

### שאלות הכנה דוח מכין מעבדה 3

1. כפי שנלמד בכיתה לפעמים נדרש מאיתנו להעביר מידע מרכיבים פריפריאליים לתוך הזכרון, באופן רגיל מה שהיה קורה מידע היה נקלט מהפריפריה ונשפך bus, המידע מהbus היה מועבר לcpu ומשם מועבר למקום הרלוונטי.  
DMA בא לנצל את העובדה שהמידע המיועד נמצא כבר על bus ומטרתו לדלג על המעבר דרך cpu דבר שחוסך המון זמן.  
רכיב ה dma ברגע קבלת בקשה אליו שולח סיבית בקרה אל cpu ומבקש "להשאיל" את bus ממנו.  
לאחר קבלת האישור של cpu ה dma משמש כמאסטר על bus ובעצם מנהל את התעבורה של המידע עליו ישירות לזכרון. כמו שניתן להסיק המערכת של DMA בעצם מדלגת על cpu כמתווך ביניים ועל כן חוסכת הרבה זמן (מנגד דורשת חומרה נוספת).  
בנוסף ה DMA בעצם משאיל את bus מהcpu ועל כן מונע את פעולתו הרציפה, קיימים מספר אופני פעולה שבאים למזער את השבת cpu בזמן זה ובעצם יוצרים מצב של עבודה בתורות.  
2. קיימים 4 אופני מיעון בעזרת DMA, עבור כל ערוץ אופן המיעון יכול להיות שונה (ערוצים בלתי תלויים). יש אפשרות להעביר מספר צירופי מידע:  
byte-to-byte, word-to-word, byte-to-word, or word-to-byte  
אפרט כעת על ארבעת האופנים:  
א. מכתובת קבועה לכתובת קבועה – על פי אופן זה תמיד כתובת המקור וכתובת היעד יהיו ידועים מראש (מדובר בהעברת byte) לדוגמא קריאה של port מסויים והעברתו באופן אוטומטי אל כתובת ספציפית בזיכרון, דוגמא לשימוש תהיה העברת המידע מלוח הלחצנים והמרתו למספר בינארי והכנסתו לערך זכרון בוא מוחזק משתנה עתידי שהתוכנית צריכה לעשות בו שימוש.  
ב. מכתובת קבועה לבלוק של כתובות – באופן זה מקור המידע הינו קבוע רק שהפעם לאחר כל הזנה היעד בעצם משתנה, לדוגמא נוכל לשמור רצף שינויים שביצענו על port או להזין בלוק שלם של כתובות לפי הזנה חיצונית שלנו.  
ג. מבלוק קבוע לכתובת ספציפית – באופן עבודה זה ניתן להעביר מבלוק כתובות ידוע מראש לכתובת ספציפית ביעד. בכל העברה הסמן של הכתובת מתקדם (כתובת בסיס והיסט). כתובת היעד נשארת קבועה, דוגמא של שימוש באופן זה יהיה העברת רצף ערכים ידוע מראש לתוך משתנה שאנו יודעים שהוא הולך להשתנות בכל איטרציה (אפשר על ידי אופן עבודה מסויים להעביר byte ואז לבצע מספר סייקלים בCPU ולחזור להעביר מידע).  
ד. מבלוק לבלוק – אופן זה מעביר בלוק של מידע מפריפריה אחת לפריפריה אחרת, אופן הביצוע הזה מאפשר לנו להעביר בלוקים של זכרון מנקודה לנקודה, דוגמא משמעותית לכך יכול להיות היכולת להעביר מידע מהזכרון הראשי ROM אל זכרון RAM שלנו.  
3. קיימות 6 שיטות להעברת מידע בעזרת DMA, כל ערוץ יכול באופן בלתי תלוי להעביר מידע לפי קינפוג מתאים. אפרט על 6 השיטות ואנסה לפרט על הטריידאופס של כל שיטה:  
א. Single transfer – העברה בודדת בכל פסיקה, בכל פעם תתבצע העברת מידע בודדת, לדוגמא בעברת המידע port ודגימת הערכים בו. באופן עבודה זה רכיב ה DMA משתלט על bus להעברה אחת של מידע ומחזיר את bus לCPU, שיטה זאת לא משביתה את CPU להרבה זמן ומיועדת להעברת מעט מידע.  
ב. Block transfer – בכל פסיקה (טריגר) רכיב ה DMA ידאג להעביר בלוק שלם של זיכרון, זאת אומרת השבתה יחסית ארוכה של BUS לטובת העברה זאת. דוגמא רלוונטית היא העברת בלוק לזכרון המסומן מהזכרון הראשי, הייתי רוצה להשבית את פעילות CPU עד לעדכון המידע הרלוונטי ורק אחרי ביצוע פעולה זאת להמשיך את עבודת ה CPU.  
ג. Burst-block transfer – באופן עבודה זה ישנה חלוקה של bus בין CPU ל DMA כך שמטרת ה DMA היא להעביר בלוק של מידע אבל כעת ישנן עצירות בהעברה לטובת החזרת ה BUS ל CPU. באופן עבודה זה לאחר העברה של 4 יחידות מידע (יכול להיות byte או word) bus עובר לשליטה של CPU שוב לטובת ביצוע שני מחזורי שעון. (מאפשר ל CPU יכולת ביצוע של 20% מהביצוע הרגיל)

- (שלו). לאחר סיום העברת המידע ה-CPU חוזר לתפקד רגיל.
- אנחנו רוצים להשתמש בשיטת עבודה זו כאשר אנחנו רוצים כן להעביר מידע באופן יחסית מיידי אבל רוצים גם שה-CPU ימשיך לבצע פעולות (פשוט בהספק נמוך יותר) ובעצם להתחלק ב-bus .
- לדוגמא כאשר רוצים לדגום ולבצע חישובים בו זמנית ולכן יש צורך להמשיך את פעולת המעבד תוך דגימה של פריפריות לזכרון.
- ד. Repeated single transfer – מתאר מצב דומה מאוד להעברה בודדת כמו שתיארנו מעל, כאשר במקרה זה ביט האפשרות לפסיקה נוספת שחוזרת על עצמה לא מתאפס ולכן אם אותו ערוץ יעלה פסיקה שוב ניתן בעצם להעביר שוב פעם את המידע בעזרת אותו ערוץ. (בשונה משיטה א' שברגע שבוצעה העברה ביט האפשרות נכבה. דוגמא מתאימה היא כאשר רוצים לדגום פריפרייה מספר פעמים לאורך התוכנית שלנו מבלי לקנפג זאת כל פעם מחדש.
- ה. Repeated block transfer – באופן דומה לשיטה מעל, אפשרות הפסיקה מהערוץ נשארת "1" ועל כל פסיקה שתגיע מהערוץ תגרום להעברת מידע בגודל בלוק. עבור שיטה זאת אנחנו רוצים לשמור על ערוץ העברת המידע פתוח. דוגמא לשימוש בשיטה זאת היא קביעת ערוץ אחד כערוץ להעברת בלוקים של זכרון כמו שתיארתי קודם העברת זכרון מה-ROM ל-RAM.
- ו. Repeated burst block transfer – בשיטה זאת כמו בקודמים סיבית האפשרות נשארת מורמת ועל כל הערוץ בעצם פתוח להעברת מידע בשיטה של burst block בכל פסיקה שתהיה, חשוב לשים לב אם מתבצעות הרבה פסיקות מסוג זה הדבר בעצם ישאיר את המעבד שלנו עם הספק של 20% באופן רציף דבר שיכול לפגום בקצב התוכנית שלנו. דוגמא שיכולה להתאים לשיטה זאת היא כאשר רוצים להעתיק כמות גדולה של קבצים או מידע באופן רציף ועדיין לשמור על תפוקת המעבד באופן בסיסי.
4. אם ברצוננו להשתמש ב-DMA לצורך העברת מידע מהמודולים DAC12, ADC12, TB עלינו להגדיר אותם כמקורות הפסיקה שלנו, (חשוב לוודא שסיבית הפסיקה של כל אחד מהרכיבים הנ"ל נמצא ב-0 לוגי כדי שאם באמת תהיה פסיקה יהיה מעבר מ-0 ל-1 לוגי ואז יתבצע פסיקה גם ב-DMA).
- בכל המקרים הנ"ל כדי בכלל לאפשר שימוש וטריגר מן הערוצים אנחנו צריכים להזין אפשרות קבלת פסיקה של DMA, חשוב להקנפג את DMA לקבלת רגישות כאשר מתבצעת עליית שעות. כעת עלינו לבחור לכל ערוץ ממי הוא מקבל את הטריגר שלו לפי הפירוט הבא:

Table 6–2. DMA Trigger Operation

DMAxTSELx	Operation
0010	A transfer is triggered when the TBCCR2 CCIFG flag is set. The TBCCR2 CCIFG flag is automatically reset when the transfer starts. If the TBCCR2 CCIE bit is set, the TBCCR2 CCIFG flag will not trigger a transfer.
0101	A transfer is triggered when the DAC12_0CTL DAC12IFG flag is set. The DAC12_0CTL DAC12IFG flag is automatically cleared when the transfer starts. If the DAC12_0CTL DAC12IE bit is set, the DAC12_0CTL DAC12IFG flag will not trigger a transfer.
0110	A transfer is triggered by an ADC12IFGx flag. When single-channel conversions are performed, the corresponding ADC12IFGx is the trigger. When sequences are used, the ADC12IFGx for the last conversion in the sequence is the trigger. A transfer is triggered when the conversion is completed and the ADC12IFGx is set. Setting the ADC12IFGx with software will not trigger a transfer. All ADC12IFGx flags are automatically reset when the associated ADC12MEMx register is accessed by the DMA controller.
1000	A transfer is triggered when the TBCCR0 CCIFG flag is set. The TBCCR0 CCIFG flag is automatically reset when the transfer starts. If the TBCCR0 CCIE bit is set, the TBCCR0 CCIFG flag will not trigger a transfer.

פה ניתן לראות את הקינפוג הדרוש לטובת בחירה של מקור מסוים לפסיקה .

אפרט על דוגמאות רלוונטיות לכל רכיב.

לדוגמא עבור רכיב טיימר B היינו רוצים שכאשר עלתה פסיקה בטיימר ערך הטיימר הנמדד יוכנס לזכרון

(מידע שיעלם אם לא נדגום) במקום לקחת את ערך הטיימר להכניסו ל CPU ולאז לשמור אותו נוכל ישירות להכניסו לזכרון .

\\ להכניס דוגמאות של DAC ADC למה אני לא סגור על זה.

5. Dma channel priorities – מושג שבא לתאר את סדר העדיפות של רכיב ה DMA לפי ערוצים.

ברירת המחדל של המערכת תהיה לתת עדיפות ל DMA0 ואז ל DMA1 ואז ל DMA2 .

מה זה בעצם אומר לנו ? -> אם מתקבלות פסיקות מכמה ערוצים בו זמנית (או כאלו שממתינות)

הערוצים בעלי העדיפות הגבוהה ביותר יטופלו קודם ורק אז נעבור לאלה בעדיפות הנמוכה יותר. (אם)

העברה מתבצעת כבר ברגע זה היא לא תעצור אם פסיקה בעדיפות גבוהה יותר עלתה).

אפשר לשלוט בסדר העדיפות של הרכיב על ידי שליטה בביט שנקרא roundrobin (משנה עדיפות בצורה סיבובית).

הצורך ביצירת עדיפות בין הערוצים היא בגלל סיווג של המידע הרלוונטי, אם למידע מסוים יש עדיפות העברה אז נעדיף לתת לו חשיבות. בנוסף יש מידע שיכול להיות שלא ימצא שם יותר אם לא נדגום אותו במהרה ולכן נרצה לתת גם על פי שיקול זה עדיפות לאותו ערוץ ספציפי.

6. DMA Transfer Cycle Time – לרכיב dma לוקח בין מחזור אחד לשניים של (MCLK) כדי להסתנכרן

לפני העברת המידע, כל העברה של byte או word דורש עוד שני מחזורי שעון לאחר הסנכרון,

ומחזור אחד של המתנה אחרי השליחה. מכיוון שרכיב DMA תלוי ב MCLK אזי הרכיב גם תלוי ביכולות

של המעבד עצמו ותלוי בשעון ההזנה למערכת, מכאן שיש כמות מחזורים שמתאימה לכל מצב עבודה של המעבד.

נדרשנו לתאר כמה מחזורי שעון נדרשים עבור שני מקרים:

עבור מצב עבודה של CPU על מוד אקטיבי ועבור מצב עבודה של מצב שינה :

CPU Operating Mode	Clock Source	Maximum DMA Cycle Time
Active mode	MCLK=DCOCLK	4 MCLK cycles
Low-power mode LPM0/1	MCLK=LFXT1CLK	5 MCLK cycles

7. מערכת הפסיקות של DMA לא מקושרת לפסיקות מהמערכת, בקשת פסיקה של מודול DMA יכולה

להתרחש אם פסיקה שאינה ניתנת למיסוך (שמגיעה מאחת הפריפריות) מאושרת על ידי רכיב DMA

בעזרת העלאת ביט ENNMI. בקשת פסיקה ממודול DMA יכולה להתבצע על ידי כל אחד מהמודולים

הפנימיים בהינתן העלאת הביטים : DMAIE ו GIE (בהינתן שרגיסטר גודל המידע הועבר וכמות המידע עברה בהצלחה)

8. בקשת פסיקה ובקשת DMA לא יכולים לקרות במקביל, הדבר הראשון שיטופל יהיה הפסיקה ורק לאחר

מכן בקשת DMA.