



## شماره تکلیف: ۴

## سوال ۱.

فکر می کنم چون توی جدول گفته شده ادرس ۳۲ بیتی پس باید ۸ رقم داشته باشه و منظور از آدرس 0x10 0000 در واقع 0x1000 0000 بوده است.

الف) با استفاده از فرمول زیر، آدرس بردار وقفه را محاسبه می کنیم:

$$\begin{aligned} \text{Interrupt Vector Address} &= \text{Start Address of VT} + 4 \times \text{Interrupt Number} \\ \Rightarrow \text{Interrupt Vector Address} &= 0x1000\ 0000 + 4 \times 32 \\ &= 0x1000\ 0000 + (1000\ 0000)_2 = 0x1000\ 0000 + 0x80 = 0x1000\ 0080 \end{aligned}$$

(ب)

نام وقفه	آدرس ۳۲ بیتی بردار وقفه در حافظه	محتوای ۱ بایت از حافظه
Reset	0x1000 0004	0xXX
	0x1000 0005	0xXX
	0x1000 0006	0xXX
	0x1000 0007	0xXX
⋮	⋮	⋮
وقفه شماره ۳۲	0x1000 0080	0x02
	0x1000 0081	0x40
	0x1000 0082	0x01
	0x1000 0083	0x20

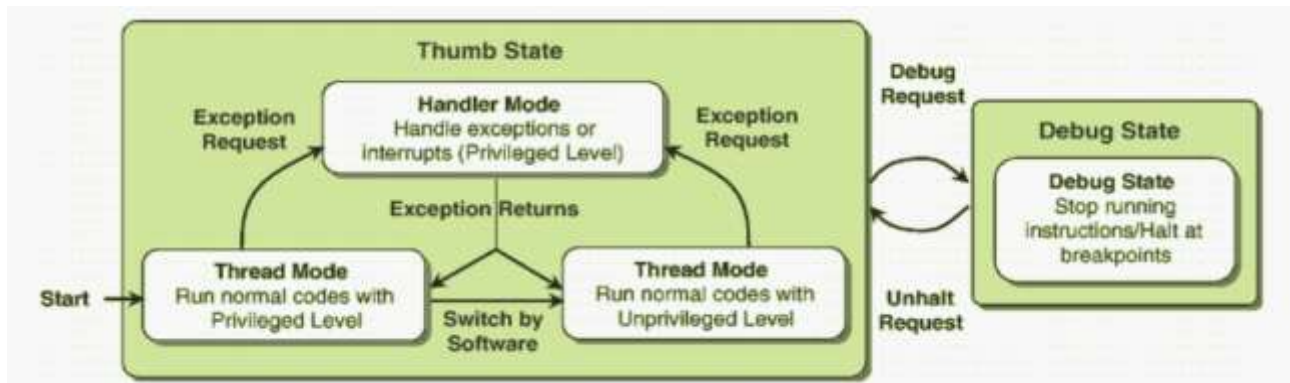
## سوال ۲.

برای انجام یک عمل Push ابتدا ۴ واحد از SP کم کرده و مقدار آن را به 0x1001 23FC می رسانیم. سپس جدول زیر را با اطلاعات مناسب پر می کنیم:

آدرس ۳۲ بیتی حافظه	محتوای ۱ بایت از حافظه
0x1001 23FC	0x44
0x1001 23FD	0x20
0x1001 23FE	0x00
0x1001 23FF	0x11



## سوال ۳.



با توجه به شکل نمودار حالات و مد های عملیاتی، برنامه به طور کلی دارای قسمت ها و در نتیجه حالت های زیر می باشد:

- روشن شدن سیستم: با توجه به start در تصویر، سیستم وارد Thread Mode با Privileged Level (سطح دسترسی ممتاز) می شود.

- اجرای برنامه های کاربردی: با توجه به سطح دسترسی برنامه، توسط نرم افزار بین دو قسمت Thread Mode با سطح دسترسی Privileged (ممتاز) و Unprivileged (غیر ممتاز) Switch می شود.

- بوجود آمدن یک رخداد یا وقفه (زدن کلید یا رسیدن به زمانهایی خاص که موجب ارسال وقفه به میکروکنترلر می شود): خروج از Thread Mode و ورود به Handler Mode با سطح دسترسی Privileged (ممتاز). هنگامی که فعالیت های خواسته شده انجام می شوند، برنامه به حالت قبلی خود (Thread Mode قبلی با سطح دسترسی ممتاز یا غیر ممتاز) باز می گردد.

- رسیدن به BreakPoint: هنگام رسیدن به BreakPoint برنامه از حالت قبلی خود خارج شده و به Debug State می رود. سپس پس از اتمام فعالیت ها به حالت قبلی خود (Handler Mode با سطح دسترسی ممتاز یا Thread Mode با سطح دسترسی ممتاز یا غیر ممتاز) باز می گردد.



#### سوال ۴.

(الف)

A: روش پیشنهادی، روش مبتنی بر سرکشی (Polling) است. در این روش پردازنده به صورت متوالی و در زمان های مشخص شده ای به تجهیزات سرکشی می کند.

B: روش پیشنهادی، روش مبتنی بر وقفه (Interrupt) است. در این روش پردازنده به محض فعال شدن یکی از تجهیزات جانبی، به سرویس دهی می پردازد و با توجه به اینکه مانند روش سرکشی، همیشه در حال بررسی تجهیزات نیست، از نظر توان مصرفی کارکرد بهتری دارد.

(ب)

روش مبتنی بر سرکشی: برای مواردی که زمان انجام فعالیت ها برایمان مهم و مشخص هستند کاربرد دارند و می توانند زمان انجام هر سک از قسمت های برنامه را مشخص کنند.

روش مبتنی بر وقفه: برای مواردی که توان مصرفی پایین نیاز داریم، این روش مفید است و منجر به بهینگی برنامه می شود.