

מעבדה מספר 2 במבנה מחשבים ספרתיים

D. שאלות הכנה תיאורטיות:

1. הטיימר basic timer1 הינו טיימר בסיסי, כלומר, היכולות שלו הינן יכולות הבסיס של טיימר של בקר שהן יצירת פסיקות באופן מחזורי כאשר אנו עובדים בתדר מסוים שאנו בוחרים, זאת לעומת הטיימר timerB שהוא בעל יכולות מורחבות הניתנות לתכנות ושימוש על ידינו, לדוגמה עבודה במספר מצבים שונים של מנייה, אפשר מספר פסיקות בתנאים שונים, מספר ערוצים הפועלים במקביל ובכל אחד מהם ישנן הגדרות בלתי תלויות ועוד.
2. במצב עבודה של compare אנו מגדירים ערך מסוים שאנו רוצים לקבל משוב כאשר אנו מגיעים עליו בטיימר, משוב זה מגיע בצורה של פסיקה ובה ניתן לממש לוגיקה בהתאם. זאת בניגוד למצב עבודה של capture שבו אנו מקבלים פסיקה על בסיס תנאי שמתקיים עבור אות חיצוני (לדוגמה שעון שמוזן בצורה חיצונית למערכת ואנו דוגמים עליות שעון שלו) ובפסיקה זו אנו גם דוגמים את ערך הטיימר באותו הרגע ובכך ניתן לדעת לדוגמה כמה זמן עבר בין עליות שעון של האות החיצוני.
3. יחידת output unit הינה יחידה הנמצאת בטיימר timerB, יחידה זו יכולה להוציא את PWM במספר מצבי עבודה שונים, היא מועילה לנו בכך שאנו יכולים לשדר מידע בצורה המוכרת של אות ריבועי כאשר השידור יכול להיות תלוי בתנאים שונים כתלות במצב העבודה.
4. שיטה ראשונה:
בשיטה זו אנו משתמשים בטיימר הבסיסי ביותר על מנת לספור חלון זמן של שנייה, בזמן הזה אנו מונים עליות שעון של הכניסה ולכן בסוף השנייה המספר שמנינו אליו מייצג את התדר (מספר מחזורים בשנייה), במקרה הזה הזמן בין חישובים עוקבים הינו שנייה
שיטה שנייה:
בשיטה זו אנו משתמשים בטיימר המתקדם timerB במצב capture input ובצורה זו אנו יכולים למדוד כמה ערכי המנייה הפנימיים של הטיימר השתנו בין עליות שעון עוקבות של הכניסה החיצונית, הכוונה בכך היא שנמנה את הערכים שאליהם הטיימר מונה ובנוסף לכך נמנה את מספר הפעמים שהיה overflow בזמן זה ובכך, בהתחשב בכך שאנו יכולים לדעת (ולתכנת) את תדר השעון שבו עובד הטיימר, נוכל לחשב את תדר השעון של הכניסה. במקרה זה זמן הרענון בין שני חישובים עוקבים הינו זמן מחזור של הכניסה החיצונית, פוטנציאלית הרבה יותר מהיר משיטה הראשונה
שיטה שלישית:
בשיטה זו אנו נשתמש בשני הטיימרים מהשיטות הקודמות באופן הבא – אנו נפתח חלון זמן של שנייה בטיימר הבסיסי ובזמן הזה אנו נדגום במצב עבודה capture עם timerB גם את אות הכניסה וגם את הטיימר הבסיסי, עבור אות הכניסה נוכל לספור עליות שעון ועל ידי כך שאנו עובדים במצב capture על הטיימר הבסיסי אנו יכולים לדעת מתי התחילה והסתיימה השנייה על ידי דגימת עליה וירידה של השעון

5. על מנת לייצר פסיקה פעם בשנייה, אנו נרצה שיעברו 32768 מחזורי שעון לפני הפסיקה (כמספר התדר), מספר זה הינו למעשה 2^{15} ולכן הקנפוג בבסיס הקסדצימלי של ערך הרגיסטר TBCCR0 הינו 0x8000.

6. פסיקות יכולות להיות פסיקות למטרת פונקציונליות או למטרת דיווח על שגיאה, לדוגמה במקרה הראשון אנו משתמשים בפסיקות של טיימרים כדי לדעת מתי עברה כמות זמן מסוימת\מתי התרחש מאורע כלשהו\טריגר לפעולה כלשהי לדוגמה ליצור אות PWM ועוד. לעומת זאת, פסיקות למטרת דיווח על שגיאה יכולות להיות לדוגמה על ידי watchdog timer, טיימר זה קיים כדי לבדוק האם פעולה מסוימת נמשכת יותר זמן ממה שהיא אמורה, כלומר הוא סופר עד איזה שהוא ערך מוגדר ובהגעה לערך זה ההנחה היא שבהכרח אם זאת כמות הזמן שעברה לפני סיום ביצוע הפעולה, בהכרח יש שגיאה ונוכל במקרה זה לגשת לרוטינת פסיקה כלשהי על מנת להתייחס לכך בהתאם (לדוגמה על ידי ביצוע פעולת reset).

7. בהמשך לשאלה בהתחלה, ברכיב outputn אנו יכולים להוציא מספר אותות PWM שונים, אשר כל אופן קשור לאופן מנייה שונה לדוגמה output, toggle/reset, set ועוד.

E. שאלות הכנה תיאורטיות מודול ADC12:

1. למודול ADC12 ישנו שעון אשר נקרא ADC12CLK הקובע את משך זמן פעולת הדגימה ופעולת ההמרה, הוא מוזן דרך המקורות MCLK, ACLK ו SMCLK שאלו שעונים פנימיים של הבקר. הסיבה לצורך בשעונים אלו היא שליטה במרווח הדגימה של האות האנלוגי בכניסה דבר המאפשר לרדת לרזולוציה גבוהה יותר במידת הצורך עבור דגימות מסוימות ומצד שני כאשר אין צורך לא נדגום בקצב מאוד גבוה ובכך נחסוך חישובים וזכרון או לחילופין נוכל לדגום לאורך אינטרוול זמן ארוך יותר.

2. המרחק בין הדגימות נקבע על ידי :

קביעת משך זמן הדגימה:

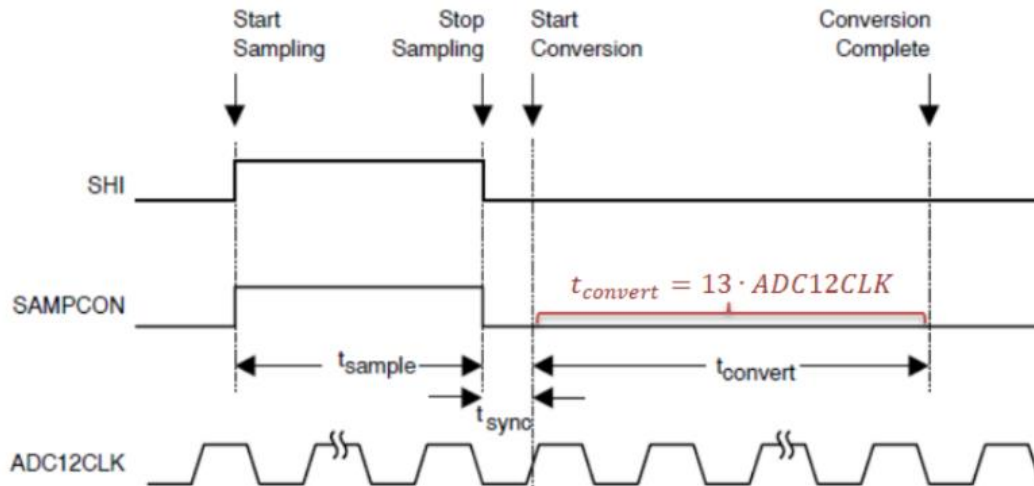
אות SHI הופך לאות SAMPCON באחת משתי אפשרויות (באופן ישיר או כטריגר – הבחירה בין השניים ע"י ביט SHP) ואות SAMPCON שולט באופן ישיר על משך מחזור הדגימה ועל תזמון תחילת ההמרה (משך זמן ההמרה הוא קבוע, 13 מחזורים של ADC12CLK). כאשר SAMPCON ב-'1' לוגי מתבצעת דגימה וכאשר SAMPCON יורד ל-'0' מתחילה ההמרה של אותה הדגימה. למעשה ישנן 2 שיטות שונות לדגימה המוגדרות ע"י ביט SHP, קביעה זו הינה עבור כל 16 הערוצים.

(1) כאשר ביט SHP=0:

אות SHI שולט באופן ישיר על אות SAMPCON ומגדיר את משך זמן הדגימה t_{sample} .

כאשר ביט SHP=0:

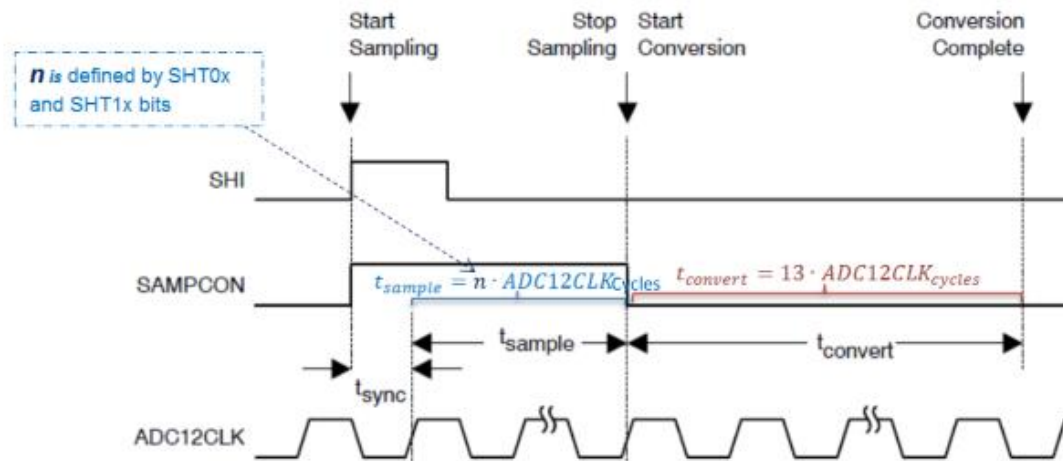
אות SHI שולט באופן ישיר על אות SAMPCON ומגדיר את משך זמן הדגימה t_{sample}



נדרש זמן נוסף t_{sync} לסנכרון בין SAMPCON ו-ADC12CLK.

כאשר ביט SHP=1:

אות SHI משמש כטריגר, בעליית אות SHI נכנס לפעולה רכיב בשם Sampling Timer משך הדגימה (משך אות SAMPCON) השווה ל- $n \cdot \text{ADC12CLK}_{\text{cycles}}$ כאשר n הוא מספר בטווח של $4 \div 1024$ (נקבע ע"י ביטים SHT0x ו-SHT1x). כאשר ביט MSC = 0 אות SHI מהווה טריגר לדגימה בודדת ולכן מתאים עבור אופן עבודה של דגימה בודדת, בלבד. כאשר ביט MSC = 1 אות SHI מהווה טריגר לדגימה ראשונה (מתאים לשאר אופני העבודה חוץ מאופן של דגימה בודדת) ושאר הדגימות נעשות באופן אוטומטי מיד לאחר סיום המרת הדגימה הקודמת.



נדרש זמן נוסף t_{sync} לסנכרון בין SAMPCON ו-ADC12CLK.

3. 4 אופני העבודה של מודול ADC12:

אופן עבודה ראשון – דגימה בודדת, לפעמים אתה רוצה להיות מסוגל למדוד בצורה חד פעמית את הערך של רכיב חיצוני ולפני הערך המתקבל לקחת החלטה בהתאם לדוגמא בדיקת זרם או ערך לוגי על קבל ולפי כך להסיק מסקנות על פעולות להמשך. אופן עבודה שני - דגימה מחזורית של הערוץ, באופן עבודה זה אנו רוצים לדגום באופן קבוע את הערוץ שלנו ולקבל המרה בקצב קבוע, במקרה של חריגות מערכים מסוימים נוכל לקבל החלטה על תיקון או איזון של המערכת. אופן עבודה שלישי – דגימה בודדת של מספר ערוצים באופן טורי, באופן עבודה זה אנו דוגמים כמה ערוצים שונים, לדוגמא מתח על קבל, זרם על נגד וכו'. במידע המגיע אלינו בנקודת זמן, סט הדגימות הבודדות האלה מאפשר לנו לקבל תמונת מצב על המערכת שלנו ועל ידי כך לקבל תנאי התחלה לדוגמא. אופן עבודה רביעי – דגימה מחזורית של מספר ערוצים באופן טורית דבר המאפשר מעקב רציף על כמות ערוצים בזמנית ועל ידי כך מאפשר לנו בקרה על כל המערכת ביחד, במקרה של חריגה של אחד הערוצים מהערכים הרצויים נוכל בעזרת פסיקה מתאימה לשנות את הערכים הרלוונטיים.

4. העקרון שלו הוא בעצם לבחור פסיקה אחת מתוך 18 אפשרויות שהיא בעצם נבחרת על ידי סדר עדיפות שנקבע מראש בארכיטקטורה. היתרון הינו סדר ושיוך של פסיקות מאותו הסוג בנוסף ליכולת להגדיר את העדיפות ביניהם. ללא השימוש בו אם נרצה לבדוק את כל הפסיקות

(18 במקרה הזה) אנו נצטרך לבדוק כתובת כתובת שזה בין היתר אומר לדעת את הכתובת של כל אחת מהפסיקות לגשת אליהן ספציפית בזכרון וכל זה לעומת לדעת את הכתובת של וקטור הפסיקות ורק לחבר לו ערך offset .

5. data format – הצורה שבה אנחנו מייצגים את האות האנלוגי במוצא מהאות הדיגיטלי בכניסה . אנחנו יכולים לפרש את הערך הדיגיטלי בכניסה (מספר בינארי) כערך שהוא תמיד חיובי או כערך שיכול להיות שלילי (במקרה של שלילי 0 הוא חצי הערך המקסימלי שכמות הביטים מאפשרת) ובכך מתאפשרת בחירה של טווח המתחים שאותם ניתן לייצג בהינתן כמות ביטים מסויימת לייצוג.

6. הרזולוציה היא רמת הדיוק שנוכל לקבל והיא נקבעת על ידי טווח המתחים הכולל חלקי מספר הרמות או לחילופין הפרש המתח בין שני רמות סמוכות. הפרש זה מהווה חסם לדיוק במתחים שכן יש לנו כמות רמות סופית לייצוג של אינסוף ערכים.

7. במודול DAC12 ישנו מנגנון כיול עצמי לכיול היסט המתח כך שגרף המוצא ביחס לערך רגיסטר DAT_DAC12 יהיה בשיפוע 1 וללא היסט. כאשר נעלה ל-1' את ביט DAC12CALON יתחיל הכיול העצמי שבסיומו ערך הביט DAC12CALON מתאפס (ניתן לוודא סיום כיול בעזרת palling).
כיול המודול יתבצע אחרי קינפוג המודול ולפני השימוש בו. בזמן הכיול ולצורך דיוק מרבי, כדאי לצמצם את פעולת הבקר.

8. ההבדל העיקרי בין 10DAC ל 12dac הוא בזמן ההמרה, זמן ההמרה של DAC 10 הוא 11 מחזורי שעון לעומת 13 של 12DAC . בנוסף ה10dac ביט SHP תמיד "1". ההבדל הנוסף הוא שכתובת ערכי ההמרה במודל 12DAC נכבטים לרגיסטרים יעודיים, ואילו ב10dac לבלוק בזכרון.

מגשים :
איילון קפל – 207807025
יעקב מסילתי - 205671852