



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

10/12/2021



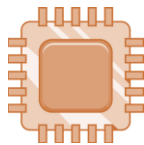
Homework 1

Lec 1-5



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2021



1) به پرسش های زیر در مورد ISA پاسخ دهید:

الف) ISA پردازنده ما باید شامل کدام گروه ها از فانکشن ها باشد تا ISA کاملی به حساب آید؟

پاسخ:

برای اینکه یک ISA کامل باشد باید دارای حداقل این سه گروه از فانکشن ها باشد:

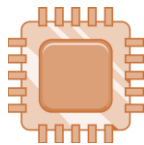
- Load / Store
- Control
- Arithmetic / Logic

با این سری از دستورات ما می توانیم تمامی دستورات مورد استفاده در زبان های سطح بالاتر را تولید کنیم.

ب) ISA چه ویژگی هایی از پردازنده را مشخص می کند (حداقل به سه مورد اشاره کنید)؟
مثال: Risc یا Cisc بودن پردازنده

پاسخ:

1. طول دستورات (دستورات چند بیتی خواهند بود)
2. طول دستورات ثابت یا متغیر (براساس Risc یا Cisc بودن پردازنده)
3. تعداد رجیسترها و تعداد بیت های آنها
4. محل قرارگیری Operand ها (رجیستر یا پشته یا حافظه)
5. نحوه برقراری ارتباط با حافظه (ممکن است چند چند مدل Load از حافظه داشته باشیم)
6. فرمت دستورات



2) به سوالات ریز در رابطه با Microcontrollers پاسخ دهید:
الف) میکرو ای که ما در درس استفاده می کنیم (SAM3X8E) از کدام یک از معماری های Harvard یا Von Neumann استفاده می کند و دلایل آن چیست (دو دلیل)?

پاسخ:

میکرو ما از معماری Harvard پیروی می کند.

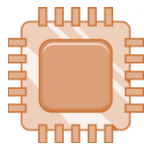
دلیل 1:

با توجه به اینکه میکرو ما بیشتر برای سیستم های نهفته استفاده می شود و مانند کامپیوترهای روزمره نیازی ندارد که هی برنامه ای که روی رم آن قرار دارد تغییر کند و معمولا برنامه ای که روی آنها قرار میگیرد تا مدت زمان زیادی نیاز به تغییر ندارد در نتیجه که این دو حافظه از هم جدا باشند سرعت پردازش ما بیشتر می شود.

دلیل 2:

با توجه به صحبت های مطرح شده در کلاس می دانیم که پردازنده های کامپیوتر های رومیزی ما زیر به طور مخفی و در cache از معماری Harvard استفاده می کنند و با توجه به اینکه میکرو ما اصلا cache ندارد و از حافظه کوچکی برخوردارست و دلیلی که معماری Von Neumann جوابگو خواهد وجود cache است پس در این پردازنده ها بهتر از معماری Harvard استفاده کنیم. با توجه به نبود cache قابلیت اجرا پایپ لاین وجود ندارد و برای اینکه این قابلیت وجود داشته باشد باید از معماری Harvard استفاده کنیم. دلیل نبود cache هم آن است که حافظه اصلی خودش به اندازه کافی بزرگ نیست که نیاز به cache و طبق دلیل 1 برنامه بر روی حافظه دستورات زیاد تغییر نمی کند پس وجود cache منطقی نیست.

نکته: با توجه به اینکه دلیل 1 بیشتر در بخش اسمبلی درس مطرح می شود نوشتن آن ضرورتی ندارد و نوشتن دلیل 2 کافی است.



ب) چند تا از برتری هایی که باعث شده ست در سیستم های نهفته از Microcontroller استفاده شود را نام ببرید (سه مورد کافی است).

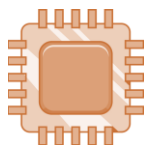
پاسخ:

- تمامی حافظه ها و I/O ها در درون یک میکروکنترلر قرار دارد و نیازی نیست این دیوایس ها را خریداری کرده و با هم اسمبل کنیم، چون بیشتر هدف ما از خرید میکروکنترلر یک واحد محاسباتی خالی نیست و بیشتر هدف استفاده خاص منظوره از آنها در یک سیستم بزرگتر است.
- با توجه به اینکه همه دیوایس ها را دارد مدار کوچکی دارد و مناسب فضای با اندازه کوچک است
- دسترسی به حافظه سریعی دارند
- قدرت پردازشی زیادی ندارند و به همین دلیل ارزان هستند و مناسب برای سیستم های نهفته
- مصرف کمتر انرژی

ج) حالت های مختلف میکرو (SAM3X8E) در Low Power Modes را نام ببرید و برای هر کدام یکی از مواقع استفاده را مثال بزنید.

پاسخ:

- Backup Mode: مثلاً دستگاه ما کلاً خاموش است و منتظر دریافت وقفه از طرف peripherals هاست. مثلاً وقتی که ماشین لباسشویی خاموش است و ما آن را روشن می کنیم .
- Wait Mode: مشابه حالت بالا ولی در این حالت سرعت بازگشت به شدت بیشتر است و در سیستم های Real-Time می تواند استفاده شود .
- Sleep Mode: وقتی که CPU کاری برای انجام ندارد و DMA در حال انتقال دیتا است.



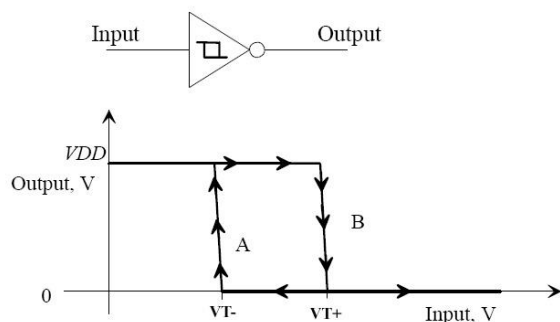
3) به سوالات زیر در مورد اجزای ریزپردازنده (SAM3X8E) پاسخ دهید:

الف) سه مدل مختلف تایمر در این ریزپردازنده را نام ببرید و موارد استفاده از هر کدام را شرح دهید.

پاسخ:

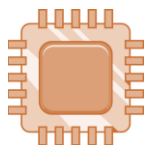
- Real-time Timer: یک شماره 32 بیتی است، ثانیه های سپری شده را می شمرد به همین دلیل به آن Real-time می گویند. می تواند برای وقفه های دوره زمانی ثابت استفاده شود. (مثلا 10 ثانیه ای)
- Real-time Clock: این تایمر برای میکرو ساعت می سازد. یعنی هم ثانیه دارد هم دقیقه هم ساعت و هم تقویم و می تواند به ما دقیق این موارد را بگوید. این تایمر به زمان اولیه دارد و بعد از شروع به شماردن می کند و هر وقت برق مدار قطع شود ریست می شود. فرق آن با RTT این است که، RTT صرفا ثانیه ها را می شمارد و برای محاسبه زمان باید خودمان بر حسب ثانیه ها بدست بیاوریم ولی RTC خودش این کار را انجام می دهد.
- Watchdog Timer: تایمر مراقب، سیستم را از Deadlock خارج می کند اگر برنامه ای که در حال اجرا باشد از تایمر Watchdog بیشتر شود، سیستم را ریست می کند تا از این وضع خلاص شود.

ب) شکل زیر نشان دهنده کدام GPIO میکرو ماست و نمودار آن را توضیح دهید.



پاسخ:

GPIO که در این نمودار نشان داده شده Triggers Schmitt Input است. این نمودار بیانگر اینست که به فرض ولتاژ ورودی مابین 0 تا 5 ولت باشد و باید آن را به دو عدد دیجیتال 0 و 1 تبدیل کند حال در این مود به این صورت است که اگر ولتاژ از برسد به 2.5، یک تعبیر نمی شد بلکه باید یک مقداری از 2.5 بالاتر برود



(میتوانیم این مقدار threshold را در این مثال 0.5 در نظر بگیریم) یعنی اگر ولتاژ به 3 برسد تازه مقدار دیجیتال آن 1 می شود.

4) به پرسش های زیر در مورد وقفه های تودرتو پاسخ دهید:

الف) NVIC چگونه وقفه های تودرتو را مدیریت می کند (از دیدگاه رجیسترهای NVIC شرح دهید)؟
پاسخ:

فرض می کنیم وقفه اول در حال اجرا باشد در نتیجه در رجیستر IABR_NVIC بیت مربوط به وقفه اول بر اساس شماره وقفه اش 1 می باشد. حال وقفه دوم رخ می دهد دو حالت پیش می آید:

الف) وقفه دوم اولویت بالاتر داشته باشد حال در رجیستر IABR_NVIC بیت مربوط به وقفه اول 0 می شود و بیت مربوط به وقفه دوم 1 می شود (رجیستر IABR_NVIC در هر زمان حداکثر یک بیت 1 دارد) و سپس در رجیستر ISPR_NVIC بیت مربوط به وقفه اول 1 می شود و منتظر می ماند تا ISR وقفه دوم کامل اجرا شود. سپس IABR_NVIC وقفه اول 1 و وقفه دوم 0 می شود و در رجیستر ICPR_NVIC بیت مربوط به وقفه اول 1 می شود تا این وقفه از حالت pending خارج شود.

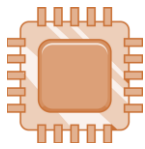
ب) وقفه اول اولویت بالاتر داشته باشد حال در رجیستر IABR_NVIC بیت مربوط به وقفه اول 1 می باشد و نیازی به تغییر ندارد سپس در رجیستر ISPR_NVIC بیت مربوط به وقفه دوم 1 می شود و منتظر می ماند تا ISR وقفه اول کامل اجرا شود. سپس IABR_NVIC وقفه اول 0 و وقفه دوم 1 می شود و در رجیستر ICPR_NVIC بیت مربوط به وقفه دوم 1 می شود تا این وقفه از حالت pending خارج شود.

ب) چهارتا از دستوراتی که تعداد کلاک بالایی برای اجرا نیاز دارند را نام ببرید و اگر درحین پردازش این دستورات وقفه ای رخ دهد چگونه با آنها برخورد خواهد شد؟

پاسخ:

Load Multiple (LDM), Store Multiple (STM), Push, Pop, MULS

در صورتی که وقفه ای در حین اجرا این دستورات رخ دهد و برنامه در شرایط بحرانی نباشد، دستورات کاملاً دراپ می شود و بعد از اجرا وقفه دوباره از اول اجرا می شوند.



5) به سوالات زیر در مورد NVIC پاسخ دهید:

الف) دلیل وجود دو حالت مختلف Active و A&P برای وقفه‌ها در NVIC را شرح دهید.

پاسخ:

دلیل وجود حالت A&P این است که اگر یک وقفه در حال اجرا باشد و باز همان دیوایس وقفه جدیدی بفرستد وضعیت آن به A&P تغییر می‌کند.

ب) فرق بین دو ویژگی Tail-chaining و Late-arriving را توضیح دهید.

پاسخ:

Tail-chaining: این ویژگی به این معناست که اگر وقفه ای در حال اجرا باشد و وقفه دیگری رخ دهد حال یکی از این وقفه‌ها به حالت pending می‌رود، پس از اتمام یکی از آنها دیگه لازم نیست رجیستر های برنامه اصلی را باز لود کرده و بعد از آن باز سیو کنیم و به سراغ وقفه دوم برویم و پردازنده باهوش عمل می‌کند و بلافاصله بعد از اتمام وقفه اول به سراغ دومی می‌رود و پس از اتمام آن رجیستر هایی که به صورت سخت افزاری سیو شده بودند را لود می‌کند.

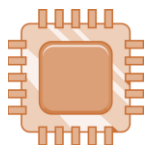
Late-arriving: فرض کنید در حین سیو کردن رجیستر های برای یک وقفه باشیم که یک وقفه با اولویت بیشتر رخ دهد در حین حالت با توجه به اینکه وقفه دوم اولویت بیشتری دارد انجام می‌شود با اینکه ما برای وقفه اول رجیستر ها را سیو کردیم (وقفه ای که دیر تر رسیده ست اجرا میشود)
Tail-chaining: مربوط به انتهای اجرا وقفه هاست در حالی که Late-arriving مربوط به ابتدای اجرای آنهاست.

ج) دلایل وجود قابلیت Masking را نام ببرید و انواع حالتی که می‌توانیم با استفاده از رجیسترهای CPU

جمعی از وقفه‌ها را باهم Mask کنیم را شرح دهید.

پاسخ:

در واقع Masking یعنی بعضی از وقفه‌ها را نادیده بگیریم و آنها را غیرفعال کنیم و به وقت نیاز باز فعالشان کنیم. مثلاً زمانی که می‌خواهیم وارد ناحیه بحرانی یا Critical Section شویم. یک نمونه دیگر نیز وقتی مثلاً وقفه‌های مربوط به یک دیوایس خارجی را مدتی قطع می‌کنیم تا روند اصلی برنامه طی شود و از طرف آن دستگاه برای مدتی وقفه نداشته باشیم.

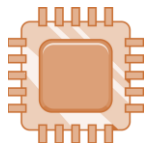


Primask: یک رجیستر تک بیتی که همه وقفه‌های با اولویت 0 به بالا را mask می‌کند.
Basepri: یک رجیستر 8 بیتی که یک عدد بین 1 تا 240 در آن قرار می‌گیرد و از آن اولویت به بعد mask می‌شود.
Faultmask: یک رجیستر تک بیتی کاملاً مشابه با Primask صرفاً با این تفاوت که وقفه‌هایی با اولویت 1- به بالا را mask می‌کند. (وقفه HardFault که یک وقفه سخت‌افزاری نیز هست)

(6) به پرسش‌های زیر در مورد NVIC Register پاسخ دهید:
الف) Vector table چیست و محتوی آن چگونه است؟
پاسخ:

Vector table یک جدول 16 ردیفی است که هر ردیف 32 برای شامل می‌شود (DCD) و در آن
ادرس شروع ISR مربوط به وقفه مورد نظر با هر خانه را نشان می‌دهد. الآن مثلاً خانه شماره 1 آدرس
ISR وقفه Reset را نشان می‌دهد.

<u>__Vectors</u>	DCD	<u>__initial_sp</u>	; Top of Stack initialization
	DCD	<u>Reset_Handler</u>	; Reset Handler
	DCD	<u>NMI_Handler</u>	; NMI Handler
	DCD	<u>HardFault_Handler</u>	; Hard Fault Handler
	DCD	<u>MemManage_Handler</u>	; MPU Fault Handler
	DCD	<u>BusFault_Handler</u>	; Bus Fault Handler
	DCD	<u>UsageFault_Handler</u>	; Usage Fault Handler
	DCD	<u>SecureFault_Handler</u>	; Secure Fault Handler
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	<u>SVC_Handler</u>	; SVC_Handler
	DCD	<u>DebugMon_Handler</u>	; Debug Monitor Handler
	DCD	0	; Reserved
	DCD	<u>PendSV_Handler</u>	; PendSV Handler
	DCD	<u>SysTick_Handler</u>	; SysTick Handler



ب) چرا وقفه‌های NVIC در سری ریزپردازنده‌های ARM از شماره 1 شروع می‌شوند نه 0؟

پاسخ:

اولین خانه Vector table با شماره 0 مربوط به آدرس شروع Stack است و وقفه‌ها به ترتیب از خانه شماره 1 به بعد شروع می‌شوند برای همین اولیه وقفه ما شماره 1 دارد نه 0. به صورت پیش‌فرض Stack از خانه آخر حافظه شروع می‌شود و به صورت کاهشی آدرس آنها کم می‌شود با هر Push و ما با دستکاری خانه شماره 0 این جدول می‌توانیم این پیش‌فرض اولیه را تغییر دهیم.

ج) با توجه به اینکه رجیستر NVIC-IPR اعداد Unsigned را در خود ذخیره می‌کند چگونه وقفه‌هایی با اولویت منفی داریم؟

پاسخ:

برای این وقفه‌های ایستا مانند Reset و ... مدار آنها به صورت سخت افزاری پیاده سازی شده است و مانند بقیه وقفه رجیستری ندارند که به آن مقدار دهیم.

- مهلت ارسال تمرین ساعت 23.59 روز 30 مهر می‌باشد.
- برای پاسخ به پرسشهای این تمرین می‌توانید در صورت نیاز به فصل 5 و 8 مرجع فنی Cortex-m3 که در مودل بارگزاری شده است مراجعه کنید.
- سوالات خود را می‌توانید از طریق تلگرام از تدریس‌یارهای گروه خود بپرسید.
- ارائه پاسخ تمرین بهتر است به روش های زیر باشد:
 - 1) استفاده از فایل docx. تایپ پاسخ‌ها و ارائه فایل Pdf
 - 2) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا
- فایل پاسخ تمرین را تنها با قالب **Hw1_StudentNumber_G[groupnumber].pdf** در مودل بارگزاری کنید.
- نمونه: Hw1_9731121_G1
- فایل زیپ ارسال نکنید.



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

10/12/2020



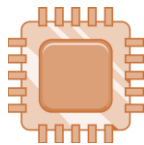
Homework 2

Lec 5-6



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2020



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

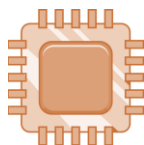
عملکرد مدارهای منطقی به گونه ای است که یک باز ولتاژی را برابر با ۱ منطقی و یک بازه دیگر را برابر با ۰ منطقی در نظر میگیرند. برای نمونه سطح منطقی و بازه ولتاژ متناظر با آن در دو تکنولوژی در جدول زیر آورده شده است.

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 متناظر با هیچ یک از سطح های منطقی نیست، پیشگیری کرد. برای اینکار ورودی را به یک ولتاژ مطمئن مثلا 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) Debouncing Filter چه زمانی به کار برده می شود؟

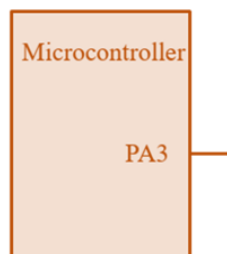
هنگامی که کلید مکانیکی فشرده یا رها می شود ممکن است که خروجی برای مدتی نوسان کند، در برخی از کاربرد ها این نوسان ها ممکن است منجر به تشخیص نادرست شوند و نهایتا رفتار های ناخواسته و نامنتظره گردد. برای نمونه در دستگاه *Treadmill* هنگامی که بخواهیم کلید *Speed-Up* را فشار دهیم، ممکن است این نوسان ها منجر 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) می‌خواهیم 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 رجیسترهای PIOA را پیکربندی کرده باشد که مقدار رجیستر PUSR:

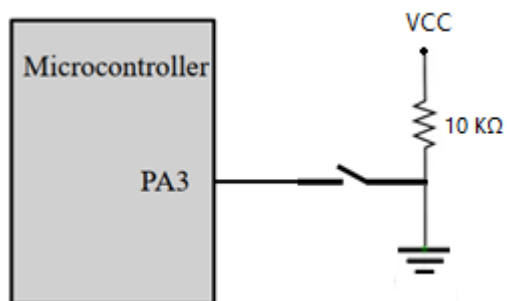
a. برابر با 0x0008 باشد.

b. برابر با 0x0000 باشد.

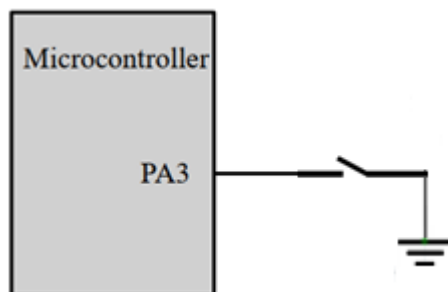


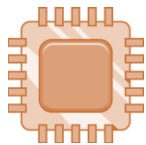
(توجه داشته باشید که شماتیک را در هر حالت با کمترین قطعه‌های ممکن بکشید.)

الف) همان‌طور که در صورت سوال گفته شده است با توجه به مقدار رجیستر PUSR مقاومت بالا کشنده برای پایه PA3 غیر فعال 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)

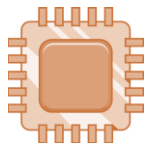




۴) برای آنکه پایه ۸ از واحد کنترلر (PIOA (PA8 را بخواهیم به صورت خروجی پیکربندی شود و حالت Open Drain (Multi Drive) و Pull-up Resistor غیر فعال باشد، مقادیری را که در رجیسترهای لازم باید نوشته شوند مشخص کنید.

```
#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 نوشت مشخص کنید.

امتیازی: برای فعال سازی وقفه ی حساس به لبه بالا رونده ی (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.

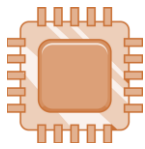
توجه کنید که برای فعال کردن این وقفه باید هم در واحد NVIC و هم در واحد کنترلر PIO پیکربندی های لازم انجام شود. یعنی برای نمونه اگر وقفه لبه بالا رونده برای پایه ای پیکربندی شود، هنگامی که سطح ولتاژ منطقی آن پایه از ۰ به ۱ تغییر می کند، وقفه واحد کنترلر PIO آن پایه به حالت Pending درمی آید، ولی برای آنکه 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 **Peripheral Identification (PID)** آن را بدست آورید.

همچنین در پیاده سازی معماری Cortex-M3 در میکروکنترلر 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 راهنما جستجو کرد.

```
#define PIN (1u1 << 8)           // 0000...100000000
#define PIOA_PER 0x400E0E00u1    // Write Only
#define PIOA_OER 0x400E0E10u1    // Write Only
#define PIOA_PUER 0x400E0E64u1   // 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 0x400E0E40u1 // Write Only
#define PIOA_AIMER 0x400E0EB0u1 // Write Only
#define PIOA_ESR 0x400E0EC0u1 // Write Only
#define PIOA_RELSR 0x400E0ED4u1 // Write Only

*((uint32_t *) PIOA_IER) = reg; // PIOA_IMR: XXXX...XX1XXXXXXXXXX
*((uint32_t *) PIOA_AIMER) = reg; // PIOA_AIMMR: XXXX...XX1XXXXXXXXXX
*((uint32_t *) PIOA_ESR) = reg; // PIOA_ELSR: XXXX...XX0XXXXXXXXXX
*((uint32_t *) PIOA_RELSR) = reg; // PIOA_FRLHSR: XXXX...XX1XXXXXXXXXX
```

پیکربندی های واحد NVIC :

```
#define PIOA_PID 11

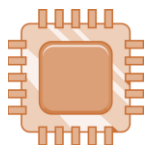
#define NVIC_ISER0 0xE000E100u1 // Read Write
#define NVIC_IPR2 0xE000E408u1 // Read Write

#define SCB_AIRCR 0xE000ED0Cu1 // Read Write*

#define AIRCR_VECTKEY 0x05FAu1
#define PRIGROUP_VALUE 0b101u1 // With this value for PRIGROUP, group priority and
// sub-priority fields in IP[n] is configured as
// bxx.yy0000

*((uint32_t *) NVIC_ISER0) |= 1u1 << PIOA_PID; // xxx...xxx1xxxxxxxxxxx
*((uint32_t *) NVIC_IPR2) |= 0b11110000u1 << 24; // xxxxxxxx11110000xx...xx

reg = *((uint32_t *) SCB_AIRCR); // value is saved for processing
reg &= ~((uint32_t)0xFFu1 << 16 | 0x07u1 << 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 می شود؟ (پاسخ شما باید مفهوم "واگیری (Pre-emption)" را در برداشته باشد).

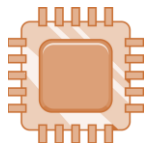
گروه بندی اولویت روشی است که در آن بیت های اولویت اختصاص یافته به هر وقفه را که در رجیستر های *Interrupt Priority* جای دارند، به دو گروه اولویت گروهی و زیر اولویت بخش بندی می کند. که شمار بیت های اختصاص یافته به هر یک از این دو گروه بستگی به مقدار بیت های *Priority Grouping* در رجیستر *Application Interrupt 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	bxxx.yyyy	[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.yyyyyyy	[7]	[6:0]	2	128
b111	b.yyyyyyyy	None	[7:0]	0	256

به اولویت گروهی، اولویت واگیری نیز گفته می شود و همان گونه که در جدول بالا نیز دیده می شود بیت های پر ارزش تر را در *PRI_N* در برمیگیرد. وقفه هایی که اولویت گروهی یکسانی دارند در یک گروه اولویت جای میگیرند. اولویت با مقدار بیت های *sub priority* و *group priority* رابطه عکس دارد. و چنانچه روال سرویس وقفه *A* با مقدار *group priority* بزرگ تر 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 *Enabled* شده باشد. در این صورت با واگیری اجرا از سرویس وقفه *A*، اجرا به سرویس وقفه *B* اختصاص داده می شود و رجیستر های *A* در پشته *push* می گردد 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 می شود زیرا با توجه به مقدار آن می توان مشخص کرد که چه وقفه هایی می تواند در صورت اعلان باعث واگیری اجرا وقفه های دیگر شوند.

زیر اولویت اما واگیری را تعیین نمی کند بلکه چنانچه چند وقفه که در یک گروه اولویت می باشند همزمان *Enabled* و *Pending* باشند در این صورت زیر اولویت تعیین میکند که ترتیب اجرای این وقفه ها چگونه باشد. و نخست وقفه ی با کمترین مقدار زیر اولویت اجرا خواهد شد و سپس وقفه های با مقدار زیر اولویت بزرگ تر.



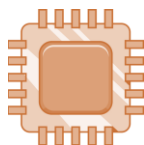
۷) بیشینه شمار وقفه هایی که NVIC می تواند پشتیبانی کند چند است؟ اولویت وقفه ها ایستا می باشد یا پویا؟

بیشینه 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 های 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 ی بیرونی در رجیستر های Interrupt Priority Registers و با توجه به مقدار خانه های PRIGROUP در رجیستر Application Interrupt and Reset Control Register آن را برنامه ریزی کرد.

۸) چنانچه در هنگام اجرای روال وقفه ی 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	۱	۰
C	۱	۳
D	۲	۱
E	۳	۰

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



- مهلت ارسال تمرین ساعت ۲۳.۵۵ روز یکشنبه چهارم آبان می باشد.
- سوالات خود را می توانید تنها از طریق ایمیل زیر بپرسید.
 - alirezasalehy@aut.ac.ir
- برای پاسخ دهی به سوال های ۵ و ۴ باید بخش هایی از فصل 10 (ARM Cortex-M3 processor) , 31 (PIO) و ۹ (Peripherals) را از راهنمای میکروکنترلر های ATSAM3 در صورت نیاز بررسی کنید.
- همچنین برای پاسخ دهی به سوال های 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 پاسخ ها و ارائه فایل Pdf
 - ۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا
 - ۳) ارائه پاسخ ها 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 **HW2-G#-9531***.pdf** در مودل بارگزاری کنید.
- نمونه: HW2-G2-9531747
- فایل زیپ ارسال نکنید.

10/28/2021



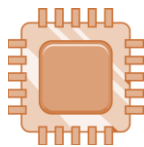
Homework 2

Lec 6-8



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2021



(الف) مفاهیم زیر را به اختصار توضیح دهید.

- full duplex
- slave receive

پاسخ:

۱- انتقال دیتا ، به صورت دو طرفه ، همزمان روی یک line انجام می پذیرد.(مانند انتقال دیتا در مکالمات تلفنی)

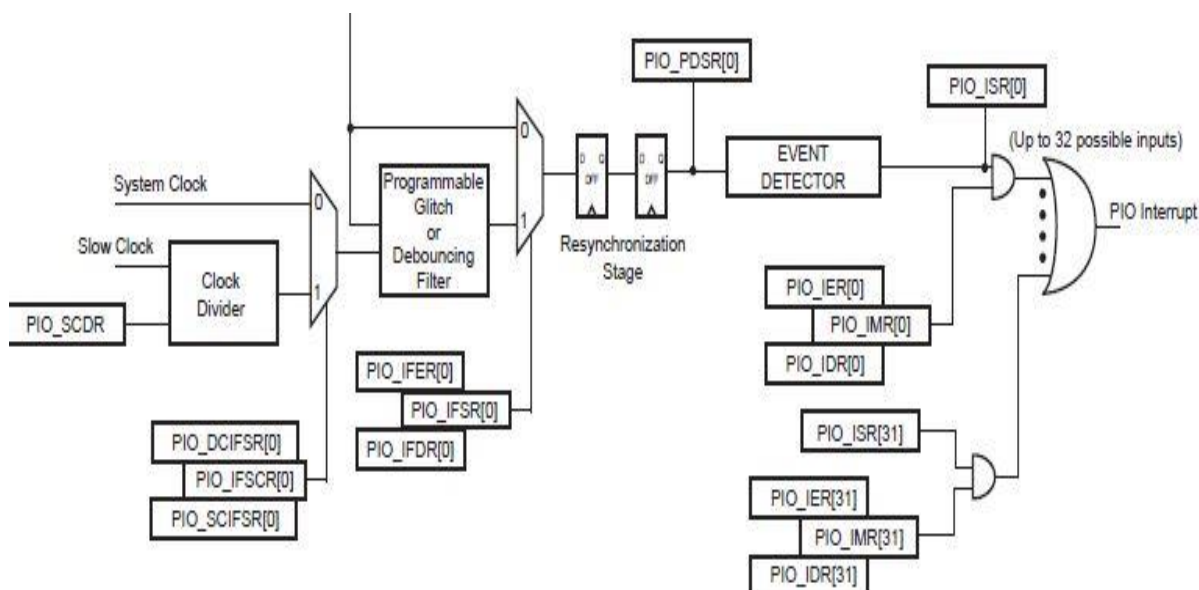
۲- در حالت c2i مستر ادرس یک slave را میفرسد و اعلام میکند داده ای قرار است ارسال شود سپس داده را میفرستد.

(ب) open drain چیست و توضیح دهید کدام پروتکل از این مفهوم تبعیت می کند.

پاسخ:

هیچ اختلال در باس نمیتواند به وجود بیاید که در آن یک دستگاه در تلاش باشد تا خط را بالا بکشد(۱) در حالی که دیگری میخواهد آن را پایین بکشد(۰)، و در نتیجه، احتمال آسیب به درایورها یا اتلاف توان بیش از حد در سیستم از بین میرود. هر خط سیگنال دارای یک مقاومت پول آپ میباشد، که سیگنال را هنگامی که هیچ دستگاهی آن را پایین نمیکشد بالا نگه دارد. پروتکل i^2C از این مفهوم تبعیت میکند

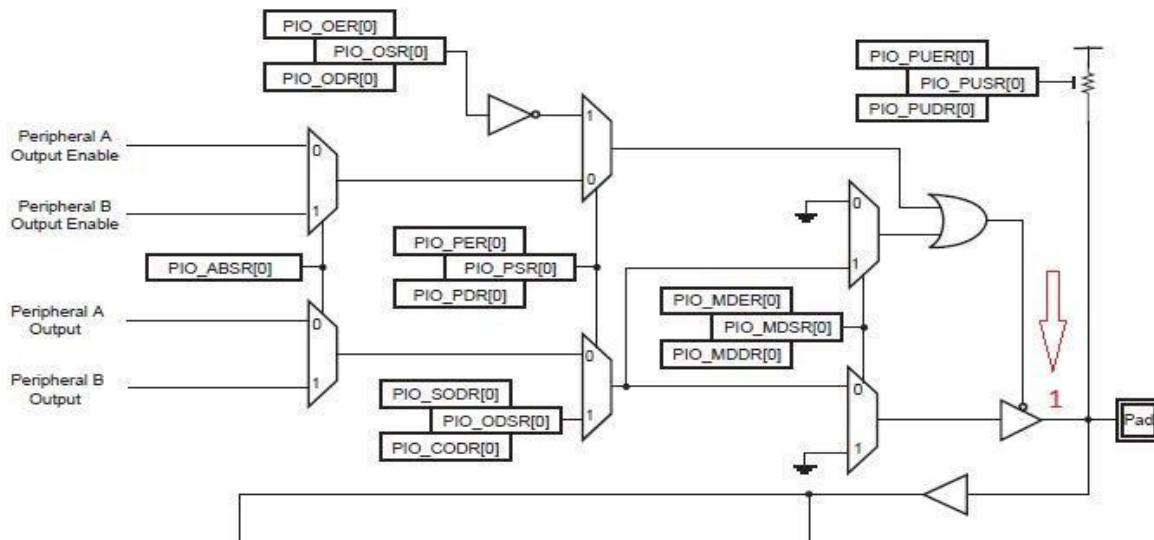
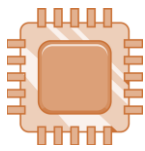
(۲) الف) در شکل زیر چگونه متوجه شویم کدام پین باعث ایجاد وقفه شده است؟



پاسخ:

در هر کدام از پین ها یک **Event Detector** وجود دارد که یک شدن آن به معنای آمدن وقفه می باشد و مقدار آن در بیت متناظر آن پایه رجیستر **PIO_ISR** ذخیره می شود و میتوان به سادگی با نگاه کردن به رجیستر 32 بیتی **PIO_ISR** متوجه شد بیت چندم آن 1 شده است و میتوان متوجه شد کدام یک از پین ها باعث ایجاد وقفه شده است

(ب) در شکل زیر اگر $PIO_PSR = 1$ باشد مقادیر خواسته شده را بیابید.



PIO_ODSR = ?, PIO_MDSR = ?, PIO_OSR = ?

پاسخ:

PIO_ODSR = 1

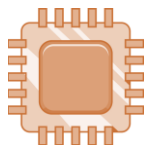
مقدار خروجی مدنظر کاربر را مشخص میکند و از آنجایی که بافر سه حالتی است یعنی از این رجیستر یک به آنجا منتقل شده

PIO_MDSR = 0

حالت مالتی درایو را مشخص میکند که در اینجا نداریم

PIO_OSR = 1

بافر سه حالتی مقدار یک دارد و فعال است یعنی خروجی **Or** برابر با صفر بوده که این در صورتی اتفاق میافتد که **OSR** مقدار یک داشته باشد.



۳) الف) کلاک PIO Controller از کدام بخش تامین میشود و قطع بودن آن چه مزیت و عیبی خواهد داشت؟

پاسخ:

کلاک از واحد (PMC) controller management power تامین میشود

مزیت: قطع بودن کلاک PIO Controller باعث ذخیره شدن انرژی میشود (کم تر شدن انرژی مصرفی)

عیب: با قطع بودن کلاک PIO Controller نمیتوانیم از همه ی ویژگی های آن استفاده کنیم برای مثال وقفه ها و حالت های مختلف آن

ب) در صورت غیرفعال بودن کلاک واحد PIO Controller ، مشخص کنید کدام یک از ویژگی های زیر در PIO Controller فعال و کدام یک غیرفعال است؟

I) نوشتن بر روی رجیسترهای PIO Controller

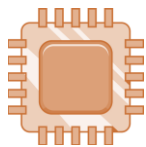
II) خواندن سطح منطقی پین های خروجی

III) استفاده از پین ها برای تولید وقفه

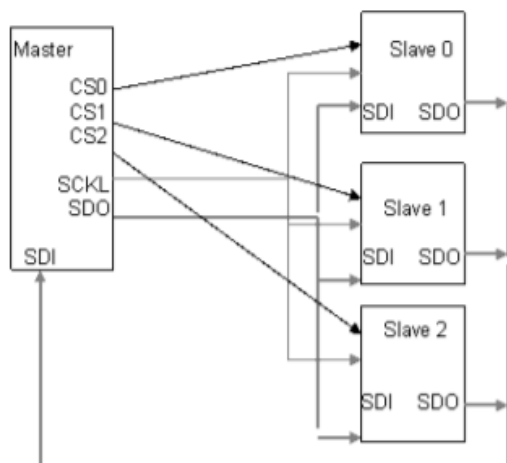
IV) pull-up کردن پین ها

پاسخ:

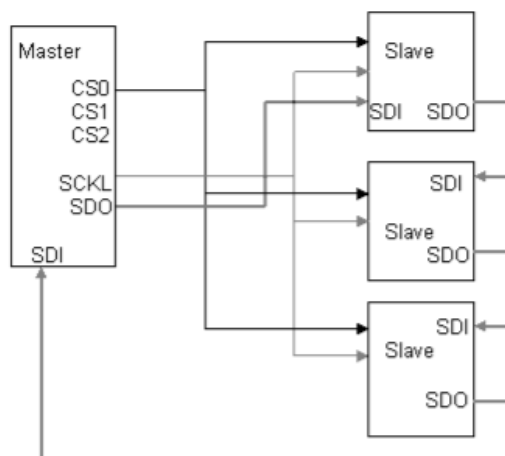
- | | |
|-------|---------|
| (I) | فعال |
| (II) | غیرفعال |
| (III) | غیرفعال |
| (IV) | فعال |



۴) برای پیاده سازی یک زنجیره دستگاه ها به کمک رابط SPI، دو توپولوژی زیر پیشنهاد شده است. این دو روش را مقایسه کنید و معایب هر کدام را ذکر کنید.



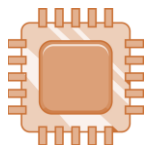
روش دوم



روش اول

پاسخ:

روش اول نیاز به SS (یا CS) کمتری نسبت به روش دوم دارد، بنابراین هزینه پیاده سازی سخت افزاری آن پایینتر است. همچنین جایگذاری قطعات در روش اول ساده تر است چرا که تنها لازم است زنجیره تشکیل شود و هر قطعه به قطعه ی بعدی نزدیک باشد در حالی که در روش دوم، همه ی قطعات باید همجواری مکانی داشته باشند. در مقابل بازدهی و سرعت روش دوم با لاتر است چرا که در روش اول، داده ها برای رسیدن از هر Slave به Master، تا انتهای مسیر را بروند و از Slave های دیگر عبور کنند. همچنین پیچیدگی در روش اول بیشتر است چرا که برای دسترسی به داده خاص، باید تشخیص داد که داده در کدام دستگاه است و چند کلاک خواندن آن داده نیاز است.

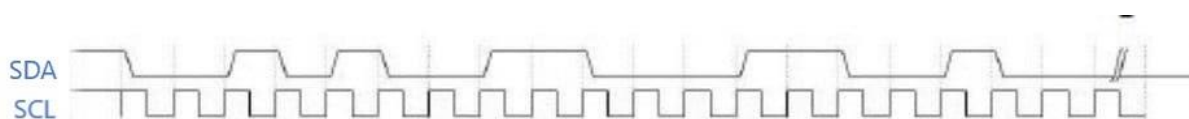


(۵) الف) سیگنال زیر نمایی از پروتکل I2C با مود ادرس ۷ بیتی است. مشخص کنید:

I) چه داده ای: 00110010

II) با کدام ادرس: 0101001

III) به صورت خواندن یا نوشتن: خواندن
منتقل میشود.



پاسخ:

در آغاز $SCL=1$ که SDA صفر میشود و به معنی شروع ارتباط است، بعد از آن ۷ بیت آدرس اسلیو، بعد از آن یک بیت خواندن یا نوشتن را مشخص میکند، یک بیت Ack اسلیو، داده ای اصلی ۸ بیتی، یک بیت ack پایان ارتباط از مستر به اسلیو

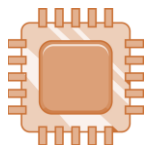
ب) در صورتی که بخواهیم ۱۰۰ بایت داده را از طریق دو رابط SPI و I2C ارسال کنیم، بازدهی (نسبت تعداد بیت داده به کل بیت ها) را در هر دو رابط محاسبه کنید. بازدهی کدام رابط بیشتر است؟

(فرض کنید در رابط SPI در هر frame حداکثر ۱۶ بیت داده می توان ارسال کرد).

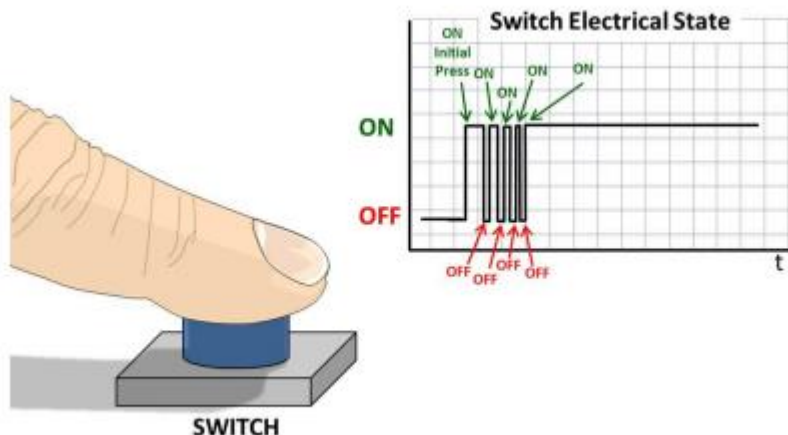
پاسخ:

SPI: در هر frame تمام ۱۶ بیت داده هستند، بنابراین به ازای هر تعداد بیت بازدهی برابر $16/16=1$ است.

I2C: برای ارسال داده نیاز به ۱ بیت شروع + ۷ بیت آدرس + ۱ بیت W/R + ۱ بیت ACK + ۱ بیت ACK به ازای هر بایت داده + ۱ بیت پایان داریم بنابراین برای ارسال ۱۰۰ بایت داده، $911 = 11 + 100 + 8 * 100$ بیت باید مبادله شود بنابراین بازدهی برابر $800/911$ می شود.
 $800/911 < 1$ در نتیجه بازدهی SPI بیشتر است.



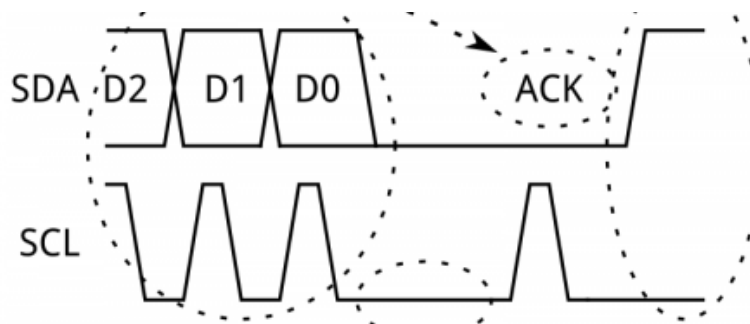
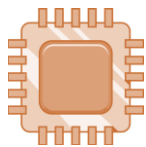
۶) الف) تصویر زیر پدیده‌ی Switch Bounce را نشان می‌دهد. نمونه‌ای از مشکلات احتمالی ناشی از این نوسان‌ها را بیان کنید و برای دوری از این مشکلات چه راه‌حلی وجود دارد؟ توضیح دهید.



پاسخ:

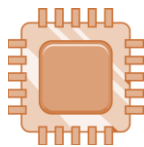
برای مثال بعد از فشردن دکمه بلند کردن صدا ممکن است صدا بیش از حد مطلوب بلند شود. زمانی که بخواهیم این نوسان‌ها را فیلتر کنیم از filter debouncing استفاده می‌شود. هنگامی که کلید مکانیکی فشرده یا رها می‌شود ممکن است خروجی برای مدتی نوسان کند، در برخی موارد این نوسان‌ها ممکن است منجر به رفتارهای ناخواسته و نامنتظره گردد. (برای نمونه ای جدی تر می‌توان به زمان فشار دادن دکمه افزایش سرعت تردمیل اشاره کرد).

ب) در مورد مفهوم clock stretching تحقیق کنید.



پاسخ :

گاهی اوقات ممکن است بعضی از دستگاههای **slave** کلاک را به اجبار پایین بکشند تا فرستادن اطلاعات بیشتر توسط **master** را به تاخیر بیندازند یا تا زمان بیشتری برای آماده سازی داده قبل از این که **master** آن را کلاک بزند درخواست کنند. به این عمل **clock stretching** گفته میشود.



- مهلت ارسال تمرین ساعت ۲۳:۵۵ روز دوشنبه ۱۷ آبان می باشد.
- سوالات خود را می توانید از طریق تلگرام از تدریسکارهای گروه خود بپرسید.
- ارائه پاسخ تمرین بهتر است به روش های زیر باشد:
(۱) استفاده از فایل docx. تایپ پاسخ ها و ارائه فایل Pdf
(۲) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا
- فایل پاسخ تمرین را تنها با قالب **HW2 - 9731***.pdf** در مدل بارگزاری کنید.
- نمونه: HW2- 9731097
- فایل زیپ ارسال نکنید.

10/25/2020



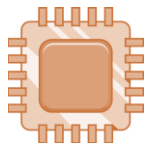
Homework 3

Lec 7-8

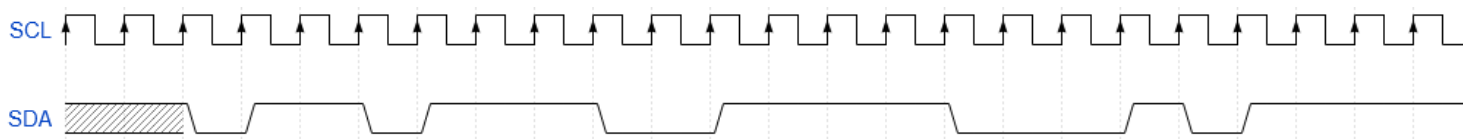


MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

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:

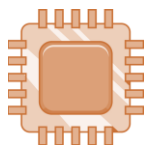
0-0110111-0-0-111100010-0-1

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

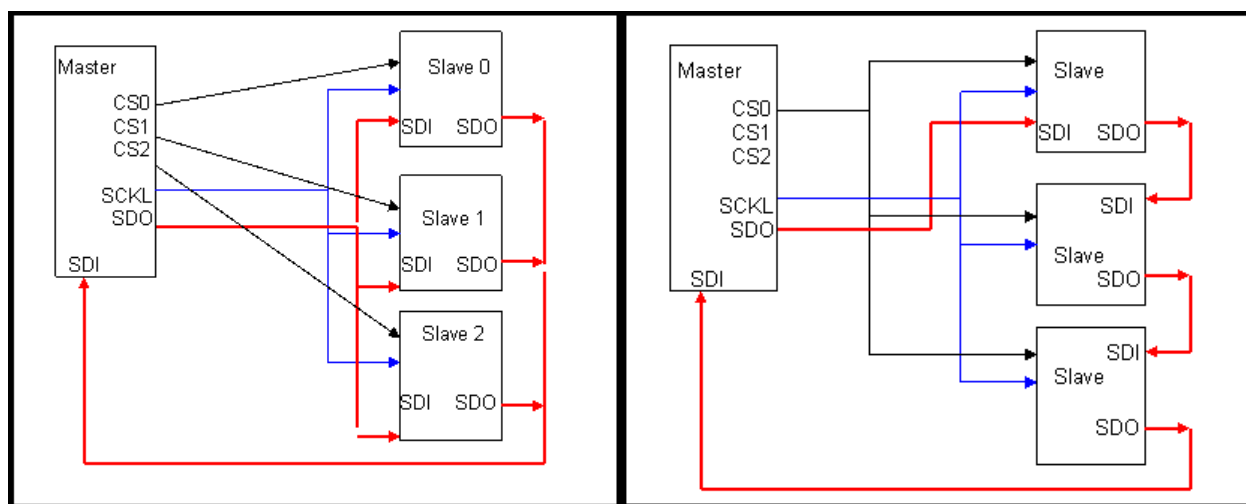
(۲) با توجه به این که هر دو رابط I2C و SPI مناسب انتقال داده با سرعت پایین و تست و دیباگ هستند، به همراه دلیل مشخص کنید که برای هر کدام از سناریوهای زیر، کدام یک از این دو رابط مناسب‌تر است:

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



۱. در این حالت، کار با رابط SPI ساده تر 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 پیکره بندی بیشتری نسبت به I2C دارد و اگر تعداد Slaveها زیاد باشد، I2C خیلی بصره تر است.
۳. برای این حالت تنها گزینه قابل قبول I2C است و استفاده از SPI در حالت معمول امکانپذیر نیست.

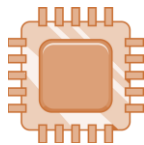
۳) برای پیاده سازی یک زنجیره از دستگاهها به کمک رابط SPI، دو توپولوژی زیر پیشنهاد شده است. این دو روش را مقایسه کنید و مزایا و معایب هر کدام را ذکر کنید:



روش دوم

روش اول

روش اول نیاز به CS (یا SS) کمتری نسبت به روش دوم دارد، بنابراین هزینه پیاده سازی سخت افزاری آن پایین تر است. همچنین جایگذاری قطعات در روش اول ساده تر است چرا که تنها لازم است زنجیره تشکیل شود و هر قطعه به قطعه بعدی نزدیک باشد در حالی که در روش دوم، همه قطعات باید همجواری مکانی داشته باشند. در مقابل بازدهی و سرعت روش دوم بالاتر است چرا که در روش اول، داده ها برای رسیدن از هر Slave به Master، 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 داده خاص، باید تشخیص داد که داده در کدام دستگاه است و چند کلاک خواندن آن داده نیاز است.

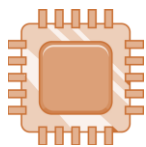
۴) مزایا و محدودیت‌های رابط I2C را نسبت به رابط SPI بیان کنید.

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

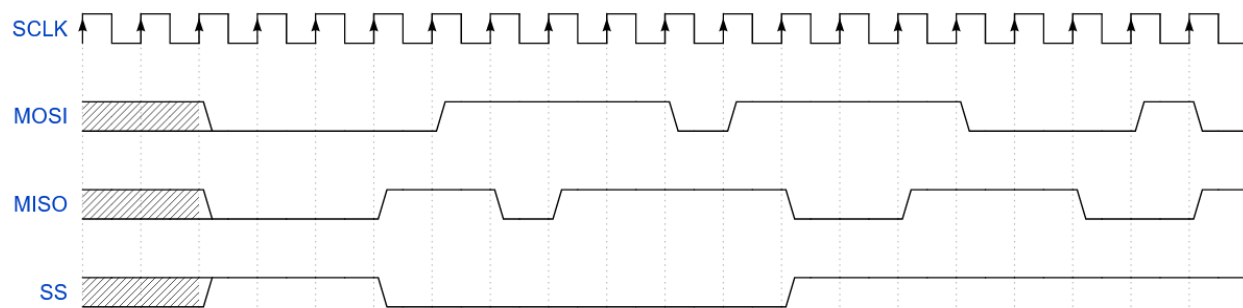
رابط SPI به دلیل نداشتن سیستم آدرس‌دهی و اطمینان از دریافت داده (نداشتن ACK و NACK)، برای Stream و انتقال داده‌ها در طولانی مدت بهتر است، در مقابل در استفاده از I2C اطمینان بالاتری از رسیدن یا نرسیدن داده‌ها وجود دارد و این درگاه برای خواندن و نوشتن روی آدرس‌ها مناسب‌تر است.

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.



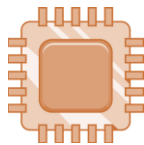
۵) در شکل موج زیر که توسط یک رابط SPI روی باس قرار می‌گیرد، زمان شروع و پایان ارسال داده، محتوا داده‌های ارسالی و نوع هر عملیات (نوشتن روی Slave یا Master) را مشخص کنید.



در انتقال با استفاده از SPI، ارتباط بین Slave و Master زمانی وجود دارد که سیگنال SS (یا CS) مقدار صفر داشته باشد. بنابراین در شکل موج بالا، تنها در ۷ کلاک 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 می باشد.

- سوالات خود را می توانید 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 می گردد.

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

- نمونه: HW2-G2-9531747

- فایل زیپ ارسال نکنید.