואחרת 0.

כלומר, לדוגמה אם ניקח $duty_cycle = 3$, המשתנה second_counter יהיה קטן ממנו 3 פעמים במחזור (בערכים 0,1,2), ולא קטן ממנו פעם אחת בערך 3, ולכן הגל שנוציא יהיה בעל $duty_cycle = 75\%$ כפי שאנו רוצים, כנ"ל עבור שאר הערכים של duty_cycle – 1,2.

לכן באופן זה על ידי שימוש בטיימרים שלמדנו ובקונפיגורציות הפנימיות שלהם (תנאים לקבלת פסיקה, הפעלת פסיקות, מצב ספירה של הטיימר, שעון פנימי של הטיימר) על ידי לבצע פעולות כלשהן לאחר כמות זמן מדויקת ובכך לממש את הנדרש.

יעקב מסילתי 205671852

איילון קפל 207807025