به‌نام خدا
تمرین سری دوم
طراحی FPU - معماری کامپیوتر
استاد : دکتر محمد رضا پورفرد
تدریس یار : جواد سبحانی
دانشگاه صنعتی امیرکبیر، دانشکده مهندسی برق
پاییز ۱۴۰۴

۱ معرفی پروژه

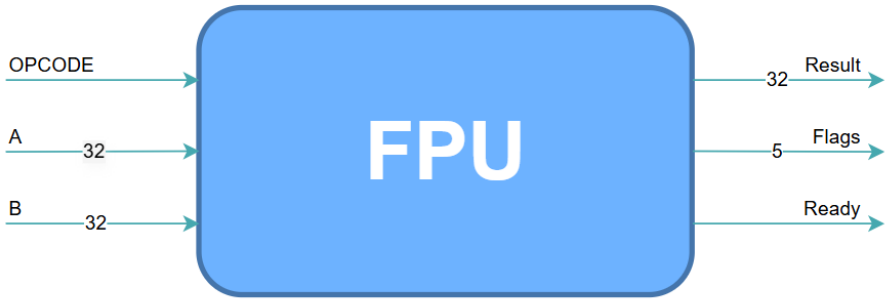
در این پروژه، هدف ما طراحی یک واحد ممیز شناور (FPU) ساده بر اساس معماری $RISC-V$ است. این ماژول به زبان $VHDL$ برنامه‌نویسی و با استفاده از $ModelSim$ شبیه‌سازی خواهد شد.

ابتدا در مورد ماژول و پارامترهای آن بحث خواهیم کرد تا اهداف طراحی که برای آن طراحی خواهید کرد، بیشتر روشن شود. سپس ساختار طراحی مورد بحث قرار می‌گیرد و یک مرور کلی از سخت‌افزار ارائه می‌شود. در نهایت، شرایط تست و مراحل شبیه‌سازی بررسی می‌شوند.

۲ ماژول FPU

۲-۱ توصیف سخت افزار

برای اینکه بتوانیم سیستم را طراحی کنیم، ابتدا باید با معماری ماژول آشنا شویم. شکل زیر بلوک دیاگرام سیستم را نشان میدهد.



شکل ۱: بلوک دیگرام سیستم

$Operation$	$Operand2$	$Operand1$	$OPCODE$
$A+B$	B	A	۰
$A-B$	B	A	۱

جدول ۱: مقادیر $OPCODE$

بر اساس شکل ۱ این سیستم دارای سه ورودی مهم می باشد:

- $OPCODE$: تک بیت برای انتخاب بیت جمع یا تفریق.
- A: ورودی ۳۲ بیت اول.
- B: ورودی ۳۲ بیت دوم.

با داشتن $OPCODE$ میتوان عملیات جمع یا تفریق را انجام داد. پس از اتمام محاسبه، نتیجع به همراه $FLAG$ ها در خروجی نمایش داده میشود. همچنین پین $OutRdy$ نیز برای یک سیکل کلاک یک شود.
جدول زیر مود های عملکرد سیستم را توصیف میکند.

$Flags$ ۲-۲

در صورت وقوع برخی اتفاقات مانند سرریز، زیرریز یا نتیجه صفر، بیت‌های خاصی در $FLAG$ تنظیم می‌شوند تا سیستم این اتفاقات را نشان دهند. جدول زیر جزئیات طراحی $FLAG$ را نشان می‌دهد:

بیت	$Flag$	$Symbol$	توضیحات	مثال
۰	$Inexact$	I	نتیجه رند شده است	$1e20 + 3.14$
۱	$Underflow$	X	حاصل کوچکتر از مینیمم قابل نمایش	$0x00000001 \; - \; 0$
۲	$Division\ by\ Zero$	Z	عدد متناهی $\div\ 0$	$5.00.0$
۳	$Overflow$	V	حاصل بزرگتر از مینیمم قایل نمایش	$1e38 \times 1e38$
۴	$Invalid\ Operation$	N	خطای محاسبه نامعتبر	$0x7FC00000 \; + \; 0$

جدول ۲: جزئیات $FLAG$ ها

لازم به ذکر است که $FLAG$ های بالا به نام $Exception\ Flag$ معروف هستند و دسته دیگری از $Flag$ ها نیز به نام $Status\ Flag$ ها وجود دارند که در این پروژه پیاده سازی نشده اند.
علاوه بر این، از جدول بالا، تقسیم بر صفر در اینجا استفاده نشده است، اما به دلیل سازگاری طراحی ذکر شده است. بنابراین در کد، جای آن (۰) باقی می‌ماند و هرگز نباید تنظیم شود.

۳-۲ ماژول $VHDL$

در زیر تعریف ماژول FPU در $VHDL$ آمده است.

```
entity FPU is
port(
    clk : in std_logic;
    rst : in std_logic;
    En  : in std_logic;
    -- Data & opcode inputs
    opcode : in std_logic;
    A      : in std_logic_vector(31 downto 0);
    B      : in std_logic_vector(31 downto 0);
    -- outputs
    Result : out std_logic_vector(31 downto 0);
    Flags  : out std_logic_vector(4 downto 0);
    OutRdy : out std_logic;
);
end entity FPU;
```

شکل ۲: تعریف ماژول

در نمودار قبلی، سیگنال‌های ساعت، تنظیم مجدد و فعال‌سازی را نشان ندادیم، اما آنها برای عملکرد صحیح دستگاه ضروری هستند.

سیگنال ریست فقط "$Result$"، "$Flags$" و "$OutRdy$" را ریست(۰) می‌کند و نیازی به ریست کردن کل رجیسترهای داخلی نیست.

علاوه بر این، برای انجام هر عملیاتی، پین "En" باید به همراه سایر ورودی‌ها تنظیم (۱) شود.

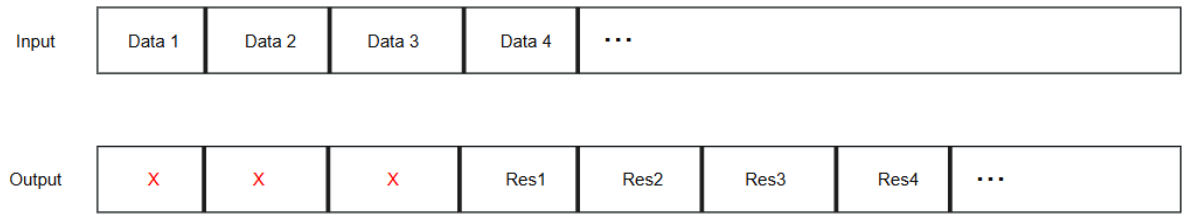
۴-۲ *Flow of Operation*

برای استفاده از ماژول جهت انجام جمع یا تفریق، مراحل زیر باید انجام شود: (با فرض وجود سیگنال کلاک)

۱. برای شروع عملیات، پین‌های $data$ و $Enable$ باید تنظیم شوند.

۲. پس از محاسبه عملیات، نتیجه در خروجی نمایش داده می‌شود و پین $OutRdy$ یک میشود. (توجه: پین $OutRdy$ باید تا زمانی که داده‌های معتبر وجود ندارند، صفر بماند.)

شایان ذکر است که پیاده‌سازی باید به صورت پایپ‌لاین انجام شود. به این معنی که داده‌های ورودی می‌توانند یکی پس از دیگری به ماژول وارد شوند و پس از یک تأخیر مشخص، داده‌های معتبر یکی پس از دیگری در خروجی ظاهر می‌شوند و پین $OutRdy$ تا زمانی که داده‌های معتبر وجود دارند، یک $(high)$ باقی می‌ماند.



شکل ۳: پایپ لاین

در زیر مثالی از سیستمی با تأخیر ۳ سیکل کلاک، اما پیاده‌سازی شده به روش پایپ‌لاین، آورده شده است.

۵-۲ *Test & Simulations*

برای بررسی اعتبار ماژول، سیستم باید تحت شرایط مختلف آزمایش شود. آزمایش‌های زیر باید انجام شوند:

- تست سرریز
- تست آندر فلو
- رند شدن اعداد
- تلاش برای عملیات نامعتبر
- جمع و تفریق معمولی
- تست پایپ لاین

برای هر یک از دسته بندی های فوق، یک مثال کافی است.

۳ شیوه بارگذاری

- فایل ارسالی باید در بخش $HW2_StudentNum$ آپلود شود.
- ساختار دایرکتوری داده شده در فایل RAR را حفظ کنید.
- یک فایل گزارش نیز به همراه کد پیوست کنید.
- فایل گزارش باید تایپ شود. نوشتن در $LaTeX$ امتیاز بیشتری خواهد داشت.

- طراحی تست در متلب امتیاز اضافی خواهد داشت.
- توضیح فایل TCL و نحوه‌ی کارکرد آن نکات تکمیلی خواهد داشت.
- کد تمیز و کامنت‌گذاری شده بسیار مورد استقبال قرار می‌گیرد.

ددلاین : ۱۰ آذر