

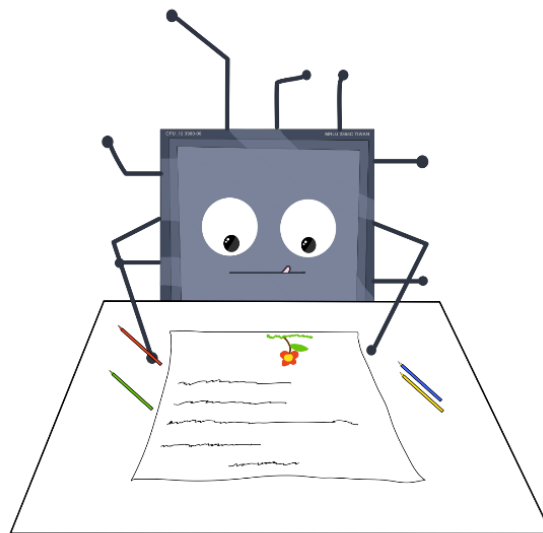


Department of Computer Engineering

Microprocessors and Assembly Language, Fall 2022, Dr. Farbeh

Homework 1 – Solutions

Lec 1-10





سوال ۱:

در ارتباط با بحث Microprocessor و Microcontroller به سوالات زیر پاسخ دهید:

الف) اجزای مختلف Microcontroller را نام ببرید. (چهار مورد)

ب) چهار تفاوت اصلی Microcontroller و Microprocessor را شرح دهید.

پ) آیا استفاده از Microcontroller ها به جای Microprocessor ها بهینه تر است؟ توضیح دهید.

پاسخ:

الف)

- Cpu
- Ram
- A/D Converter
- Internal Oscillator
- Memory
- I/O Ports
- Timers



(ب)

Microcontroller	Microprocessor
مرکز و قلب سیستم‌های نهفته	مرکز و قلب سیستم‌های کامپیوتری
در کنار حافظه داخلی و اجزای دیگر، شامل پردازش‌گرهای مختلف از جمله ریزپردازنده نیز می‌شود.	یک پردازش‌گر است. حافظه و سایر اجزا به صورت خارجی به آن متصل هستند.
بر خلاف ریزپردازنده از مدار کوچکتر برخوردار است.	به دلیل اتصال خارجی اجزا، از مدار بزرگتر برخوردار است.
به دلیل مدار کوچکتر در سیستم‌های جمع و جور به کار می‌روند.	به دلیل مدار بزرگتر در سیستم‌های جمع و جور به کار نمی‌روند.
به دلیل مصرف کمتر انرژی، مناسب برای سیستم‌های باتری دار است.	مصرف انرژی به دلیل اتصالات خارجی بالا است.
برخلاف ریزپردازنده دسترسی به حافظه به دلیل برخورداری از حافظه داخلی نسبتاً سریع است.	دسترسی به حافظه کند است.
بر اساس معماری هاروارد است. (حافظه برنامه و داده جدا است)	بر اساس معماری فون نیومن است. (برنامه و داده‌ها در یک حافظه ذخیره می‌شوند)

پ) بله با توجه به مقایسه بالا، از آن‌ها می‌توان در سیستم‌های فشرده استفاده کرد. همچنین حافظه کمتری نیاز دارند و باتوجه به انرژی که مصرف می‌کنند مناسب برای سیستم‌های باتری‌دار هستند اما نباید از این نکته غافل شد که اگر نیاز به دسترسی حجیمی از داده‌ها در زمان کم داریم، باید از ریزپردازنده‌ها استفاده کنیم.



سوال ۲:

به پرسش‌های زیر در مورد NVIC پاسخ دهید:

(الف) Vector table چیست و محتوای آن چیست و آدرس آن در کجا ذخیره شده است؟

(ب) با در نظر گرفتن این موضوع که NVIC-IPR اعداد بدون علامت در خود ذخیره می‌کند، چگونه وقفه‌هایی با اولویت منفی داریم؟

(پ) ۴ حالت کاری وقفه NVIC را نام برده و حالت Active and Pending را به صورت مختصر شرح دهید.

(ت) توضیح دهید چرا برای ذخیره اولویت‌بندی وقفه‌ها نیاز به ۶۰ رجیستر ۳۲ بیتی داریم؟ (منظور همان رجیسترهای

NVIC_IPR0 – NVIC_IPR59 است)

پاسخ:

(الف) یک جدول ۱۶ ردیفه است که شماره هر ردیف آن شماره وقفه مربوطه است. محتوای هر ردیف آن جدول آدرس ISR مربوط به آن وقفه است. متغیری هم با اسم Vectors داریم که در SCB ذخیره شده است که آدرس شروع همین Vector Table است. کاربرد این جدول این است که هنگامی که وقفه ای رخ می‌دهد، به کمک این جدول می‌توان آدرس ISR مربوط به آن را پیدا کرد. مثلاً در سطر یک در ابتدا یک DCD وجود دارد که در زبان اسمبلی یک فضای یک بعدی ۳۲ بیتی را اختصاص می‌دهد و سپس رو به روی آن Reset_handler قرار دارد که لیبل هست که به آدرس شروع ISR مربوط به آن وقفه اشاره می‌کند.

(ب) این وقفه‌های ایستا مانند Reset و ... مدارشان به صورت سخت‌افزاری پیاده‌سازی شده است و مانند بقیه وقفه‌ها، رجیستری ندارند که به آن به صورت نرم‌افزاری و دستی مقدار دهیم.

(پ)

- Inactive
- Pending
- Active
- A&P

در این حالت چنانچه یک وقفه در حال رسیدگی باشد و همزمان دوباره سیگنال وقفه آن از دستگاه به پردازنده فرستاده شود، در این صورت وقفه در حالت A&P قرار خواهد گرفت.

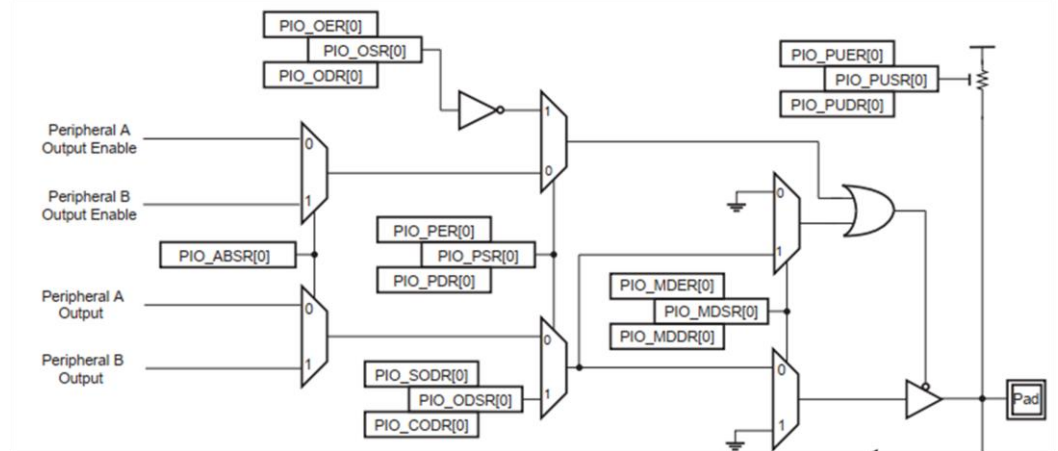


ت) برای اولویت‌بندی هر یک از وقفه‌ها به ۸ بیت نیاز داریم (حداکثر حالت) حال ۲۴۰ وقفه هم می‌توانیم داشته باشیم. بنابراین 240×8 برابر می‌شود با ۱۹۲۰. حال باید این ۱۹۲۰ بیت را در رجیسترهای ۳۲ بیتی ذخیره کنیم در نتیجه به ۶۰ رجیستر ۳۲ بیتی نیاز خواهیم داشت.

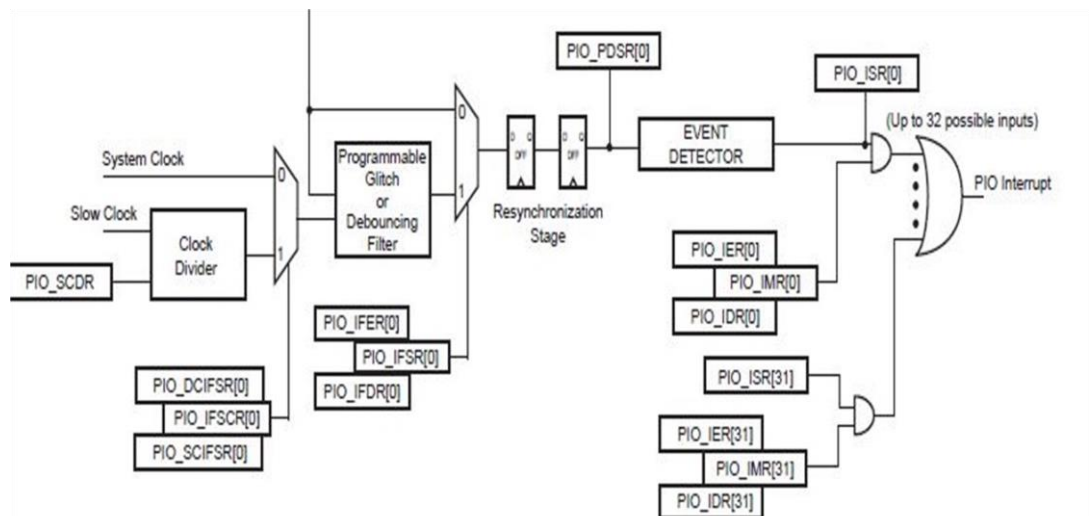


سوال ۳:

الف) با توجه به تصویر رو به رو که منطق کنترلی پایه شماره صفر یک واحد PIO را نشان می دهد، اگر بخواهیم خروجی پایه تحت کنترل Peripheral A باشد (مقدار پایه ورودی بافر سه حالت برابر با Peripheral A Output باشد و مقدار پایه فعال سازی آن برابر با Peripheral A Output Enable باشد)، بیت صفر هر یک از رجیسترهای PIO_ABSR، PIO_MDSR و PIO_PSR باید چه مقداری داشته باشد؟ (پاسخ خود را به ترتیب به صورت PIO_PSR[0]PIO_ABSR[0]PIO_MDSR[0] مانند XXX بنویسید)



ب) در شکل زیر Flip Flop ها چه کاربردی دارند و چرا فقط در مدار مربوط به ورودی از آن ها استفاده می شود؟





پاسخ:

(الف)

مقدار رجیستر $PIO_ABSR[0]$ یکی از peripheral های A یا B را انتخاب می کند. با توجه به صورت سوال، مقدار این رجیستر برابر با صفر است تا Peripheral A انتخاب شود ($PIO_ABSR[0] = 0$). مقدار رجیستر $PIO_PSR[0]$ مشخص می کند که پایه مورد نظر به خروجی متصل شود یا از یکی از peripheral ها مقدار بگیرد. در اینجا مقدار این رجیستر برابر با صفر است تا حالت peripheral فعال شود ($PIO_PSR[0] = 0$). در نهایت مقدار رجیستر $PIO_MDSR[0]$ ورودی بافر سه حالته را مشخص می کند. در اینجا مقدار این رجیستر برابر با صفر است تا ورودی بافر سه حالته به Peripheral A Output متصل شود ($PIO_MDSR[0] = 0$). با توجه به موارد تعیین شده، خروجی OR که به پایه فعال سازی بافر متصل است برابر با مقدار Peripheral A Output Enable است؛ زیرا ورودی دیگر آن برابر با صفر است و ورودی بافر نیز به Peripheral A output متصل است. پاسخ نهایی به فرم 000 خواهد بود.

(ب)

وجود Flip Flop ها در مدار ورودی باعث می شود تا داده ورودی با کلاک درون سیستم همگام شود (در واقع عمل resynchronization را انجام می دهند). داده ای که از دنیای خارج می آید با کلاک سیستم همگام نیست و می دانیم خروجی Flip Flop همواره با کلاک سیستم همگام است. به همین دلیل برای همگام سازی داده ورودی از دو Flip Flop استفاده می کنیم. در مدار خروجی، داده مورد نظر از درون خود سیستم انتخاب می شود که با کلاک درونی سیستم همگام است و به همین دلیل در مدار خروجی به Flip Flop نیازی نداریم.



سوال ۴:

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

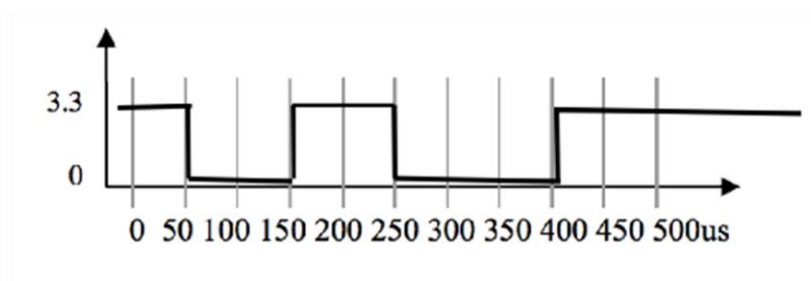
پاسخ:

- 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.
- 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 به صرفه تر است.
- برای این حالت تنها گزینه قابل قبول I2C است و استفاده از SPI در حالت معمول امکان پذیر نیست.



سوال ۵:

الف) فرض کنید شکل زیر یک فریم ۱۰ بیتی است (یک بیت شروع، ۸ بیت داده و یک بیت پایان). با فرض این که خط قبل و بعد از انتقال داده بیکار است، ۸ بیت داده را مشخص کنید.



ب) مقدار baud rate در این کانال چند bit/s است؟

ج) در حالت کلی فرض کنید که BR مقدار baud rate یک کانال UART بر حسب bits/s است. رابطه ماکزیمم bandwidth و BR چیست؟

پاسخ:

الف) هشت بیت از ۱۰۰ شروع و تا ۵۰۰ ادامه دارد: ۰۱۱۰۰۰۱۱

ب)

$$BR = \frac{1}{50\mu s} = \frac{20kb}{s}$$

ج)

$$\begin{aligned} \text{Bandwidth} &= \text{Bitrate} / N \\ N &= \text{number of bit per symbol} \\ \text{Bitrate} &= \text{baudrate} \times K \\ K &= \text{number of bit per baud} \end{aligned}$$

بنابراین:

$$BW = \frac{K}{N} BR$$

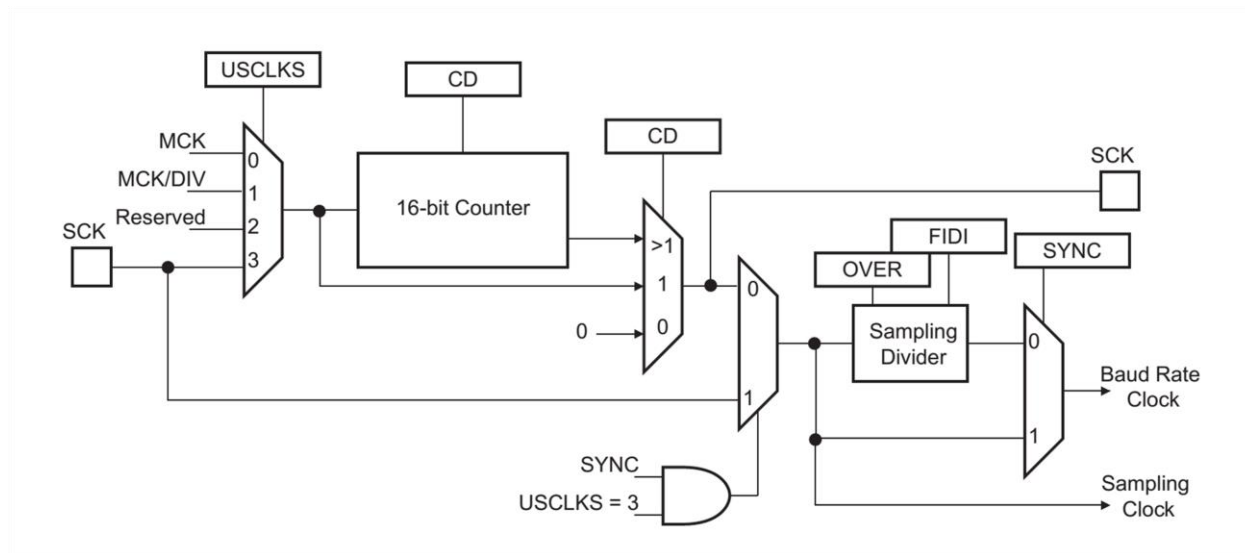


سوال ۶:

به سوالات زیر در مورد baud rate generator برای ارتباط USART پاسخ دهید.

$MCK = 512\text{MHz}$

$MCK/DIV = 128\text{MHz}$



الف) در حالت آسنکرون در صورتی که $\text{baud rate} = 8\text{Kbps}$ باشد، رجیسترهای OVER، SYNC، USCLKS و CD چه مقادیری باید داشته باشند؟

ب) فرض کنید در حالت سنکرون $\text{baud rate} = 2\text{Kbps}$ باشد، مقادیر ممکن برای رجیسترهای SYNC، USCLKS و CD را بدست



پاسخ:

(الف)

$SYNC = 0$

حالت اول: اگر $USCLK = 0$ (یعنی MCK انتخاب شود)

اگر $OVER = 0$:

$$baud\ rate = \frac{MCK}{8(2 - OVER)CD} = \frac{512 * 10^6}{8 * 2 * CD} = 8 * 10^3 \rightarrow CD = 4000$$

اگر $OVER = 1$:

$$baud\ rate = \frac{MCK}{8(2 - OVER)CD} = \frac{512 * 10^6}{8 * CD} = 8 * 10^3 \rightarrow CD = 8000$$

حالت دوم: اگر $USCLK = 1$ (یعنی $\frac{MCK}{DIV}$ انتخاب شود)

اگر $OVER = 0$:

$$baud\ rate = \frac{\frac{MCK}{DIV}}{8(2 - OVER)CD} = \frac{128 * 10^6}{8 * 2 * CD} = 8 * 10^3 \rightarrow CD = 1000$$

اگر $OVER = 1$:

$$baud\ rate = \frac{\frac{MCK}{DIV}}{8(2 - OVER)CD} = \frac{128 * 10^6}{8 * CD} = 8 * 10^3 \rightarrow CD = 2000$$



ب) چون در حالت سنکرون هستیم مقدار over برای ما بی اثر است.

$$SYNC = 1$$

حالت اول اگر $USCLK = 0$:

$$baud\ rate = \frac{MCK}{CD} = \frac{512 * 10^6}{CD} = 2 * 10^3 \rightarrow CD = 256000 > 2^{16} = 65536$$

از آنجا که مقدار CD بیشتر از ۱۶ بیت می شود این حالت امکان پذیر نیست.

حالت دوم اگر $USCLK = 1$:

$$USCLK = 1$$

$$baud\ rate = \frac{MCK}{CD * DIV} = \frac{128 * 10^6}{CD} = 2 * 10^3 \rightarrow CD = 64000 < 2^{16} = 65536$$

در نتیجه مقادیر $USCLK = 1$, $SYNC = 1$ و $CD = 64000$ خواهد بود.



سوال ۷ (امتیازی):

در مورد وقفه‌ها به سوالات زیر پاسخ دهید:

الف) سیاست tail chaining و late-arriving و pulse & level را در NVIC بررسی کنید.

ب) دلایل وجود Masking را نام برده و ۳ نوع masking را که می‌توانیم تعدادی از وقفه‌ها را همزمان Mask کنیم را به صورت مختصر شرح دهید.

پ) فرض کنید می‌خواهیم وقفه شماره ۲ را Disable کنیم و وقفه شماره ۱۱ را Enable کنیم و اولویت آن‌را به ۶ تغییر دهیم. محتوای کدام یک از رجیسترهای NVIC تغییر می‌کند؟ (وقفه‌ها از شماره ۱ شروع می‌شوند)

پاسخ:

الف) در Tail chaining این‌گونه عمل می‌کنیم که مثلاً در حال اجرای یک ISR هستیم و یک وقفه جدید می‌آید که باید بعد از این وقفه به آن رسیدگی کنیم. حال باید بعد از اتمام وقفه اول، state قبلی سیستم را از stack لود کنیم و سپس بلافاصله دوباره همان‌ها را به خاطر رسیدگی به وقفه جدید به stack بازگردانیم که کاری کاملاً بیهوده است اما به کمک ویژگی tail chaining پس از اتمام وقفه اول بلافاصله وقفه دوم را شروع می‌کنیم و دیگر انتقالات بیهوده نداریم.

در Late arriving هم هنوز ISR وقفه فعلی شروع نشده و فعلاً در حال انتقال state فعلی پردازنده به stack هستیم و حال اگر وقفه جدیدی که اولویت بالاتری داشته باشد، برسد با استفاده از تکنیک late-arriving ابتدا ISR وقفه با اولویت بالاتر را اجرا می‌کنیم. یعنی ذخیره سازی در stack به قصد وقفه اول بود اما پس از ذخیره سازی در stack به وقفه دوم می‌پردازیم.

کاربرد سیاست pulse and level هم این است که چون سیگنال‌های وقفه‌ها از دیوایس‌های ورودی می‌آیند و تغییر وضعیت در یک سیگنال ورودی ما را متوجه وقفه می‌کند، این قابلیت به ما کمک می‌کند تا تنظیم کنیم چه نوع تغییری را به عنوان وقفه تشخیص دهیم. مثلاً حساس به لبه و

ب) از قابلیت masking می‌توان بسته به شرایط به صورت نرم افزاری کاری کرد که تعدادی یا اکثر وقفه‌ها نتوانند اجرای برنامه یا قسمتی از قطعه کد شما را متوقف کنند مثلاً هنگامی که حال اجرای critical section هستیم می‌توانیم قبل از اجرای آن وقفه‌ها را متوقف کنیم و پس از اتمام دوباره اجازه اجرا به وقفه را بدهیم.



در پردازنده مورد مطالعه ۳ نوع masking داریم که بر شرح زیر است:

• PRIMASK

در این روش یک رجیستر یک بیتی داریم که اگر مقدار آن را یک کنیم در اصل اولویت اجرای برنامه ما صفر می‌شود؛ یعنی اگر در حال اجرای قطعه کد مهمی هستیم و نمی‌خواهیم توسط وقفه‌ای متوقف شود باید این رجیستر را ست کنیم تا اولویتمان صفر شود. حال اولویت تمام وقفه‌های قابل تنظیم صفر و ۱ و به بالا بود و چون ما اولویتمان صفر شده دیگر وقفه دیگری نمی‌تواند کار ما را متوقف کنند.

• BASEPRI

• یک رجیستر هشت بیتی است که اگر بر فرض مقدار آن را K تنظیم کنیم، آنگاه اولویت قطعه کد ما همان K می‌شود و وقفه‌هایی با ارزش $k-1, k-2$ و ... می‌توانند ما را متوقف کنند. اما وقفه‌هایی با ارزش $k, k+1$ و ... دیگر نمی‌توانند ما را متوقف کنند. همچنین اگر مقدار این رجیستر صفر باشد، یعنی BASEPRI کلاً غیرفعال است.

• FAULTMASK

• یک رجیستر یک بیتی است که اگر مقدار آن را ست کنیم مشابه همان PRIMASK عمل می‌کند اما وقفه ۱- هم نمی‌تواند کار ما را قطع کند. در اصل انگار اولویت برنامه ما اولویت ۱- را دارد و فقط دو وقفه ۲- و ۳- می‌توانند کار ما را قطع کنند یعنی HardFault هم نمی‌تواند کار ما را قطع کند.

(پ)

NVIC_ICER0: 0x00000002 for Masking "2"

NVIC_ISER0: 0x00000400 for enable "11"

NVIC_IPR2: 0x00060000 for change priority of "11" to 6