

میزکار < درس‌ها < مهندسی کامپیوتر < نیمسال اول 1400 < 14001 - 3103013 - ریزپردازنده و زبان اسمبلی (2) < امتحان پایان ترم < امتحان پایان ترم

|              |                                |
|--------------|--------------------------------|
| شروع         | چهارشنبه، 22 دی 1400، 1:30 عصر |
| وضعیت        | پایان یافته                    |
| پایان        | چهارشنبه، 22 دی 1400، 3:29 عصر |
| زمان صرف شده | 1 ساعت 59 دقیقه                |
| جمع نمره     | 7.80 از 8.00                   |
| نمره         | 19.50 از 20.00 (98%)           |

با فرض این که پایپ لاین نداریم، مراحل پردازش یک interrupt یا exception را بنویسید و هرکدام را توضیح دهید.

مراحل به صورت زیر است:

Other code (background) is running

Interrupt trigger occurs

Processor does some hard-wired processing

Processor executes ISR (foreground), including return-from-interrupt instruction at end•

Processor resumes other code

(Finish current instruction (except for lengthy instructions •

(Push context (8 32-bit words) onto current stack (MSP or PSP •

xPSR, PC, LR (R14), R12, R3, R2, R1, R0 •

Switch to handler/privileged mode, use MSP •

Load PC with address of exception handler •

Load LR with EXC\_RETURN code •

Load IPSR with exception number •

Start executing code of exception handler •

توضیح هر کدام:

• کد قسمت عادی برنامه در حالت اجراست ..

• interrupt رخ می دهد.

• پردازنده کمی کار سخت افزاری انجام می دهد.

• بیشتر instruction ها از نوع کوتاه هستند، ولی اگر از نوعی باشند که زیاد زمان میبرند (مثل LDM)، آن هارا ول میکنیم و به interrupt می پردازیم

• دو استک پوینتر داریم، MSP, PSP که با توجه به هر کدام که اکتیو است کانتکست سوئیچ را انجام می دهیم.

• برای این کار باید از MSP استفاده شود

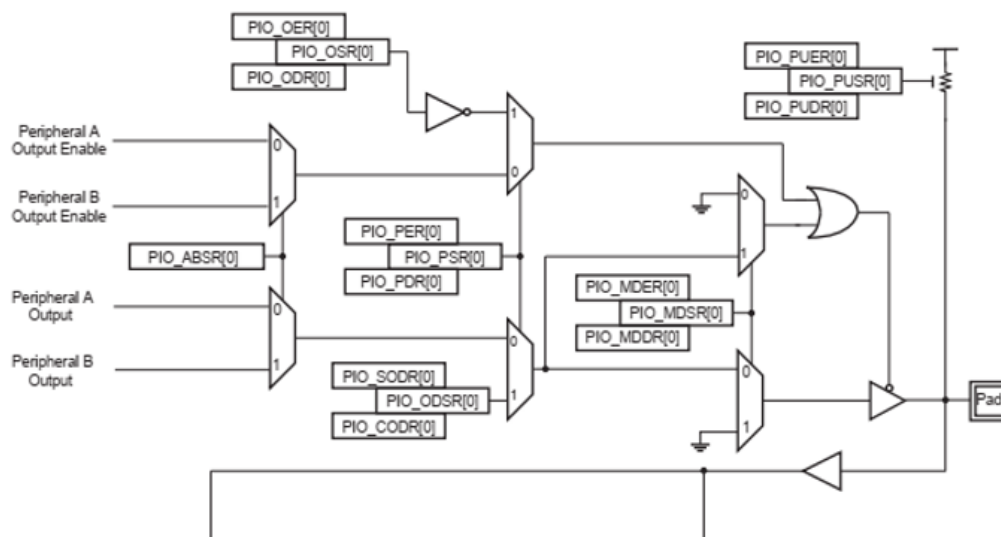
• به کد interrupt handler رجوع کرده و interrupt را انجام می دهیم.

• دوباره PC را لود میکنیم. (با ادرس دستوری که قبل از interrupt در حال انجام بود)

• به بقیه کارها به صورت عادی می پردازیم.

interrupt از دید برنامه به صورت شفاف است. (یعنی برنامه ما متوجه وقوع آن نمی شود)

تصویر زیر منطق کنترلی پایه شماره صفر واحد PIO را نشان می دهد؛ فرض کنید Peripheral A متصل به خط MOSI مربوط به SPI Controller است. اگر SPI Controller نقش Master داشته باشد، مقدار بیت صفر رجیستر های Status لازم برای برقراری اتصال بین MOSI و پین را تعیین کنید.



برای این کار پین ما باید به عنوان خروجی باشد، یعنی  $PIO\_OER=1$  که به تبع آن  $PIO\_OSR=1$  نیز هست. چون peripheral A مربوط به SPI Controller است باید  $PIO\_ABSR=0$  باشد که A انتخاب شود. همچنین باید peripheral status register نیز peripheral را انتخاب کند یعنی  $PIO\_PSR=0$  در نهایت  $PIO\_MDSR$  نیز برابر صفر قرار می دهیم که خروجی از Peripheral A output به خروجی منتقل شود.

همچنین چون در حالت master هستیم  $MSTR\ bit = 1$  است در  $SPI\_MR$

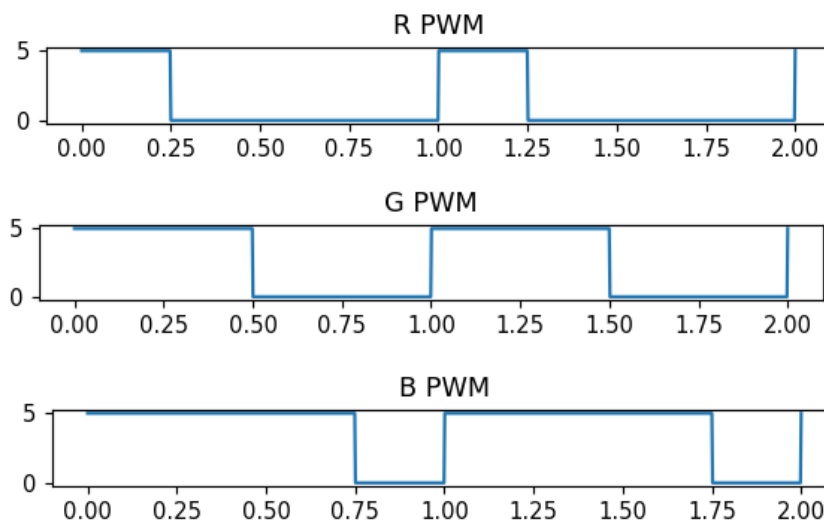


سؤال 3

کامل

نمره 1.00 از 1.00

میدانیم که یکی از کاربردهای PWM کنترل رنگ چراغ‌های LED است. سیگنال‌های PWM مربوط به هر یک از رنگ‌های RGB در زیر آورده شده است. ابتدا duty cycle هر یک را بدست آورید و سپس رنگ نهایی حاصل را بنویسید. (ماکسیم مقدار هر یک از رنگ‌های RGB برابر با ۲۵۵ است. رنگ نهایی را به فرمت (R, G, B) بنویسید)



$$a) (1.25 - 1.00) / (1.25 - 0.25) = 25\% \implies 63.75 = 64$$

$$b) (1.50 - 1.00) / (1.50 - 0.50) = 50\% \implies 127.5 = 128$$

$$c) (1.75 - 1.00) / (1.75 - 0.75) = 75\% \implies 191.25 = 191$$

(rgb(64, 128, 191



سؤال 4

کامل

نمره 0.90 از 1.00

فضای آدرس دهی پردازنده ARM به صورت زیر تقسیم بندی شده است. مقدار اختصاص داده شده به هر بخش را محاسبه کنید. همچنین ما به 1 مگابایت حافظه برای Off-Chip DRAM هم نیاز داریم. آدرس فضای حافظه‌ای نیز برای این بخش، مشخص نمایید.

- a) 0x00200000 – 0x00200FFF: EEPROM
- b) 0x40000000 – 0x40007FDF: SRAM
- c) 0x00010000 – 0x0008FF00: Flash
- d) 0xFFFFCD00 – 0xFFFFFFFF: Peripherals

a) 0x00200000 – 0x00200FFF: EEPROM

فضای حافظه برابر است با تعداد بایت هایی که اختصاص داده ایم، داریم:  $0x00200FFF - 0x00200000 + 1 = 4096 \text{ byte}$   
 که برابر است با:  $4\text{kByte} = 12^2$

b) 0x40000000 – 0x40007FDF: SRAM

$0x40007FDF - 0x40000000 + 1 = 32736 \text{ bytes}$

c) 0x00010000 – 0x0008FF00: Flash

$0x0008FF00 - 0x00010000 + 1 = 524033 \text{ bytes}$

d) 0xFFFFCD00 – 0xFFFFFFFF: Peripherals

$0xFFFFFFFF - 0xFFFFCD00 + 1 = 4026544896 \text{ bytes}$

برای تخصیص  $1\text{M byte} = 2^{20} \text{ bytes}$  خواهیم داشت:

$0x100000 = 20^2$

from 0x00000000 to 0x000FFFFF

$0x000FFFFF = 2^{20} - 1$

سؤال 5

کامل

نمره 1.00 از 1.00

یک روال بنویسید که عدد ذخیره شده در R0 را به اندازه مقدار R1، همراه با Carry بچرخاند و خروجی را در R10 بریزد.

چون مشخص نشده که به کدام سمت بچرخانیم فرض را به این میگذاریم که سمت راست باشد، همچنین برای این که carry هم درگیر شود باید از پسوند S استفاده کنیم، خواهیم داشت:

MOVS R10, R0, ROR R1



جدول زیر مقادیر شماری از واژه های حافظه (Memory Word) را نشان می دهد.

| Memory Word Address | Memory Content (Word) |
|---------------------|-----------------------|
| 0x3000201C          | 0x00000000            |
| 0x30002018          | 0xFFFFFFFF            |
| 0x30002014          | 0x3000201C            |
| 0x30002010          | 0x00000000            |
| 0x3000200C          | 0x30002014            |
| 0x30002008          | 0x00000080            |
| 0x30002004          | 0x70605040            |
| 0x30002000          | 0x30201000            |

فرض کنید که رجیستر های آورده شده در زیر نیز مقدار های نمایش داده شده را دارند.

$r0 = 0x30002000$

$r1 = 0x00000004$

$r2 = 0x0000FFFF$

$r9 = 0x3000200C$

$r13 (sp) = 0x30002008$

در این صورت به بگویید با اجرای هر یک از دستور های زیر نتیجه چیست. ( هر تغییری در مقدار رجیستر ها یا محتوای حافظه، مگر رجیستر PC)

الف) `LDRSB r9, [r0, #4]!`

ب) `LDR r9, [r9]`  
`LDR r9, [r9]`  
`LDR r9, [r9]`

الف)

دستور LDRSB دستور لود کردن از حافظه با نگهداری علامت اعداد می باشد.

دستور الف از دستورات pre-indexed addressing mode به همراه جایگذاری می باشد.

$$r0 = r0 + \#4 = 0x30002000 + \#4 = 0x30002004$$

$$r9 = \text{contnet of } 0x30002004 \text{ (is positive)} = 0x70605040$$

ب) با فرض اینکه دستورات الف از دستورات ب متمایز باشند دستور را اجرا می کنیم:

$$r9 = 0x3000200C$$

`[LDR R9, R9`

$$r9 = \text{mem } (0x3000200C) = 0x30002014$$

`[LDR R9, R9`

$$r9 = \text{mem } (0x30002014) = 0x3000201C$$

`[LDR R9, R9`

$$r9 = \text{mem } (0x3000201C) = 0x00000000$$



سؤال 7

کامل

نمره 1.00 از 1.00

یک تابع اسمبلی بنویسید که از R0 یک کاراکتر ASCII را میخواند و اگر یک عدد بود، R0 را False (صفر) میکند.

کاراکتر اسکی عدد صفر برابر است با ۴۸ یا 0x30

کاراکتر اسکی عدد ۹ برابر است با ۵۷ یا 0x39

کدی می نویسیم که اگر محتویات رجیستر بین این دو عدد باشد R0 را ۱ و در غیر اینصورت صفر کند.

```
MOV r1, r0
```

```
CMP r1, #0x30
```

```
BHS HIGHER_OR_SAME
```

```
MOV r0, #0
```

```
B OVER
```

```
HIGHER_OR_SAME
```

```
CMP r1, #0x39
```

```
BHI OVER
```

```
MOV r0, #1
```

```
OVER
```

```
HERE B HERE
```

```
END
```

همچنین می توانیم به صورت زیر نیز عمل کنیم:

```
'CHARACTER_0 EQU '0
```

```
'CHARACTER_9 EQU '9
```

```
MOV R2, #CHARACTER_0
```

```
MOV R3, #CHARACTER_9
```

```
MOV R1, R0
```

```
CMP R1, R2
```

```
BHS HIGHER_OR_SAME
```

```
MOV R0, #0
```

```
B OVER
```

```
HIGHER_OR_SAME
```

```
CMP R1, R3
```

```
BHI OVER
```

```
MOV R0, #1
```

```
OVER
```

```
HERE B HERE
```

```
END
```



سؤال 8

کامل

نمره 1.00 از 1.00

برنامه ای بنویسید که فاکتوریل R0 را به صورت بازگشتی محاسبه کند و مقدار آن را در R1 ذخیره کند.

PARAMN EQU 20

PARAMERESULT EQU 16

FACTORIAL

STR LR, [SP, #-4]! ; Save Return Address

STR IP, [SP, #-4]! ; Save Stack Frame Pointer

STR R0, [SP, #-4]! ; Save General Register

STR R1, [SP, #-4]! ; Save General Register

STR R1, [SP, #-4]! ; Save General Register

MOV IP, SP ; Set Up New Stack Frame

LDR R0, [IP, PARAMN] ; If N &lt;= 1 Then

CMP R0, #1

BGT MAIN

MOV R0, #1 ; Store 1 into Result

[STR R0, [IP, PARAMERESULT

B DONE

MAIN

[LDR R0, [IP, PARAMN

SUB R0, R0, #1

(STR R0, [SP, #-4]! ; push (N - 1

SUB SP, SP, #4 ; reserve space

BL FACTORIAL

LDR R0, [SP], #4 ; pop result

(ADD SP, SP, #4 ; discard (N - 1

[LDR R1, [IP, PARAMN

!(MUL R0, R1, R0 ; Compute N \* (N-1

STR R0, [IP, PARAMERESULT] ; Store into Result

DONE

LDR R1, [SP], #4 ; Restore General Register

LDR R0, [SP], #4 ; Restore General Register

LDR IP, [SP], #4 ; Restore Stack Frame

LDR PC, [SP], #4 ; Return

HERE B HERE

END

Previous activity

[▶ پروژه امتیازی](#)[رفتن به ...](#)

Next activity

[◀ اعتراض به امتحان پایان ترم](#)[اطلاعات تماس](#)[support.aut.ac.ir](mailto:support.aut.ac.ir) [021-66967416-64545947-5948-5949-5495](tel:021-66967416-64545947-5948-5949-5495) [دریافت نرم افزار تلفن همراه](#)