

# میانترم طراحی سیستم‌های دیجیتال

پارسا محمدیان – ۹۸۱۰۲۲۸۴

۳۱ اردیبهشت ۱۴۰۰

## فهرست مطالب

۳	۱ سوال ۳
۲	۱.۱ پیاده سازی پردازنده subleq . . . . .
۲	۲.۱ نوشتن کد مرتب سازی . . . . .
۲	۳.۱ اجرای کد بر روی پردازنده . . . . .
۲	۲ سوال ۷
۳	۱.۲ جزئیات پیاده سازی . . . . .
۴	۲.۲ جزئیات تست . . . . .
۴	۳ سوال ۹
۶	فهرست منابع

## ۱ سوال ۳

### ۱.۱ پیاده سازی پردازنده subleq

برای آشنایی با این معماری از منابع [۴] و [۶] و [۲] استفاده کردم. برای پیاده سازی این پردازنده، ماژول subleq را ساختم و برای سهولت مموری را درون آن به صورت آرایه بیت وکتور در نظر گرفتم که قابلیت لود کردن دارد. بقیه جزئیات پیاده سازی در کد با کامنت توضیح داده شده است.

### ۲.۱ نوشتن کد مرتب سازی

از آنجایی که Assemble کردن کد این ماشین کار طاقت فرسایی است، به زبان پایتون کدی برای اسمبل کردن کد این ماشین می نویسیم. نمونه کد ورودی Assembler در کد ۱ آمده است. برای راحتی کار کد ورودی را در قالب JSON، می نویسیم. خروجی این برنامه دستورالعمل ها به باینری و آدرس آنها است. اگر پارامتر -d به این برنامه پاس داده شود داده ها به همراه آدرسشان در خروجی قرار می گیرند.

```
1 {
2   "code": [
3     {"instruction": "", "op1": "", "op2": "", "jaddr": ""},
4     ...
5   ],
6   "data": [
7     {"identifer": "value",
8     ...
9   ]
10 }
```

Listing 1: Subleq assembler input code

برای Assemble کردن کد برنامه را با پارامترهای زیر اجرا می کنیم.

```
python assembler.py -d -i "input.json" -o "output.txt"
```

حال کد باینل سورت و دیتا را می نویسیم و به کمک این زبان آن را به کد ماشین متناظر تبدیل می کنیم. کد این بخش در فایل sort.json موجود است.

### ۳.۱ اجرای کد بر روی پردازنده

اگر پارامتر -v را به اسمبلر پاس دهیم کد لود کردن در مموری را برای وریلاگ به ما می دهد. این کد را در فایل sort.v و در ماژولی با همین نام قرار می دهیم. سپس با شبیه سازی نتیجه را بررسی می کنیم.

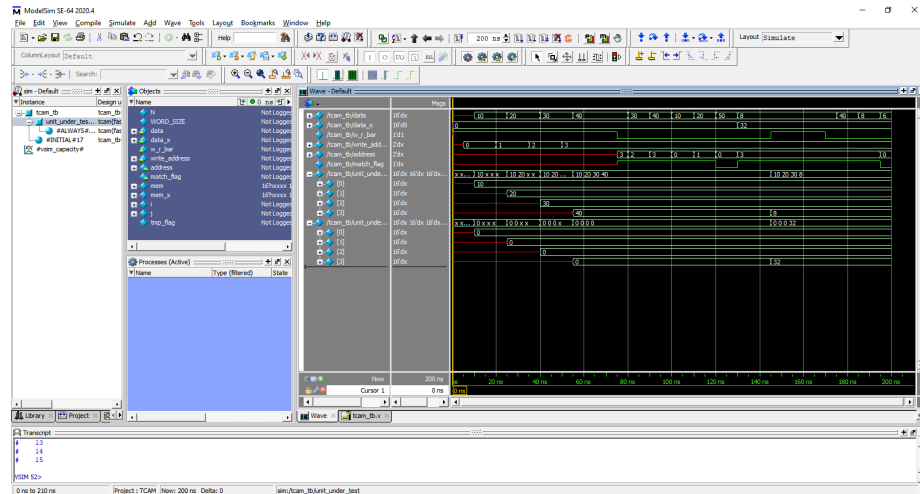
## ۲ سوال ۷

### ۱.۲ جزئیات پیاده سازی

برای آشنایی با TCAM از منابع [۵] و [۱] استفاده کردم. همچنین از تمرین پنجم درس ساختار و زبان کامپیوتر پائیز ۱۳۹۹ دکتر ارشدی که در مورد CAM بود نیز استفاده کردم. سپس برای پیاده سازی از توصیف رفتاری استفاده کردم. در کد از دو پارامتر N و WORD\_SIZE استفاده شده است که N طول آدرس است. نکته قابل توجه این است که علی رغم اینکه وریلاگ در منطق خود X دارد ولی در سنتز چنین چیزی وجود ندارد چون یک سیم حتی اگر مقدارش را ندانیم دارای مقدار است. پس باید خودمان منطق سه تایی را پیاده سازی کنیم. برای این کار برای هر داده علاوه بر خود داده، وکتور دیگری در نظر گرفته شده که ۱ بودن هر خانه از آن مشخص کننده X بودن خانه متناظر در داده است. با این تفصیل سراغ پیاده سازی می رویم. پیاده سازی در فایل tcam.v موجود است.

## ۲.۲ جزئیات تست

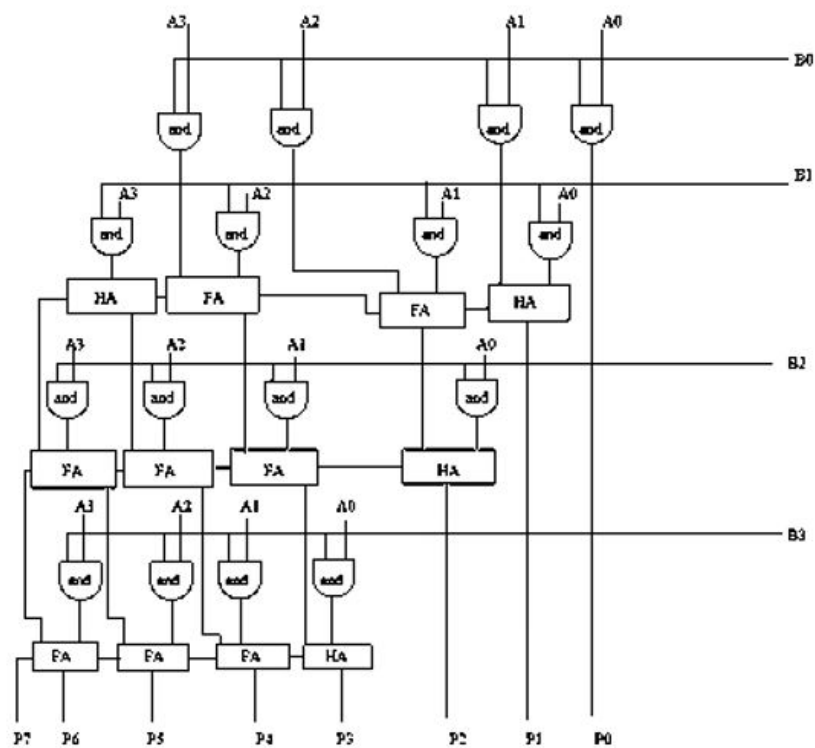
برای تست فایل `tcam_tb.v` نوشته شده است. در این تست ابتدا مقادیری در حافظه ذخیره شده که همه بیت های آن ها مشخص است. سپس هر یک از مقادیر فراخوانی شده است و آدرس متناظر دریافت شده است. پس از این ها به سراغ تست نوشتن با X میرویم. سپس با دو مقدار متفاوت که در بیت نامعلومشان اختلاف دارند آدرس را میگیریم. Wave مربوط به این تست در شکل ۱ نشان داده شده است.



شکل ۱: شکل موج تست TCAM

## ۳ سوال ۹

برای حل این سوال ابتدا با Array Multiplier از طریق [۷] و [۳] آشنا شدم. سپس برای پیاده سازی از شکل ۲ که در منبع اول موجود بود استفاده کردم. لازم به ذکر است که برای سادگی در کد، ماژول part را به صورت ترکیب fa و گیت and تعریف کردم. در این صورت به ازای هر بخش از این ماتریکس از این ماژول استفاده کردم. برای مثال اگر فقط نیاز به گیت and دو ورودی دیگر را صفر می‌دهیم. جزئیات دیگر کد پایتون تقریباً واضح است. برای تست یکی ضرب‌کننده ۴ در ۴ می‌سازیم و برای آن تست بنچ می‌نویسیم. تست بنچ در ماژول stimulus است.



©Elprocus.com

شكل ٢: Array Multiplier

## فهرست منابع

- [1] *Content-addressable memory*. URL: [https://en.wikipedia.org/wiki/Content-addressable\\_memory#Ternary\\_CAMs](https://en.wikipedia.org/wiki/Content-addressable_memory#Ternary_CAMs). (accessed: 16.05.2021).
- [2] jaredkrinke. *SIC-1 programming game*. URL: <https://jaredkrinke.itch.io/sic-1>. (accessed: 21.05.2021).
- [3] Tutorials Point (India) Ltd. *Multiplication Using Array Multiplier*. URL: <https://www.youtube.com/watch?v=gTxgiJHBfsI>. (accessed: 16.05.2021).
- [4] *One-instruction set computer*. URL: [https://en.wikipedia.org/wiki/One-instruction\\_set\\_computer](https://en.wikipedia.org/wiki/One-instruction_set_computer). (accessed: 21.05.2021).
- [5] Jessica Scarpati. *Ternary content-addressable memory (TCAM)*. URL: <https://searchnetworking.techtarget.com/definition/TCAM-ternary-content-addressable-memory>. (accessed: 16.05.2021).
- [6] *Subleq*. URL: <https://esolangs.org/wiki/Subleq>. (accessed: 21.05.2021).
- [7] *What is 4x4 Array Multiplier and Its Working*. URL: <https://www.elprocus.com/4x4-array-multiplier-and-its-working/>. (accessed: 16.05.2021).