

به نام خدا

پروژه پایانی درس معماری کامپیوتر

دانشگاه شهید بهشتی

ترم پاییز سال تحصیلی ۹۸-۹۹

توجه: این پروژه شامل بخش ۱ و بخش ۲ می باشد. بخش ۱ اجباری است و همه دانشجویان باید به صورت یکنفره انجام دهند. بخش دوم اختیاری است و دانشجویانی که نمره اضافی بخواهند می توانند به صورت یکنفره بخش ۲ را انجام دهند.

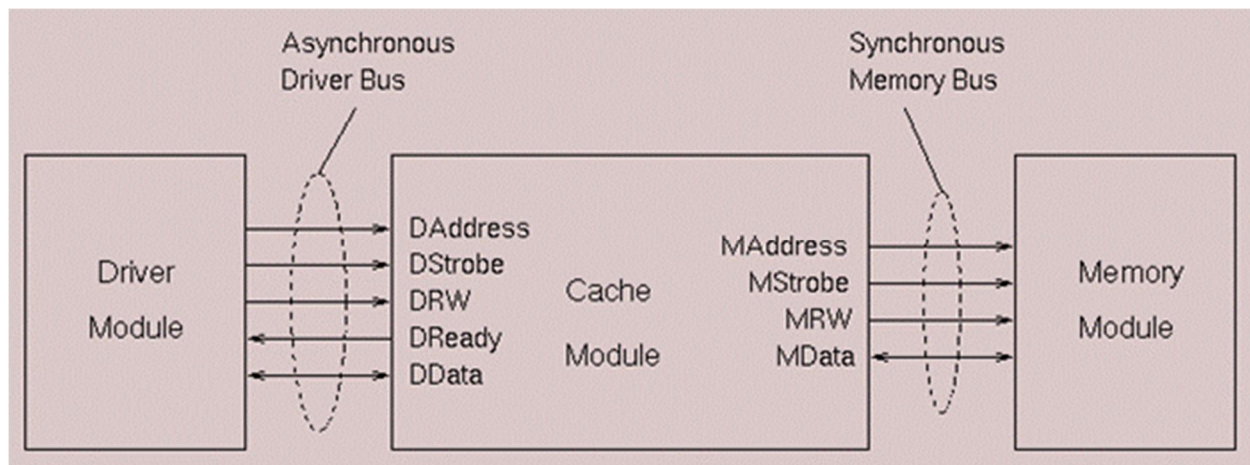
مهلت ارسال فایل ها برای بخش ۱ (اجباری) و بخش ۲ (اختیاری) روز دوشنبه ۹۸۱۱۰۷ ساعت ۲۳:۵۹ دقیقه می باشد. همچنین روز تحویل حضوری پروژه که در آن ضریب تسلط نیز سنجیده می شود روز سه شنبه ۹۸۱۱۰۸ می باشد. با توجه به آنکه روز چهارشنبه تعطیل رسمی است، و نمره های شما باید نهایی شود امکان تمدید وجود نخواهد داشت. لطفاً اگر سوالی دارید با TA های خود مطرح نمایید.

هدف:

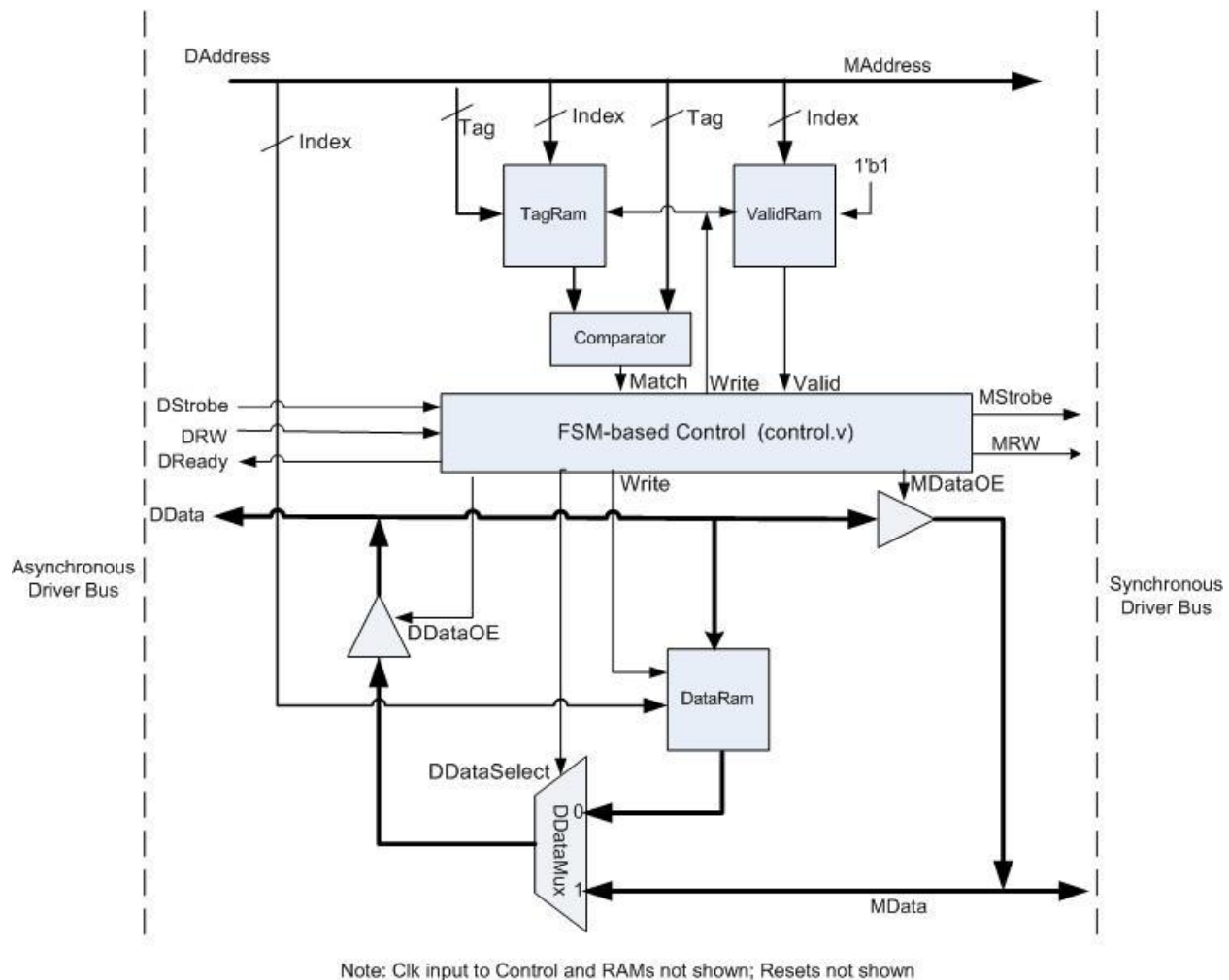
این پروژه در نظر دارد درک شما از ساختار حافظه نهان (cache) را افزایش دهد. بدین منظور مدلی اولیه از حافظه نهان که به صورت direct map cache است به همراه کدهای Verilog آن به شما تحویل می شود. شما در درجه اول باید این کدها را متوجه شوید و سپس تغییرات لازم را در آن اعمال نمایید.

۱- بخش اول پروژه: (اجباری):

کد direct map cache با روش write-through در فایل Cache-proj-part1.zip در اختیار شما قرار داده می شود. بلوک دیاگرام این ساختار حافظه در شکل زیر دیده می شود. توجه کنید که driver عنصری مانند یک پردازنده ساده است که از ساختار حافظه ما (که شامل cache و main memory) است استفاده می کند و دستورات خواندن و نوشتن برای آن صادر می کند. همانطور که در بخش های مختلف درس نیز عنوان شد از دید driver (و یا پردازنده) کل ساختار حافظه ما (شامل cache و main memory) به عنوان یک حافظه نسبتاً سریع با ساختار main memory دیده می شود و دلیل اطلاق اصطلاح حافظه نهان و یا cache نیز همین موضوع می باشد.



ساختار داخلی دقیق مازول حافظه نهان مطابق شکل زیر است. این ساختار و کد Verilog فوق را مطالعه نمایید و دقیق کنید و مطمئن شوید که آنرا به طور کامل درک نموده اید قبل از آنکه جلوتر بروید. سائز حافظه نهان به صورت پارامتری در فایل های header تعریف شده است و می توانید آنرا تغییر دهید. آدرس های حافظه به صورت byte addressable می باشد و لی همه ی دسترسی ها به حافظه به صورت یک کلمه (word) چهار بایتی انجام می شود. همچنین سائز حافظه نهان در cache.h به صورت پارامتری و نشان دهنده ی تعداد word بیان شده است. توجه کنید که به همراه کد Verilog دو فایل بزرگ که هر کدام حدود یک میلیون دسترسی به حافظه دارد را شامل می شود. این دسترسی ها از اجرای برنامه های واقعی استخراج شده است. در واقع فایل large_cc.trc شامل دسترسی ها به حافظه برای بخشی از یک کامپایلر C++ می باشد. همچنین فایل large_spice.trc شامل بخشی از دسترسی های برنامه spice برای شبیه سازی کارکرد مدارها می باشد. دقت کنید که نحوه انتخاب این فایل های trace از درون فایل trace.h انجام می شود.



خروج هایی که در انجام بخش اول باید تحویل دهید:

الف) مدل direct map cache فوق را برای هر کدام از این trace file ها و برای cache سائیزهای زیر شبیه سازی نمایید:

1K (1024) words, 2K (2048) words, 4K (4096) words, 8K (8192) words, 16K (16,384) words, and 32K (32,768) words

بنا براین شبیه سازها گراف های زیر را توسط نرم افزار متلب و یا اکسل و یا هر نرم افزار دیگری رسم و به عنوان خروجی بخش اول پروژه تحویل نمایید

1. Overall cache miss rate versus cache size. Show curves for both traces on the same graph.
2. Read miss rate versus cache size. Show curves for both traces on the same graph.
3. Write miss rate versus cache size. Show curves for both traces on the same graph.
4. Memory read time versus cache size. Show curves for both traces on the same graph.
5. Memory write time versus cache size. Show curves for both traces on the same graph.

ب) گزارشی دو صفحه ای که در آن روند انجام کار را توضیح داده باشید، همچنین چگونه و با چه ابزاری کد را شبیه سازی کردید و گراف ها را با چه ابزاری رسم نموده اید را توضیح دهید. همچنین تحلیل ۲ خطی برای هر کدام از نمودارهای ۱ الی ۵ بالا ارایه نمایید که روند تغییرات را توجیه نماید.

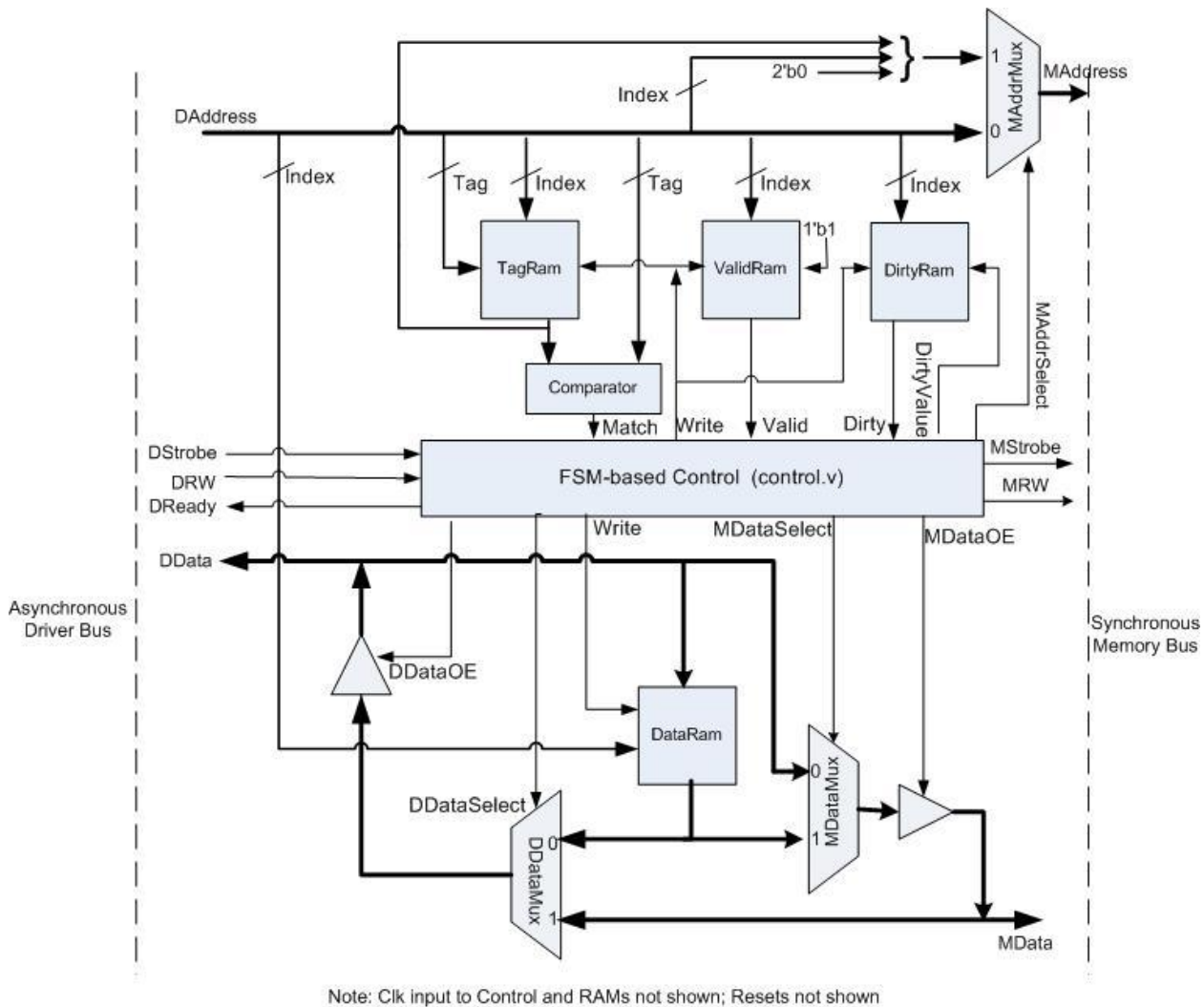
۲- بخش دوم پروژه (اختیاری):

در بخش دوم پروژه هدف آن است که write policy در بخش اول که write-through بود را به write-back تبدیل نمایید.

توجه کنید که در این بخش نیز یک کد تقریبا کامل شده در فایل Cache-proj-part2.zip گنجانده شده است. ساختار حافظه نهان تغییر یافته در شکل زیر دیده می شود: توجه کنید که تنها بخشی از کد که کامل نشده است اختصاص سیگنال های کنترلی در بخش task UpdataSignal در فایل Control.v می باشد. بنابراین کاری که در این بخش باید انجام دهید تکمیل این بخش از کد Verilog در فایل Control.v برای کارکرد صحیح ماژول cache می باشد. لازم نیست ماژول cache.v و یا دیگر ماژول ها را تغییر دهید.

توجه کنید برای آنکه راحتتر بتوانید کد تغییر یافته خودتان را دیباگ نمایید یک فایل trace کوتاه ۱۲ تایی به نام short_test.trc در فایلها گنجانده شده است. همچنین در فایل dbgflags.h پارامترهایی در نظر گرفته شده که با فعال کردن آنها اطلاعات بیشتری حین شبیه سازی می توانید ببینید. دقت کنید که این برای آن است که بتوانید بهتر و راحتتر دیباگ کنید. توجه نمایید هنگام استفاده از trace file های طولانی این پارامترها فعال نباشند، چون باعث می شوند شبیه سازی به شدت کند شود. این پارامترها فقط برای دیباگ خوب است استفاده شود.

همچنین برای آنکه تا حدی مطمئن شوید Cache جدید شما به درستی کار کرده است می توانید خانه هایی از حافظه که در بخش اول پروژه برای trace file های بزرگ خوانده شده بود را با همین خانه هایی که در این بخش برای همین فایل ها از حافظه خوانده می شود را مقایسه نمایید و باید شبیه به هم باشند. البته درست بودن این مقایسه شرط کافی نیست ولی شرط لازم است برای آنکه Cache شما درست کار کرده باشد!



خروج هایی که در انجام بخش دوم باید تحویل دهید:

الف) کد تغییر یافته *control.v*

ب) گراف های زیر:

1. Read miss rate of the modified cache versus cache size, using the same range of cache sizes as in part 1. Show curves for both traces on the same graph.
2. Write miss rate of the modified cache versus cache size, using the same range of cache sizes as in part 1. Show curves for both traces on the same graph.
3. Memory write time of the modified cache and of the original cache versus cache size, using the same range of cache sizes as in part 1. Show curves for each trace/policy on the same graph, four curves total.

4. Total time (memory read time plus memory write time) for the modified cache and the original cache versus cache size, using the same range of cache sizes as in part 1. Show curves for each trace/policy on the same graph, four curves total.

پ) گزارشی که در آن تحلیل ۲ خطی برای هر کدام از نمودارهای ۱ الی ۴ بالا ارایه نمایید که روند تغییرات را توجیه نماید. همچنین یک تحلیل یک الی دو صفحه ای که تغییرات و تفاوت‌های کارایی در دو نوع cache بخش ۱ و بخش ۲ را توضیح و توجیه نماید.

موفق باشید