

سوال (۱)

ما سه مدل interrupt داریم :

مدل اول (IRQs) Interrupt requests :

ناهمزمان (Asynchronous): به کدی که پردازنده در حال حاضر در حال اجرای آن است مربوط نمی شود. مثال ها: وقفه اعلام شده است، کاراکتر در serial port یا ADC دریافت می شود ، مبدل تبدیل را تمام می کند.

مدل دوم (NMI) Non-maskable Interrupt :

شبیه به IRQ هستند اما غیرفعال نمی شود(non-maskable)

مدل سوم (Exceptions, Faults, software interrupts (Generated by core) :

همزمان (synchronous): نتیجه اجرای دستورات خاص هستند.

مثال ها: دستورالعمل های تعریف نشده، سرریز برای یک دستور داده شده رخ می دهد.

مدل چهارم SysTick Timer :

تایمر systick یک تایمر شمارش معکوس 24 بیتی با بارگذاری مجدد خودکار است.

به طور کلی برای ارائه یک وقفه دوره ای برای زمانبندی RTOS استفاده می شود.

رجیستر BASEPRI برای اولویت دهی وقفه ها استفاده میشود. اولویت دهی وقفه زمانی معنی دار میشود که چند تا وقفه هم زمان به سیستم ما اعمال شوند. بعضی از وقفه ها اولویتشون از قبل تنظیم شده و ثابت است مثال reset همیشه بالاترین اولویت که 3- است را دارد. ولی اولویت بقیه ادوات جانبی و peripheral ها قابل تنظیم هستند. عدد اولویت هرچه کمتر باشد اولویت بیشتر است و عدد اولویت هرچی بالاتر باشد سطح اولویت پایین تر است.

اولویت دهی وقفه ها :

حالت اول : در آن واحد و هم زمان چند تا درخواست روتین وقفه با هم برسد. هرچه شماره اکسپشن پایین تر باشد اولویت بالاتر است و سرویس دهی طبق اولویت انجام میشود .

حالت دوم : یک درخواست وقفه یا اکسپشنی بیاد در حالی که الان در ترد مود نباشیم و در سرویس دهی یک روتین دیگر باشیم یعنی در حال اجرای اکسپشن هندلر یک اکسپشن دیگه باشیم. اون موقع پردازنده چک میکند کسی که درخواست داده اولویتش از اولویت اون وقفه ای که الان دارد سرویس دهی میکند بالاتر یا پایین تر. اگر اولویتش بالا تر بود اجرای روتین وقفه فعلی متوقف میشود و وضعیت استک میشود و به صورت تو در تو وقفه دیگری سرویس دهی میشود. اما اگر اولویت آن کوچکتر مساوی باشد از قابلیت پندینگ استفاده میکنیم یعنی بیت پندینگ اکسپشن جدید فعال میشود و در استیت پندینگ باقی میماند. صبر میکنیم تا اجرای اکسپشن هندلر فعلی به اتمام برسد سپس اولویت وقفه قبلی که توی سیستم تنظیم شده بود بازیابی میشود زیرا ممکن است وقفه دیگری به صورت تو در تو اینترپت داده بوده باشد و متوقف کرده باشد پس نگاه میکنیم قبلی چیست. اگر برگشته بود توی مود ترد یا برگشته بود توی یک اکسپشن هندلر دیگری که اونم تو در تو بود، دوباره اولویت هارا چک میکنیم اگر اولویت وقفه آخری که آمده بود بالاتر بود اجرا میشود و اگر نبود دوباره به وضعیت فعلی اجرای برنامه یا اکسپشن هندلر میرویم تا در نهایت اولویت به سطحی برسد که اکسپشنی که در لحظه اومده بود بتواند سرویس دهی شود.

سوال (2)

تفاوت اصلی بین وقفه و سرکشی در این است که در صورت وقفه، دستگاه به CPU اطلاع می دهد که نیاز به توجه دارد در حالی که در مورد سرکشی، CPU به طور مداوم وضعیت دستگاه را بررسی می کند تا دریابد که آیا نیاز به توجه دارد یا خیر.

Interrupt	Polling
Efficient - code runs only when necessary	Slow - need to explicitly check to see if switch is pressed
Fast - hardware mechanism	Wasteful of CPU time - the faster a response we need, the more often we need to check
Scales well <ul style="list-style-type: none">- ISR response time doesn't depend on most other processing.- Code modules can be developed independently.	Scales badly - difficult to build system with many activities which can respond quickly. Response time depends on all other processing.

سوال (3)

هنگامی که پردازنده Cortex-M یک درخواست exception را می پذیرد، پردازنده باید آدرس شروع کنترل کننده exception را تعیین کند.

این اطلاعات در vector table در حافظه ذخیره می شود. به طور پیش فرض، vector table از آدرس حافظه 0 شروع می شود و vector address بر اساس exception number در چهار ضرب می شود.

Vector table مورد استفاده در راه اندازی همچنین حاوی مقدار اولیه main stack pointer (MSP) است. این مورد نیاز است زیرا برخی استثناها مانند NMI ممکن است زمانی اتفاق بیفتد که پردازنده به تازگی از حالت تنظیم مجدد خارج شده که یعنی قبل از اجرای هر مرحله initialization.

Vector table ها در پردازنده های Cortex-M با Vector table ها در پردازنده های سنتی ARM مانند ARM7TDMI متفاوت است. در پردازنده های سنتی ARM، جداول برداری حاوی دستورالعمل هایی مانند دستورالعمل های شاخه ای برای انشعاب به کنترل کننده های مناسب هستند، در حالی که در Cortex-M، Vector table حاوی آدرس های آغازین exception handlers است.

معمولاً آدرس شروع (x00000000) باید حافظه بوت باشد و یا فلش مموری یا دستگاه های ROM خواهد بود و در زمان اجرا نمی توان مقدار را تغییر داد. با این حال، در برخی از برنامه ها بهتر است که بتوان exception vectors را در زمان اجرا تغییر داد یا تعریف کرد. برای انجام این کار، پردازنده های Cortex-M3 و Cortex-M4 از ویژگی به نام Vector Table Relocation پشتیبانی می کنند.

Vector Table Relocation ویژگی ای دارد که یک رجیستر قابل برنامه ریزی به نام Vector Table Offset (VTOR) Register را ارائه می دهد. این رجیستر آدرس شروع حافظه مورد استفاده را به عنوان Vector Table تعریف می کند.

سوال 4)

در هنگام وقوع وقفه EXC_RETURN Code در LR قرار میگیرد.

از زمان وقوع وقفه (با فرض عدم وجود وقفه ی دیگر) تا زمان شروع اجرای اولین روتین سرویس وقفه، حداکثر 12 سیکل CPU طول میکشد.

منابع سوالات: کتاب YIU JOSEPH و اسلاید های استاد و صحبت های استاد.