



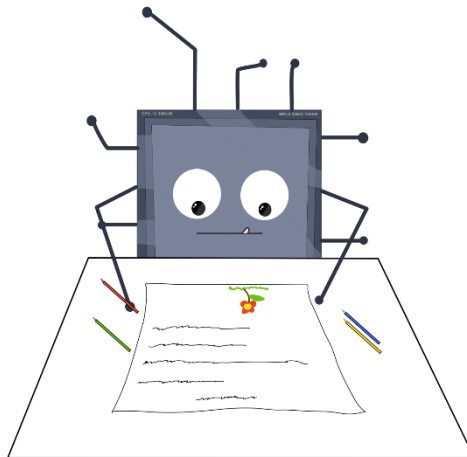
Department of Computer Engineering

Microprocessors and Assembly Language, Fall 2022, Dr. Farbeh

---

# Midterm Answers

---





## سوال 1:

الف) میکروبی که ما در درس استفاده می‌کنیم (SAM3X8E) از کدام یک از معماری‌های Harvard یا Von Neumann استفاده می‌کند و دلایل آن چیست (دو دلیل)؟

ب) چند مورد از برتری‌هایی که باعث شده است در سیستم‌های نهفته از Microcontroller استفاده شود را نام ببرید (سه مورد کافی است).

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

## پاسخ:

میکرو ما از معماری Harvard پیروی می‌کند.

### دلیل 1:

با توجه به اینکه میکرو ما بیشتر برای سیستم‌های نهفته استفاده می‌شود و مانند کامپیوترهای روزمره نیازی ندارد که برنامه‌ای که روی رم آن قرار دارد تغییر کند و معمولاً برنامه‌ای که روی آن‌ها قرار می‌گیرد تا مدت زمان زیادی دچار تغییر نمی‌شود، در نتیجه این دو حافظه از هم جدا هستند تا سرعت پردازش بیشتر باشد.

### دلیل 2:

با توجه به صحبت‌های مطرح شده در کلاس می‌دانیم که پردازنده کامپیوترهای ما معماری Von Neumann است. دلیلی که این معماری جوابگو است، وجود cache می‌باشد که در لایه‌های پایین‌تر و به طور مخفیانه از معماری هاروارد استفاده می‌کند. اما چون میکرو ما حافظه بزرگی ندارد و همان‌طور که در دلیل 1 گفته شد، تغییرات زیادی بر روی دستورات ذخیره شده در حافظه نداریم، cache هم نداریم، پس معماری این میکرو هاروارد است و حافظه دستورات از حافظه دیتا جدا می‌باشد.



(ب)

- تمامی حافظه ها و I/O ها در درون یک میکروکنترلر قرار دارد و نیازی نیست این دیوایس ها را جداگانه 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 برای فضای با اندازه کوچک است.
- دسترسی به حافظه سریع است.
- قدرت پردازشی زیادی ندارند و به همین دلیل ارزان و مناسب برای سیستم های نهفته هستند.
- مصرف کمتر انرژی
- و هر یک از موارد گفته شده در اسلاید

(ج)

- Backup Mode 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.



## سوال 2:

به سوالات زیر در مورد NVIC پاسخ دهید:

الف) دو حالت مختلف Active و A&P برای وقفه ها در NVIC را شرح دهید.

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

ج) انواع حالاتی که می توانیم با استفاده از رجیسترهای CPU جمعی از وقفه ها را باهم Mask کنیم را شرح دهید.

## پاسخ:

(الف)

حالت Active زمانی است که ISR در حال اجرا شدن باشد.

حالت A&P برای زمانی است که یک وقفه در حال اجرا باشد و باز همان دیوایس وقفه جدیدی بفرستد.

(ب)

Tail Chaining: این ویژگی به این معناست که اگر وقفه ای در حال اجرا باشد و وقفه دیگری رخ دهد، حال یکی از این وقفه ها به حالت pending می رود؛ پس از اتمام یکی از آن ها دیگر لازم نیست رجیسترهای برنامه اصلی را باز لود کرده و بعد از آن باز سیو کنیم و به سراغ وقفه دوم برویم و پردازنده باهوش عمل می کند و بلافاصله بعد از اتمام وقفه اول به سراغ دومی می رود و پس از اتمام آن رجیسترهایی که به صورت سخت افزاری سیو شده بودند را لود می کند.

Late Arriving: فرض کنید در حین سیو کردن رجیسترهای یک وقفه باشیم که یک وقفه با اولویت بیشتر رخ دهد. در این حالت با توجه به اینکه وقفه دوم اولویت بیشتری دارد، این وقفه انجام می شود با اینکه ما برای وقفه اول رجیستر ها را سیو کردیم (وقفه ای که دیرتر رسیده است اجرا میشود). Tail chaining مربوط به انتهای اجرای وقفه هاست در حالی که Late arriving مربوط به ابتدای اجرای آن هاست.

(ج)

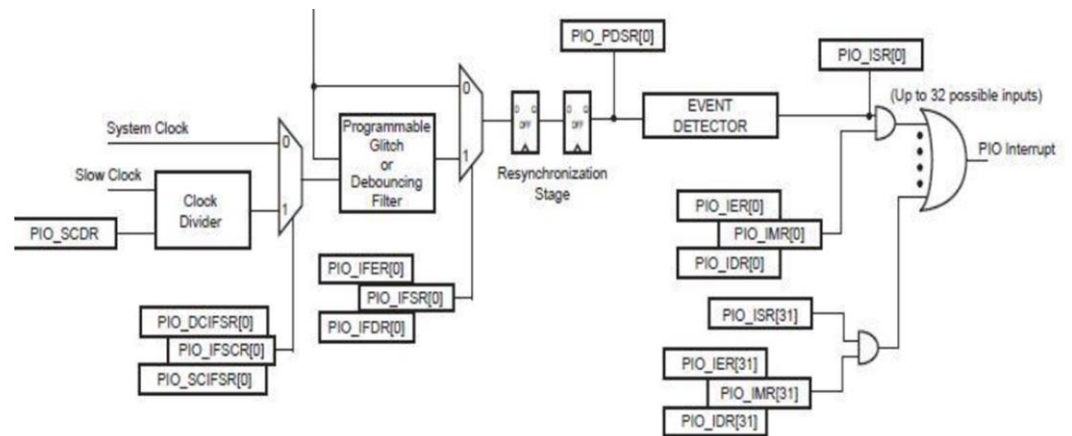
Primask: یک رجیستر تک بیتی که همه وقفه های با اولویت 0 به بالا را mask می کند.

Basepri: یک رجیستر 8 بیتی که یک عدد بین 1 تا 240 در آن قرار می گیرد و از آن اولویت به بعد mask می شود.

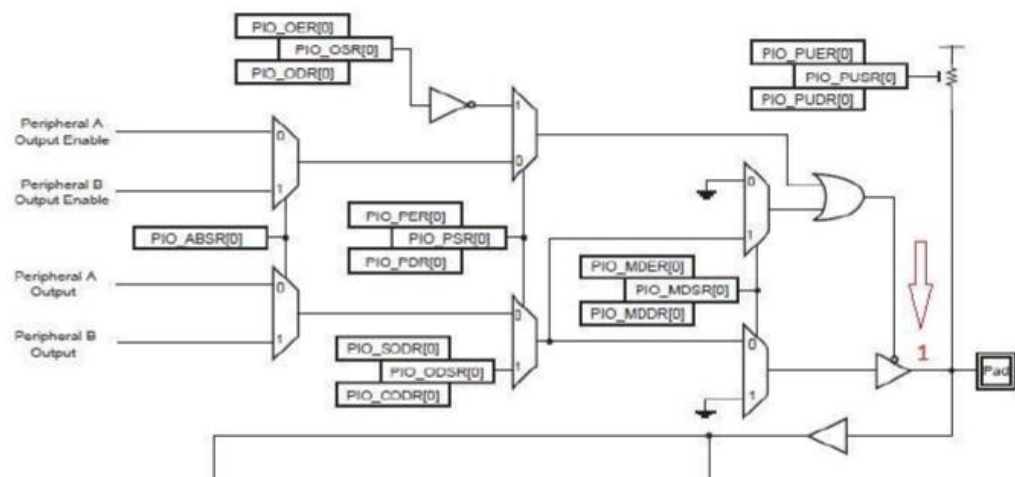
Faultmask: یک رجیستر تک بیتی کاملاً مشابه با Primask صرفاً با این تفاوت که وقفه هایی با اولویت منفی 1 به بالا را mask می کند. (وقفه Hard Fault که یک وقفه سخت افزاری نیز هست).

### سوال 3:

الف) در شکل زیر چگونه متوجه شویم کدام پین 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:

(الف)

در هر یک از پین‌ها که تعدادشان 32 تا است، یک Event Detector وجود دارد که در صورت یک شدن، یعنی یک event آمده است. در هر کدام از پین‌ها یک ذخیره می‌شود و میتوان به سادگی با نگاه کردن به PIO\_ISR مقدار آن در بیت متناظر آن پایه رجیستر متوجه شد بیت چندم آن 1 شده است و میتوان متوجه شد کدام یک از پین PIO\_ISR رجیستر 32 بیتی‌ها باعث ایجاد وقفه شده است.

(ب)

$PIO\_ODSR = 1$  مقدار خروجی مدنظر کاربر را مشخص میکند و از آنجایی که بافر سه حالته یک 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 حالته 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.



#### سوال 4:

- الف) مزایا و محدودیت های رابط I2C را نسبت به رابط SPI بیان کنید.
- ب) مزایا و محدودیت های رابط UART را نسبت به رابط های SPI بیان کنید.
- ج) چهار تفاوت پروتکل های UART و USART را بیان کنید.

#### پاسخ:

الف)

مزایا:

- I2C تنها به دلیلی داشتن دو پین SDA و SCK، معماری ساده تری دارد.
- I2C، مستعد پذیرش نویز کمتری است.

محدودیت:

- در SPI بیت stop و start وجود ندارد بنابراین داده ها می توانند بدون وقفه و به طور مداوم منتقل شوند.
- SPI، full-duplex می باشد.
- دارای سرعت بیشتری نسبت به I2C می باشد.
- در SPI به دلیل وجود MISO و MOSI داده ها می توانند به صورت همزمان ارسال و دریافت شوند.

ب)

SPI بیشترین فاصله ی ممکن را در بین UART و SPI و I2C دارد در حالی که UART فاصله ی کمتری در حدود 50 feet را پوشش می دهد. یکی از بزرگترین تفاوت ها این است که UART یک نوع سخت افزار است در حالی که SPI یک پروتکل است. تعداد pin های مورد نیاز در UART برابر 2 عدد است در حالی که SPI حداقل به 4 pin نیاز دارد و در نتیجه در هنگام طراحی اگر تعداد pin ها محدود باشد انتخاب SPI چندان مناسب نیست. UART تنها ارتباط 1 به 1 را ساپورت می کند ولی SPI به جهت داشتن master/slave می تواند تعداد slave ها را افزایش داده و تعداد ارتباط های بسیار بیشتری را ساپورت کند. SPI به طرز قابل توجهی از UART سریعتر است به طوری که حتی در برخی مواقع ممکن است SPI سه برابر سریعتر عمل کند. از لحاظ قیمت به پارامترهای زیادی بستگی دارد اما به طور کلی SPI ارزان تر است. همچنین SPI به طور synchronous است اما UART به شکل asynchronous عمل می کند. از نظر اندازه نیز دیوایس های SPI نسبت به چیپ های UART فضای کمتری را اشغال می کنند.



(ج)

۱. سرعت USART بیشتر از UART است.
۲. USART برای انجام عملیات خود هم از سیگنال‌های دیتا و هم از clock استفاده می‌کند در حالی که UART تنها از سیگنال‌های دیتا استفاده می‌کند.
۳. در USART، انتقال داده به صورت بلاک انجام می‌شود در حالی که در UART، به صورت بایت‌هایی (یک بایت در هر لحظه) است.
۴. USART می‌تواند به گونه‌ای عمل کند که همان UART باشد ولی UART این قابلیت را ندارد.
۵. USART پیچیده‌تر از UART است.
۶. در USART، انتقال داده با یک نرخ ثابتی انجام می‌شود ولی در UART، با سرعت‌های متفاوتی ممکن است انتقال داده انجام شود.



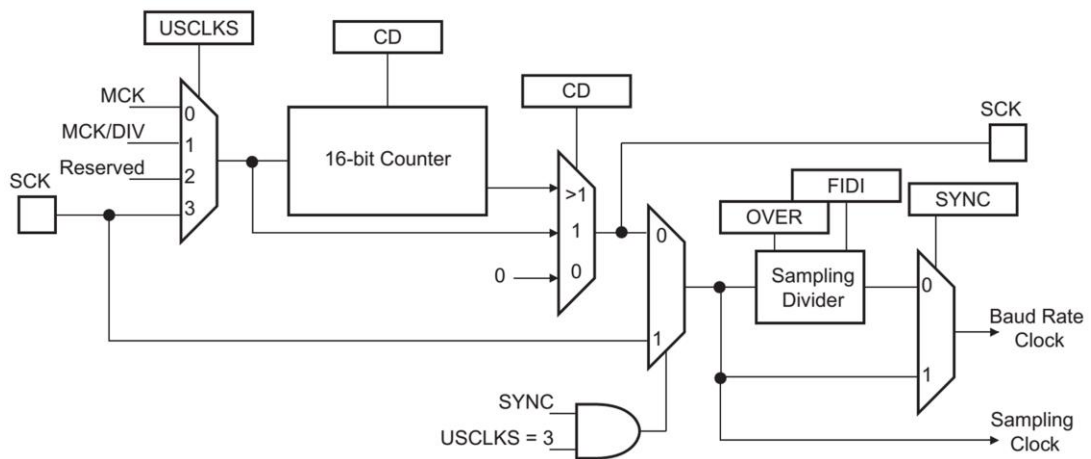


سوال 5:

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

MCK = 4GHz

MCK/DIV = 512MHz



الف) اگر در حالت آسنکرون باشیم و در صورتی که baud rate = 4Kbps باشد رجیسترهای OVER, CD, USCLKS, SYNC چه مقادیری باید داشته باشند؟

ب) اگر در حالت سنکرون باشیم و فرض کنیم baud rate = 32Kbps باشد مقدار رجیسترهای CD, USCLKS, SYNC چقدر باید باشد؟



پاسخ:

الف)

$$SYNC = 0$$

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

اگر  $OVER = 0$

$$baud\ rate = \frac{MCK}{8(2 - OVER)CD} = \frac{4 * 10^9}{8 * 2 * CD} = 4 * 10^3 \rightarrow CD = 62500$$

اگر  $OVER = 1$

$$baud\ rate = \frac{MCK}{8(2 - OVER)CD} = \frac{4 * 10^9}{8 * CD} = 4 * 10^3 \rightarrow CD = 125000 > 2^{16} = 65536$$

از آنجا که مقدار CD بیشتر از 16 بیت می شود این حالت امکان پذیر نیست.

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

اگر  $OVER = 0$

$$baud\ rate = \frac{\frac{MCK}{DIV}}{8(2 - OVER)CD} = \frac{512 * 10^6}{8 * 2 * CD} = 4 * 10^3 \rightarrow CD = 8000$$

اگر  $OVER = 1$

$$baud\ rate = \frac{\frac{MCK}{DIV}}{8(2 - OVER)CD} = \frac{512 * 10^6}{8 * CD} = 4 * 10^3 \rightarrow CD = 16000$$



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

$$SYNC = 1$$

حالت اول اگر  $USCLK = 0$ :

$$baud\ rate = \frac{MCK}{CD} = \frac{4 * 10^9}{CD} = 32 * 10^3 \rightarrow CD = 125000 > 2^{16} = 65536$$

از آنجا که مقدار CD بیشتر از 16 بیت می شود این حالت امکان پذیر نیست.

حالت دوم اگر  $USCLK = 1$ :

$$USCLK = 1$$

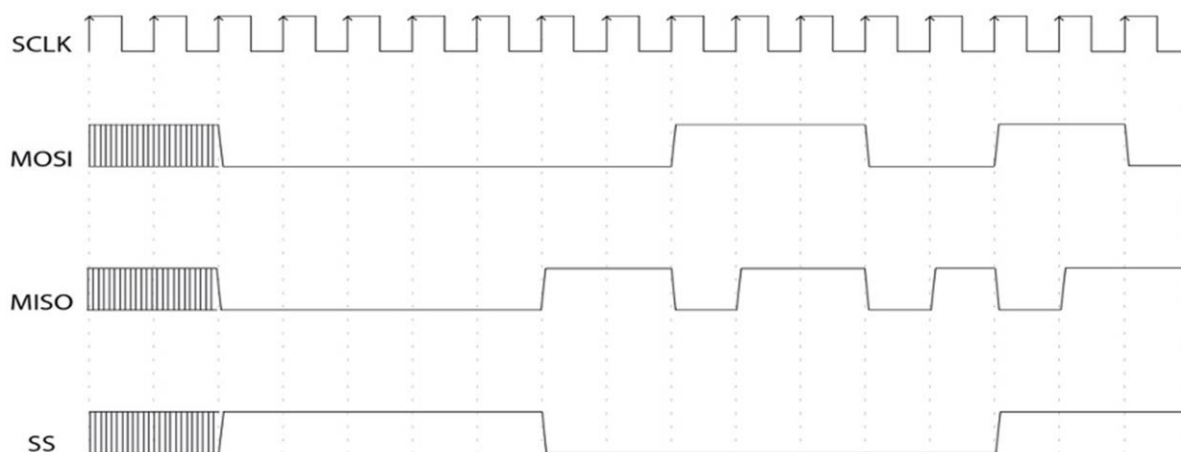
$$baud\ rate = \frac{MCK}{CD * DIV} = \frac{512 * 10^6}{CD} = 32 * 10^3 \rightarrow CD = 16 * 10^3 = 16000 < 2^{16} = 65536$$

در نتیجه مقادیر  $USCLK = 1$ ,  $SYNC = 1$  و  $CD = 16000$  خواهد بود.



## سوال 6:

در شکل موج زیر که توسط یک رابط SPI روی باس قرار می گیرد، زمان شروع و پایان ارسال داده، محتوای داده ارسالی و نوع هر عملیات (نوشتن روی slave یا master) را مشخص کنید.



## پاسخ:

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 7 کلاک انتقال داده صورت میگیرد.

سیگنال های ارتباطی به صورت زیر هستند:

این سیگنال، داده انتقالی از مستر به اسلیو است.  $MOSI = 0011100$

این سیگنال، داده انتقالی از اسلیو به مستر است.  $MISO = 1101101$



سوال 7:

اگر بخواهیم با پروتکل I2C یک بایت داده بر روی دستگاه Slave بنویسیم، آنگاه از Master به Slave چند بیت فرستاده خواهد شد؟

پاسخ:

18 بیت، یک بیت برای start، 7 بیت برای مشخص کردن آدرس، یک بیت برای نوع داده (خواندنی/نوشتنی)، 8 بیت data، یک بیت

stop