بهار ۱۳۹۶

میان‌ترم

طراحی مدارهای واسطه

مهشید علی‌نوری

پرهام الوانی

سوال ۱

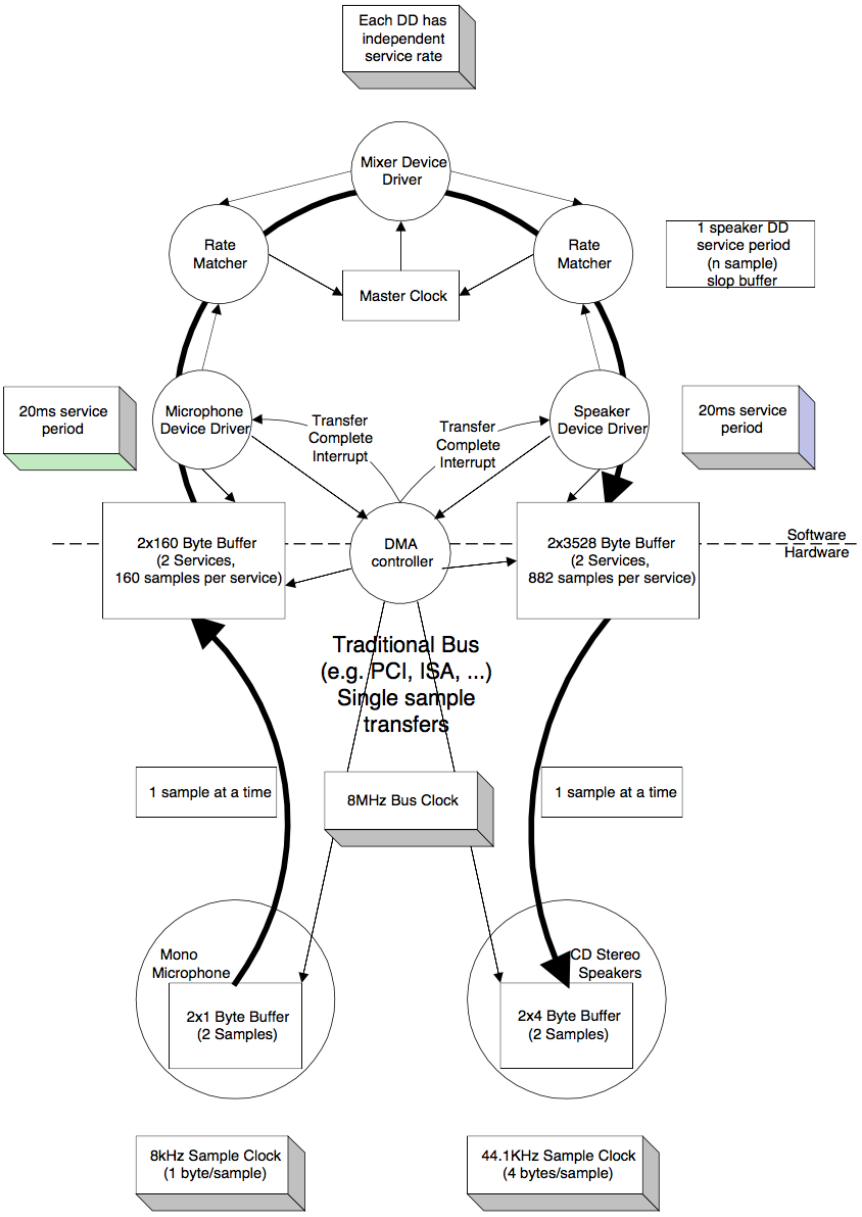
در ارتباطات isochronous نیاز است که ارسال‌کننده و دریافت‌کننده از نظر داده‌ای و زمانی با یکدیگر همگام باشند. USB در ارتباطات isochronous از باز ارسال پشتیبانی نمی‌کند بنابراین امکان ندارد که همگامی زمانی در تاخیر باز ارسال از دست برود. با این وجود نیاز است که همگامی در زمان ارسال عادی داده و در زمانی که در bus خطا رخ داده است حفظ شود.

برای همگام‌سازی در بیشتر سیستم‌های isochronous مانند PSTN از یک clock مشترک برای همه‌ی موجودیت‌های سیستم استفاده می‌گردد، اما USB مدل clockای تعریف می‌کند که اجازه می‌دهد تعداد زیادی دستگاه مختلف روی یک bus وجود داشته باشند و هزینه‌ی معقولی برای پیاده‌سازی داشته باشد.

در ادامه دو مثال برای مقایسه‌ی پیاده‌سازی ارتباطات isochronous به صورت کلی (باس‌های PCI، ISA و ..) و در USB آورده شده است.

به صورت کلی برای ارتباطات isochronous از یک clock مشترک استفاده می‌گردد. در این حالت دستگاه‌های جانبی داده‌ها را به وسیله‌ی DMA با درایورهای نرم‌افزاری مبادله می‌کنند و DMA مرز بین لایه نرم افزار و سخت افزار قرار می‌گیرد. درمثال ارائه شده clock مورد استفاده Bus در این 8MHz می‌باشد. در صورتی که هریک از endpoint ها دیتا داشته باشند آن را روی bus قرار میدهند و DMA آن را به بافر انتقال میدهد و در نهایت بخش نرم افزاری را با استفاده از interrupt باخبر می‌سازد و device driver ها دیتا را از روی بافر می‌خوانند. واضح است که در بخش نرم‌افزاری mixer driver با master clock سیستم کار می‌کند و نرخ درخواست دیتا یا فرستادن دیتا با آن مشخص می‌شود و عملکرد این بخش از bus و بخش سخت‌افزاری مستقل است. هر یک از device ها با نرخ مختص خودشان دیتا رو روی bus قرار می‌دهند یا از روی آن برمیدارند. به طور کلی در این مدل دو clock در سیستم مورد استفاده قرار می‌گیرد که یکی همان clock بخش نرم افزاری

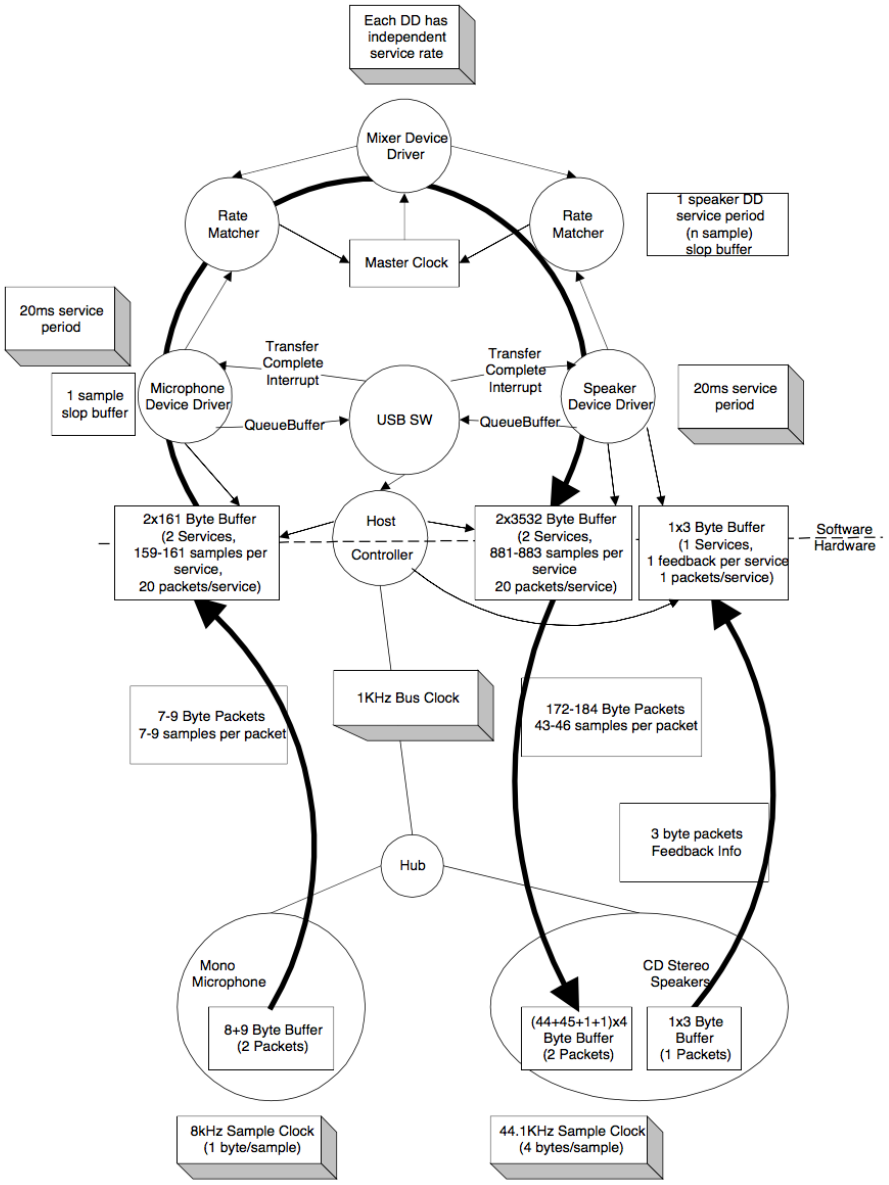
است که فرکانس بالایی دارد و دیگری clock مربوط به bus میباشد.



شکل 1 ارتباطات isochronous در حالت non-usb

به صورت کلی برای ارتباطات isochronous از یک clock مشترک استفاده می‌گردد. در این حالت دستگاه‌های جانبی داده‌ها را به وسیله‌ی DMA با درایورهای نرم‌افزاری مبادله می‌کنند. clock مورد استفاده Bus در این 8MHz می‌باشد. در این سیستم قسمت نرم افزاری با clock سیستم کار می‌کند و میکروفن و بلندگو نیز clockهای خود را دارند که با یکدیگر نا همگام هستند. نرم افزار mixer نیاز دارد که ورودی و خروجی را با نرخ مشخصی دریافت کند و برای این کار از rate matcher استفاده می‌شود. دستگاه‌های ورودی، دستگاه‌های خروجی

و DeviceDriverهایشان می‌بایست بتوانند داده‌ها را در وقفه‌های سخت‌افزاری DMA بخوانند یا بنویسند. این دستگاه‌ها از دو بافر سیستمی استفاده می‌کنند به صورتی که یک بافر منحصرا تحت دسترسی DMA می‌باشد و بافر دیگر در صورت خالی شدن بافر DMA به آن داده می‌دهد یا از آن داده می‌خواند. زمان سرویس مدت زمانی است که برای یک پردازه در نظر گرفته می‌شود (در اینجا پردازه درایور نرم افزاری دستگاه می‌باشد) و این زمان در سیتسم عامل‌های مختلف متفاوت می‌باشد و می‌بایست به گونه‌ای انتخاب شود که بار وقفه‌ها خیلی زیاد نباشد.



شکل 2 ارتباطات isochronous در حالت usb

این در حالیست که در مدل ارائه شده برای usb سه clock تعریف میشود.

- Sample Clock
- Bus Clock
- Service Clock

sample clock همان clock مشخص کننده نرخ جابجایی sample ها در بخش نرم افزاری است پس در این مورد تفاوتی در سیستم های usb و غیر usb وجود ندارد.

bus clock را می‌توان معادل bus clock ۸مگاهرتزی در حالت غیر usb دانست. با این تفاوت که در مدل usb فرکانس آن پایین تر از کلاک sample گیری در حالات غیر usb میباشد در حالیکه عموما فرکانس کلاک bus بیشتر است. ولی چون در ارائه پروتکل usb تلاش بر بود که هزینه ها کاهش پیدا کند و endpoint با حداقل پیچیدگی قابل پیاده سازی باشند و نیازی به سنکرون سازی کلاک device ها با کلاک host نباشد در این مدل نیز فرکانس پایینی برای bus clock در نظر گرفته شده است. در USB در حالت HighSpeed فرکانس برابر 8kHz و در حالت FullSpeed برابر 1kHz می‌باشد.

service clock مربوط به بخش پاسخگویی وقفه ها در بخش نرم افزاری است و در سیستم‌های usb و غیر usb تفاوتی ندارد.

برای همگام‌سازی در USB دستگاه‌های انتهایی به سه دسته تقسیم می‌شوند:

۱. ناهمگام: این دسته از دستگاه‌های انتهایی نمی‌توانند خود را با clock یا SoF همگام کنند، آن‌ها داده‌ها را به وسیله‌ی یک feedback که با توجه به اینکه فرستنده یا گیرنده هستند می‌تواند ضمنی یا صریح باشد.

۲. همگام: این دسته از دستگاه‌های انتهایی می‌توانند clock خود را با host همگام کنند و می‌بایست با استفاده از SoF از درستی کلاک خود اطمینان حاصل کنند. این دسته از دستگاه‌ها روی نرخ ثابتی که از ابتدا تعیین می‌کنند فعالیت خواهند کرد.

۳. انطباقی: این دسته از دستگاه‌های انتهایی تواناترین دستگاه‌ها هستند و می‌توانند به عنوانو گیرنده یا فرستنده با هر نرخی در بازه‌ی فعالیتشان کار کنند. این دسته از دستگاه‌ها نیز می‌بایست به وسیله‌ی یک feedback نرخ خود را اطلاع دهند و در صورت لزوم آن را کاهش یا افزایش دهند.

در مقایسه دو مدل ارائه شده میتوان گفت که در هردو بخش نرم افزاری از لحاظ فرکانس کاری DeviceDriverها تفاوتی نداشته است. در مدل usb، host controller به عنوان واسط نرم افزار و سخت افزار که به USB SW در بخش نرم افزاری متصل است و ارتباط بخش نرم افزاری سیستم به طور مستقیم با USB SW است. یکی از دلایل آن این است که عموما سیستم عامل نمیتوان تعداد زیادی ارتباط های isochronous را پشتیبانی کند چون ممکن است به ازای هر sample یک وقفه به device driver اعمال میشود. به همین دلیل ابتدا پردازش Sample ها در در بخش نرم افزاری صورت می‌گیرد و سپس به host controller داده میشود.
تفاوت چشمگیر این دو مدل وجود فیدبک در مدل usb است. همانطور که در توضیح Adaptive endpoints گفته شد برای مشخص کردن فرکانس کاری و گزارش دادن به host نیاز به فیدبک از سمت endpoint ها به سیستم میباشد. و برای یک device آسنکرون باید بصورت صریح این فیدبک به host فرستاده شود تا host بتواند بصورت مداوم با نرخ ارسال sample ها همگام شود. این امر از بروز overflow و underflow جلوگیری میکند یعنی حافظه های بافر هیچگاه خالی و بلا استفاده نمیمانند و سرریز هم نمیکنند.

سوال ۲

در استاندارد USB برای ارتباط‌های isochronous همگام‌سازی‌ها مستقل از سرعت clock و با استفاده از SoF صورت می‌گیرد و نرخ ارسال داده‌ها توسط endpointها مشخص می‌شود، بنابراین افزایش سرعت USB و افزایش فرکانس تفاوتی در پیاده‌سازی ایجاد نمی‌کند. از آنجایی که endpointها در اولین فاز اطلاعات تنظیمات خود را با host هماهنگ می‌کنند می‌توانند نرخ‌های ارسال و دریافت مختلفی داشته باشند و از بین آن‌ها نرخ مناسب را انتخاب کند. (منظور از نرخ مناسب نرخی است که دستگاه و host از آن پشتیبانی می‌کنند.)