تمرین اول ریزپردازنده‌ها و اسمبلی

اشکان شکیبا (۹۹۳۱۰۳۰)

سوال اول

الف) حجم کمتر، کارایی بالا، توان مصرفی حداقلی، دسترسی سریع به حافظه، هزینه کم

ب) microprocessor یک پردازنده کامل برای اجرای برنامه‌های پیچیده است که عموما در کامپیوترهای شخصی، سرورها و موارد مشابه استفاده می‌شود و تنها کار پردازش داده‌ها و دستورها را انجام می‌دهد.

microcontroller شامل یک پردازنده نسبتا کوچک‌تر به همراه حافظه، ورودی / خروجی و سایر بخش‌های مورد نیاز است که عموما در دستگاه‌های کوچک‌تر و ساده‌تر (سیستم‌های نهفته) استفاده می‌شود.

پ) بسته به کاربرد، می‌تواند باشد.

همان‌طور که در بخش قبل گفته شد، در دستگاه‌های کوچک و ساده که نیاز به پردازش سنگین نیست، مانند دستگاه‌های پزشکی، خودروها، لوازم خانگی و ...، استفاده از microcontroller بسیار بهینه‌تر است. در حالی که در دستگاه‌های پیچیده‌تر اینطور نیست.

سوال دوم

الف)

Accumulator Machine, Register-Memory Machine, Load-Store Machine, Stack Machine

ب)

variable-length instruction: دستورات پیچیده‌تر و گسترده‌تری می‌توان نوشت اما نیاز به چند مرحله fetch و decode دارد.

fixed-length instructions: طول دستورات ثابت است و انعطاف کمتری دارند، اما راحت‌تر fetch و decode شده و قابلیت pipelining بیشتری دارند.

پ) دستورات control، دستورات arithmetic و logic، دستورات load و store

سوال سوم

الف) ممکن است سایر بخش‌ها بخواهند بدون پردازنده به کار خود ادامه دهند، مانند direct memory access.

ب) از glitch filters به منظور حذف پالس‌های ناخواسته و کوتاه که کمتر از یک کلاک دارند در سیگنال‌های ورودی و خروجی استفاده می‌شود؛ در حالی که debouncing برای از بین بردن خطاهای ناشی از bounce سوییچ‌ها و تنها برای سیگنال‌های ورودی مورد استفاده قرار می‌گیرد و از یک شمارنده برای تاخیر ورودی و تولید سیگنالی پایدار استفاده می‌کند.

پ) real-time timer که ثانیه‌ها را شمرده و می‌تواند به صورت دوره‌ای، وقفه ایجاد کند.

real-time clock که به عنوان یک تقویم میلادی استفاده می‌شود و برای فعال‌سازی رویدادها کاربرد دارد.

watchdog timer که شمارنده‌ای رو به کاهش دارد و در صورت عدم انجام عملیات مورد نظر در زمانی تعیین شده، که می‌تواند ناشی از مشکلاتی چون dead lock باشد، سیستم را ریست می‌کند.

سوال چهارم

الف) وقفه reset با اولویت ۳، که آسنکرون است و وظیفه ریست سیستم را دارد.

وقفه non-maskable با اولویت ۲، که آسنکرون است و تنها وقفه reset قادر به توقف آن است.

وقفه hard fault با اولویت ۱، که سنکرون است و در هنگامی که کنترلر خطاهای قابل تنظیم غیرفعال شده استفاده می‌شود.

وقفه memory management با اولویت صفر و قابل تعیین.

ب) ابتدا مقدار 0xFF بر روی رجیستر نوشته می‌شود و سپس مقداری که به آن بازگردانی می‌شود خوانده می‌شود که اگر کمتر از 0xFF باشد، بیت‌های اولویت‌بندی را تعیین می‌کند.

سوال پنجم

برای وقفه A، نیاز به ۴۵ کلاک برای اجرا و ۱۵+۱۵ کلاک برای load و save هست. یعنی مجموعا ۷۵ کلاک.

برای وقفه B نیز نیاز به ۳۰ کلاک برای اجرا و ۱۵+۱۵ کلاک برای load و save هست. یعنی مجموعا ۶۰ کلاک.

در مجموع زمان کل فرآیند ۱۳۵ کلاک می‌شود که ۷۵ کلاک از آن صرف اجرای وقفه‌ها شده است؛ بنابراین بازده به صورت زیر است:

۷۵ / ۱۳۵ = ۵۵٪

(البته با در نظر گرفتن ۱۵ کلاک برای اولین load هنگام گرفتن کنترل از برنامه و ۱۵ کلاک دیگر برای آخرین save هنگام بازگشت به برنامه، مجموعا ۱۶۵ کلاک و بازدهی ۴۵% خواهیم داشت)

سوال ششم

الف) بسته به مقدار PIO_PSR، اگر یک باشد خروجی برابر با مقدار PIO_ODSR و اگر صفر باشد بستگی به peripheral output enable دارد، که اگر یک باشد خروجی ۱ و اگر نه مقدار peripheral output خواهد بود.

ب)

PIO_PSR = 1, PIO_OSR = 1, PIO_MDSR = 0

سوال هفتم

الف) دلیل ۳ حالته بودن بافر قرمز این است که در حالت ورودی نباید داده‌ای از آن منتقل شود.

دلیل ۲ حالته بودن بافر آبی نیز این است که مستقل از درستی ورودی، می‌توان از آن صرف نظر کرد؛ پس نیازی به بافر ۳ حالته نیست.

ب)

PIO_ABSR = 0, PIO_PSR = 0, PIO_PUSR = 0, PIO_MDSR = 0