



بسمه تعالی

**درس طراحی سیستم‌های نهفته مبتنی بر FPGA**  
**آزمایش ۴: طراحی یک شتاب‌دهنده سخت‌افزاری با رابط Avalon**  
پروفسور دانشکده‌های فنی دانشگاه تهران  
دانشکده مهندسی برق و کامپیوتر  
دکتر بیژن علیزاده

دستیاران آموزشی:

n.aghapour.s@gmail.com

farrokhi77parham@gmail.com

mahdib25@gmail.com

نگار آقاپور

پرهام فرخی

مهدی بحرینی

پاییز ۱۳۹۸

---

**مدت آزمایش: چهار جلسه**

**اهداف آزمایش:**

✓ آشنایی با رابط Avalon و مفاهیم پیشرفته در Nios II

✓ آشنایی با طراحی یک شتاب‌دهنده سخت‌افزاری

**مقدمه**

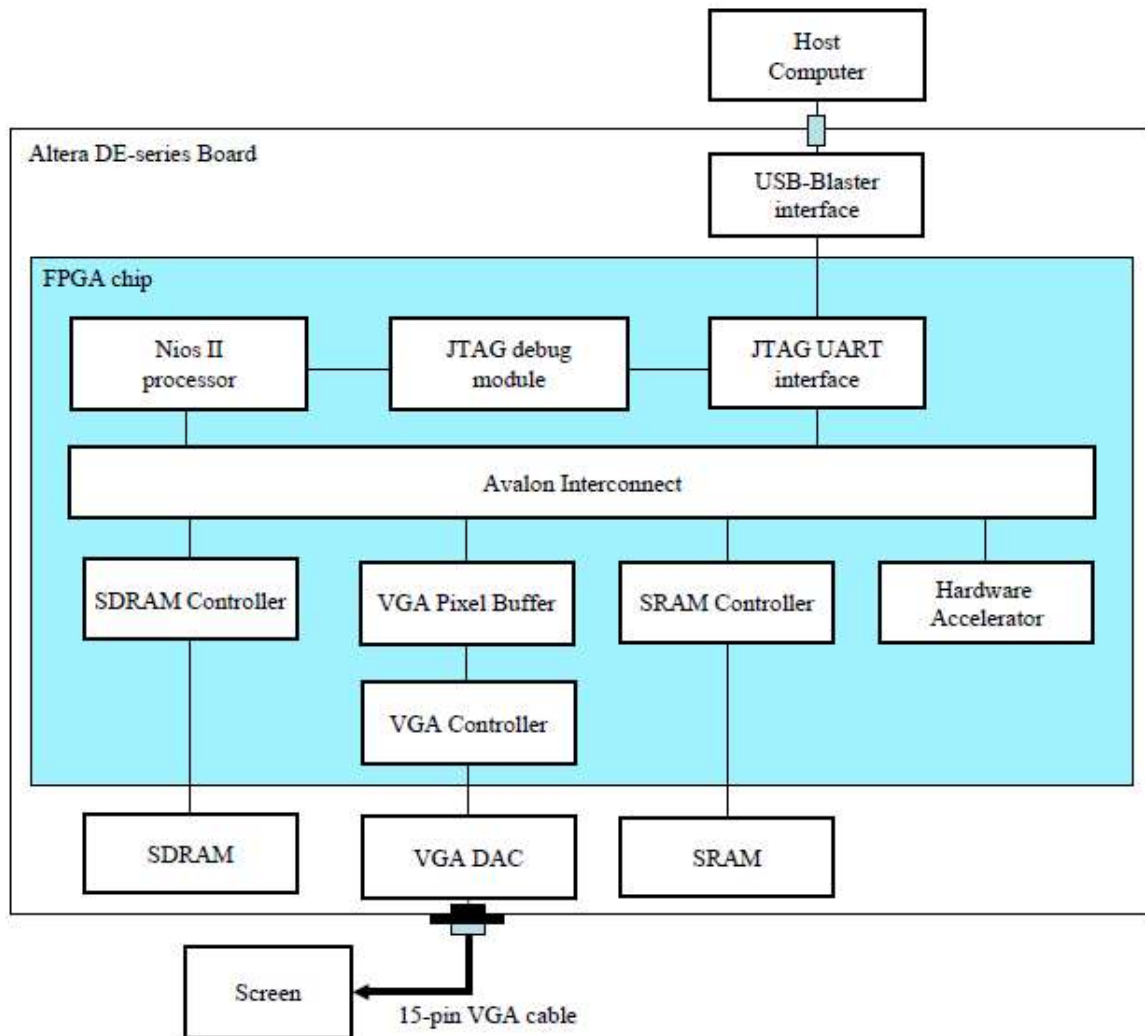
در این آزمایش با مفهوم شتاب‌دهی سخت‌افزاری<sup>۱</sup> آشنا خواهید شد. شتاب‌دهنده‌های سخت‌افزاری مدارهایی هستند که به منظور برداشتن وظایف خاصی از دوش پردازنده طراحی شده‌اند. برای مثال الگوریتم محاسبه‌ی FFT را در اندازه‌های بزرگ توسط نرم‌افزار نمی‌توان به صورت بی‌درنگ انجام داد. با سپردن این کار به سخت‌افزار، اولاً سخت‌افزار می‌تواند همان کار را در زمان کمتری انجام دهد (مشابه روند مشاهده شده با دستورات اختصاصی) و ثانیاً نرم‌افزار همزمان به انجام سایر وظایف پردازشی خود می‌پردازد (بر خلاف روند دستورات اختصاصی).

**شرح آزمایش**

شکل ۱ یک سیستم نمونه با شتاب‌دهنده سخت‌افزاری را نشان می‌دهد. تنها تفاوت این شکل با سیستم DE2 Media Computer در وجود واحد شتاب‌دهنده سخت‌افزاری است. این شتاب‌دهنده پس از دریافت دستور انجام عملیات مورد نظر، می‌تواند در صورت لزوم مستقیماً از طریق باس Avalon با سایر واحدهای سخت‌افزاری نیز در ارتباط باشد.

---

<sup>1</sup> Hardware Acceleration



شکل ۱ سیستمی با شتاب‌دهنده‌ی سخت‌افزاری

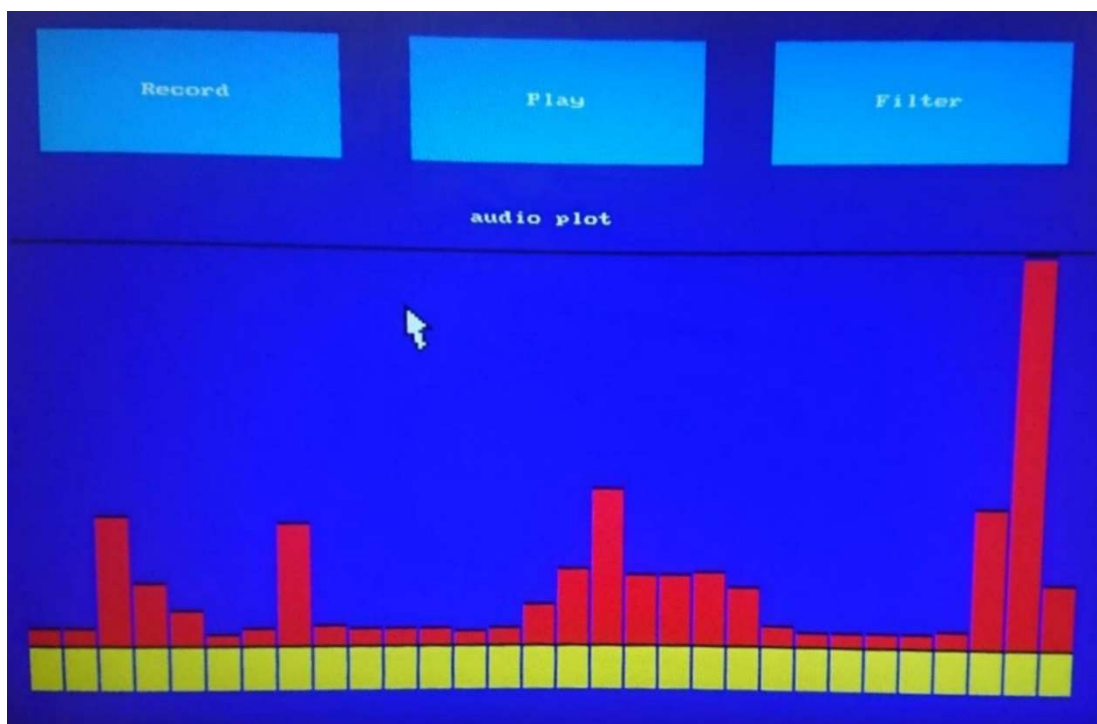
بخش‌های زیر را به ترتیب انجام دهید:

### ۱- طراحی سیستم مورد نظر به صورت نرم‌افزاری

در این بخش بر پایه‌ی سیستم DE2 Media Computer و نرم‌افزاری که در آزمایش دوم طراحی کرده‌اید، در Nios II Software Build Tools for Eclipse سیستمی پیاده کنید که:

- با کلیک روی دکمه‌ی Record، ۱۰ ثانیه از صدای ورودی ضبط شود. سپس بلافاصله پس از ضبط متوسط اندازه‌ی دامنه‌ی سیگنال در  $N$  بازه با زمان  $10/N$  ثانیه، به صورت مستطیل‌های کنار هم رسم شود (در شکل ۲ مستطیل‌های قرمز هستند). برای رسم این نمودار تابع Plot\_Audio را بنویسید و آن را پس از اتمام ضبط صدا فراخوانی کنید. کد شما باید با هر  $N$  دلخواهی کار کند (یعنی یا پارامتری باشد یا ورودی بگیرد).

- با کلیک روی دکمه‌ی Play، همراه با پخش صدای ضبط شده مستطیل‌هایی در پایین مستطیل‌های میانگین نمایش داده شود که نشانگر مکان فعلی پخش صدا باشد (در شکل ۲ مستطیل‌های زرد هستند).



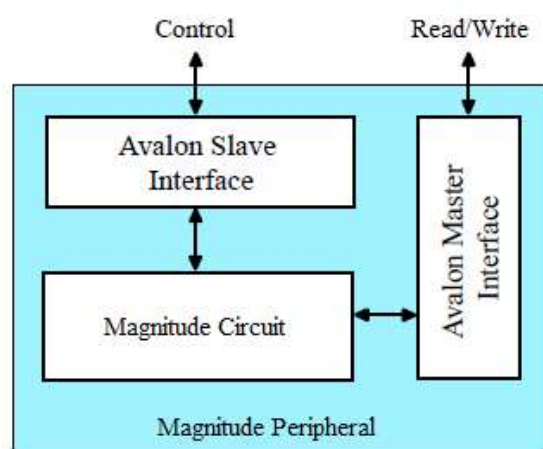
شکل ۲ نمونه‌ای از رابط گرافیکی سیستم مورد نظر

توجه نمایید که نمونه‌های ۲۴ بیتی صدا در بیت‌های MSB از آرایه‌های ۳۲ بیتی `l_buf` و `r_buf` ذخیره شده است. بنابراین برای جمع اندازه نمونه‌ها لازم است از متغیر با عرض بیت بالاتر (۶۴ بیتی در نرم‌افزار) استفاده کنید. زمان اجرای الگوریتم محاسبه‌ی متوسط را برای مقایسه‌های بعدی اندازه بگیرید و در گزارش ذکر کنید. توجه کنید که این کار حتما و حتما با استفاده از کتابخانه‌های مرتبط با زمان انجام شود. کارهایی مانند کرنومتر و ساعت گرفتن قابل قبول نبوده و نمره‌ای به آن‌ها تعلق نمی‌گیرد. همچنین حتما باید روند طی شده برای به دست آوردن زمان به طور کامل گزارش شود.

## ۲- طراحی شتاب‌دهنده سخت‌افزاری

در این بخش به طراحی سخت‌افزار شتاب‌دهنده و انجام عملیات بخش ۱ با استفاده از این سخت‌افزار خواهیم پرداخت. در شکل ۳ شمای کلی مدار جانبی سخت‌افزار محاسبه‌ی متوسط دامنه رسم شده است که می‌بایست به سیستم DE2 Media Computer اضافه شود. این شتاب‌دهنده شامل سه بخش است: مدار محاسبه‌ی دامنه، یک رابط Avalon Slave و یک رابط Avalon Master. پردازنده از طریق رابط Slave (با رابط Master خود)، آدرس حافظه شروع بافر صدای راست و چپ، آدرس مکان ذخیره‌سازی جواب، تعداد بازه‌ها (Num) و تعداد نقاط در هر بازه (Size) را در اختیار مدار محاسبه دامنه قرار می‌دهد. با نوشتن یک در

رجیستر Go مدار محاسبه دامنه، کار خود را شروع می‌کند. این مدار از طریق رابط Master خود، داده‌ها را از حافظه خوانده و پردازش می‌کند. پس از اتمام پردازش، نتیجه را در آدرسی از حافظه که توسط پردازنده مشخص شده است، ذخیره کرده و سیگنال Done را فعال می‌کند. پردازنده به صورت polling مقدار این سیگنال را بررسی کرده و به محض یک شدن آن مستطیل‌های بیانگر دامنه‌ی سیگنال را رسم می‌کند. همانطور که مشاهده می‌شود این روند مشابه عملیات DMA<sup>2</sup> است با این تفاوت که جهت سهولت کار از وقفه استفاده نشده است و به جای آن polling انجام می‌شود. برای درک بهتر ساختار در صورت نیاز به کدهای Verilog پیوست شده مراجعه کنید. توجه کنید که رعایت این ساختار اجباری است (رعایت نکردن این ساختار و یکپارچه‌کردن کد باعث کسر نمره خواهد شد). برای اطمینان از ساختار درست کد خود می‌توانید از کدهای پیوست شده استفاده کنید.



شکل ۳ شمای کلی مدار جانبی محاسبه‌ی متوسط دامنه

مراحل زیر را به ترتیب انجام دهید:

## ۲-۱ طراحی رابط Avalon Memory-Mapped Slave

همانطور که در شکل ۱ نشان داده شده است، در سیستم مبتنی بر Nios II، برای ارتباط میان پردازنده و سایر اجزا از گذرگاه Avalon استفاده می‌شود که در آن پردازنده‌ی Nios II نقش Master (پایه) را دارد. برای آنکه دستورات لازم توسط پردازنده به شتاب‌دهنده‌ی سخت‌افزاری ارسال شود، لازم است شتاب‌دهنده شامل یک رابط Slave (پیرو) باشد. در این بخش لازم است یک رابط Avalon Memory-Mapped Slave طراحی کنید که قابلیت خواندن از رجیسترهای شکل ۴ و نوشتن در آن‌ها را داشته باشد. برخی از این رجیسترها را در بخش بعد استفاده خواهیم کرد. این رابط و فایل testbench مربوطه را به کمک Modelsim (یا هر محیط طراحی سخت‌افزار دیگر) طراحی و درستی‌سنجی کنید.

<sup>2</sup> Direct Memory Access

Slave Address	31	30..12	11..1	0	
00	Done	Size	Num	Go	Config. Reg.
01					Right Addr.
10					Left Addr.
11					Out Addr.

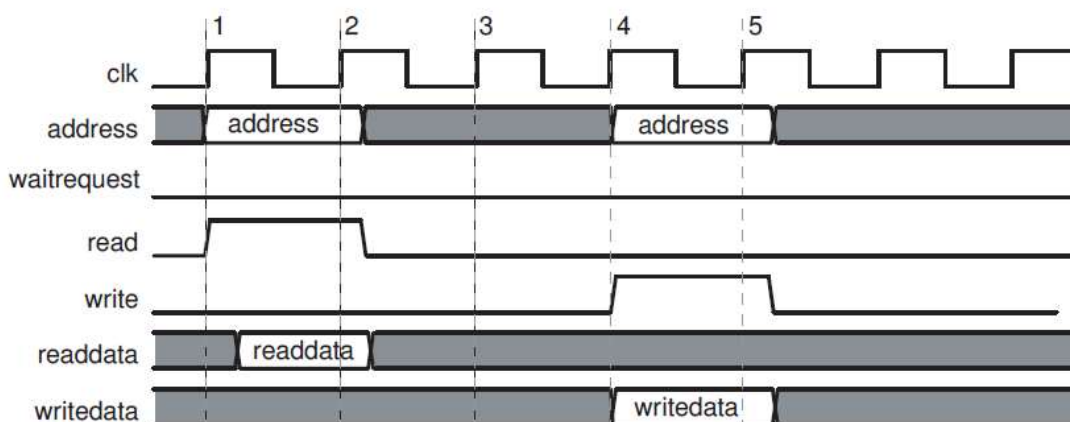
شکل ۴ رجیسترهای مورد استفاده در رابط Slave

به منظور طراحی و درستی‌سنجی رابط Slave لازم است با پروتکل مورد استفاده در گذرگاه Avalon آشنایی داشته باشید. گذرگاه Avalon یک رابط استاندارد و راحت برای اتصال واحدهای سخت‌افزاری درون FPGA است که انواع مختلفی مانند High Speed و Memory Mapped را شامل می‌شود. در این آزمایش از رابط Avalon Memory-Mapped استفاده خواهیم کرد. در این رابط عملیات نوشتن و خواندن، توسط سیگنال waitrequest که توسط Slave کنترل می‌شود، انجام می‌شود. برای شروع یک انتقال، Master سیگنال‌های address، byteenable، read/write و writedata را در هنگام پالس کلاک منتشر می‌کند. زمانی که یک Slave این سیگنال‌ها را دریافت می‌کند و chipselect آن فعال است، اطلاعات را ذخیره می‌کند یا اطلاعات درخواست شده را در خروجی (readdata) می‌گذارد. اگر Slave تا لبه‌ی کلاک بعدی قادر به پاسخ‌گویی نباشد، می‌تواند سیگنال waitrequest را قبل از لبه‌ی بالارونده‌ی کلاک منتشر کند. بدین ترتیب انتقال اطلاعات به تأخیر می‌افتد و Master، آدرس و سیگنال‌های کنترلی را ثابت نگه می‌دارد.

هنگام پیاده‌سازی رابط Slave جهت سهولت تشخیص واسط‌ها توسط Qsys، اسامی تمامی سیگنال‌های آن را با پیشوند avs\_avalonslave\_ مانند avs\_avalonslave\_read انتخاب کنید. در این قسمت پروژه، رابط Slave شما نیازی به انتشار waitrequest ندارد زیرا فرض بر این است که می‌تواند در یک کلاک یا تعداد کلاک ثابتی پاسخ دهد (و باید این کار را انجام دهد). شکل ۵ زمان‌بندی مورد نیاز در گذرگاه Avalon را (با عملکرد بدون waitrequest و با Wait Cycle=0) نشان می‌دهد. دقت کنید سیگنال chipselect که در طول انتقال خواندن/نوشتن فعال می‌شود، در این شکل نشان داده نشده است. همچنین بر خلاف این شکل، طرح شما می‌تواند Wait Cycle غیر صفر برای خواندن یا نوشتن داشته باشد. روند شکل ۵ بدین صورت است:

- ۱) Master سیگنال‌های read و address را روی لبه‌ی بالارونده‌ی کلاک منتشر می‌کند. در همان سیکل، Slave سیگنال‌های Master را decode می‌کند و readdata را ارائه می‌کند.
- ۲) Master، سیگنال readdata را روی لبه‌ی بالارونده‌ی کلاک می‌خواند و آدرس و سیگنال‌های کنترلی را غیر فعال می‌کند. در این لحظه انتقال پایان می‌یابد.
- ۳) هیچ سیگنال کنترلی منتشر نمی‌شود.
- ۴) Master، سیگنال‌های address، write و writedata را روی لبه‌ی بالارونده‌ی کلاک منتشر می‌کند. سیگنال‌های Master ثابت باقی مانده و Slave آن‌ها را decode می‌کند.
- ۵) Slave، سیگنال writedata را روی لبه‌ی بالارونده‌ی کلاک می‌گیرد. Master سیگنال‌های writedata، address و write را غیر فعال می‌کند و انتقال پایان می‌یابد.

برای آشنایی بیشتر با رابط Avalon به اسلایدهای درس و در صورت نیاز به فصل ۳ از مرجع [۱] مراجعه کنید.



شکل ۵ زمانبندی Avalon برای انتقال خواندن و نوشتن بدون سیگنال waitrequest

قسمت زیر خواسته‌ی اصلی نیست و در صورتی که به مشکلی خوردید که نتوانستید آن را رفع کنید به سراغ این قسمت بروید:

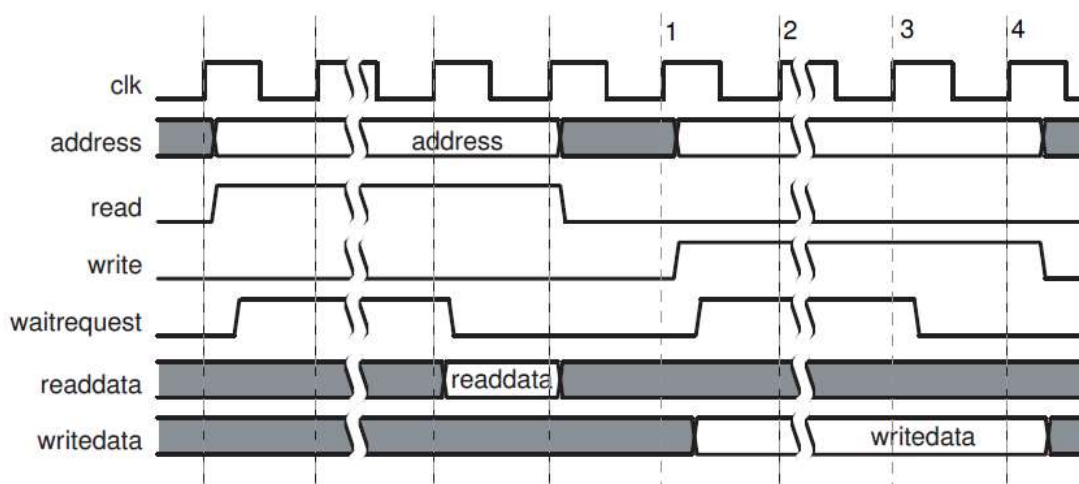
برای آنکه رابط Slave خود را بر روی FPGA تست کنید، خروجی LEDR[17:0] و ورودی SW[1:0] را به رابط Avalon Slave خود اضافه کنید به طوری که بیت‌های ۰ تا ۱۷ رجیسترها به LEDR وصل شوند (با استفاده از یک مالتی‌پلکسر که با سوئیچ‌های SW[1:0] کنترل می‌شود). این سخت‌افزار را در محیط Qsys با کلیک بر روی New Component... (از منوی فایل یا برگه Component Library) به سیستم DE2 Media Computer اضافه کنید (این روند کاملاً مشابه روند اضافه کردن Custom Instruction است با این تفاوت که اینترفیس Avalon Slave استفاده می‌شود و SW و LEDR به صورت Conduit تعریف می‌شوند). اینترفیس Avalon Slave سخت‌افزار خود را به باس data از Nios II متصل کنید. توجه کنید که در Qsys باید PIOهای مربوط به LEDهای قرمز و سوئیچ‌ها را غیر فعال کنید (تیک کنار آن‌ها را بردارید) و در مازول خود پورت‌های مربوطه را export کنید. فایل DE2\_Media\_Computer.v را در پروژه Quartus متناظر با تغییرات ایجاد شده در خروجی Qsys (پورت‌های فایل nios\_system.v تغییر یافته است) تغییر دهید. پروژه را کامپایل کرده و FPGA را پروگرام نمایید. نرم‌افزاری بنویسید که در رجیسترهای مذکور مقداری را بنویسد و آن را خوانده و با مقدار نوشته شده مقایسه کند. همچنین مطمئن شوید مقدار ۱۸ بیت پایین رجیسترها به درستی توسط LEDها نشان داده می‌شود.

## ۲-۲ طراحی رابط Avalon Memory-Mapped Master

در این بخش رابط Master را برای ارتباط با کنترلر SDRAM پیاده‌سازی خواهیم کرد. مدار بخش قبل را به گونه‌ای تغییر دهید که با نوشتن در رجیستر Go (توسط پردازنده) به تعداد مشخص شده (Num) هر دفعه از آدرس راست و چپ به اندازه مشخص شده (Size) بخواند و جمع کند و نتیجه را در آدرس خروجی

بنویسد. پس از این کار سیگنال Done فعال می‌شود. پس از ساخت و انتقال سیستم به FPGA، با نوشتن نرم‌افزار مناسب، صحت عملکرد آن را تست کنید.

در طراحی رابط Master در نظر داشته باشید که سیگنال waitrequest می‌تواند توسط کنترلر SDRAM منتشر شود. در این حالت رابط Master باید تمامی سیگنال‌های کنترلی خود را ثابت نگه دارد تا زمانی که waitrequest غیر فعال گردد. مشابه بخش قبل، اسامی سیگنال‌های مربوط به رابط Master را با پیشوند avm\_avalonmaster\_[signal] وارد نمایید. شکل ۶ زمانبندی Avalon را برای انتقال خواندن و نوشتن به همراه سیگنال waitrequest نشان می‌دهد.



شکل ۶ زمانبندی Avalon برای انتقال خواندن و نوشتن به همراه سیگنال waitrequest

## ۲-۳ طراحی مدار محاسبه دامنه

مطابق توضیحات بخش‌های قبل، مداری طراحی کنید که با دریافت آدرس بافر راست و چپ صدا، آدرس قرارگیری نتیجه‌ی عملیات، تعداد بازه‌های محاسبه‌ی دامنه و تعداد نقاط هر بازه، پس از نوشتن یک رجیستر Go شروع به کار کرده، مجموع قدر مطلق داده‌ها را در هر بازه محاسبه کرده و در مکان مربوطه در آدرس نتایج بنویسد. در انتهای کار سیگنال Done، یک می‌شود. عملکرد این سیستم باید دقیقاً مشابه بخش ۱ (بخش طراحی نرم‌افزاری) باشد. با انتقال سیستم طراحی شده به FPGA، سرعت عملکرد سخت‌افزار خود را نسبت به نرم‌افزار بسنجید و زمان اجرا را در هر دو حالت گزارش کنید. توجه نمایید که عملکرد و اعداد بدست آمده توسط نرم‌افزار و شتاب‌دهنده دقیقاً باید یکسان باشند. همچنین برای ارتباط با سخت‌افزار نوشتن مستقیم در رجیسترها صحیح نبوده و منجر به کسر نمره خواهد شد.

ابتدا HAL مورد نیاز برای ارتباط با سخت‌افزار خود را بسازید و سپس با استفاده از توابع آن، کد اصلی را بنویسید. تابعی که در نهایت باید در کد اصلی استفاده شود در پیوست با نام amplitude\_calculation.c قرار داده شده است. کتابخانه خواسته شده را در یک فایل با نام amplitude\_circute.h به صورت جداگانه بنویسید و به فایل‌های پروژه اضافه کنید (خواسته‌ی آزمایش).

**بخش امتیازی (%۲۰):** رابط Master را به گونه‌ای طراحی کنید که امکان Burst داشته باشد. از Burst‌های حداکثر ۱۶ تایی برای انتقال داده استفاده کنید. فاصله بین گرفتن سیگنال Go تا صدور سیگنال Done را در هر دو حالت با و بدون Burst با اضافه کردن یک شمارنده اندازه گرفته و باهم مقایسه نمایید.

### نکات مهم:

(۱) قسمت‌هایی که باید در گزارش ذکر شود، عبارتند از:

- شکل صفحه‌ی نهایی
  - کد تابع قسمت نرم‌افزاری به صورت پیوست جداگانه
  - کد تابع قسمت سخت‌افزاری به صورت پیوست جداگانه
  - زمان اجرای کد نرم‌افزاری به همراه روش به دست آوردن آن
  - زمان اجرای کد سخت‌افزاری به همراه روش به دست آوردن آن
  - کدهایی که به عنوان HAL برای ارتباط با سخت‌افزار نوشته شده‌اند به صورت پیوست جداگانه
  - کدهای سخت‌افزاری نوشته شده
  - تصاویری از نتایج شبیه‌سازی کدهای سخت‌افزاری (یا از طریق نرم‌افزارهای جداگانه مانند ModelSim یا از طریق signal tab خود نرم‌افزار Quartus)
- (۲) پیروی از قالب خاصی در گزارش مد نظر نیست، اما ترجیحاً می‌توانید از قالب ارائه شده برای تکالیف کامپیوتری استفاده نمایید.

### مراجع

[1] Altera, "Avalon Interface Specification", Chapter 3: Avalon Memory-Mapped Interfaces, May 2007. Available: [http://www.altera.com/literature/manual/mnl\\_avalon\\_spec\\_1\\_3.pdf](http://www.altera.com/literature/manual/mnl_avalon_spec_1_3.pdf)

موفق باشید

۹۸/۹/۲۰