

9/27/2020

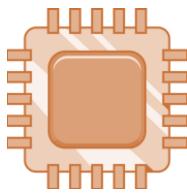


Homework 1

Lec 1-2-3-4



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE



MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

Homework 1



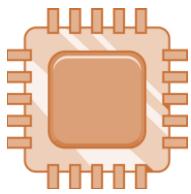
دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

۱) مجموعه دستورالعمل های RISC و CISC را مقایسه کنید و سپس جدول زیر را تکمیل کنید.

RISC: Reduced Instruction Set Computer

CISC: Complex Instruction Set Computer

	RISC	CISC
اندازه دستورالعملها		
نرخ کلاک		
نوع واحد کنترل		
دسترسی به حافظه		
ایراد یابی		
تعداد ثبات		
تعداد ترانزیستور و مقدار مصرف انرژی		
گذرگاه		
خط لوله		
پشتیبانی از مدل های آدرس دهنده		
Fetch & Decode		
پیچیدگی		



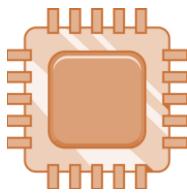
MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

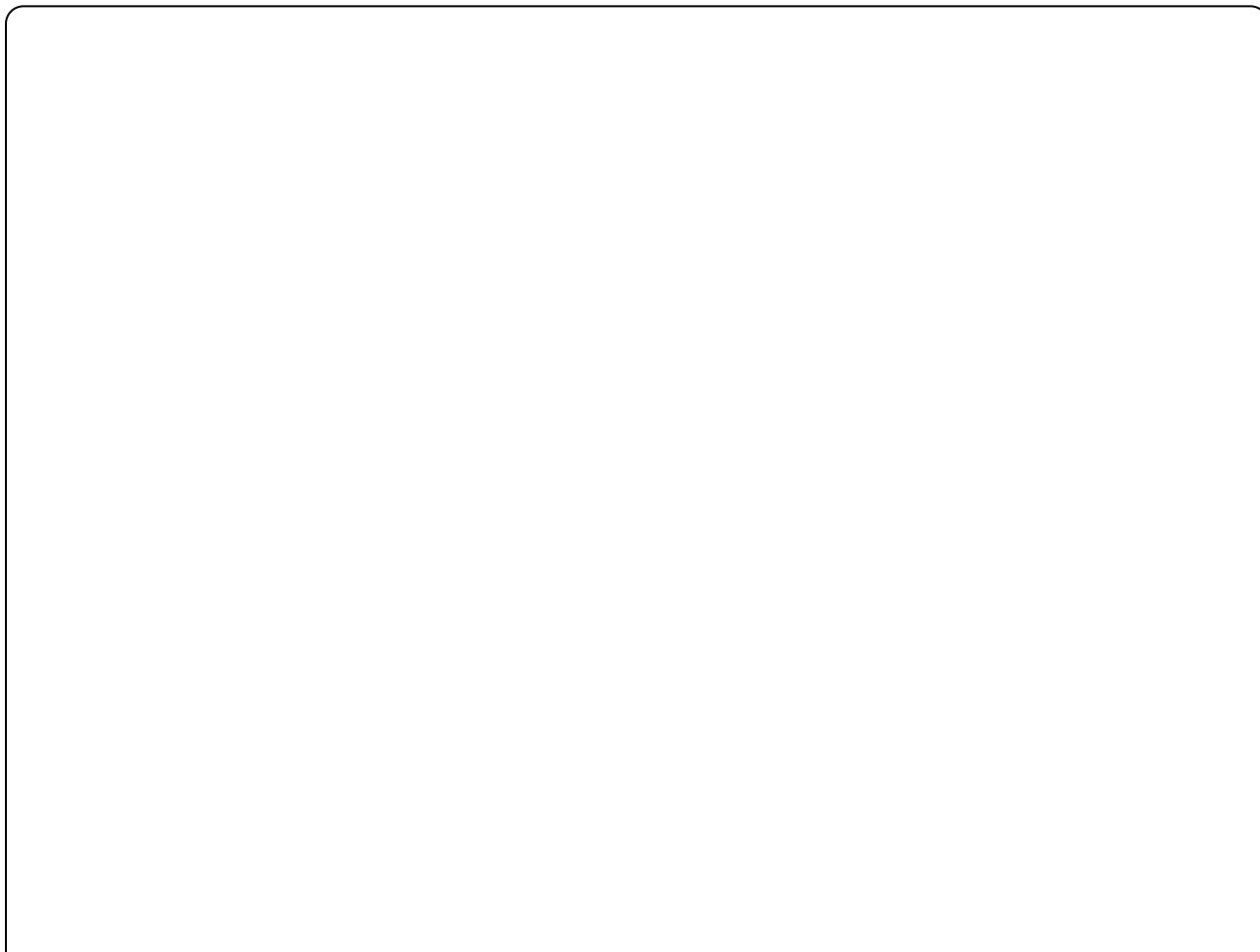
Homework 1



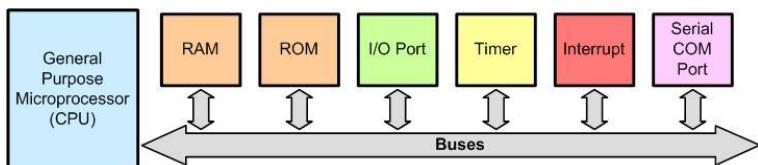
۲) وظیفه Instruction Set Architecture چیست و در بردارنده چه اطلاعاتی است؟



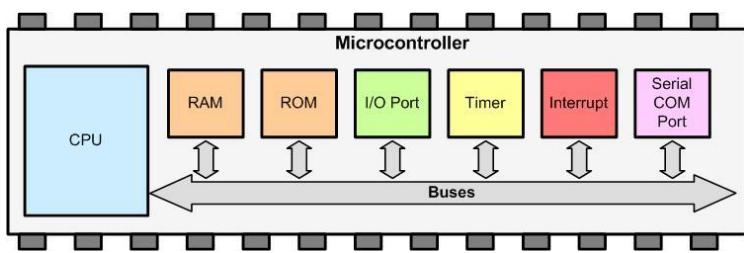
۳) تفاوت Microprocessor و Microcontroller چیست؟

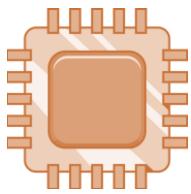


Microprocessor



Microcontroller (MCU)





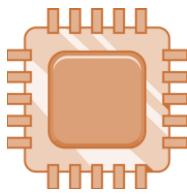
MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

Homework 1



۴) وقهه های تودرتو در ARM چگونه انجام می شود؟ (در پاسخ شما باید روند آن و نقش اجزایی همانند stack و nvic و priority register بیان شده باشد.)



MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

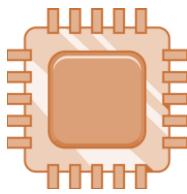
Homework 1



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

۵) وقفه‌ها با Procedure Call ها شباهت بسیاری دارند، چهار مورد از خصوصیات وقفه‌ها را نام ببرید.

۶) اگر حین یک ناحیه بحرانی وقفه‌ای از راه برسد چه اتفاقی می‌افتد؟ اگر در حین سرویس دادن به یک وقفه، وقفه دیگری از راه برسد چطور؟



MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

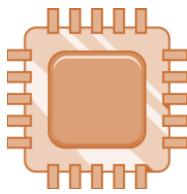
Homework 1



۷) در NVIC وقفه‌ها در چهار حالت می‌توانند باشند. آنها را نام برد و مختصری توضیح دهید.

۸) وقفه‌ها به دو گروه pulse و level-sensitive هم دسته‌بندی می‌شوند. تفاوت این وقفه‌ها در چیست؟

(امتیازی)



- مهلت ارسال تمرین ساعت ۲۳.۵۵ روز یکشنبه بیستم مهر می‌باشد.
- سوالات خود را می‌توانید از طریق ایمیل از تدریسیاران بپرسید.
- a.varaste.n@gmail.com
- behdadaspacey2099@gmail.com
- ارائه پاسخ تمرین به سه روش ممکن است:
 - (۱) استفاده از فایل **.docx**, تایپ پاسخ‌ها و ارائه فایل **Pdf**
 - (۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا
 - (۳) ارائه پاسخ‌ها به روش سنتی
- ارائه تمرین به روش اول و دوم به ترتیب شامل ۱۰٪ و ۵٪ نمره امتیازی می‌گردد.
- فایل پاسخ تمرین را تنها با قالب **HW1-G#-9731***.pdf** در مودل بارگزاری کنید.
- فایل زیپ ارسال نکنید.



MICROPROCESSOR AND
ASSEMBLY LANGUAGE
Dr. Farbeh

Homework 1



۱) مجموعه دستورالعمل های RISC و CISC را مقایسه کنید و سپس جدول زیر را تکمیل کنید.

RISC: Reduced Instruction Set Computer

CISC : Complex Instruction Set Computer

پردازنهای RISC، دارای سهیله دستورات ساده و کامپونت یافته ای می‌باشند. این دستورات، فرستهای آبجکتیو لسته و بنابراین، اتصالات decade کرنخ آن را ساده‌تر ایام می‌نمایند. همچنان، دستورات RISC می‌توانند سطح دریک clock cycle افزایش نمایند.

پردازنهای CISC، معمولیه بزرگ و پیچیده‌ای از دستورات پردازنهای را در بر می‌گیرند و از فرسته مدل‌های آرسن (MC) متنوعی استفاده می‌کنند. (هر مرتب دستورات بینری ثابت نمی‌شوند).

همچنان، برخلاف پردازنهای RISC، دستورات در هیئتین کلاک ایام می‌شوند.

~~(برخلاف پردازنهای RISC، دستورات در هیئتین کلاک ایام می‌شوند.)~~

	RISC	CISC
اندازه دستورالعمل ها	تعداد دستورالعمل های کم با فرمت ثابت	تعداد بالای دستورالعمل ها با فرمت های متفاوت
نرخ کلاک	دستورات در یک کلاک اجرا میشوند	اجرای یک دستور ممکن است چندین کلاک طول بکشد.
نوع واحد کنترل	Hardwired	Microprogrammed
دسترسی به حافظه	دسترسی به حافظه از طریق دستورات مشخص (load/store) (انجام میشود. (تعداد کمتر دسترسی)	دستورات دسترسی به حافظه به صورت مشخص از باقی دستورات جدا نشده اند. (تعداد بالای دسترسی)
ایراد یابی	امکان خطای کمتر و پیچیدگی کمتری دارد؛ بنابراین دیباگ آن راحتتر است.	امکان خطای بالاتر و پیچیدگی بالا؛ به همین خاطر ایرادیابی آن مشکل تر است.
تعداد ثبات	تعداد بالای ثبات ها	تعداد پایین ثبات
تعداد ترانزیستور و مقدار مصرف انرژی	تعداد کمتر ترانزیستور و مصرف پایینتر انرژی	تعداد بالاتر ثبات نسبت به RISC و مصرف انرژی بیشتر
گذرگاه	گذرگاه های داده و دستورات جدا	گذرگاه مشترک
خط لوله	از خط لوله زیاد استفاده میشود و پیاده سازی آن راحتتر است.	پیاده سازی آن پیچیده است و زیاد استفاده نمیشود.
پشتیبانی از مدل های آدرس دهی	تعداد کمتر مدهای آدرس دهی	مدل های آدرس دهی متعددی در آن به کار رفته است.
Fetch & Decode	ساده تر انجام میشود	نسبت به RISC پیچیده تر است.
پیچیدگی	به طور کلی (از لحاظ دستورات و ..) ساده تر است	پیچیدگی بالاتری دارد.

MICROPROCESSOR AND
ASSEMBLY LANGUAGE

Dr. Farbeh

Homework 1



(۲) وظیفه Instruction Set Architecture چیست و در بردارنده چه اطلاعاتی است؟

- ISA: مجموعه‌ای از دستورات که توسط ماسین استفاده می‌شوند تا برنامه را اجرا نمایند.
- لهم در واقع کی واسطه بین ساخت افزار نرم افزار بسیار آرایه و زبان ماسین (Assembly) بسیار بود.
- ISA مک مخصوص ایجاد سازی سخت افزار را ارائه می‌ردد و تقسیم می‌کند که دستورات حلقه پیاده‌سازی شوند.
- و ISA دستورات، رجیسترها، دسترسی به حافظه، ID را مشخص می‌کند.
- همین درستور اولیه حیث نوع (OpCode) کارهای کاربردی ایجاد سیتم باشید سیستم تو منبع می‌گیرد.
- لهم مثلاً دستورات Load/Store، Arithmetic/logic، ...
- و ISA نوع طول دستورات، نوع بیاناتها در دستور، OpCode، نوع دسترسی به حافظه، فرمت عبارت،
- کلیونی قدرتمندی محدوده‌ها در دستور، مدل آدرس (Physical Address) و ... را هم مشخص می‌کند.

MICROPROCESSOR AND
ASSEMBLY LANGUAGE
Dr. Farbeh

Homework 1



(۳) تفاوت Microprocessor و Microcontroller چیست؟

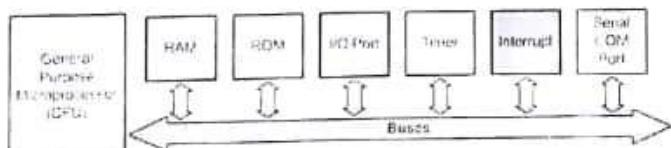
که Microprocessor، معملاً در دارای CPU است. اما که Microcontroller در جمله داخلی فقط سیستمی مانند ROM، RAM و ... را ندارد و این تفاوتات به صورت طاری بآن سهل محسوب شوند.

اما که Microcontroller در این سیستمی دارای قابلیتی مانند ROM، RAM و پورت های I/O و ... (برسیل پاسن آمده است).

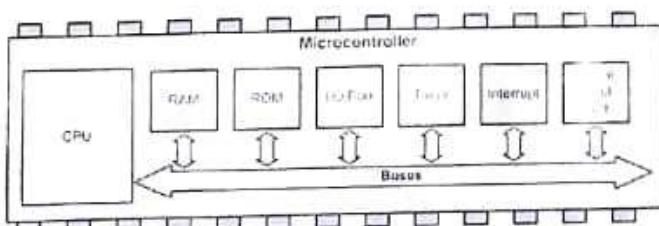
میکروکنترولرها برای انجام کم و فلیپه مخفف طرایی ساخته اند برای اجرای خانگی، توزیع ها و ... و چیزی مخفیهای آنها سخن دارند، مثلاً حافظه های زیارتی و پورت های I/O مقدار زیادی دارند و ...، اما ریز پردازنده ها و ظرفی مخفی مبارزه و مترانه صیغه های فلیپه را انجام نمی دارند. از ریز پردازنده های داخلی کامپیووتر های سطحی و ... استفاده می شود. در آن حالت ریز پردازنده ها، وظایف سختی را انجام می دهند، مثلاً RAM های بالا، پورت های I/O مقدار زیادی دارند.

همینطور که سرعت کلاک یک ریز پردازنده بینتر از میکروکنترولر است؛ میراث که کامپیوuter های سیمیه تری انجام می دهند.

Micropocessor



Microcontroller (MCU)



MICROPROCESSOR AND
ASSEMBLY LANGUAGE
Dr. Farbeh

Homework 1

۴) وقفه های تودرتو در ARM چگونه انجام می شود؟ (در پاسخ شما باید روند آن و نقش اجزایی همانند stack و nvic و priority register بیان شده باشد).

دستگاهی که بی دقت واریستور، CPU حالت یکتین خود را ذمیه کرده تا به دقته رسیدن نماید.
در درآفل stack، اطلاعات لازم را ذخیره می کند. (عمل stacking (اطلاعات مانند سفاره PC و ...))
حال آنرا درین امری بی دقت، دقته دلیری دس واریستور، با استفاده از مستود NVIC، وقفه های
تودرتو را هنوز می نشوند.
میتوانیم این رخدان داری، متراند اولویت بلای سرومه ابررسانی است. اولویت دقتها
با priority register و سختپن سفاره در ISR ها
ملکیت کارها، این اهمیت است که وقفه های بالاتر، زودتر از وقفه های
با اولویت کمتر انجام می شوند؛ هنوز اگر دقت، با اولویت کمتر زودتر آشده باشد و در حالت Active باشد.
(مانند حالت) دقته بالاتر است، زودتر انجام می شوند و سپس به دقته دلیر پرداخت می شوند.
او سیاست tail-chaining هم استفاده می کند.

لئن دستگاه NVIC مربوط به دقته بالاتر است، بر این رسم و دقته دلیر حالت pending

(ست بحای) باز آن حالت یکتین CPU از stack، مستقیماً رسیدن به دقته دوم
می پردازد.

MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

Homework 1



۵) وقفه‌ها با (ال.) Procedure ها شباهت بسیاری دارند، چهار مورد از خصوصیات وقفه‌ها را نام ببرید.

۱. وقفه‌ها میتوانند بین هر دو دستور در بین دو دستور اجرا یابند (در حالی که procedure ها را میتوانند تراویراند)
۲. این وقفه‌ها از دین سینه در حال اجرا ممکن است (transparent) نامهای است (له خلی - اتفاقاً کاریش مستفاده نمی‌کند).
۳. این وقفه‌ها عمرها به رخواست سینه در حال اجرا است (با هفتمین وقفه بحاطر event بین‌دنی که ممکن است رخ دهد).
۴. سینه نفعی، تعیین ایندیکاتور هفتم وقفه (چیزی که حافظه می‌بردند) هر وقفه ۱۵۰۰ مخصوصاً دارد (ال. call procedure ها در هفتم وقفه، تضابع وقفه بین طاسه سینه در حال اجرا).

۶) اگر حین یک ناحیه بحرانی وقفه‌ای از راه برسد چه اتفاقی می‌افتد؟ اگر در حین سرویس دادن به یک وقفه، وقفه دیگری از راه برسد چطور؟

بالنسبة NVIC، این قابلیت را بمالدوده (Priority Masking) برخواهد که کمترین (معنی آنها را نماید که بزرگ‌تر باشند) و محدوده نداشته باشند که توسط وقفه‌ها دیگر امتحان شوند،

این بخواهیم داشت که ناحیه اجرا شده و سینه را باشند که توسط وقفه‌ها دیگر امتحان شوند، برخواهد priority masking را که کمترین (معنی آنها را نماید که بزرگ‌تر باشند) و محدوده نداشته باشند که توسط وقفه‌ها دیگر امتحان شوند، برخواهد ورکار سینه خالی نمایند. به این فرم از ناحیه بحرانی، حالت آمن (interrupt) را به طایعت داری بخواهیم می‌کردیم درین آنچنانچه وقفه‌ها را می‌گذاشتند.

آخرین سه وسیله داریم که وقفه، وقفه دیگری برسم، اولویت وقفه دوم کمتر است، و آن اولویت آن سالات باشد، به آن سه وسیله داریم که وقفه اولی نباید بزرگ‌دد.

MICROPROCESSOR AND
ASSEMBLY LANGUAGE

Homework 1

Dr. Farbeh



۷) در NVIC وقفه‌ها در چهار حالت می‌توانند باشند. آنها را نام برد و مختصری توضیح دهید.

Inactive : زمانی که وقفه‌ای هنوز وارد نشده است. (معنی لکی به interrupt نهاده است)
در حال حاضر در خواسته نداده) → در این حالت، استیت مربوط به آن نوع وقفه، "غیرفعال" است.

pending : آن را سئون شان نداده وارد ISR می‌بینیم
interrupt : خواهد بود؛ این استیت CPU نسبت به آن را سئون شان نداده وارد ISR می‌بیند
مربوط به آن نشده است.

ISR : Active .
ISR مربوط به دفعه روحان امراز است.

Active & pending : A&P .
که ISR مربوط به کمی وقفه روحان امراز است،
آن را سئون نمایم که دلیل هم طوریست که در حالت Pending (مستور)

۸) وقفه‌ها به دو گروه pulse و level-sensitive هم دسته‌بندی می‌شوند. تفاوت این وقفه‌ها در چیست؟

(امنیتی)

edge-triggered → pulse ↗

pulse-triggered →

Pulse-interrupt ها، هفچاهی لبی بالا (روزه، پاسخ روزه) ملاک برداشته باید، ISR مستمر بـ دلالت
مکنه است، level-sensitive ، ISR بازهم بـ سطح سلیمان (که + High است) را می‌کند.

pulse محسس به

level-sensitive : هم حساس به سطح



10/12/2020

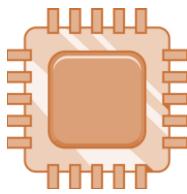


Homework 2

Lec 5-6



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE



MICROPROCESSOR AND ASSEMBLY LANGUAGE

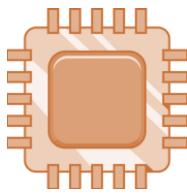
Dr. Farbeh

Homework 2



۱) مقاومت های Pull-Down و Pull-Up چه کاربردی دارند؟

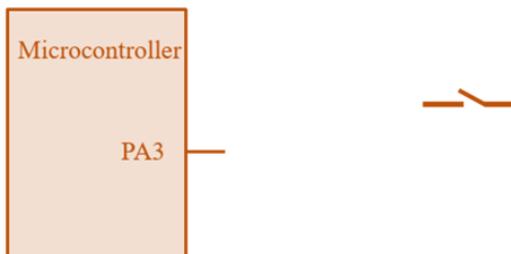
۲) چه زمانی به کار بردہ می شود؟ Debouncing Filter

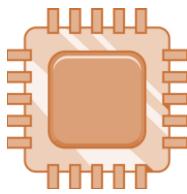


۳) می خواهیم از پایه PA3 ورودی بگیریم. کلید نشان داده در شکل زیر را به گونه‌ای به این پایه متصل کنید که هنگامی که برنامه از این پایه صفر منطقی را می‌خواند کلید فشار داده شده باشد و هنگامی که ۱ منطقی را می‌خواند کلید رها شده (آزاد) باشد. در صورتی که برنامه به گونه‌ای رجیستر های PIOA را پیکربندی کرده باشد که مقدار رجیستر PUSR

a. برابر با 0x0004 باشد.

b. برابر با 0x0000 باشد.





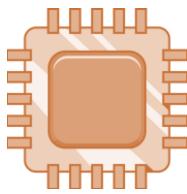
MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

Homework 2



۴) برای آنکه پایه ۸ از واحد کنترلر PIOA (PA8) را بخواهیم به صورت خروجی پیکربندی شود و حالت Open Pull-up Resistor و Drain (Multi Drive) غیر فعال باشد، مقادیری را که در رجیسترها لازم باید نوشته شوند مشخص کنید.



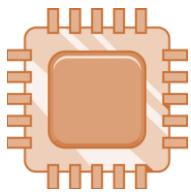
۵) برای آنکه پایه PA8 را به صورت ورودی پیکربندی و مقاومت Pull-Up درونی آن فعال باشد ، مقدار هایی را که باید در رجیستر های مورد نیاز نوش特 مشخص کنید.

امتیازی: برای فعال سازی وقفه حساس به لبه بالا روندهی (Rising Edge Interrupt) نیز مقداردهی رجیستر های لازم را انجام دهید به گونه ای که اولویت گروهی (Group Priority) و زیر اولویت (Sub-priority) وقفه متناظر با این واحد کنترلر PIO را برابر با ۳ قرار دهید.

توجه کنید که برای فعال کردن این وقفه باید هم در واحد NVIC و هم در واحد کنترلر PIO پیکربندی های لازم انجام شود. یعنی برای نمونه اگر وقفه لبه بالا رونده برای پایه ای پیکربندی شود، هنگامی که سطح ولتاژ منطقی آن پایه از ۰ به ۱ تغییر می کند، وقفه واحد کنترلر PIO آن پایه به حالت Pending درمی آید، ولی برای آنکه پس از آن به حالت Active درآید و تابع گرداننده آن اجرا شود نیاز است که در NVIC نیز تنظیمات لازم انجام شود.

برای آنکه خانه های متناظر با PIOA را در رجیستر های NVIC مشخص کنید، باید شماره **PID Identification** Peripheral را بدست آورید.

همچنین در پیاده سازی معماری Cortex-M3 در میکرو کنترلر های AT91SAM از ۸ بیت اولویت وقفه تنها ۴ بیت آن به کار برده شده است، یعنی برای نمونه بیشترین مقدار برای اولویت گروهی برابر با 0b1111 یا ۱۵ می باشد که هنگامی رخ می دهد که بیت های اختصاص یافته به اولویت گروهی برابر با بیشترین مقدار خود یعنی ۴ بیت باشد. در صورت نیاز به اطلاعات بیشتر می توان برای پیکربندی رجیسترها کلیدواژه ها را در راهنمای جستجو کرد.

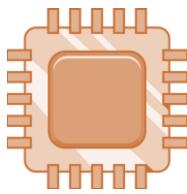


MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

Homework 2





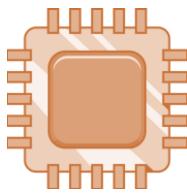
MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

Homework 2



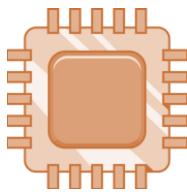
- ۶) گروه بندی اولویت (Priority Grouping) چیست و در کدام رجیستر پیکربندی می‌شود؟ (پاسخ شما باید مفهوم "واگیری" (Pre-emption) را در برداشته باشد).



۷) بیشنیه شمار وقفه هایی که NVIC می تواند پشتیبانی کند چند است؟ اولویت وقفه ها ایستا می باشد یا پویا؟

۸) چنانچه در هنگام اجرای روال وقفهی شماره A با اولویت بندی های زیر وقفه های B , C , D و E نیز با ویژگی های زیر به صورت همزمان رخ داده باشند، ترتیب اجرای روال وقفه ها را مشخص کنید. (امتیازی)

	زیر/اولویت	نام وقفه	اولویت گروهی
B	۱	.	
C	۱	۳	
D	۲	۱	
E	۳	.	



- مهلت ارسال تمرین ساعت ۲۳.۵۵ روز یکشنبه چهارم آبان می‌باشد.
- سوالات خود را می‌توانید تنها از طریق ایمیل زیر بپرسید.
 - alirezasalehy@aut.ac.ir
- برای پاسخ دهی به سوال‌های ۵ و ۴ باید بخش هایی از فصل 10 (ARM Cortex-M3 processor) و (PIO 31 , (ARM Cortex-M3 processor) را از راهنمای میکروکنترلر های ATSAM3 (Peripherals) ۹ را از راهنمای میکروکنترلر های ATSAM3 در صورت نیاز بررسی کنید.
- همچنین برای پاسخ دهی به سوال‌های ۷ و ۵ و ۶، فصل ۵ (Exceptions) و فصل ۸ (NVIC) را از راهنمای معماری ARM Cortex-M3 بررسی کنید.
- این منابع در مودل درس در بخش Books قرار داده شده است.
 - ارائه پاسخ تمرین به سه روش ممکن است:
 - (۱) استفاده از فایل .docx. تایپ پاسخ‌ها و ارائه فایل Pdf
 - (۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا
 - (۳) ارائه پاسخ‌ها به روش سنتی
- ارائه تمرین به روش اول و دوم به ترتیب شامل ۱۰٪ و ۵٪ نمره امتیازی می‌گردد.
- فایل پاسخ تمرین را تنها با قالب **HW2-G#-9731***.pdf** در مودل بارگزاری کنید.
- نمونه: HW2-G2-9531747
- فایل زیپ ارسال نکنید.



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

10/12/2020



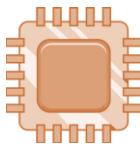
Homework 2

Lec 5-6



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2020



(۱) مقاومت های Pull-Down و Pull-Up چه کاربردی دارند؟

عملکرد مدارهای منطقی به گونه ای است که یک باز ولتاژی را برابر با ۰ منطقی و یک بازه دیگر را برابر با ۱ منطقی در نظر میگیرند. برای نمونه سطح منطقی و بازه ولتاژ منتظر با آن در دو تکنولوژی در جدول زیر آورده شده است.

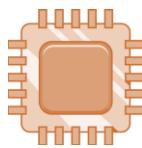
Examples of binary logic levels

Technology	L voltage	H voltage	Notes
CMOS ^[3]	0 V to 1/3 V _{DD}	2/3 V _{DD} to V _{DD}	V _{DD} = supply voltage
TTL ^[3]	0 V to 0.8 V	2 V to V _{CC}	V _{CC} = 5 V ±10%

از این رو برای اینکه بتوان به درستی ورودی دریافت کرد باید ورودی این مدارها در بازه ولتاژی مناسبی قرار داشته باشد در غیر این صورت امکان تشخیص ورودی نادرست در گیت های منطقی وجود دارد. که این تشخیص نادرست در تشخیص ورودی دلخواه و مورد انتظار تاثیر منفی می گذارد. از این رو باید از ورودی دادن به گیت های منطقی در بازه های ولتاژی که منتظر با هیچ یک از سطح های منطقی نیست، پیشگیری کرد. برای اینکار ورودی را به یک ولتاژ مطمئن مثل ۰ یا ولتاژ تغذیه با یک مقاومت وصل می کنند. از آن جا که ورودی می تواند هر دو حالت صفر و یک را داشته باشد برای جلوگیری از اتصال کوتاه در مدار ورودی از این مقاومت ها استفاده می شود.

(۲) چه زمانی به کار برد می شود؟ Debouncing Filter

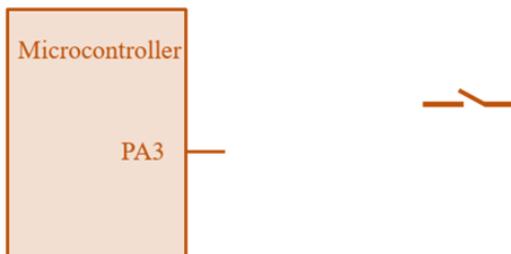
هنگامی که کلید مکانیکی فشرده یا رها می شود ممکن است که خروجی برای مدتی نوسان کند، در برخی از کاربردها این نوسان ها ممکن است منجر به تشخیص نادرست شوند و نهایتاً رفتار های تاخوسته و نامنتظره گردد. برای نمونه در دستگاه *Treadmill* هنگامی که بخواهیم کلید *Speed-Up* را فشار دهیم، ممکن است این نوسان ها منجر به افزایش ناگهانی و بیش از اندازه سرعت شوند. زمانی که به خواهیم این نوسان ها را فیلتر کنیم از این روش استفاده می شود.



(۳) می خواهیم از پایه PA3 ورودی بگیریم. کلید نشان داده در شکل زیر را به گونه‌ای به این پایه متصل کنید که هنگامی که برنامه از این پایه صفر منطقی را می‌خواند کلید فشار داده شده باشد و هنگامی که ۱ منطقی را می‌خواند کلید رها شده (آزاد) باشد. در صورتی که برنامه به گونه‌ای رجیستر های PIOA را پیکربندی کرده باشد که مقدار رجیستر PUSR

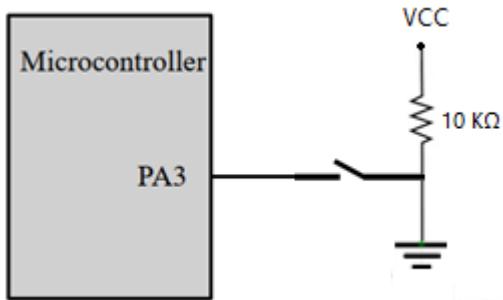
a. برابر با 0x0008 باشد.

b. برابر با 0x0000 باشد.

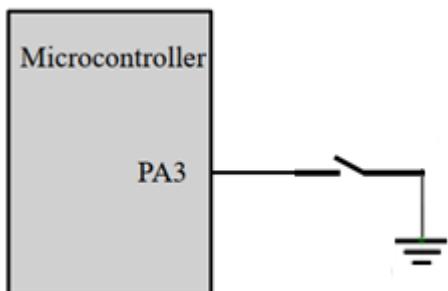


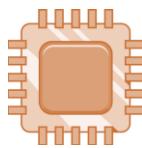
(توجه داشته باشید که شماتیک را در هر حالت با کمترین قطعه های ممکن بکشید.)

الف) همان طور که در صورت سوال گفته شده است با توجه به مقدار رجیستر PUSR مقاومت بالا کشنده برای پایه PA3 غیرفعال شده است. در نتیجه باید به صورت بیرونی مدار را به پایه افزود. (در پاسخ باید مقدار مقاومت نیز مشخص گردد و عدد معقولی باشد)



(ب)

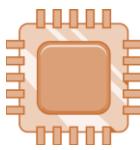




۴) برای آنکه پایه ۸ از واحد کنترلر (PA8) PIOA را بخواهیم به صورت خروجی پیکربندی شود و حالت Open Drain را که در رجیسترها لازم باید نوشته شوند مشخص کنید.

```
#define PIN (1ul << 8)           // 0000...100000000
#define PIOA_PER 0x400E0E00ul      // Write Only
#define PIOA_OER 0x400E0E10ul      // Write Only
#define PIOA_PUDR 0x400E0E60ul     // Write Only
#define PIOA_MDDR 0x400E0E54ul     // Write Only

*((uint32_t *) PIOA_PER) = reg;    // PIOA_PSR: XXXX...XX1XXXXXXXXX
*((uint32_t *) PIOA_OER) = reg;    // PIOA_OSR: XXXX...XX1XXXXXXXXX
*((uint32_t *) PIOA_PUDR) = reg;   // PIOA_PUSR: XXXX...XX1XXXXXXXXX
*((uint32_t *) PIOA_MDDR) = reg;   // PIOA_MDSR: XXXX...XX0XXXXXXXXX
```



۵) برای آنکه پایه PA8 را به صورت ورودی پیکربندی و مقاومت Pull-Up درونی آن فعال باشد ، مقدار هایی را که باید در رجیستر های مورد نیاز نوشته مشخص کنید.

امتیازی: برای فعال سازی وقفه حساس به لبه بالا روندهی (Rising Edge Interrupt) نیز مقداردهی رجیستر های لازم را انجام دهید به گونه ای که اولویت گروهی (Group Priority) و زیر اولویت (Sub-priority) وقفه متناظر با این واحد کنترلر PIO را برابر با ۳ قرار دهید.

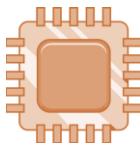
توجه کنید که برای فعال کردن این وقفه باید هم در واحد NVIC و هم در واحد کنترلر PIO پیکربندی های لازم انجام شود. یعنی برای نمونه اگر وقفه لبه بالا رونده برای پایه ای پیکربندی شود، هنگامی که سطح ولتاژ منطقی آن پایه از ۰ به ۱ تغییر می کند، وقفه واحد کنترلر PIO آن پایه به حالت Pending درمی آید، ولی برای آنکه پس از آن به حالت Active درآید و تابع گرداننده آن اجرا شود نیاز است که در NVIC نیز تنظیمات لازم انجام شود.

برای آنکه خانه های متناظر با PIOA را در رجیستر های NVIC مشخص کنید، باید شماره **PID Identification** Peripheral را بدست آورید.

همچنین در پیاده سازی معماری Cortex-M3 در میکرو کنترلر های AT91SAM از ۸ بیت اولویت وقفه تنها ۴ بیت آن به کار برده شده است، یعنی برای نمونه بیشترین مقدار برای اولویت گروهی برابر با 0b1111 یا ۱۵ می باشد که هنگامی رخ می دهد که بیت های اختصاص یافته به اولویت گروهی برابر با بیشترین مقدار خود یعنی ۴ بیت باشد. در صورت نیاز به اطلاعات بیشتر می توان برای پیکربندی رجیسترها کلیدواژه ها را در راهنمای جستجو کرد.

```
#define PIN (1ul << 8)           // 0000...100000000
#define PIOA_PER 0x400E0E00ul      // Write Only
#define PIOA_OER 0x400E0E10ul      // Write Only
#define PIOA_PUER 0x400F0E64ul     // Write Only

*((uint32_t *) PIOA_PER) = reg;   // PIOA_PSR: XXXX...XX1XXXXXXXXX
*((uint32_t *) PIOA_ODR) = reg;   // PIOA_OSR: XXXX...XX0XXXXXXXXX
*((uint32_t *) PIOA_PUER) = reg;  // PIOA_PUSR: XXXX...XX0XXXXXXXXX
```



امتیازی

```
#define PIOA_IER 0x400E0E40ul      // Write Only
#define PIOA_AIMER 0x400E0EB0ul     // Write Only
#define PIOA_ESR 0x400E0EC0ul      // Write Only
#define PIOA_REHLSR 0x400E0ED4ul    // Write Only

*((uint32_t *) PIOA_IER) = reg;   // PIOA_IMR:      XXXX...XX1XXXXXXXXX
*((uint32_t *) PIOA_AIMER) = reg; // PIOA_AIMMR:   XXXX...XX1XXXXXXXXX
*((uint32_t *) PIOA_ESR) = reg;   // PIOA_ELSR:    XXXX...XX0XXXXXXXXX
*((uint32_t *) PIOA_REHLSR) = reg; // PIOA_FRLHSR: XXXX...XX1XXXXXXXXX

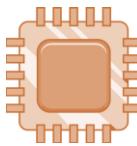
#define PIOA_PID 11

#define NVIC_ISER0 0xE000E100ul      // Read Write
#define NVIC_IPR2 0xE000E408ul       // Read Write

#define SCB_AIRCR 0xE000ED0Cul      // Read Write*
#define AIRCR_VECTKEY 0x05FAul
#define PRIGROUP_VALUE 0b101ul
// With this value for PRIGROUP, group priority and
// sub-priority fields in IP[n] is configured as
// bxx.yy0000

*((uint32_t *) NVIC_ISER0) |= 1ul << PIOA_PID; // xxx...xxxx1xxxxxxxxxxxxx
*((uint32_t *) NVIC_IPR2) |= 0b11110000ul << 24; // xxxxxxxx11110000xx...xx

reg = *((uint32_t *) SCB_AIRCR);           // value is saved for processing
reg &= ~((uint32_t)0xFFul << 16 | 0x07ul << 8); // Clears PRIGROUP & VECTKEY fields
reg |= AIRCR_VECTKEY << 16 | PRIGROUP_VALUE << 8; // Writes KEY & PRIGROUP
fields
*((uint32_t *) SCB_AIRCR) = reg;           // Result is written back to AIRCR
```



۶) گروه بندی اولویت (Priority Grouping) چیست و در کدام رجیستر پیکربندی می‌شود؟ (پاسخ شما باید مفهوم "واگیری" (Pre-emption) را در برداشته باشد.)

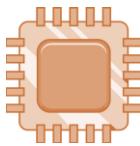
گروه بندی اولویت روشی است که در آن بیت‌های اولویت اختصاص یافته به هر وقفه را که در رجیستر‌های *Interrupt Priority* جای دارند، به دو گروه اولویت گروهی و زیر اولویت بخش بندی می‌کند. که شمار بیت‌های اختصاص یافته به هر یک از این دو گروه بستگی به مقدار بیت‌های *Priority Grouping Register* در رجیستر *Application Interrupt and Reset Control Register* دارد.

Table 5-3 Priority grouping

Interrupt priority level field, PRI_N[7:0]					
PRIGROUP[2:0]	Binary point position	Pre-emption field	Subpriority field	Number of pre-emption priorities	Number of subpriorities
b000	bxxxxxx.y	[7:1]	[0]	128	2
b001	bxxxxx.yy	[7:2]	[1:0]	64	4
b010	bxxxx.yyy	[7:3]	[2:0]	32	8
b011	bxxxx.yyy	[7:4]	[3:0]	16	16
b100	bxxx.yyyyy	[7:5]	[4:0]	8	32
b101	bxx.yyyyyy	[7:6]	[5:0]	4	64
b110	bx.yyyyyy	[7]	[6:0]	2	128
b111	b.yyyyyyy	None	[7:0]	0	256

به اولویت گروهی، اولویت واگیری نیز گفته می‌شود و همان‌گونه که در جدول بالا نیز دیده می‌شود بیت‌های پر ارزش تر را در *PRI_N* در بر می‌گیرد. وقفه‌هایی که اولویت گروهی یکسانی دارند در یک گروه اولویت جای می‌گیرند. اولویت با مقدار بیت‌های *sub group priority* و *priority* رابطه عکس دارد. و چنانچه روال سرویس وقفه A با مقدار *group priority* در حال اجرا باشد که وقفه B با مقدار *group priority* کمتر اعلان گردد و به حالت *pending* درآیند. در صورتی که وقفه B در *NVIC* پیشتر شده باشد. در این صورت با واگیری اجرا از سرویس وقفه A، اجرا به سرویس وقفه B اختصاص داده می‌شود و رجیستر *Enabled* می‌گردد تا پس از پایان یافتن روال سرویس B در صورت امکان اجرا خود را دنبال کند. از این رو به این اولویت گروهی اولویت واگیری نیز گفته می‌شود زیرا با توجه به مقدار آن می‌توان مشخص کرد که چه وقفه‌هایی می‌توانند در صورت اعلان باعث واگیری اجرا وقفه‌های دیگر شوند.

زیر اولویت اما واگیری را تعیین نمی‌کند بلکه چنانچه چند وقفه که در یک گروه اولویت می‌باشند همزمان *Pending* و *Enabled* باشند در این صورت زیر اولویت تعیین می‌کند که ترتیب اجرای این وقفه‌ها چگونه باشد. و نخست وقفه‌ی با کمترین مقدار زیر اولویت اجرا خواهد شد و سپس وقفه‌های با مقدار زیر اولویت بزرگ تر.



۷) بیشنیه شمار وقفه هایی که NVIC می تواند پشتیبانی کند چند است؟ اولویت وقفه ها ایستا می باشد یا پویا؟

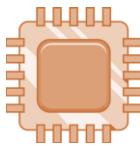
بیشنیه شمار وقفه های خارجی که معماری Cortex-M3 پشتیبانی می کند ۲۴۰ می باشد، جایگاه های ۰ تا ۱۵ در جدول بردار وقفه برای وقفه های درونی پردازنده می باشد. از این رو شمار کل وقفه هایی که در این معماری می تواند وجود داشته باشد. ۲۵۶ می باشد.

اولویت ها به صورت پویا پیکربندی می شود. که می توان با مقدار دهی کردن خانه های متناظر به هر وقفه بیرونی در رجیستر های Application Interrupt and PRIGROUP و با توجه به مقدار خانه های Interrupt Priority Registers آن را برنامه ریزی کرد.

۸) چنانچه در هنگام اجرای روال وقفه شماره A با اولویت بندی های زیر وقفه های B , C , D و E نیز با ویژگی های زیر به صورت همزمان رخ داده باشند، ترتیب اجرای روال وقفه ها را مشخص کنید. (امتیازی)

	نام وقفه	اولویت گروهی	زیر اولویت
A		1	4
B		1	.
C		1	3
D		2	1
E		3	.

A -> B -> C -> D -> E



- مهلت ارسال تمرین ساعت ۲۳.۵۵ روز یکشنبه چهارم آبان می‌باشد.
- سوالات خود را می‌توانید تنها از طریق ایمیل زیر بپرسید.
 - alirezasalehy@aut.ac.ir
- برای پاسخ دهی به سوال‌های ۴ و ۵ باید بخش هایی از فصل ۱۰ (ARM Cortex-M3 processor) و (PIO) ۳۱ را از راهنمای میکروکنترلر های ATSAM3 (Peripherals) ۹ را از راهنمای میکروکنترلر های ATSAM3 در صورت نیاز بررسی کنید.
- همچنین برای پاسخ دهی به سوال‌های ۶ و ۷ و ۸، فصل ۵ (Exceptions) و فصل ۸ (NVIC) را از راهنمای معماری ARM Cortex-M3 بررسی کنید.
- این منابع در مودل درس در بخش Books قرار داده شده است.
- ارائه پاسخ تمرین به سه روش ممکن است:
 - (۱) استفاده از فایل .docx. تایپ پاسخ‌ها و ارائه فایل Pdf
 - (۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا
 - (۳) ارائه پاسخ‌ها به روش سنتی
- ارائه تمرین به روش اول و دوم به ترتیب شامل ۱۰٪ و ۵٪ نمره امتیازی می‌گردد.
- فایل پاسخ تمرین را تنها با قالب **HW2-G#-9531***.pdf** در مودل بارگزاری کنید.
- نمونه: HW2-G2-9531747
- فایل زیپ ارسال نکنید.

10/25/2020



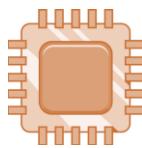
Homework 3

Lec 7-8

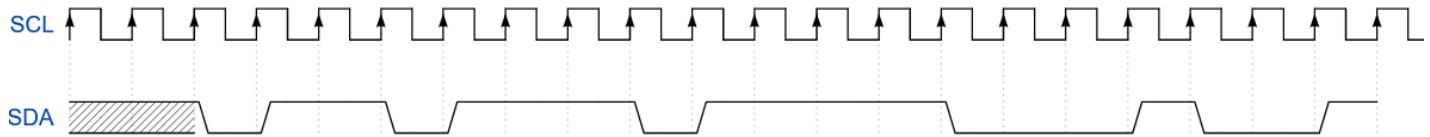


MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2020

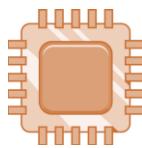


- ۱) در شکل موج زیر که توسط یک رابط I2C روی بس قرار می‌گیرد، آدرس Slave، محتوا داده ارسالی، نوع عملیات (Read/Write) و سایر بیت‌ها را مشخص کنید. (فرض کنید بس در حالت idle بوده است و آمدن این رشته بیتی شروع کار است.)

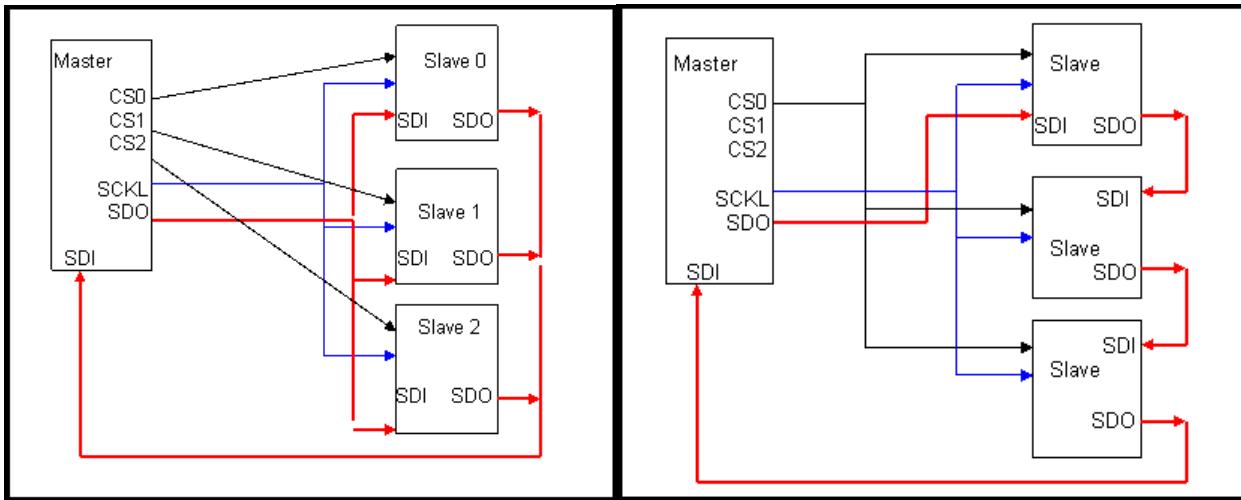


- ۲) با توجه به این که هر دو رابط I2C و SPI مناسب انتقال داده با سرعت پایین و تست و دیباگ هستند، به همراه دلیل مشخص کنید که برای هر کدام از سناریوهای زیر، کدام یک از این دو رابط مناسب‌تر است:

- ۱) یک Master در ارتباط با یک Slave
- ۲) یک Master در ارتباط با چند Slave
- ۳) چندین Master در ارتباط با یک یا چند Slave

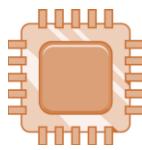


۳) برای پیاده‌سازی یک زنجیره از دستگاه‌ها به کمک رابط I2C، دو تولوژی زیر پیشنهاد شده است. این دو روش را مقایسه کنید و مزایا و معایب هر کدام را ذکر کنید:



روش دوم

روش اول



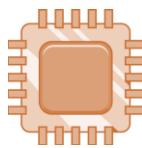
MICROPROCESSOR AND ASSEMBLY LANGUAGE

Dr. Farbeh

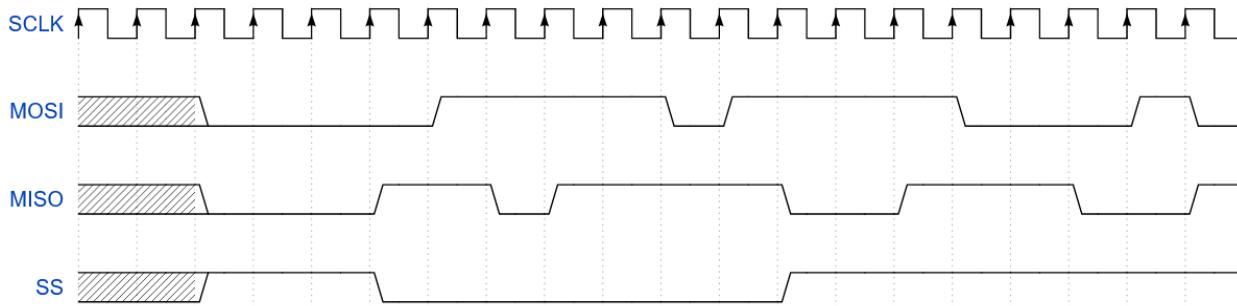
Homework 3



۴) مزایا و محدودیت‌های رابط I2C را نسبت به رابط SPI بیان کنید.



۵) در شکل موج زیر که توسط یک رابط SPI روی باس قرار می‌گیرد، زمان شروع و پایان ارسال داده، محتوا داده ارسالی و نوع هر عملیات (نوشتن روی Slave یا Master) را مشخص کنید.



مهلت ارسال تمرین ساعت ۲۳.۵۵ روز یکشنبه هجدهم آبان می‌باشد.

سوالات خود را می‌توانید تنها از طریق ایمیل زیر بپرسید.

○ M.ChamanMotlagh@gmail.com

ارائه پاسخ تمرین به سه روش ممکن است:

(۱) استفاده از فایل .docx. تایپ پاسخ‌ها و ارائه فایل Pdf

(۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا

(۳) ارائه پاسخ‌ها به روش سنتی

ارائه تمرین به روش اول و دوم به ترتیب شامل ۱۰٪ و ۵٪ نمره امتیازی می‌گردد.

فایل پاسخ تمرین را تنها با قالب **HW3-G#-9*****.pdf** در مدل بارگزاری کنید.

نمونه: HW3-G2-9531888

فایل زیپ ارسال نکنید.

10/25/2020



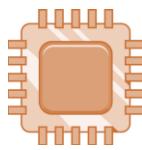
Homework 3

Lec 7-8

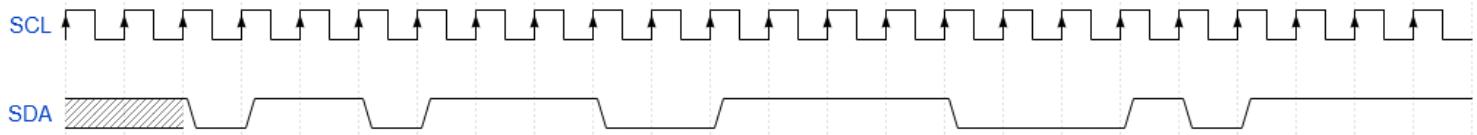


MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2020



۱) در شکل موج زیر که توسط یک رابط I2C روی باس قرار می‌گیرد، آدرس Slave، محتوا داده ارسالی، نوع عملیات (Read/Write) و سایر بیت‌ها را مشخص کنید. (فرض کنید باس در حالت idle بوده است و آمدن این رشته بیتی شروع کار است.)



رشته بیت قرار گرفته روی SDA به صورت زیر است:

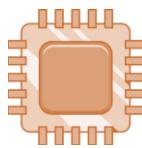
0-0110111-0-0-111100010-0-1

به ترتیب از چپ به راست معنا هر بخش به صورت زیر است:

0 به منظور بیت شروع، مقدار 0110111 به منظور آدرس (0x37)، مقدار 0 به معنی خواندن دیتا در نوشتن توسط Master، بیت 0 بعدی به منظور Ack است، مقدار 111100010 (0x742=) داده Slave ارسالی است، مقدار 0 به منظور ACK در پایان داده‌ها و مقدار 1 پایانی به منظور بیت پایان انتقال است.

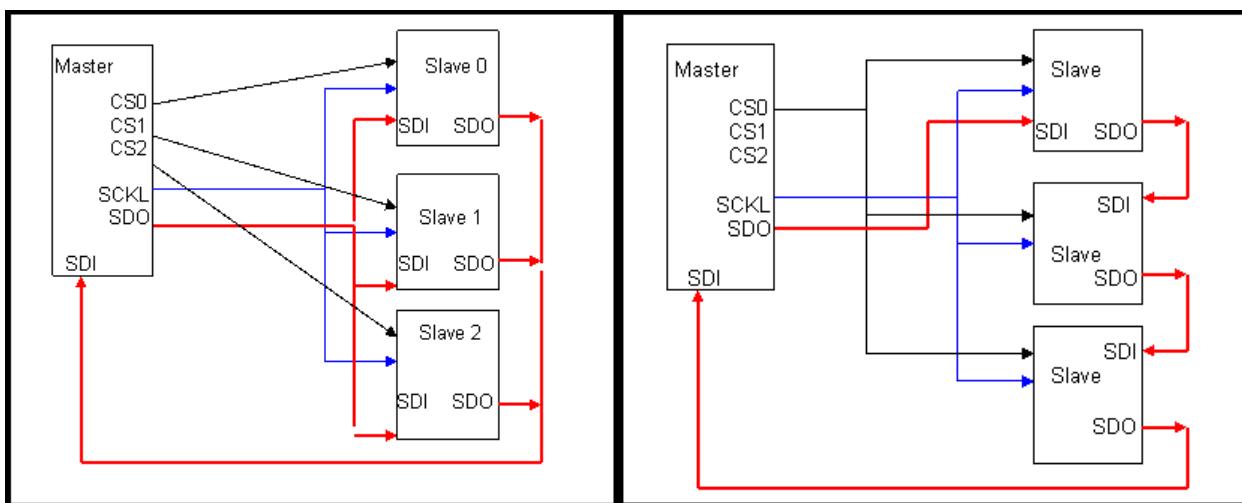
۲) با توجه به این که هر دو رابط I2C و SPI مناسب انتقال داده با سرعت پایین و تست و دیباگ هستند، به همراه دلیل مشخص کنید که برای هر کدام از سناریوهای زیر، کدام یک از این دو رابط مناسب‌تر است:

- یک Slave در ارتباط با یک Master
- یک Slave در ارتباط با چند Master
- چندین Slave در ارتباط با یک یا چند Master

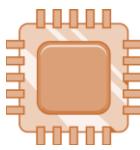


۱. در این حالت، کار با رابط SPI ساده‌تر است و همچنین به دلیل عدم نیاز به آدرس دهی، بازدهی و سرعت ارتباط در SPI بالاتر است.
۲. در این حالت SPI نیاز به سخت افزار (پیاده سازی SS‌ها) و پیکره‌بندی بیشتری نسبت به I2C دارد و اگر تعداد Slave‌ها زیاد باشد، I2C خیلی بصره‌تر است.
۳. برای این حالت تنها گزینه قابل قبول I2C است و استفاده از SPI در حالت معمول امکان‌پذیر نیست.

(۳) برای پیاده‌سازی یک زنجیره از دستگاه‌ها به کمک رابط SPI، دو تopolوژی زیر پیشنهاد شده است. این دو روش را مقایسه کنید و مزایا و معایب هر کدام را ذکر کنید:



روش اول نیاز به CS (یا SS) کمتری نسبت به روش دوم دارد، بنابراین هزینه پیاده‌سازی سخت افزاری آن پایین‌تر است. همچنین جایگذاری قطعات در روش اول ساده‌تر است چرا که تنها لازم است زنجیره تشکیل شود و هر قطعه به قطعه‌ی بعدی نزدیک باشد در حالی که در روش دوم، همه‌ی قطعات باید هم‌جواری مکانی داشته باشند. در مقابل بازدهی و سرعت روش دوم بالاتر است چرا که در روش اول، داده‌ها برای رسیدن از هر Master به Slave، تا انتهای مسیر را بروند و از Slave‌های دیگر عبور کنند. همچنین پیچیدگی کار با



در روش اول پیچیده‌تر است چرا که برای دسترسی به داده‌خاص، باید تشخیص داد که داده در کدام دستگاه است و چند کلک خواندن آن داده نیاز است.

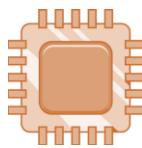
۴) مزایا و محدودیت‌های رابط I2C را نسبت به رابط SPI بیان کنید.

هر دو درگاه ارتباطی، مناسب انتقال داده‌ها با سرعت پایین هستند و مهم‌ترین استفاده آن‌ها تست و دیباگ است.

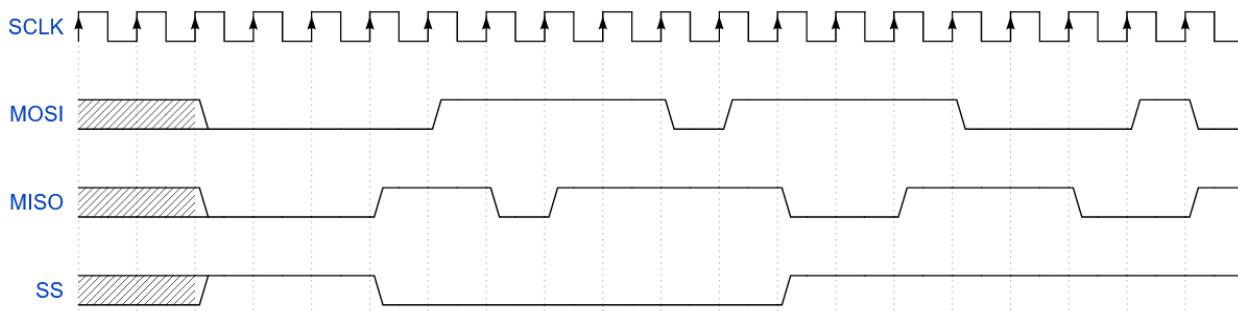
رابط SPI به دلیل نداشتن سیستم آدرس‌دهی و اطمینان از دریافت داده (نداشتن ACK و NACK)، برای Stream و انتقال داده‌ها در طولانی مدت بهتر است، در مقابل در استفاده از I2C اطمینان بالاتری از رسیدن یا نرسیدن داده‌ها وجود دارد و این درگاه برای خواندن و نوشتمن روی آدرس‌ها مناسب‌تر است.

رابط I2C سربار بیشتری دارد و قابلیت استفاده در حالت Full-duplex (ارتباط دو طرفه همزمان) را ندارد، بنابراین در حالت کلی سرعت SPI بالاتر از I2C است.

I2C امکان Multi Master بودن را در حالت معمول داراست، ولی در رابط SPI امکان انجام این کار وجود ندارد.



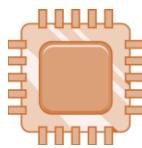
۵) در شکل موج زیر که توسط یک رابط SPI روی باس قرار می‌گیرد، زمان شروع و پایان ارسال داده، محتوا داده ارسالی و نوع هر عملیات (نوشتن روی Master یا Slave) را مشخص کنید.



در انتقال با استفاده از SPI، ارتباط بین Master و Slave زمانی وجود دارد که سیگنال SS (یا CS) مقدار صفر داشته باشد. بنابراین در شکل موج بالا، تنها در ۷ کلک انتقال داده صورت می‌گیرد. سیگنال‌های ارتباطی به صورت زیر هستند:

MOSI = 0111101 (=0x3D) → این سیگنال، داده انتقالی از مستر به اسلیو است.

MISO = 1101111 (=6F) → این سیگنال، داده انتقالی از اسلیو به مستر است.



• مهلت ارسال تمرین ساعت ۲۳.۵۵ روز یکشنبه هجدهم آبان می‌باشد.

• سوالات خود را می‌توانید تنها از طریق ایمیل زیر بپرسید.

○ M.ChamanMotlagh@gmail.com

• ارائه پاسخ تمرین به سه روش ممکن است:

(۱) استفاده از فایل **.docx**. تایپ پاسخ‌ها و ارائه فایل **Pdf**

(۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا

(۳) ارائه پاسخ‌ها به روش سنتی

• ارائه تمرین به روش اول و دوم به ترتیب شامل ۱۰٪ و ۵٪ نمره امتیازی می‌گردد.

• فایل پاسخ تمرین را تنها با قالب **HW3-G#-9*****.pdf** در مدل بارگزاری کنید.

• نمونه: HW2-G2-9531747

• فایل زیپ ارسال نکنید.



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

11/9/2020



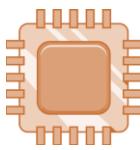
Homework 4

Lec 9-10-11



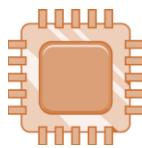
MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2020

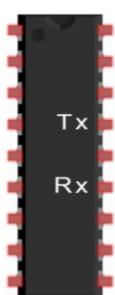


۱) میکروکنترلری در اختیار داریم که در بازه ولتاژی [-3.3, 3.3] کار می‌کند و می‌تواند ولتاژهای آنالوگ را در 256 محدوده گستته قرار دهد. حال سنسور آنالوگی به این میکروکنترلر وصل می‌کند که ولتاژی برابر V_{in} به میکروکنترلر منتقل می‌کند. اگر V_{in} برابر 2.2V باشد، مقدار عددی دیجیتال این سنسور (عددی بین 0 تا 255) چند خواهد بود؟

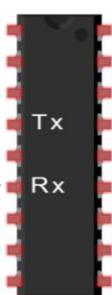
۲) قصد داریم از طریق رابط UART یک میکروکنترلر را به یک Device متصل کنیم. هدف ما ارسال داده‌ها با نرخ 8 kb/s است ولی در کنار داده‌ها نیازمند به بیت Parity نیز هستیم. اگر فرکانس کلک میکروکنترلر اصلی 2.88 MHz برابر باشد، چه مقداری برای CD مناسب است؟



UART 1



UART 2



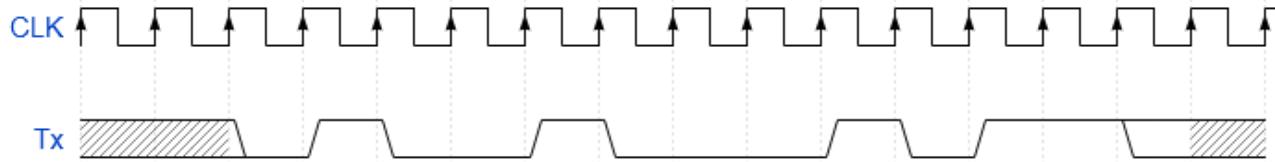
۳) توبولوژی مقابله را برای ارتباط UART دو میکروکنترلر در نظر بگیرید:

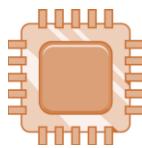
ارسال اطلاعات به اندازه $256\mu s$ به طول بینجامد و رابط

UART در حالت RESET نباشد، ابتدا رشته اطلاعات جابه‌جا

شده بین دو میکروکنترلر را مشخص کنید. سپس مقدار Baud

rate مربوطه را محاسبه کنید.





۴) مزایا و محدودیت‌های رابط UART را نسبت به رابط‌های SPI بیان کنید.

• مهلت ارسال تمرین ساعت ۲۳.۵۵ روز پنجشنبه ۶ آذر می‌باشد.

• سوالات خود را می‌توانید تنهایاً از طریق ایمیل زیر پرسید.

○ M.ChamanMotlagh@gmail.com

• ارائه پاسخ تمرین به سه روش ممکن است:

(۱) استفاده از فایل .docx. تایپ پاسخ‌ها و ارائه فایل Pdf

(۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا

(۳) ارائه پاسخ‌ها به روش سنتی

• ارائه تمرین به روش اول و دوم به ترتیب شامل ۱۰٪ و ۵٪ نمره امتیازی می‌گردد.

• فایل پاسخ تمرین را تنها با قالب **HW4-G#-9*****.pdf** در مدل بارگزاری کنید.

• نمونه: HW4-G2-9531888

• فایل زیپ ارسال نکنید.



11/9/2020



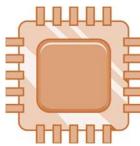
Homework 4

Lec 9-10-11



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2020



- ۱) میکروکنترلری در اختیار داریم که در بازه ولتاژی [-3.3, 3.3] کار می‌کند و می‌تواند ولتاژهای آنالوگ را در 256 محدوده گستته قرار دهد. حال سنسور آنالوگی به این میکروکنترلر وصل می‌کند که ولتاژی برابر V_{in} به میکروکنترلر منتقل می‌کند. اگر V_{in} برابر 2.2V باشد، مقدار عددی دیجیتال این سنسور (عددی بین 0 تا 255) چند خواهد بود؟

$$N_{ADC} = 255 \times \frac{2.2 - (-3.3)}{3.3 - (-3.3)} \rightarrow N_{ADC} = 212.49$$

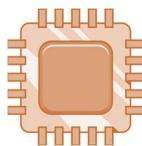
مقدار به دست آمده را به سمت پایین گرد میکنیم:

$$N_{ADC} = 212$$

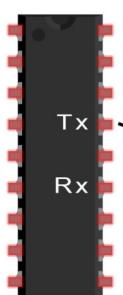
- ۲) قصد داریم از طریق رابط UART یک میکروکنترلر را به یک Device متصل کنیم. هدف ما ارسال داده‌ها با نرخ 8 kb/s است ولی در کنار داده‌ها نیازمند به بیت Parity نیز هستیم. اگر فرکانس کلک میکروکنترلر اصلی 2.88 MHz برابر باشد، چه مقداری برای CD مناسب است؟

چون به بیت parity نیز نیاز داریم و نرخ ارسال داده 8 کیلو بیت بر ثانیه است، بایستی در هر packet که ارسال می‌شود، ۱۱ یا ۱۲ بیت وجود داشته باشد (بسته به تعداد بیت‌های stop). اگر تعداد بیت‌های هر بسته را ۱۲ تا در نظر بگیریم:

$$12 \times 10^3 = \frac{2.88 \times 10^6}{CD \times 16} \rightarrow CD = 240 \div 16 = 15$$



UART 1



UART 2



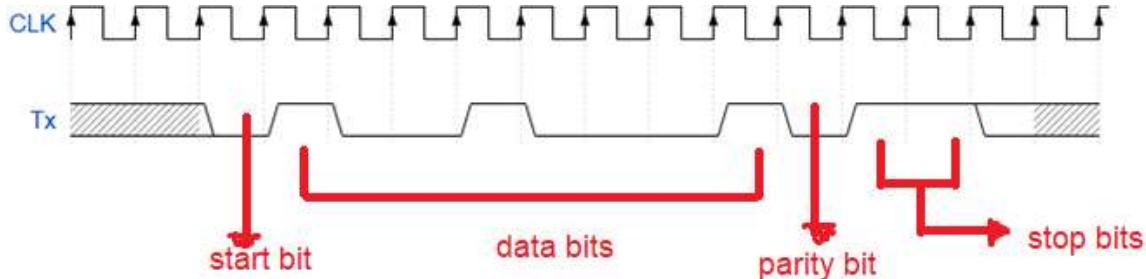
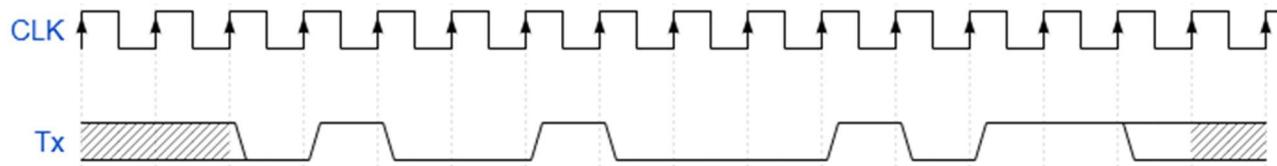
۳) توبولوژی مقابله را برای ارتباط UART دو میکروکنترلر در نظر بگیرید:

ارسال اطلاعات به اندازه $256\mu s$ به طول بینجامد و رابط

UART در حالت RESET نباشد، ابتدا رشته اطلاعات جایه جا

شده بین دو میکروکنترلر را مشخص کنید. سپس مقدار Baud

rate مربوطه را محاسبه کنید.

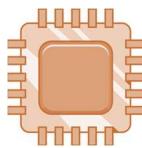


Start bit – 10010001 – parity bit – 2 stop bits

Baud rate = speed of data transfer

$$T = 256 \times \frac{10^{-6}}{8} = 32 \times 10^{-6}$$

Baud rate = $1/32 \times 10^6 b/s$



۴) مزایا و محدودیت‌های رابط UART را نسبت به رابط‌های SPI بیان کنید.

رابط SPI، برخلاف UART تنها ارتباط بین ۲ دیوایس را فراهم می‌کند و از سیستم master and slave پشتیبانی نمی‌کند.

سرعت انتقال داده هم در UART از SPI کمتر است. در SPI از بیت‌های start و stop استفاده نمی‌شود بنابراین در آن داده نیز بدون وقفه ارسال می‌شود.

در UART، با استفاده از parity bit، وجود خطا در داده بررسی می‌شود که در رابط SPI، هیچ گونه بررسی ای برای وجود خطا انجام نمی‌شود.

تعداد پین‌هایی که در UART برای اتصال به میکروکنترلر اشغال می‌کند از SPI کمتر است.

مهلت ارسال تمرین ساعت ۲۳:۵۵ روز پنجشنبه ۶ آذر می‌باشد.

سوالات خود را می‌توانید تنها از طریق ایمیل زیر پرسید.

○ M.ChamanMotlagh@gmail.com

ارائه پاسخ تمرین به سه روش ممکن است:

(۱) استفاده از فایل .docx. تایپ پاسخ‌ها و ارائه فایل Pdf

(۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا

(۳) ارائه پاسخ‌ها به روش سنتی

ارائه تمرین به روش اول و دوم به ترتیب شامل ۱۰٪ و ۵٪ نمره امتیازی می‌گردد.

فایل پاسخ تمرین را تنها با قالب **HW4-G#-9*****.pdf** در مدل بارگزاری کنید.

• نمونه: HW4-G2-9531888

• فایل زیپ ارسال نکنید.

12/9/2020



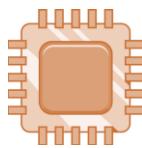
Homework 5

Lec 12-13-14-15



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2020



- ۱) پنج دستگاه جانبی که میکروکنترلر به وسیله PWM آنها را کنترل می‌کند را نام ببرید. دلیل ترجیح آنها به استفاده از PWM چیست؟
- ۲) چهار بخش مختلف حافظه on-chip را روی حافظه مشخص کنید و کاربرد هریک را توضیح دهید.
- ۳) همه صفت (Attribute) های AREA Directive را شرح دهید.
- ۴) ساختار حافظه را رسم کنید. فرض می‌کنیم که جای ذخیره Data از 0x02000000 شروع شود.

AREA myData, Data

ALIGN 4

a DCB 1

b DCB 1

c DCB 1

ALIGN 8,5

d DCB 2

۵) یک کد اسمنبلی بنویسید که تمامی رجیسترهاي عمومی را صفر کند.

۶) پانزده عدد از خانه‌ی 0x04000000 شروع می‌شوند. با فرض اینکه جمع آنها از یک Word بیشتر نمی‌شود، برنامه ای بنویسید که تمام آنها را باهم جمع کند و در رجیستر R2 بریزد.

۷) این کد اسمنبلی string_copy است. آن را کامل کنید و کاملاً کامنت گذاری کنید

AREA string_copy, CODE, READONLY

ALIGN

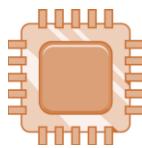
ENTRY

AREA myData, DATA, READWRITE

ALIGN

srcStr DCB "The source string.",0

dstStr DCB "The destination string.",0



پاسخ هر یک از تمرین های عملی را در یک روال در فایل main.s بنویسید.

- مهلت ارسال تمرین ساعت ۲۳.۵۵ روز پنجم شنبه ۲۷ آذر می باشد.
- سوالات خود را می توانید تنها از طریق ایمیل زیر بپرسید.
○ behdadaspacey2099@gmail.com
- ارائه پاسخ تمرین به سه روش ممکن است:
 - ۱) استفاده از فایل .docx. تایپ پاسخها و ارائه فایل Pdf
 - ۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا
 - ۳) ارائه پاسخها به روش سنتی
- ارائه تمرین به روش اول و دوم به ترتیب شامل ۱۰٪ و ۵٪ نمره امتیازی می گردد.
- فایل پاسخ تمرین را تنها با قالب **HW5-G#-9*****.pdf** در مدل بارگزاری کنید.
- نمونه: HW5-G2-9531888
- فایل زیپ ارسال نکنید.

پاسخنامه تمرین ۵ میکروپروسسور

پاییز ۹۹

سوال اول:

موتورهای servo - نور LED - Voltage regulator DC - رگولاتورها - موتور linear

سوال دوم:

- **On-chip peripheral and I/O registers**

برای آدرس دهی به رجیسترها I/O و peripheral میکرو مثل رجیسترها UART/USART و ... I2C.NVIC controller

- **On-chip data SRAM**

برای نگه داری داده ها (داده های معمولی که روزمره را آنها کار می کنیم)

- **On-chip EEPROM (critical data)**

برای نگهداری داده های مهم و حساس (که میخواهیم با قطع برق از بین نرونده مانند password حافظه هی غیر فرار و non-volatile است

- **On-chip Flash ROM (program space)**

برای نگه داری کد (برنامه یا دستورات) حافظه هی غیر فرار و non-volatile است

سوال سوم

Code

آن region را از جنس کد در نظر گرفته که به صورت پیشفرض آن را readonly قرار می دهد و دستورات این flash را در region میریزد

Data

آن region را از جنس داده در نظر گرفته که به صورت پیشفرض آن را readwrite قرار می دهد و دستورات این SRAM را در region میریزد

Readonly

آن region را فقط برای خواندن تعریف می کند و اطلاعات این region قرار نیست در runtime تغییر کند و این اطلاعات را در flash میریزد

ReadWrite

آن region را فقط برای خواندن و نوشتمن تعریف می کند و اطلاعات این محدوده را در SRAM میریزد

Align x

آدرس شروع از مضارب x قرار می دهد یعنی اگر در حافظه ادرس خالی بعدی مضرب x نباشد خانه های حافظه خالی را رد می کند و در خانه ای که مضرب x است می نویسد

سوال چهارم: حذف

سوال پنجم

```
1 MOV R0, #0
2 MOV R1, #0
3 MOV R2, #0
4 MOV R3, #0
5 MOV R4, #0
6 MOV R5, #0
7 MOV R6, #0
8 MOV R7, #0
9 MOV R8, #0
10 MOV R9, #0
11 MOV R10, #0
12 MOV R11, #0
13 MOV R12, #0
14 END
```

سوال ششم

```
1      LDR    R0, =0x04000000
2      MOV    R2, #0
3      MOV    R3, #15 ; R3 is loop counter
4      STR    R3, [R0] ; storing a number in memory
5      MOV    R4, #3 ; preparing another number
6      STR    #3, [R0, #4] ; stored on the next word
7
8 HERE
9      LDR    R1, [R0]
10     ADD   R2, R2, R1
11     ADD   R0, R0, #4 ; going to next word
12     SUBS  R3, R3, #1 ; decrease counter
13     BNE   HERE ;branch if not equal
14
15     END
```

سوال هفتم

```
1 srcStr  DCB    'H', 'E', 'L', 'L', 'O'
2 dstStr  DCB    'G', 'O', 'O', 'D', 'B', 'Y', 'E'
3      MOV    R0, #0x100
4      ;      above should be "ADR R0, firstStr" but we address manually
5      MOV    R1, #0x108
6      ;      above should be "ADR R1, secondStr" but we address manually
7      SUB    R3, R1, R0 ; distance between words
8 Loop
9      LDRB  R2, [R0], #1 ; load word and move one byte
10     STRB  R2, [R1], #1 ; store word and move one byte
11     SUB    R3, R3, #1 ; is distance between words closed?
12     CMP    R3, #0 ; is distance between words closed?
13     BNE   Loop ; if not, loop again
14
15     END
```



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

12/18/2020

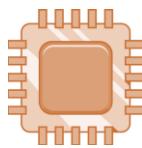


Homework 6



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2020



۱) کد اsemblی ای بنویسید که $R0 \times R1$ را ضرب در $R0$ کند. (بدون استفاده از دستور ضرب (MUL)

۲) کد اsemblی ای بنویسید که $R1 \times R0$ را به توان $R0$ برساند. (بدون استفاده از دستور ضرب (MUL)

۳) برنامه ای در اsemblی آرم بنویسید که کد زیر را پیاده کند:

```
for (i=0; i<15; i++)  
    j=j+j;
```

۴) برنامه اsemblی آرم معادل کد زیر را بنویسید:

```
int *ptr;  
int sum = 0;  
for (int i=0; i<20; i++)  
    sum += *(ptr++);
```

۵) برنامه ای در اsemblی آرم بنویسید که قدرمطلق یک عدد را بیابد.

۶) برنامه ای بنویسید که $R1$ را از صفر تا مقدار ذخیره شده در $R0$ یکی بالا بیاورد، و در هر مرحله اگر عدد مشاهده شده بر سه بخش پذیر بود، $R2$ را به علاوه یک کند.

۷) یک برنامه ای اsemblی بنویسید که در رجیستر $R0$ عدد ۲ به توان مقدار $R1$ را بنویسد.

۸) برنامه اsemblی ای بنویسید که عدد $R0$ را تقسیم بر عدد $R1$ کند. خارج قسمت را در $R2$ و باقی مانده را در $R3$ برجیزد. (استفاده از هر دستوری مجاز است)

• مهلت ارسال تمرین ساعت ۲۳.۵۵ روز دوشنبه ۱۶ دی می باشد.

• سوالات خود را می توانید تنها از طریق [ایمیل](#) زیر بپرسید.

○ behdadaspacey2099@gmail.com

پاسخنامه تمرین ۶ میکروپروسسور

پاییز ۹۹

سوال اول

```
1      MOV    R0, #10
2      MOV    R1, #15
3      MOV    R4, #0
4      BL     MULTIPLY
5      END
6
7 MULTIPLY
8      MOV    R2,#0 ; i = 0
9 L1      CMP    R2,R1 ; while i!=R1
10     BEQ    FINISH
11     ADD    R2, R2, #1 ; i++
12     ADD    R4,R4,R0
13     B     L1
14 FINISH   MOV    PC, LR
15     END
```

سوال دوم

```
1      MOV    R1,#2
2      MOV    R0,#3
3      MOV    R3,R1
4      MOV    R2,R0
5 LOOP1   CMP    R2,#1
6      BLS    FINISH
7      SUB    R2,R2,#1
8      MOV    R4, R1
9      MOV    R5, R3
10     LOOP2  CMP    R5, #1
11     BLS    LOOP1
12     SUB    R5,R5,#1
13     ADD    R1,R1, R4
14     B     LOOP2
15 FINISH   END
```

سوال سوم

```
1      MOV      R0,#0
2      MOV      R1,#15
3      MOV      R2,#1
4 L1     CMP      R0,R1
5      BEQ      FINISH
6      ADD      R2,R2,R2
7      ADD      R0,R0,#1
8      B       L1
9 FINISH
10     END
```

سوال چهارم

```
1      ADR      R0,0X400
2      MOV      R1,#0
3      MOV      R2,#21
4
5 LOOP
6      CMP      R2,#1
7      BEQ      FINISH
8      ADD      R0,R0,#4
9      LDR      R3,[R0],#4
10     ADD      R1,R1,R3
11     SUB      R2,R2,#1
12     B       LOOP
13 FINISH
14 END
```

سوال پنجم

```
1      MOV      R0 ,#-10
2      MOV      R1 ,R0 ,ASR #31
3      ; 31 bit arithmetic shift right of R0 if R0
4      ; is positive the result is 0x0000 and if R0 negative the result is 0xFFFF
5      EOR      R2,R1,R0
6      ; xor the input with 0x0000 or 0xFFFF if the input is positive
7      ; the result of xor is input and if the input is negative the result of xor is
8      ;ones complement of input
9      SUB      R2,R2,R1 ; this line is to calculate 2s complement
```

سؤال ششم

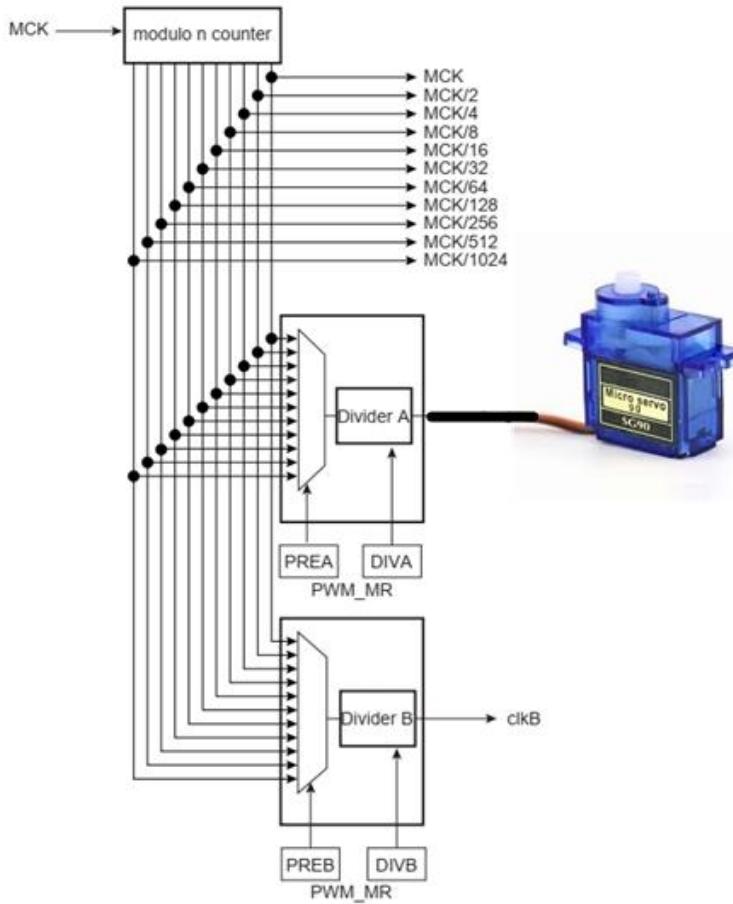
```
1      MOV      R1 ,#0
2      MOV      R0 ,#10
3      MOV      R6 ,R0 ;COUNTER TO LOOP 10 TIMES
4      MOV      R2,#0
5
6 LOOP1      ADD      R1,R1,#1
7          ;GO      TO DO DIVISION TO FIND REMINDER
8      MOV      R3,R1
9      MOV      R4,#3
10     MOV      R5,#0
11 L1      CMP      R3 ,R4
12     BLO      FINISH
13     SUB      R3,R3,R4
14     ADD      R5,R5,#1
15     B       L1
16
17 FINISH      CMP      R3,#0
18     BEQ      L2
19     SUB      R6,R6,#1
20     CMP      R6,#0
21     BNE      LOOP1
22
23 END      ; IF      REMINDER IS 0
24
25 L2      ADD      R2,R2,#1
26     SUB      R6,R6,#1
27     CMP      R6,#0
28     BNE      LOOP1
29
30 END
```

سؤال هفتم

```
1      MOV      R1,#8
2      SUB      R1,R1,#1
3      MOV      R0,#2
4      LSL      R0,R0,R1
```

سوال هشتم

```
1      LDR    R0, =21 ; R0 = 21 (numerator)
2      ; it will contain remainder
3      MOV    R1, #10 ; R1 = 10 (denominator)
4      MOV    R2, #0  ; R2 = 0 (quotient)
5 Label1   CMP    R0, R1 ; Compare R0 with R1 to see if less than 10
6      BLO    FINISH ; if R0 < R1 jump to finish
7      SUB    R0, R0, R1 ; R0 = R0 - R1 (division by subtraction)
8      ADD    R2, R2, #1 ; R2 = R2 + 1 (quotient is incremented)
9      B     Label1 ; goto L1 (B is discussed in the next section)
10
11 FINISH
12      MOV    R3, R0
13      END
```



۱) یک سرورموتور به صورتی که در شکل زیر آمده به میکروکنترلری که با ولتاژ ۵ ولت و کار می‌کند متصل شده است. فرض کنید دور موتور در حالتی که ولتاژ ۵ ولت به طور پیوسته به آن اعمال شود، ۱۰۰۰۰ دور بر ثانیه است. برای آنکه حرکت این موتور بدون تکانه محسوس باشد (حرکت موتور یکنواخت باشد)، لازم است برق آن با فرکانس حداقل ۱KHz قطع و وصل شود. اگر بخواهیم این موتور ۶۰۰ دور در ثانیه بچرخد،

الف) مقدار Duty Cycle کلاک اعمال شده به موتور را بدست آورید.

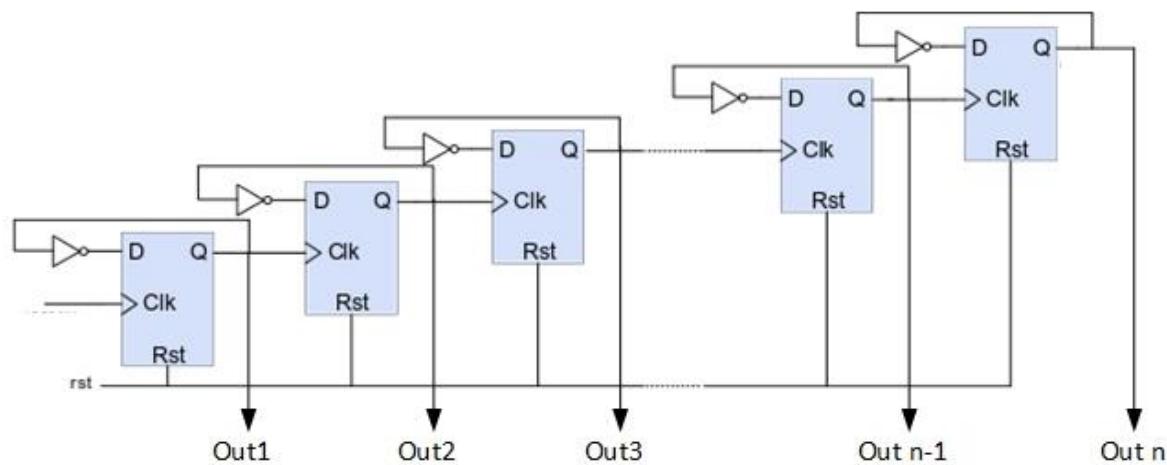
$$\text{Duty Cycle} =$$

ب) با فرض اینکه PREA و DIVA ثبات های ۱۶ بیتی باشند، مقدار آن ها را به نحوی تنظیم کنید که با اعمال حداقل فرکانس، موتور با دور یکنواخت کار کند.

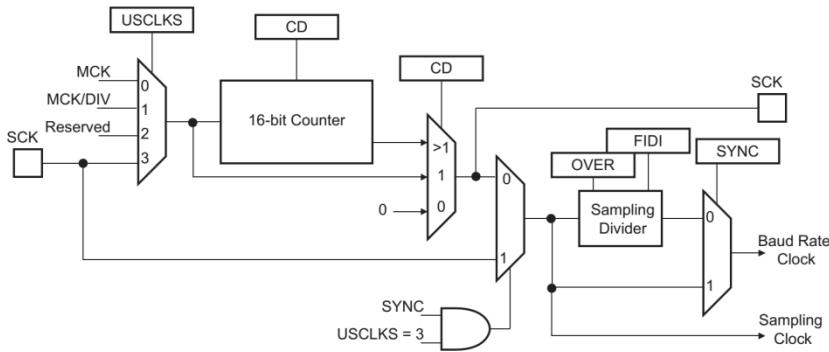
$$\text{PREA} =$$

$$\text{DIVA} =$$

۲) در یک جمله بیان کنید مدار شکل زیر چه نقشی در میکروکنترلر دارد.



۳) مدار شکل زیر برای تنظیم فرکانس ارتباط USART را در نظر بگیرید. برای برقراری یک ارتباط ناهمگام (آسنکرون) با Baud Rate برابر 100 Kbps، جدول زیر را کامل کنید.



MCK	200 MHz
MCK/DIV	50 MHz
CD	
SYNC	
USCLKS	
OVER	

ب) با فرض اینکه بخواهیم از ثبات ۱۶ بیتی FP برای دقیق‌تر کردن فرکانس ارسال استفاده کنیم، مقادیر FP و CD را مشخص کنید.
 $CD =$

$$FP =$$

ج) اگر هر بسته ۸ بیتی داده، در قالب یک بیت شروع، یک بیت خاتمه و یک بیت توازن ارسال شود، زمان لازم برای ارسال یک داده ۱۲۸ بایتی توسط میکروکنترلر چقدر خواهد بود؟

$$T_{Transmit} =$$

د) پنهانی باند (Bandwidth) یا نرخ ارسال داده‌ی مفید) را محاسبه کنید.

$$Bandwidth =$$

۴) شکل زیر را در نظر بگیرید. فرض کنید بخواهیم پایه‌یی عالم‌منظوره‌ای که در شکل دیده می‌شود را به صورت متناوب برای ارسال خروجی Peripheral A و Peripheral B یک داده‌ی تکبیتی با دوره تناوب ۳ ثانیه استفاده کنیم. یعنی در هر ثانیه یکی از سه عملکرد ذکر شده توسط این پایه انجام شود. مختصراً توضیح دهید چگونه باید پیکربندی و برنامه‌ریزی این کار را انجام داد.

توجه: نیاز نیست کد نوشته شود یا مقدار تک‌تک ثبات‌ها در هر لحظه ذکر شود.

نکته: می‌توانید از وقفه‌ها و تایمرها کمک بگیرید.

۵) دو تفاوت بین کامپیوترهای RISC و CISC را بیان کنید.

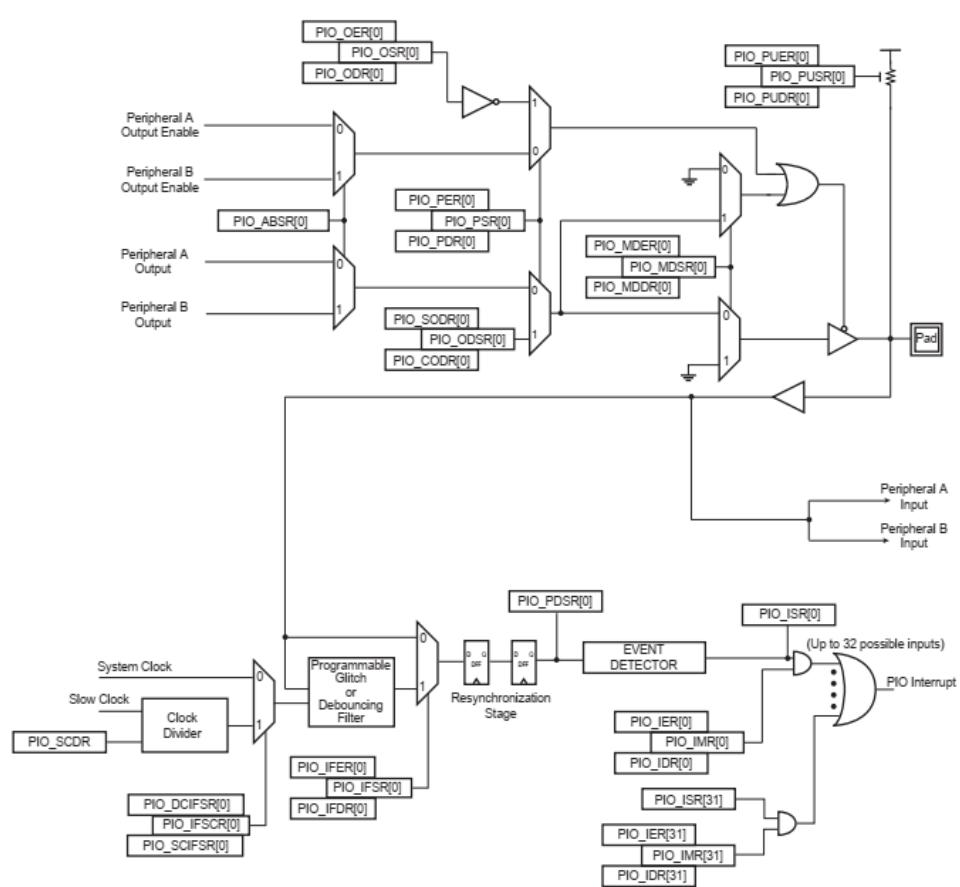
۶) قطعه برنامه زیر را در نظر بگیرید:

```
While (tmp[i] == k)
    i += 1;
```

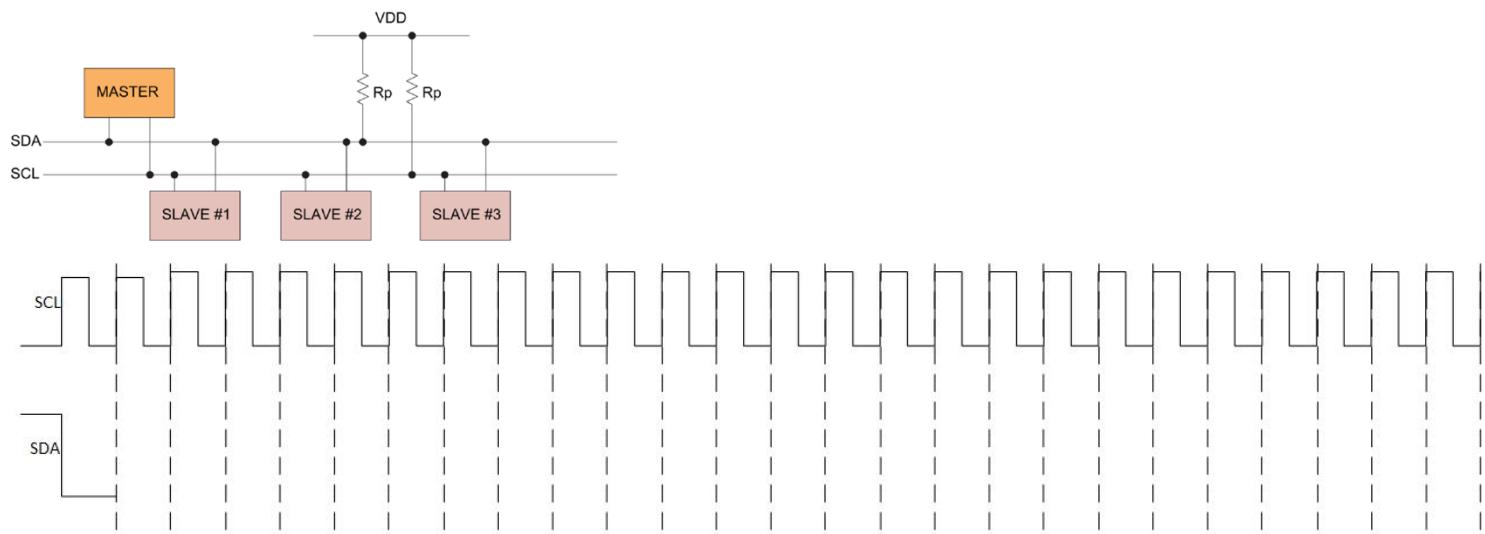
فرض کنید متغیرهای i و k در رजیسترهای R1 و R2 و آدرس شروع آرایه tmp در رجیستر R3 قرار دارد. کد اسambilی معادل قطعه برنامه فوق را بنویسید.

۷) قدرت محاسباتی و سرعت پردازنده‌های شرکت ARM بسیار کمتر از پردازنده‌های شرکت Intel است، اما فروش آن‌ها بسیار بیشتر است. دلیل آن را توضیح دهید.

۸) سه تفاوت تراشه مبتنی بر ریزپردازنده با میکروکنترلر را بیان کنید.



۹) در ارتباط I²C که در شکل زیر آمده، آدرس ۷ بیتی واحدها دوباره شماره‌ی آنها در شکل زیر است. اگر واحد MASTER بخواهد سه بایت داده از واحد SLAVE #3 بخواند، شکل موج زیر را تکمیل کنید. فرض کنید داده ارسالی 0xAB456F باشد.



۱۰) ورودی یک مبدل آنالوگ به دیجیتال ولتاژی در بازه‌ی [0v, 5v] را به اعداد ۰۱بیتی تبدیل می‌کند. اگر ورودی این مبدل از یک حسگر دما که بازه [-20°C, 80°C] را می‌تواند تشخیص دهد آمده باشد و دمای فعلی 30°C باشد، چه عددی به عنوان خروجی مبدل خواهد دید و معادل دقیقاً چه دمایی است؟

$$[D9-D0] =$$

$$\text{Detected Temperature} =$$

۱۱) معادل اسمبلی قطعه کد زیر را بنویسید.

```
int v[10], s;
int *p;
s = 17;
for (p = &v[3]; *p != 0; p++){
    s = s + *p;
}
```

۱۲) برنامه‌ای به زبان MIPS بنویسید که با توجه به نمره‌ی دانشجو که در ثبات R3 ذخیره شده، رده‌ی آن را چاپ کند. یک نمره مطابق جدول زیر به رده‌های مختلف دسته بندی شود.

17 to 20	A
14 to 17	B
10 to 14	C
0 to 10	D

۱۳) قطعه کد زیر را در نظر بگیرید. با فرض اینکه در حین اجرای دستورات برنامه، زمانی که $PC = 0x003F2204$ است وقفه‌ای از واحد USART1 وارد شود، محتوای حافظه در آدرس‌های جدول زیر را مشخص کنید.

__Vectors	DCD	__initial_sp	; Top of Stack initialization
	DCD	Reset_Handler	; Reset Handler
	DCD	NMI_Handler	; NMI Handler
	DCD	HardFault_Handler	; Hard Fault Handler
	DCD	MemManage_Handler	; MPU Fault Handler
	DCD	BusFault_Handler	; Bus Fault Handler
	DCD	UsageFault_Handler	; Usage Fault Handler
	DCD	SecureFault_Handler	; Secure Fault Handler
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	SVC_Handler	; SVCall Handler
	DCD	DebugMon_Handler	; Debug Monitor Handler
	DCD	0	; Reserved
	DCD	PendSV_Handler	; PendSV Handler
	DCD	SysTick_Handler	; SysTick Handler
	DCD	WWDG_IRQHandler	; Window Watchdog
	DCD	PVD_IRQHandler	; PVD EXTI Line detect
	DCD	TAMPER_IRQHandler	; Tamper
	DCD	USART1_IRQHandler	; USART1 Handler
....			
....			
....			
0x40008000	USART1_IRQHandler	ld R5, [R0] ; USART1_IRQHandler begin	
		...	
0x40004000	TAMPER_IRQHandler	ld R5, [R0] ; TAMPER_IRQHandler begin	
		...	
		...	

0x00000000	
0x00000001	
0x00000002	
0x00000003	
0x00000021	
0x00000021	
0x00000022	
0x00000023	
0x00000048	
0x00000049	
0x0000004A	
0x0000004B	
0x0000004C	
0x0000004D	
0x0000004E	
0x0000004F	

۱۴) فرض کنید آدرس شروع آرایه‌ی A (که محتوای داده‌های چهار بایتی است) در ثبات R3 و متغیر h در ثبات R5. کد اسمبلی معادل عبارت زیر را بنویسید.

$$A[6] = h + A[3];$$

۱۵) به سوالات زیر به صورت بسیار مختصر پاسخ دهید.

الف) (درست/نادرست) آدرس مقصد دستور BNE تمام فضای آدرس دهی ۳۲ بیتی پردازنده می‌تواند باشد.

ب) (درست/نادرست) اجرای دستور ADDEQ مشروط به صفر بودن بیت Z است.

ج) مقدار بیت‌های Z و C را پس از اجرای قطعه کدهای زیر مشخص کنید.

(a)
MOV R0, #50
MOV R1, #40
CMP R0, R1

(b)
MOV R1, #0xFF
MOV R2, #0x6F
CMP R1, R2

(c)
EOR R0, R0, R0
EOR R1, R1, R1
CMP R0, R1

C =	C =	C =
Z =	Z =	Z =

۵) مشخص کنید کدام یک از دستورات زیر مجاز نیست یا نادرست است.

a) MOV R2, #0x50000	
b) MOV R2, #0x50	
c) MOV R17, #25	
d) ADD R2, #20, R1	
e) ADD R5, R16, R3	

۶) مزیت استفاده از دستور BX بر دستور B چیست؟