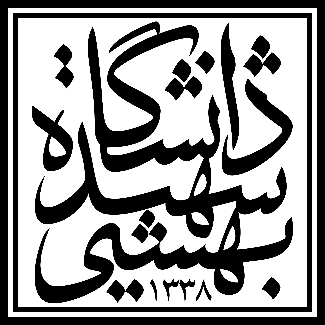
**به­نام خدا**



تمرین 5 معماری

نوید رئیــــــــــــــــــــــــــس­زاده – محمد بهرامــــــــــــــــــــــــــــی.

* **سوال 1 :**

هنگامی که رجیستر فایل ما دارای یک ورودی باشد نیاز است که ابتدا با اضافه کردن یک مالتیپکسر سه به یک سه ورودی rs1, rs2 , rd را به تک ورودی رجیستر فایل متصل کنیم و یک سیگنال دو بیتی کنترل(RegFileInCtrl) را به کنترل یونیت اضافه کنیم.

در هیچ سایکلی همزمان read و write را نیاز نداریم ولی در دستورات R-type و beq ما نیاز داریم که هم rs1 و هم rs2 را بخوانیم که نیاز است در دو سایکل این مورد انجام شود و در دو رجیستر nonarchitectural بعد از رجیستر فایل ذخیره شوند.

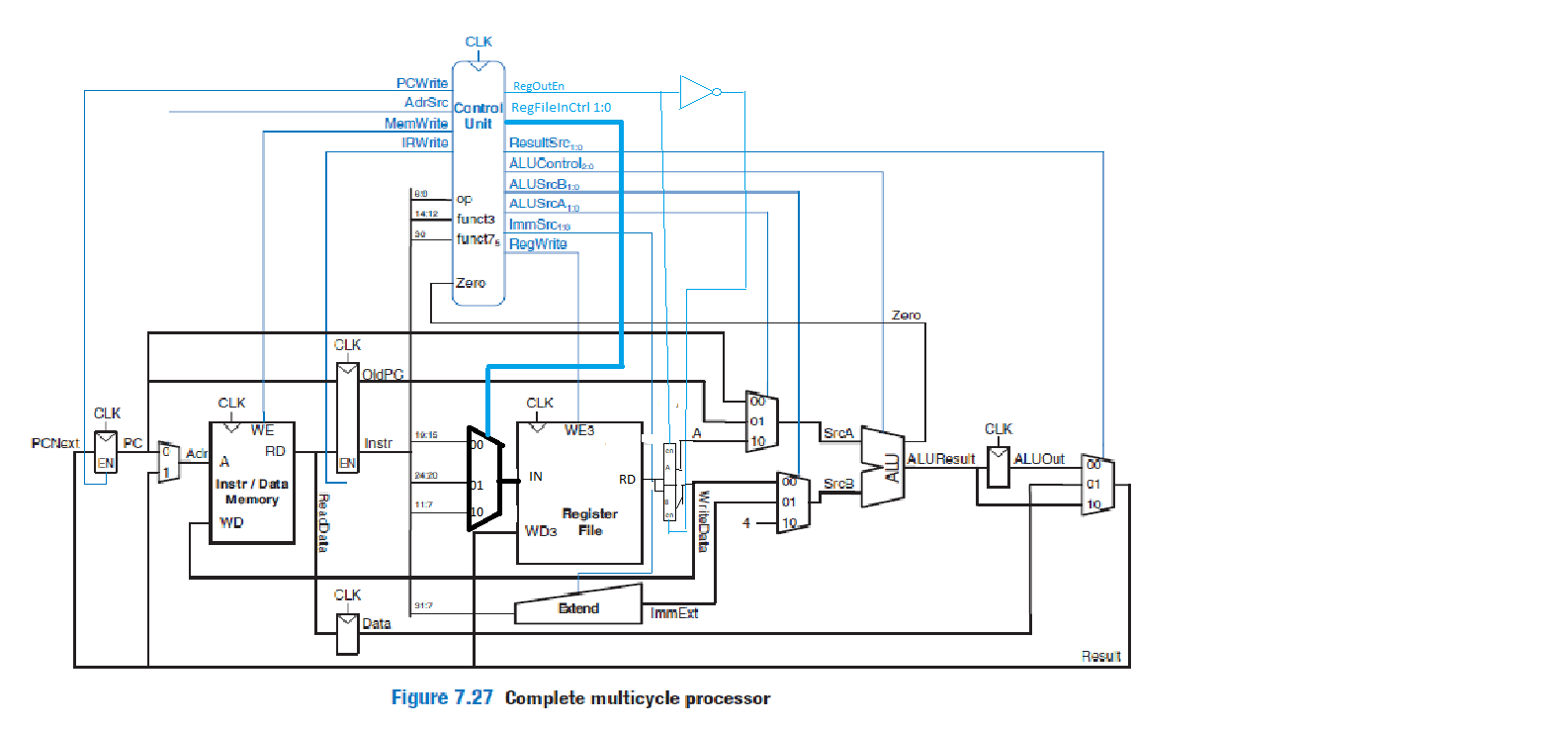
چون یک ورودی داریم یه خروجی هم داریم که در سایکل اول نیاز است تا rs1 را بخوانیم و در رجیستر A ذخیره کنیم و در سایکل دوم rs2 را.

برای این منظور از خروجی یک سیم به رجیستر A و یک سیم به Bمیبریم و در سایکل اول enable رجیستر A را فعال کرده و در سایکل دوم enable رجیستر دوم را که نقیض enable اولی است.نام enable را RegOutEn میگذاریم.

در FSM مان یک استیت اضافه میکنیم که اگر دستور R-type یا beq بود بعد از استیت Decode به آن استیت میرویم تا rs2 را بخوانیم و بعد از آن اگر دستور beq بود به استیت BEQ رفته و اگر دستور R-type بود به ExecuteR میرویم.

در قسمت هایی که نیاز به write کردن هست ، باید کنترل مالتیپلکسر را 10 قرار دهیم.

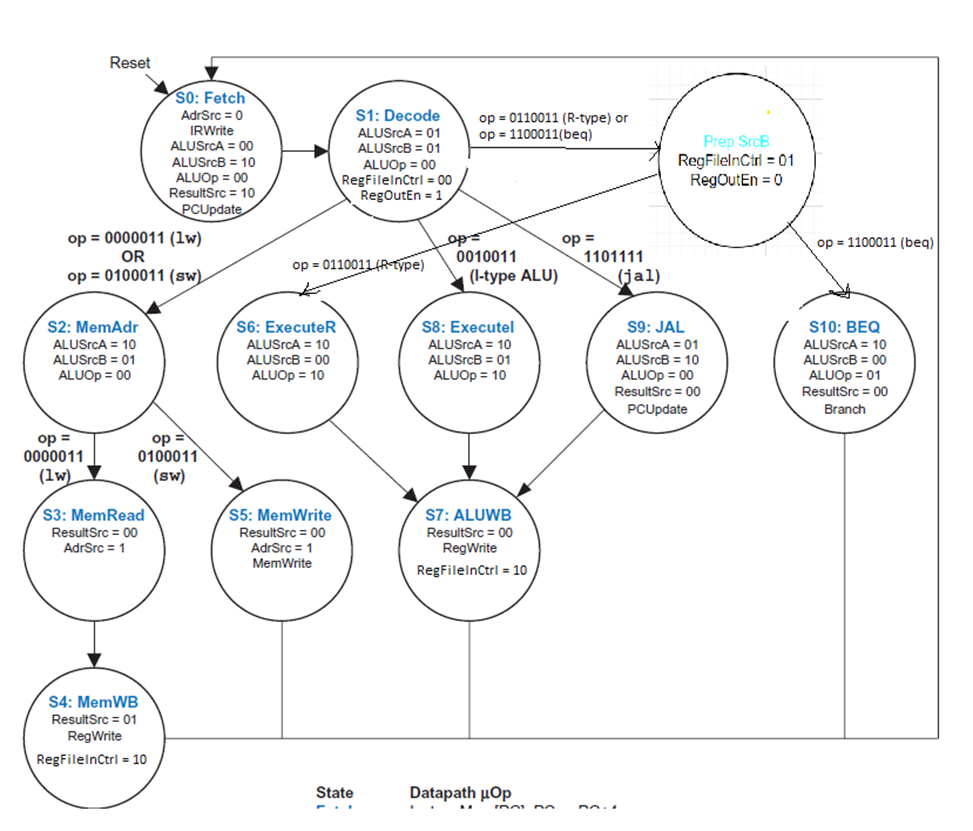
DataPath:



FSM:

نام استیت اضافه شده را PrepB قرار میدهیم و پس از ست کردن دستورات برای خواندن B اگر opcode برای beq بود به استیت BEQ رفته و اگر برابر با R-type بود به استیت ExecuteR میرویم

در دو استیت ALUWB, MemWB باید مقدار RegFileInCtrl را برابر 10 قرار دهیم تا مقداری که باید رایت شود انتخاب شود



**پی­نوشت: تصاویر FSM و Data Path به صورت جداگانه و با نسخه­ی باکیفیت درر فایل­بندی موجود است.**

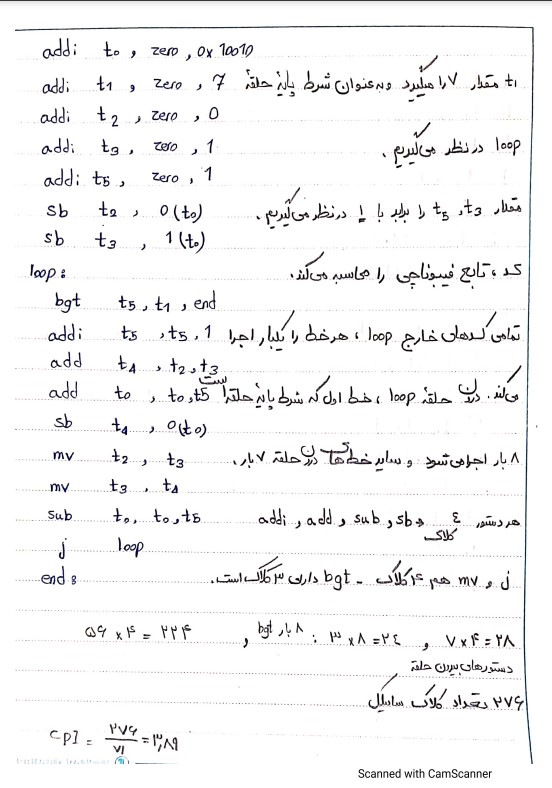
* **سوال دوم :**

ابتدا تعداد اجرای هر خط را می­شماریم. هر خط بیرون حلقه­ی loop یکبار اجرا می­شود و داخل حلقه­ی for نیز 7 بار اجرا میگردد. البته که شرط حلقه 8 بار اجرا میگردد.

تابع فوق تابع فیبوناچی است و جمع دو عدد قبل از هر عدد را محاسبه میکند و به عنوان جروجی 7امین عضو فیبوناچی را باز میگرداند.

برای محاسبه­ی تعداد کلاک­ها، تعداد اجرای هر خط را باید در تعداد کلاک­های هر instruction ضرب کنیم. تعداد کلاک­های هر instruction را میتوان بر اساس data path آن یافت.

برای محاسبه­ی CPI هم تعداد کلاک­ها را بر تعداد اینستراکشن­ها تقسیم می­کنیم.



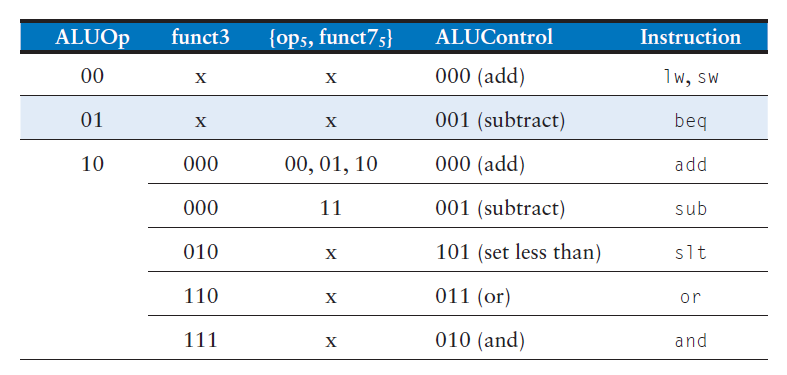
سوال سوم:

دستور srai یک نوع دستور I-type است.



تنها تغییری که باید اعمال شود تغییر در ALU است تا بتواند دستور srai را اجرا کند.

ششمین دستوری که ALU میتواند اجرا کند shift right arithmetic خواهد بود که به آن کد 100 را اختصاص میدهیم (در گذشته به slt مقدار 101 را اختصاص داده بودیم در صورتی که کد 100 هنوز استفاده نشده بود). حال باید با ALU Decoder Truth Table را متناسب با srai تغییر دهیم.





Opb5 = 0 و funct7b5 =1 بودن.

نیازی به تغییر دیتاپث و FSM نیست و فقط ALU باید تغییر میکرد .