



پروژه درس ساختار کامپیوتر و میکروپروسسور

دکتر موحیدیان

فاز چهارم

طراحی Cache

تابستان 99

در قسمت های قبل دیدیم که تاخیر در خواندن از حافظه دیتا می تواند سرعت پردازنده را بسیار پایین بیاورد. با طراحی پردازنده با معماری پایپ لاین دیدیم که در بسیاری از برنامه هایی که با تعداد زیادی خواندن و نوشتن حافظه همراهند، سرعت حافظه به نوعی bottle neck سرعت کل اجرا می شود. یک راه حل این مشکل طراحی حافظه ی Cache است. در این تمرین هدف ، طراحی یک حافظه Cache می باشد که به حافظه RAM و Processor شبیه سازی شده ای که در اختیار شما قرار داده شده است وصل شود.

واحد Top شما باید شامل دو واحد Processor و Memory System باشد . واحد Memory System ، clock و reset را می گیرد و Hit و خروجی مموری را در هر لحظه نمایش می دهد. در ادامه در مورد Hit و خروجی مموری توضیح می دهیم.

نحوه عملکرد هر کدام از واحد ها در قسمت های بعد توضیح داده شده.

## • Processor :

در این تمرین این واحد را در اختیار شما قرار داده ایم . این واحد از الگوریتمی برای تولید اعداد استفاده میکند. خروجی این ماژول شبیه ساز درخواست های یک پردازنده برای خواندن و نوشتن حافظه است.

کد این واحد به صورت زیر می باشد:

```
module Processor(  
    clk,  
    start,  
    RWB,  
    Address,  
    Data  
);  
  
input clk;  
input start;  
output RWB;  
output reg[5:0] Address;  
output reg[7:0] Data;  
reg Addreg;  
always@(posedge clk)  
begin  
    begin  
        Data[0] <= start ? 1'b0 : Data[7]^ Data[6];  
        Data[6:1] <= start ? 5'b0: Data[5:0];  
        Data[7] <= start ? 1'b1 : Data[6];  
        Address[0] <= start ? 1'b0 : Address[5]^ Addreg;  
        {Addreg,Address[3:1]} <= start ? 4'b0: Address[3:0];  
        Address[4]<=0;  
        Address[5] <= start ? 1'b1 : Addreg;  
    end  
end  
  
assign RWB = Data[5];  
  
endmodule
```

برای راه اندازی این واحد نیاز است که سیگنال start در لبه ی بالارونده کلاک اول 1 شود و در باقی زمان همواره صفر باقی بماند.

## • Memory System :

این واحد شامل دو واحد Cache و Ram می باشد که هر کدام به صورت زیر می باشند:

### ▪ Ram :

کد این واحد در اختیار شما قرار گرفته است. مشخصات این رم به صورت زیر است:

1. Ram ما شامل 64 داده 8 بیتی می باشد.
2. برای خواندن در صورتی که ورودی readEn برابر یک باشد ، داده هایی که با آدرس readAddress مشخص می شود ، بر روی خروجی ReadData قرار می گیرد . در غیر این صورت خروجی به صورت High Impedance می باشد.
3. برای نوشتن نیز در لبه مثبت Clock در صورتی که ورودی writeEn یک باشد ، مقدار ورودی WriteData در خط حافظه ای که با آدرس writeAddress مشخص شده است، ذخیره می شود.
4. در لبه مثبت کلاک در صورت یک بودن ورودی Reset ، تمام اطلاعات موجود در حافظه صفر می شود.

### ▪ Cache :

هدف اصلی این پروژه طراحی این قسمت از System Memory می باشد. این واحد شامل مشخصات زیر است:

1. در این قسمت باید یک واحد حافظه نهان از نوع 2-way-set-associative پیاده سازی کنید.
2. ظرفیت این حافظه نهان را 16 بایت و اندازه هر بلوک را 1 بایت در نظر بگیرید.
3. در این واحد سیاست نوشتن write-back و سیاست جایگزینی LRU است.
4. کد وریلاگ شما باید خروجی نشان دهنده ی hit یا miss شدن را داشته باشد.
5. در لبه ی بالا رونده کلاک در صورت 1 بودن Reset تمام بلاک های cache ریست می شوند.

Way 1			Way 0			
V	Tag	Data	V	Tag	Data	
0			0			Set 3
0			0			Set 2
1	00...00	mem[0x00...24]	1	00...10	mem[0x00...04]	Set 1
0			0			Set 0

Two-way-set-associative cache

#### ▪ نحوه کار این دو واحد:

ابتدا آدرس و دیتایی که پروسسور تولید کرده بود به داخل cache برده می شود. در صورتی که این داده hit شده بود، Cache آپدیت می شود. در غیر این صورت در صورتی که عملیات read باشد داده را از مموری می خوانیم و هم به خروجی کلی مموری و هم به cache می دهیم. در صورتی که عملیات write باشد هم داده را در cache می نویسیم و با توجه به الگوریتم write-back ادامه عملیات ها را انجام می دهیم. خروجی مموری در واقع خروجی واحد MemorySystem در حالت Read و Hit نشانگر hit شدن یا miss شدن درخواست می باشد.

## خواسته های پروژه:

- 1- ابتدا 2 واحد System Memory و Processor را به همدیگر وصل کنید . این پروسسور تعدادی read و write رندوم درخواست می کند. تعداد hit های این cache را در مدت زمان 1000 نانو ثانیه با دوره ی کلاک 10 نانو ثانیه در گزارش قرار بدهید.
- 2- ساختار cache پیاده سازی شده را توضیح دهید. همچنین در مورد روش هایی که برای پیاده سازی write-back و LRU به کار گرفتید نیز توضیح مختصری دهید.
- 3- با توجه به الگوریتم تولید داده های تصادفی، روشی پیشنهاد دهید (با تغییر ساختار cache در عین ثابت بودن سایز) که بتوان تعداد hit ها را افزایش داد.

## نکات مهم پروژه:

- 1- برای تست مدار لازم است که ورودی Start در ابتدا برای 1 کلاک برابر 1 شود.
- 2- کد های واحد های Ram و Processor را تغییر ندهید.
- 3- در صورتی که بخواهید می توانید TopTB را تغییر دهید.
- 4- نمره این فاز امتیازی است و تحویل حضوری ندارد. اما طراحی شما حتما باید تست بنچ های داده شده را درست اجرا کند.
- 5- کل طراحی های خود را به همراه گزارش مختصری از ایده های پیاده سازی شده در قالب فایلی با اسم cache\_#Student.rar در CW آپلود کنید.