سوال اول

الف)

در واحد ISA در پردازنده نیازمند سه تابع Load/Store و Control و Arithmetic/Logic است تا بتواند مانند یک ISA کامل کار کند.

ب)

هر واحد ISA مشخصات و ویژگی های کلیدی دارد تا بتواند به خوبی عمل کند. به عنوان مثال:

- نحوه دسترسی به حافظه (Memory)، اینکه چطور با حافظه ارتباط برقرار شود نظیر ارتباط مستقیم و ...
- طول هر دستورالعمل (Instruction Length)، اینکه Risc یا Cisc باشد بر روی این ویژگی تاثیر دارد (معمولا Cisc دستورات طولانی تری دارد)
- فرمت دستورالعمل ها (Instruction Format) که مشخص میکند دستورالعمل چگونه در ISA ذخیره شود و در نتیجه چطور توسط سیستم باید خوانده شود.
- Operand Format که مشخص میکند چه اوپراتور هایی داشته باشیم، به چه تعداد و چقدر حجیم میتوانند باشند.

سوال دوم

الف)

میکرو SAM3X8E از طراحی Harvard استفاده میکند. در طراحی Von Neumann یک موری برای ذخیره کردن دستورات و دیتا داریم در صورتی که در طراحی Harvard دو موری داریم: یکی برای دستورات و دیگری برای دیتا. در نتیجه دو تا Memory bus خواهیم داشت در نتیجه پردازنده میتواند همزمان هم Read و هم Write انجام دهد (در طراحی Von Neumann این امکان را نداریم.) تفاوت دوم این است که در طراحی Von Neumann برای اجرای هر دستور به دو cycle نیاز داریم اما در طراحی Harvard تنها به یک کلاک نیاز است.

ب)

(۱) به آن دلیل است که در Microcontroller ها علاوه بر پردازنده مرکزی، موری و IO هم داریم اما در Microprocessors ها تنها پردازنده موجود است. (2) به همین علت Microcontroller ها مدار کوچک تری دارند چون IO داخل آنها تعبیه شده در صورتی که Microprocessors نیاز به IO خارجی دارند پس مدار بزرگ تری خواهند داشت. (3) Microcontroller بر اساس طراحی Harvard ساخته شده اند که مردن تر و با مزیت های بیشتری است ولی Microprocessors بر اساس طراحی Von Neumann ساخته شده اند.

ج)

این کرو در حالت Low Power Modes سه حالت دارد: Backup Mode و Wait Mode و Sleep Mode. در Backup Mode که کمترین مقدار مصرف انرژی را دارد پردازنده، IO، موری و کلاک های نظیر آنها خاموش است و این حالت را

تقریبا در Hibernate بودن دستگاه مشاهده میکنیم. در حالت Sleep Mode صرفا کلاک پردازنده خاموش است و تمام IO ها مشغول به کار هستند (به وسیله DMA) که در نتیجه بیشترین مصرف انرژی و سریع ترین زمان فعال شدن مجدد را دارد. در حالت Wait Mode تمام کلاک های پردازنده و IO نمیخورد اما خود پردازنده، IO و مموری آماده فعالیت هستند.

سوال سوم

الف)

این سه تایمر Real-time Timer و Real-time Clock و Watchdog Time هستند. اولی به طول متناوب وقفه ایجاد میکند تا ثانیه ها را بشمارد که قابل تغییر نیست (Read-only است) و در آلارم و شمارش کوتاه مدت استفاده میشود. دومی برای ذخیره تاریخ است به طوری که اگر 60 ثانیه گذشت به دقیقه اضافه کرده و همینطور به ساعت، روز، هفته و ... اضافه میکند؛ به عبارتی تاریخ را از 200 سال گذشته با یاد دارد. سومی برای جلوگیری از هنگ کردن سیستم استفاده میشود به طوری که اگر یک برنامه برای مدتی بیشتر از مقدار Watchdog Timer هنگ بماند سیستم آن را میبندد؛ این تایمر تا 16 ثانیه را میشمارد.

ب)

این شکل مربوط به Input Schmitt Trigger است. از آن جایی که ولتاژ هیچگاه 0 یا 5 تمیز نیست (نویز دارد) دستگاه های حساس سریع آسیب میبینند. برای جلو گیری از این اتفاق این ماژول یک محدوده قابل قبول برای ولتاژ در نظر میگیرد. برای مثال ولتاژ 0–2.5 را 0 و ولتاژ 2.5–5.5 را 1 منطقی در نظر میگیرد؛ به این سبک، دستگاه های حساس آسیب نمیبینند. (این اعداد فرضی بوده و میتوانند بسته به دستگاه تغییر کنند)

سوال چهارم

الف)

شیوه برخورد NVIC با وقفه ها به این صورت است که به صورت Dynamic آنها را اولویت بندی میکند و هدف آن است که وقفه ها با اولویت بالاتر همیشه قبل از وقفه ها با اولویت کمتر اجرا شوند (به ترتیب اولویت) حتی اگر دیرتر Trigger شده باشند. به عنوان مثال اگر دو وقفه تو در تو بوده و وقفه داخل اولویت بالاتری داشته باشد، ابتدا Process State در Stack ذخیره شده و Process State برای وقفه دوم (که اولویت بالاتری دارد) تخصیص داده میشود. بعد اتمام آن Process State را از Stack خوانده و دوباره در پردازنده Restore میکنیم تا اولویت اول ادامه کار خود را بکند. اما اگر اولویت دو وقفه یکسان باشد تفاوتی بین آنها نمی گذارد. این رویه برای n وقفه تو در تو صادق است.

ب)

دستورات Load Multiple و Store Multiple و Push و Pop و MULS دستورات طولانی بوده و نیاز به تعداد بیشتری کلاک برای اتمام شدن دارند. در صورتی که یکی از این وقفه ها رخ دهند دستور فعلی را Pending کرده، به وقفه جواب میدهد (آن را اجرا میکند) و سپس ISR را اجرا کرده و از وقفه خارج میشود (الان وقفه انجام شده) و دوباره به دستوری که ndingPe شده بود باز میگردد.

سوال پنجم

الف)

در حالت Active زمانی که پردازنده مشغول به انجام یک وقفه هست وقفه دیگری به پردازنده اعمال نمیشود در صورتی که در A&P هین انجام این وقفه، امکان پذیرش وقفه دیگر توسط پردازنده امکان پذیر است. این زمانی اتفاق می افتد که وقفه ای در حال انجام است و نمیخواهیم در حین آن اختلالی در پردازش رخ دهد.

ب)

در Tail Chaining پردازنده وقفه ها را پشت هم و بدون معطلی انجام میدهد. به طوری که وقتی وقفه اول انجام شد دیگر پردازنده مقادیر داخل Register ها را pop نمیکند (چون تاثیری ندارد) و در عوض مقادیر Register های وقفه بعدی را push کرده. این هیچ تاثیری روی اختلال وارد کرده بر روی عملکرد یا Stack ندارد، بلکه سرعت پردازش را بالاتر میبرد.

در Late Arriving اگر وقفه با اولویت بالاتر هنگام اجرا شدن یک وقفه رخ دهد، وقفه جدید انجام شده State ذخیره شدن را Skip کرده و درجا اجرا میشود. این روش قابلیت آن را میدهد که به محض تمام شدن وقفه با اولویت بالا، وقفه اولیه به وسیله Tail Chaining اجرا شود.

ج)

در پردازنده، CPU Masking به طوری است که یک بیت به نام Mask بر روی هر هسته CPU قرار میگیرد و به Job Scheduler میگوید که کدام CPU را برای یک process یا thread خاص استفاده کند. این شیوه زمانی بدرد میخورد که میخواهیم یک برنامه فقط از یک یا چند هسته استفاده کند و نه بیشتر. پس برای محدود کردن هسته ها برای هر process استفاده میشود.

برای Interrupt Masking چندین روش داریم: primask و basepri و faultmask. در صورتی که PRIMASK را معادل 1 بگذاریم اولویت 0 برای وقفه در نظر گرفته میشود. با BASEPRI میتوان اولویت یک وقفه را عوض کرد و مقدار صفر برای این 8 بیت قابلیت Masking را برای وقفه غیر فعال میکند. در FAULTMASK اگر 1 شود اولویت 1– برای وقفه در نظر گرفته میشود و باعث میشود نسبت به بسیاری از وقفه های دیگر اولویت بگیرد.

سوال ششم

الف)

یک Interrupt Vector Table جدول است (یا به عبارتی یک mapping) برای متصل کردن هر وقفه به Interrupt Hander مربوطه که آدرس آن در این جدول ذخیره میشود. این آدرس به کد مربوط به Handler اشاره دارد. این جدول هر بار که یک وقفه رخ میدهد مورد استفاده قرار گرفته و با توجه به آدرس Handler مربوطه، دستورات مورد نیاز اجرا میشود.

ب)

زیرا خانه شماره ۰ برای Stack تعبیه شده و در نتیجه بقیه وقفه ها باید از شماره 1 شروع شوند که در صورت ریست شدن اولین ماژول لود شده است.

ج)

زیرا این وقفه ها قابل تغییر نیستند و این روش، راهی برای نشان دادن این موضوع است. این وقفه ها سیستمی هستند (در سخت افزار نوشته شده اند و configurable نیستند). وقفه های قابل تغییر پذیر از اعداد مثبت (۱، ۲، ...) شروع میشوند.

کیوان ایپچی حق – ۹۸۳۱۰۷۳