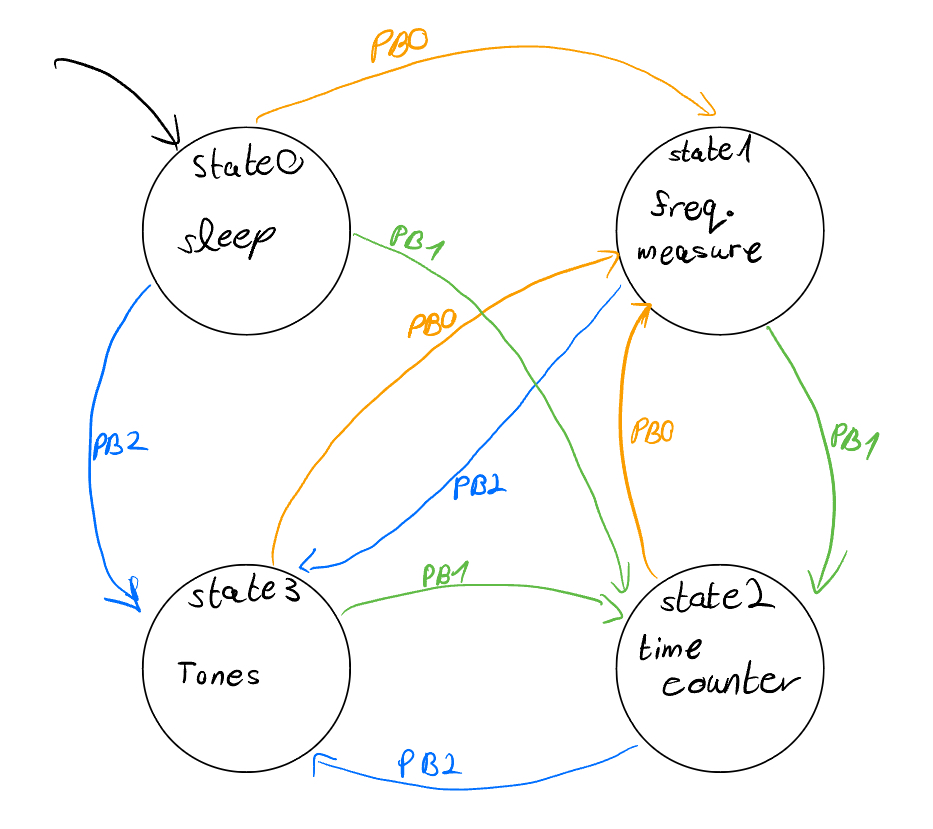
Preparation Report

Lab 2 – Timers ADC & DAC

ירדן לוי 206212276

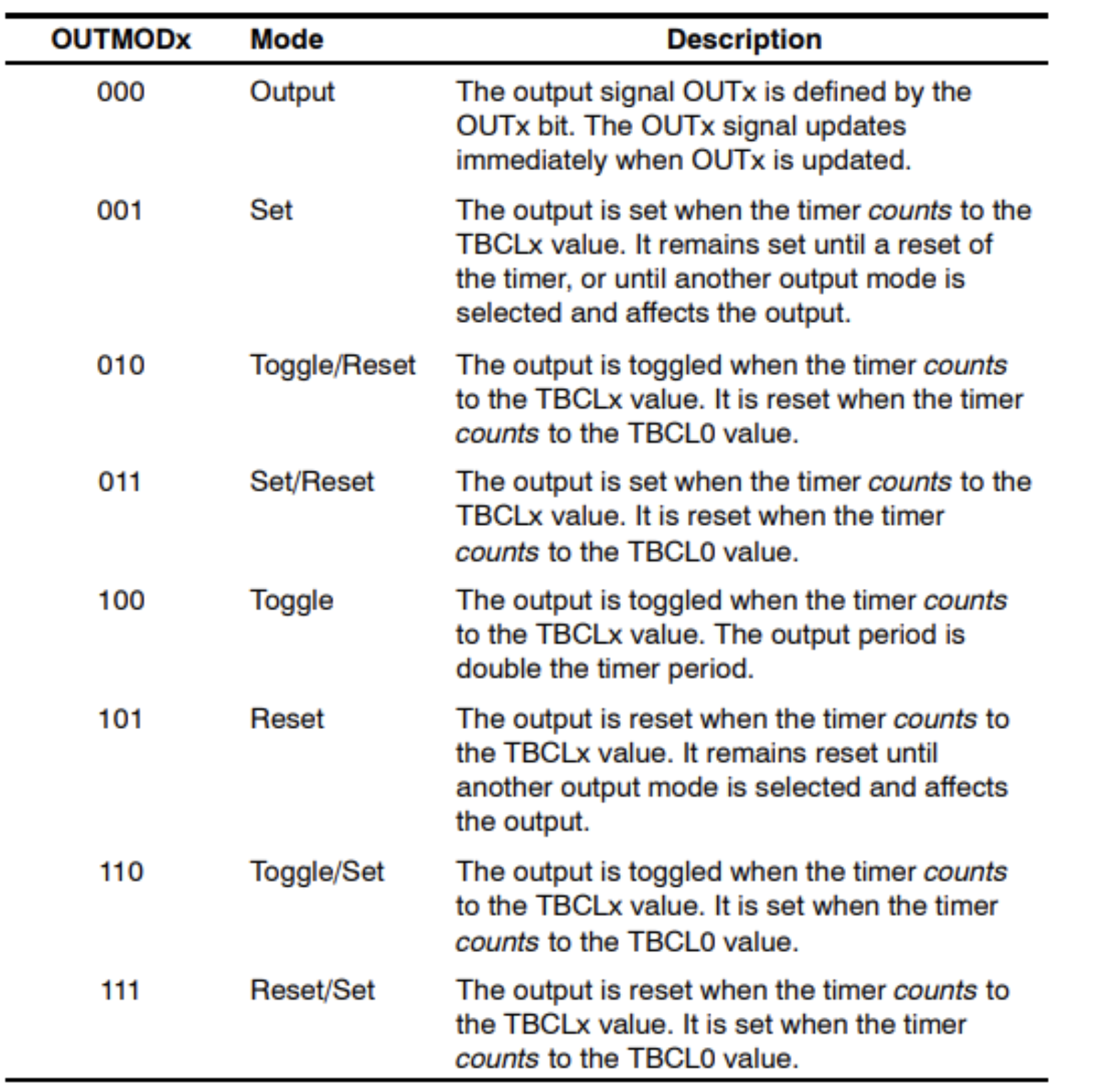
נחמן מימון 321730558

# דיאגרמת מצבים



# שאלות תיאורטיות

## TIMERS

1. **הסבר את ההבדל בייעוד של הטיימרים Basic Timer1, TimerB**Basic Timer1: מספק תזמון ל-LCD ומספק תזמונים עם תדר נמוך, יכול לשמש עבור פונקציות RTC. בעל 2 טיימרים של 8 ביט ויכול לפעול עם שניהם או במוד 16 ביט מאוחד.  
   Timer B: זהו טיימר בעל שימושים נרחבים יותר ומיועד לשימוש של פעולות מורכבות יותר. ניתן לקנפוג לכמה מצבי עבודה כמו Capture/Compare, ובנוסף מסוגל לייצר PWM ולהעלות פסיקה – מה שחשוב לצורך יצירת פסיקות בזמנים מדודים עבור שימושים הדורשים זאת.
2. **מה משמעות ומטרת השימוש של אופן Capture ו- Compare**Capture: בנמב זה הטיימר ״לוכד״ את ערכו כאשר מגיעה פסיקה מרגל ספציפית המחוברת לביט של הטיימר. את הערך הוא שומר ברגיסטר ייעודי ומעלה דגל המודיע שהתבצעה לכידה של הערך. מצב זה שימושי למדידת רוחב פולס מסויים, לדוגמה, או למדידת זמן ריצה של התוכנית.  
   Compare: כשהטיימר מגיע לערך מסוים הוא יכול להוציא PWM או להעלות פסיקה. מצב זה טוב לצורך תזמון פסיקה במרווחים מדודים וידועים מראש (ולהוצאת אות PWM על כל שימושיו)
3. **הסבר מהי מטרת יחידת output unit ומה הייתרון שלה?**לכל אחד משני המצבים של capture/compare יש output unit. יחידה זו מוציאה גל בהתאם לפסיקה שהועלתה מאחד משני המצבים. ניתן לקנפג את הגל באופני עבודה שונים. היתרון של יחידה זו היא שבאמצעותה ניתן לשדר גל ריבועי עם פרמטרים הנקבעים על ידינו בהתאם לצרכי המערכת, בקלות יחסית.
4. **מנה 3 שיטות למימוש מונה תדר של שעון חיצוני המחובר למעבד. עבור כל שיטה, רשום מהו זמן הריענון (זמן מינימלי בין חישובים עוקבים)**שיטה 1 – נחבר את השעון החיצוני לרגל כלשהי המקונפגת כinput. נצא לפסיקה בכל עליית שעון ונקדם מונה ב-1. במקביל, נקנפג את הטיימר הפנימי שייצא לפסיקה לאחר פרק זמן של שניה אחת. לאחר שעבר הזמן, ויצאנו לפסיקה עקב המונה הפנימי, ערך הטיימר הוא בדיוק תדר השעון החיצוני ע״פ הגדרה (מספר מחזורי שעון לשנייה אחת).  
     
   שיטה 2 – נחבר את השעון החיצוני לכניסת TBCLK של TimerB . נקנפג אותו כך שבעליית שעון של השעון החיצוני, ערך הרגיסטר הרגיסטר המונה (TBR) יתעדכן ל-1 ותחל ספירה. במקביל, עם טיימר אחר (basic timer לדוג׳) נצא לפסיקה לאחר שנייה אחת ולאחר מכן נבדוק מה הערך של TBR ונחסר מהערך ההתחלתי שלו, ונקבל את מספר המחזורים לשנייה של השעון החיצוני.  
     
   שיטה 3 – capture mode , נשתמש במצב זה ונחבר את השעון החיצוני לכניסה ונקנפג שיתבצע capture כל עליית שעון (מספיקות לנו 2 עליות). נחסר את הערך ההתחלתי של הטיימר מערכו הסופי וניתן יהיה לחשב כאשר Count הוא ערך החיסור.
5. **כאשר TimerB מוזן ע״י ACLK (32768Hz) והוא מקונפג ל- Compare מהו ערך הרגיסטר TBCCR0 לצורך אפשור פסיקה פעם בשנייה?**לטובת פסיקה של פעם בשנייה עם ACLK נרצה שלאחר 32768 עליות שעון נקבל פסיקה, כלומר, ערך הרגיסטר TBR ייצא לפסיקת compare כאשר יגיע לערך   
   ולכן נקנפג:
6. **הסבר את סוגי הפסיקות במודול זה ומה מטרת כל פסיקה**  
   ניתן לצאת מפסיקה ל-2 וקטורים:  
   TimerB0 – וקטור זה מעורר ע״י דגל CCIFG של TBCCL0 והוא בעדיפות הגבוהה יותר.  
   TimerB1 – וקטור זה מעורר ע״י דגלי CCIFG של TBCCR1-6. כמו כן מעורר ע״י דגל TBIFG הקשור לפסיקת הטיימר מרגיסטר TBR.
7. **הסבר את סוגי אותות PWM אותם ניתן להפיק במוצא הבקר ומה ההבדל ביניהם**mode0 – OUT bit value: כאשר יש התאמה בין ערך הטיימר לערך ההשוואה, המצב לא משנה את המוצא אלא משאיר אותו בערך שהוגדר בביט OUT  
     
   mode1 – SET: כאשר יש התאמה בין ערך הטיימר לערך ההשוואה, המוצא נקבע ל-1.  
     
   mode2 – Toggle/Reset: כאשר יש התאמה בין ערך הטיימר לערך ההשוואה, המוצא מתהפך (1🡨0 / 0🡨1). אם הטיימר במצב עולה-יורד, המוצא ייתאפס כאשר הטיימר חוזר לערך הבסיסי שלו.  
     
   mode3 – Set/Reset: כאשר יש התאמה בין ערך הטיימר לערך ההשוואה, אם הטיימר במצב עולה אז המוצא נקבע ל-1, אם הטיימר במצב עולה-יורד המוצא יחזור ל-1 בירידה חזרה למצב הבסיסי שלו.  
     
   mode4 – Toggle: כאשר יש התאמה בין ערך הטיימר לערך ההשוואה, ערך המוצא מתהפך (1🡨0 / 0🡨1).  
     
   mode5 – Reset: (1🡨0 / 0🡨1), המוצא מתאפס (עובר ל-0).  
     
   mode6 – Toggle/Set: כאשר יש התאמה בין ערך הטיימר לערך ההשוואה, אם הטיימר במצב עולה – המוצא מתהפך, ואם הטיימר במצב עולה-יורד המוצא נקבע ל-1 כאשר הטיימר חוזר לערך הבסיסי שלו.  
     
   mode7 – Reset/Set: כאשר יש התאמה בין ערך הטיימר לערך ההשוואה במצב עולה , המוצא מתאפס. אם הטיימר במצב עולה-יורד, המוצא נקבע ל-1 כאשר הטיימר חוזר לערך הבסיסי שלו.  
     
     
     
   

## ADC12

1. **מהם מקורות שעון ההמרה ADC12CLK ומה הצורך בכולם?**  
   יש ארבעה מקורות ל-ADC12CLK: ADC12OSC, ACLK, MCLK, SMCLK  
   לכל מקור תדר עבודה שונה, לכן יש לנו מנעד רחב של אפשרויות עבודה עם תדרים שונים לצורך ביצוע הדגימות. כל מקור קובע את זמן הדגימה וזמן ההמרה של הדגימה.
2. **פרט והסבר בקצרה את השיטות לקביעת מרחק בין הדגימות של מודול ADC12**מקור השעון ADC12CLK הוא התדר הפוטנציאלי של ביצוע הדגימות, אותו ניתן לחלק עד יחס של 1:8 (חלוקה ב-8). בנוסף, אנחנו יכולים לקבוע את זמן הדגימה ע״י האות SHI שגם לו ארבעה מקורות.. גם את תדר הדגימה אנחנו יכולים לחלק ע״י הביטים SHT0,SHT1.
3. **הסבר בקצרה את 4 אופני העבודה של מודול ADC12, רשום דוגמה על הצורך בשימוש בכל אחד מהאופנים  
   דגימה בודדת –** תתבצע דגימה רק כאשר נעלה ביט בתוכנה או ע״י הפסקת יחידת הדגימה והפעלתה. נשתמש כאשר נקרצה לדגום ערך בודד ללא תלות בזמן שעון קבוע.  
   **דגימה מחזורית –** נדגום את ערך הערוץ במחזור קבוע שנקבע על ידינו באמצעות בחירת מקור השעון וקנפוג מול מודול הדגימה. עלינו לעבד את הערך הנגדם לפני שתגיע דגימה נוספת ותדרוס אותו.  
   **דגימה בודדת באופן טורי -**  בעת קבלת טריגר לביצוע דגימה, מיד עם סיום ההמרה של הערך הראשון ושמירתו ברגיסטר, יידגמו ערכים נוספים באופן טורי (אחד אחרי השני) וישמרו ברגיסטרים שונים.  
   **דגימה מחזורית באופן טורי -**  בעת קבלת טריגר לביצוע הדגימה, הדגימה מתבצעת באופן טורי, כלומר מספר ערכים נדגמים אחד אחרי השני ונשמרים ברגיסטרים שונים. בסוך הדגימה האחרונה בקבלת טריגר נוסף, תחל סדרת דגימות נוספת נוספת ותדרוס את הדגימות הקודמות.
4. **הסבר את העיקרון והיתרון של שימוש ברגיסטר ADC12IV ומה התשלום בזמן הריצה ללא השימוש בו?**כאשר ערך נדגם ומומר, הוא נשמר באחד מרגיסטרי הזכרון המיועדים לאחסון הערכים. כאשר נכתב ערך לזכרון מסוים, ADC12IV מקבל ערך מסויים שבעזרתו, בווקטור הפסיקה של המודול, נוכל להוסיף ערך זה ל-PC כך שנגיע בדיוק לקטע קוד המטפל בערך השמור באותו רגיסטר זכרון מסויים. החסכון בזמן הריצה מתבטא בכך שאנחנו לא צריכים לבדוק באמצעות קוד איזה רגיסטר התעדכן וקיבל את הערך החדש, אלא נוכל לקפוץ (באופן ״עיוור״) ישר לקוד המתאים לטיפול בערך החדש הנשמר ברגיסטר המסויים.
5. **הסבר את המושג data format במודול ADC12 והצורך בשימוש בו**data format זהו אופן הייצוג של המידע המומר בDAC. ישנן 2 אפשרויות לייצוג:  
   ייצוג בינארי לא מסומן – הערך היוצא הוא אי-שלילי  
   ייצוג בינארי בשיטת המשלים ל-2 – הערך מקבל ערכים שליליים או חיוביים כל שניתן לייצג מתח בעל ערך שלילי.   
   נשתמש בייצוג בינארי אם אין חשיבות לערכים שליליים, אלא יש רק חשיבות להבדל בין הדגימות השונות.
6. **הסבר מהי רזולוציה עבור מודול DAC12 ואיך קובעים אותה**רזולוציה עבור מודול DAC12 היא מספר הביטים הניתנים לייצוג אות דיגיטלי באופן אנלוגי. יש אפשרויות לייצור ע״י 8/12 ביטים, ככל שמספר הביטים המייצגים גבוה יותר, הרזולוציה יותר טובה. את הרזולוציה קובעים ע״י הביט DAC12RES ברגיסטר DAC12CTL.
7. **הסבר את המושג self-calibration ומתי תרצה להשתמש בו?**לעיתים האות האנלוגי במוצא מה-DAC12 לא תואם לערכים המומרים על ידו, מאחר וקיים היסט. על כן, באפשרותנו לכייל את הערכים כך שיתאימו לתוצאה הרצויה. נשתמש בכיול כאשר הוא קריטי למשימה שלנו. לדוגמה, אם יש היסט חיובי ואנו רוצים לייצג ערך מקסימלי, הוא לא ייתקבל במוצא האות האנלוגי מאחר והוא עבר את מתח הרפרנס המקסימלי.
8. **הסבר את ההבדל העקרוני בין המודולים ADC10, ADC12**ההבדל העקרוני הוא קודם כל בכך ש-ADC10 תומך ברזולוציה של עד 10 ביטים בעוד ש ADC12 תומך עד 12 ביטים. ההבדל השני הוא שADC12 מאפשר תמיכה בדגימה מחזורית טורית של ערכים ושמירתם ברגיסטרים יעודיים, בעוד שADC10 מסוגל לשמור ערכים אלו רק בבלוקים הנמצאים בזיכרון הראשי.