

תוכן עניינים:

| | | |
|----|------------------------------------------------------|---|
| A. | נושאי המעבדה: | 2 |
| B. | חומר הכנה למעבדה: | 2 |
| C. | שאלות הכנה – MSP430 DMA module: | 2 |
| D. | חלק מעשי – כתיבת קוד המערכת (באופן גנרי ופורטאבילי): | 3 |
| E. | צורת הגשה דוח מכין: | 6 |
| F. | צורת הגשה דוח מסכם: | 6 |

DMA (Direct Memory Access)

A. נושאי המעבדה:

בניסוי מעבדה זה נעסוק בנושא DMA.

באופן כללי בעזרת מודול DMA נוכל להעביר בלוקי מידע ממודלי-חומרה (peripheral hardware) או מהזיכרון אל מודלי-חומרה או לזיכרון וזאת ללא שימוש ב-CPU. מאחר ומבחינה סטטיסטית העברת מידע רב היא בעלת התנהגות לוקאלית (המידע מתפרס על פני ערכי כתובות סמוכים) שימוש ב-CPU מיותר ו"גוזל" זמן והספק יקרים הנדרשים לצורך ייבוא נתונים. בבקר **MSP430x4xx** (בקר משפחה 4, בשונה מבקר משפחה 2 בו לא קיים מודול DMA) ישנו מודול DMA אחד בעל שלושה ערוצים שונים בלתי-תלויים, כאשר בו זמנית ניתן להפעיל רק אחד מהם. קובצי קוד לדוגמה בשימוש DMA נמצאים במודל.

B. חומר הכנה למעבדה:

- ♦ המשימה מבוססת על החלק התיאורטי בפרק DMA עמודים 377-405 בספר **MSP430x4xx family user guide** (הנמצא במודל תחת לשונית חומר עזר) + קוד לדוגמה הנמצא במודל תחת לשונית LAB3.
- ♦ הפעלת קוד לדוגמה והבנתו במקביל לחומר התיאורטי הנלמד.
- ♦ על בסיס הנ"ל לבצע את מטלת המעבדה היישומית.

C. שאלות הכנה – MSP430 DMA module:

- 1) הסבר/י בפירוט את המוטיבציה לשימוש ב DMA ביחס ל- CPU לצורך העברת נתונים (מהו ה Tradeoff).
- 2) הסבר/י בפירוט את ארבע השיטות המעון, רשמו דוגמה מתאימה עבור כל אחת מהשיטות.
- 3) הסבר/י בפירוט את שש השיטות להעברת מידע בשימוש DMA, רשמו דוגמה מתאימה עבור כל אחת מהשיטות.
- 4) הסבר/י כיצד ניתן להשתמש ב DMA לצורך העברת מידע מהמודולים DAC12, ADC12, TimerB, רשמו דוגמה מתאימה עבור כל אחת מהשיטות.
- 5) הסבר/י את המושג **DMA Channel Priorities** ומדוע יש צורך בו.
- 6) הסבירו כיצד מורכב ה- **DMA Transfer Cycle Time** (זמן העברה בפועל + תקורה) במקרים הבאים:
 - Case 1:
CPU Operating Mode - Active mode
CPU Clock Source MCLK - DCOCLK
 - Case 2:
CPU Operating Mode - Low-power mode LPM0/1
CPU Clock Source MCLK - DCOCLK
- 7) הסבר/י באילו תנאים תתבצע בקשת פסיקה של מודול DMA.
- 8) הסבר/י האם מודול פריפריאלי יכול לבצע בקשת DMA ובקשת פסיקה בו זמנית.

D. חלק מעשי – כתיבת קוד המערכת באופן גנרי ופורטאבילי (בסביבת פיתוח CCS בלבד):

הקדמה:

בניסוי זה, נממש מערכת Embedded מבוססת מאיץ חומרה DMA לצורך ביצוע מניפולציה ועיבוד מידע על מבנה נתונים. ביצוע עיבוד מידע על מבני נתונים דורש פעולות load, store בכמות גדולה דבר אשר אינו מתאים לביצוע בעזרת CPU מבחינה סיבוכיות זמן וניצול אנרגיה. בניסוי זה נלמד לשלב מימוש אלגוריתם לצורך עיבוד מידע המבוסס על שימוש ב CPU ובמאיץ חומרה DMA בצורה נכונה, מבחינה זמן תגובת המערכת וצריכת הספק במערכת Low Power MCU.

דרישת חיבורי החומרה:

חיבורי חומרה של המערכת:

1. הלחצנים PB2 - PB0 מחוברים לרגלי הבקר P1.0 – P1.2 בהתאמה
2. מסך LCD נדרש לחבר את D7-D4 לרגליים P1.4-P1.7 בהתאמה (אופן עבודה של ה- LCD בארבע סיביות של מידע) + שלושת קווי הבקרה של ה- LCD לרגליים P2.5, P2.6, P2.7 (קוד עבור LCD נתון במודל, עליכם לעדכן לצרכיכם).

3. חיבור ה- Keypad ל P10 ואת רגל הפסיקה לרגל P2.1 (ראו נספח בנושא Keypad)

ארכיטקטורת תוכנה של המערכת:

1. ארכיטקטורת התוכנה של המערכת נדרשת להיות מבוססת *Simple FSM* (כמתואר בדו"ח מכין 1) המבצעת אחת מתוך ארבע פעולות בהינתן בקשת פסיקה חיצונית של לחיצת לחצן מתוך שלושת הלחצנים.
2. קוד המערכת נדרש להיות מחולק לשכבות כך שהוא יהיה נייד (portable) בקלות בין משפחות MSP430 ע"י החלפת שכבת ה- BSP בלבד.
3. טרם שלב כתיבת הקוד נדרש לשרטט גרף דיאגרמת FSM מפורטת של ארכיטקטורת התוכנה של המערכת ולצרפה לדו"ח מכין. המצבים אלו הצמתים והקשתות אלו המעברים ממצב למצב בגין בקשות פסיקה.

4. תזכורת: אסור לבצע השהייה ע"י שימוש ב poling למעט עבור debounce ברוטינת שירות של בקשות

פסיקה בגין לחצנים.

דרייברים שכבת ה HAL תוספות לתמיכה במערכת:

1. כתיבת פונקציות ה driver של ה LCD צריכות להיות ממוקמות ב HAL בעוד שלדוגמה פונקציה לכתיבת מחרוזת המבוססת עליהן צריכה להיות ממוקמת בשכבת ה API
2. כתיבת פונקציות ה driver של ה Keypad צריכות להיות ממוקמות ב HAL בעוד שלדוגמה פונקציה לקליטת מחרוזת מה Keypad המבוססת עליהן צריכה להיות ממוקמת בשכבת ה API
3. מצב 2 של המערכת מבוסס תכונת DMA של העברת מידע מטווח כתובות בזיכרון אחד לטווח כתובות אחר בזיכרון, הגדירו לכך דרייבר מתאים
4. מצב 3 של המערכת מבוסס תכונת DMA של העברת מידע מטווח כתובות בזיכרון לרכיב פריפריאלי, הגדירו לכך דרייבר מתאים

מבנה נתונים סטטיים של המערכת:

נדרש להגדיר מטריצת מסד נתונים (מערך דו-ממדי סטטי) בשכבת ה main מטיפוס char באופן הבא:

```
Char data_base[M][N] = {
    "An apple a day keeps the doctor away",
    "Give someone the benefit of the doubt",
    "Time flies when you're having fun",
    "He who pays the piper calls the tune",
    "The pen is mightier than the sword",
    "The pot calling the kettle black",
    "Wrap your head around something",
    "We'll cross that bridge when we come to it",
    "A bird in the hand is worth two in the bush",
    "Do unto others as you would have them do unto you",
    "You can catch more flies with honey than you can with vinegar",
    "eod" // data bade delimiter string
};
```

ומערך סטטי נוסף מטיפוס char בשם **idiom_recorder** בגודל של 32 לאחסון הקלטת פתגם/ביטוי/ניב ע"י המשתמש דרך ה- Keypad (ראו מצב 1).

:(state=idle=0)

בלחיצת RESET או בסיום ביצוע כל המצבים, הבקר נמצא/חוזר למצב שינה (Sleep Mode).

:(state=1) PB0 על לחצן

לצורך מימוש idiom recorder נדרש להדפיס בקשה למשתמש דרך ממשק מתאים על גבי מסך ה- LCD לצורך קלט של פתגם (באמצעות הקלדה) עד מקסימום 32 תווים דרך הקלדתם על גבי ה Keypad ולאחסן במערך **idiom_recorder** נדרש לבצע חיווי על גבי מסך ה LCD של התווים אותם מקליד המשתמש.

הערה: המצב מוגדר להסתיים בלחיצה על לחצן המשויך למצב אחר.

Keypad

| | | | |
|-------|-------|-------|-------|
| 1/G | 2/H | 3/I | C/J |
| 4/K | 5/L | 6/M | D/N |
| 7/O | 8/P | 9/Q | E/R |
| A/S/T | 0/U/V | B/W/X | F/Y/Z |

בלחיצה על לחצן PB1 (state=2):

הגדירו פונקציה להיפוך מחרוזת (תמונת מראה) בעזרת DMA בלבד. הפונקציה מקבלת מצביע לבסיס נתונים $data_base$ (המוגדר לעיל), אינדקס i למחרוזת (כאשר $0 \leq i < N - 1$) ומצביע למערך הפלט. הפונקציה מבצעת אתחול פרמטרים של ה DMA ולאחר מכן מעבירה שליטה ל DMA לצורך היפוך המחרוזת לתוך מערך פלט בשם `strMirror` בעזרת DMA בלבד, בשימוש אופן עבודה Block transfer. להלן מבנה הפונקציה:

```
char* dma_strMirror(char** data, int idx, char* strout) {
    dma initializations;
    dma transfers execution;
    return strout;
}
```

1. בסיום הפונקציה נדרש להדפיס את מערך `strMirror` על גבי מסך ה LCD (בשימוש כתיבה לשתי השורות עם יכולת גלילה ע"י מקש Keypad לבחירתם).
 2. בכניסה למצב זה, נדרש לבקש מהמשתמש לבחור בעזרת ה Keypad אינדקס $0 \leq i < 11$ (אינדקס אחרון נועד לכתובת תחימה של מבני הנתונים) לבחירת שורה במטריצת `data_base`.
 3. כתובת סוף מחרוזת i היא למעשה כתובת של מחרוזת $i + 1$ פחות 1.
- המצב אטומי ומוגדר להסתיים לאחר הדפסת מערך `strMirror` על גבי מסך ה LCD**
 לדוגמה עבור בחירת אנדקס 9 תוכן מערך הפלט `strMerge` הוא:
 "Yuo nac htac erom seilf htiw yenoh naht uoy nac htiw iverag"

בלחיצה על לחצן PB2 (state=3):

בעזרת DMA בלבד, בשימוש אופן עבודה Single transfer נדרש להציג על מערך של שמונה לדים את איברי המערך הבא {128,224,32,126,83,44,253,113,160} איבר אחר איבר משמאל לימין עם השהייה של חצי שנייה בעזרת טריגר של TimerB.

הדרכה: נגדיר מערך באורך 8 לשמירת ערכי ההדפסה ללדים, העברת המידע תהיה בנפרד תא אחר תא במערך ל PORT9 בשימוש DMA ותזמון המעבר יהיה בעזרת טריגר חומרה TACCR2_CCIFG במרווחי זמן של 500ms.

המצב אטומי ומוגדר להסתיים בסיום מחזור הצגה של המערך הנתון.

E. צורת הגשה דוח מכין:

- הגשת מטלת דוח מכין תיעשה ע"י העלאה למודל של תיקיית zip מהצורה **id1_id2.zip** (כאשר $id1 < id2$), רק הסטודנט עם הת"ז id1 מעלה את הקבצים למודל.
- התיקייה תכיל את שני הפרטים הבאים בלבד:
 - ✓ קובץ `pre_labx.pdf` – מכיל תשובות לחלק תיאורטי דו"ח מכין
 - ✓ תיקייה בשם `CCS` - מכילה שתי תיקיות, אחת של קובצי source (קבצים עם סיומת *.c) והשנייה של קובצי header (קבצים עם סיומת *.h).

F. צורת הגשה דוח מסכם:

- הגשת מטלת דוח מכין תיעשה ע"י העלאה למודל של תיקיית zip מהצורה **id1_id2.zip** (כאשר $id1 < id2$), רק הסטודנט עם הת"ז id1 מעלה את הקבצים למודל.
- התיקייה תכיל את שני הפרטים הבאים בלבד:
 - ✓ קובץ `final_labx.pdf` – מכיל תיאור והסבר לדרך הפתרון של מטלת זמן אמת.
 - ✓ תיקייה בשם `CCS` - מכילה שתי תיקיות, אחת של קובצי source (קבצים עם סיומת *.c) והשנייה של קובצי header (קבצים עם סיומת *.h).

בהצלחה.