

«آز یازده (آخر معماری کامپیوتر) (نسخه ساده سازی شده ی مانوا»

«رایان ترتیب» 9736098 و «سیرعلی بلالی» 9837709

توضیح کلی: یک ماژول Top داریم که در آن از Command باس و کنترل یونیت Instance

گرفته شده، از خط 50 تا 60 اجرای Common باس است و از 90 تا 50 هم اجرای کنترل یونیت تعریف شده.

تعدادی سیگنال تعریف شده که برای ارتباط Common باس و کنترل یونیت است و در نهایت از خط 177 تا 145،

Instance که با باس گرفته شده، ورودی و خروجی ها بهش داده شده اند، خط 177 تا 220 هم همین کارها برای

کنترل یونیت انجام می گیرد.

از خط 54 تا 4 مربوط به ورودی ها و خروجی های ماژول Common باس است، از خط 60 تا 164،

تعدادی اجزا تعریف شده مربوط به رجیسترهایی که می خواهیم استفاده کنیم، ALU و Memory، از خط 177 تا 166

(دستی)

مربوط به سیگنال هایی است که تعریف کردیم. خط 177، یک سیگنال داریم، C-B که مشخص کننده باس ارتباط

بین Registerها است و یک سیگنال که به ترتیب ترتیبی 2، 1 و 0 است. این برای قسمت ای است

که با باس

که مشخص کننده قرارگیری چه چیزی بر روی باس است. از خط 193 تا 210 هم Instance های مختلف گرفته

قرار داده شده Memory، AR، PC، DR، AC، ALU، رجیسترهای E، IR، TR و خروجی ها.

برای بررسی Memory، آدرس اش را از AR می گیرد، آدرس اش می شود AR، Data ورودش همان Data روی

باس است، Data خروجی اش می شد Write & Read Out اش هم که به همین صورت است.

enable

clips

AR و PC هر دو درازده بیت هستند، یعنی Register ها 76 بیت هستند. برای بررسی Data و AR ورودی این همان

Common باسی می شود و از 76 تا 77 مفر (12 بیت)، Data خروجی این می شود و AR Load و AR می شود و AR

، Clock، AR Reset، Reset، Clock، Increments، AR Increment و به همین صورت. از خط 273 تا 220 م،

برای آریستریه پیاپی سانی شده اند مقدار 5 ای که داریم آن 50 می بود AR تری می شود آن 50 یا دو بود PC

تری می شود، سه بود DR و آن هفت بود Memory out.

ماژول Memory خط 6 تا 72 ورودی و خروجی ها است Memory Type تری می کردیم در خط 76، به این صورت

که 72 تا 73 خانی 16 بیت است و در خط 74 از این نوع، نمونه تری می که مقدار خانی های 5 تا 5 و 70 تا 73 این را

5 تا 5، دستورات این 70 تا 73 م، Data ها است. به عنوان مثال در خانی مفر نوشته شده، اضافه کند به 70.

در خانی یک نوشته شده که به خانی یازده ضرب کند و بعدش and کند با 72، سیف راست به، خانی 73 را Load

کند و نهایت پاک این کند. اولین بار مقدار 70 میاد، از خانی 70 و با مقدار خانی 76 میاد که 2 باشد و ضرب می شود

و 20 می شود و بعد and می شود با مقدار خانی 72 که 4 است و مقدار 4 می شود، سیف راست داده می شود و به 2

تبدیل شده، مقدار 73 از خانی 73 Load می شود و در نهایت مفری شده خط 33 و 34 هم مکانیم خواندن و

نوشن است که پیاپی سانی شده.

همی Register کانی که داریم مل هم هستند و بی توضیح داده می شود، Register AR، خط 6 تا 73،

ورودی ها هستند، کیا سیال Temp تری شده که اینجا 72 بیت و بعد نفیم که Temp آن چیزی است که مقدار را در خروجی

دخیره می کند

clips

گفتیم آدرس‌های بالا رونده‌ی Clock آمد، آدرس Reset بود، مقدار آدرس را می‌گذاشت، Increment بود و یک واحد به

مقدار اضافه کند آدرس ما بود، مقدار ورودی را به Temp بریزد، در نهایت هم در خط 37 Temp را به خروجی دادیم.

تیمار ما و کلیه کارهای ما در ALU است، ورودی‌ها و خروجی‌ها مشخص است، با توجه به اینکه عملیات

جمع، Increment و ضرب ممکن است که بیت کلی اضافه برآید، تولید کنند که برای آن‌ها سیگنال‌های متفاوتی از

خط 77 تا 20 در نظر گرفته شده و در خط 25 تا 27 هم، به عنوان مقدار دادیم تا عملیات‌ها انجام بشوند.

بعد مشخص کردیم که کدام یک از این عملیات‌ها باید به خروجی مستقل شود. صفر آدرس ضرب، یک باشد جمع، با توجه به ALU

و And و Or و ... و C out هم به همین صورت مشخص شده در نهایت از خط 50 تا 52 هم،

مقدار ~~ALU~~ ^{Zers} با توجه به اینکه مقدار خروجی صفر است یا نه مشخص شده و C out temp هم به output temp

و C out داده شده.

در ماژول Control unit و در Decoder یک Sequence استفاده شده، Decoder برای تولید سیگنال 3 to 8 Counter (C)

های D و Decoder برای تولید سیگنال‌های A و SC هم شروع به شماردن می‌کند و در نهایت با استفاده از خروجی 4 to 16

SC، Aهای time slot‌های زمانی تولید می‌شود. قسمت مهم این ماژول از خط 94 به بعد است، به این صورت که

لتر T می‌شود به اتفاقی رخ دهد، آدرس یک بود، آدرس دو بود، آدرس سه بود و از 4 متفاوت خواهد بود.

(1) Opcode‌ها: با کامنت جلوی هر کدام نوشته شده که چه دستوری است، مثلاً اگر که IR ایپ 15، صفر باشد و

Do یعنی Do، یک باشد و IR ایپ 15 هم صفر باشد، وقتی Do، یک باشد، یعنی 000 باشد عقلاً آن 3 بیت.

4. بیت برای Opcode داریم. بیت پرازشش اُس می شود IR ایست و آن سه بیت دیگر با D مشخص می شود.

اگر D0 یک باشد یعنی آن 000 بود، می شود 4 مغز (عملیات ضرب)، 001 بود، می شود عملیات جمع و متمم که D1 باشد.

و متمم که D2 یک باشد یعنی یک مغز و آن 3 بیت بعدی عدد 2070 And و به همین صورت تا خط 394.

از این خط به بعد IR ایست ها، یک می شوند و حالا D0 یک باشد 000 یعنی 6 با 000 عملیات Load است.

1 با 001 عملیات Store است و به همین صورت. برای تکلیف از آن ها، متناظراً سیگنال هایی که باید ^{مقداری} ~~بلیز~~ باشند.

T0 و T1 بررسی می شوند و در T0 و S2 و S1 و S0 که چون می شوند 070، یعنی مقدار PC روی Bus قرار می گیرد، AR هم Load

یک می شود یعنی اینکه مقدار PC ای که روی Bus قرار گرفته، در AR قرار می گیرد و بعداً چیزها هم می شوند. در T1،

S2، S1 و S0 هر سه تا 1 هستند و Read Enable و PC ^{increment} هم یک باشد و IR ^{Load} هم یک باشد. یعنی مقدار Memory

روی Bus قرار می گیرد و Read Enable هم یک است یعنی عملیات خواندن انجام می شود پس آن مقداری که در AR

قبلاً گذاشته شده، الآن می رود سراغ حافظه، مقدار آن خانه خوانده می شود، AR به عنوان آدرس اُس بوده،

روی Bus قرار می گیرد، IR ^{Load} هم یک هست پس نه IR، لود می شود، PC ^{increment} هم یک است پس یک واحد هم

به PC اضافه می شود، به همین صورت باقی دستورات قابل Decode هستند.

(2) توضیح چگونه اجرای چرخه Fetch، Decode، با کنترل یونیت بررسی می کنیم، همون چرخه مربوط به T0 = 1

execute

T1 = 1 وقتی است که برای Fetch استفاده می شود چون که اولش PC مقدارش وارد AR می شود و در T0

در T1 هم، مقدار Memory خوانده شده و برگردانده می شود.

مست decode عملی کاری هست که بعد از اینکه Instruction در IR قرار می گیرد در خط 87 جاری می شود

15

ایمپ

Controller و آن Decoder در حقیقت انجام می دهد که D را بر مبنای سیگنال با استفاده از D خود IR 3 to 8

با عملیات decode انجام می دهیم و با توجه به آن مشخص می شود که هر دستوری باید چه کاری را انجام دهد، این از خط

260 تا آخر Control، هر Else ای که نوشته شده برای هر دستور مختلف ایمپ، این مست decode است که IF unit

گفته می شود چه کاری باید انجام می شود و سیگنال ها مقداری می گیرند.

مست execute، به ازای هر دستور ما یک ALU op مشخص می کنیم، برای ALU op دستور Move و

موازی است، Add، یک است، And 2 است و به همین صورت، این مقداری است که داده می شود و

ALU مشخص می شود چه کاری باید انجام شود و از روی آرگومان های ورودی است این کار را انجام می دهد

③ RTL و توضیح: عملی های RTL در فایل وجود دارد، RTL بین ساینی Register ها در حال

جایگاه شدن هستند و توضیح مختصر از آن در مورد بخش Common Bus: تمام Register ها، واحد ALU و

...

واحد ~~پردازش~~ آریستر که برای نمایی داده شده است. بین اینها Register ها هستند که روی Bus قرار دارند

می شود و با توجه به سیگنال های کنترلی یک مقدار از Bus برداشته می شود و در یکی از Register ها به دلخواه.