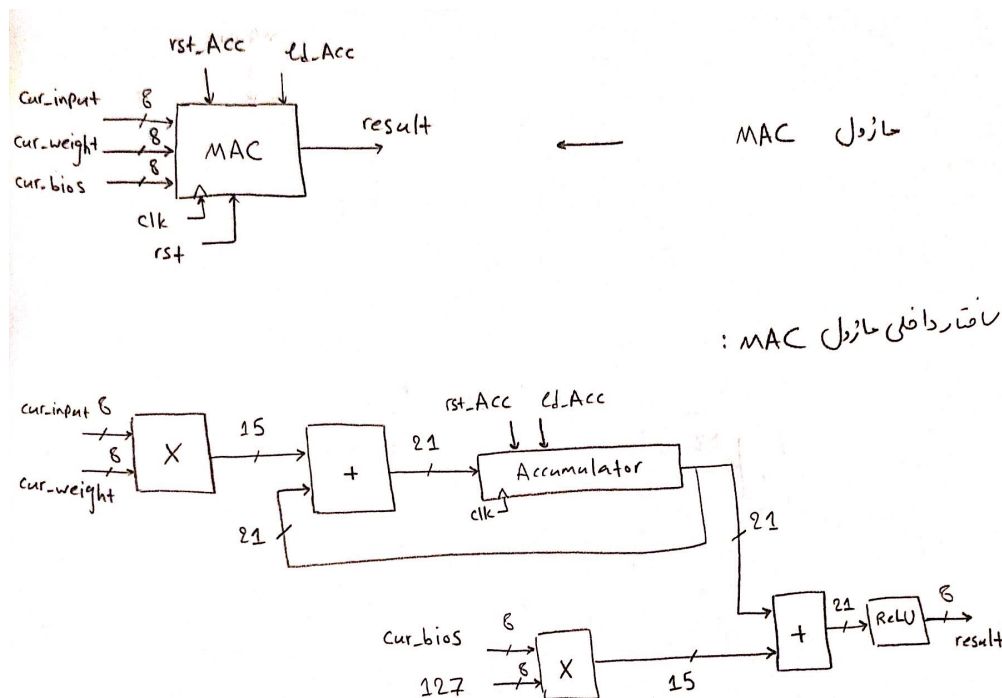


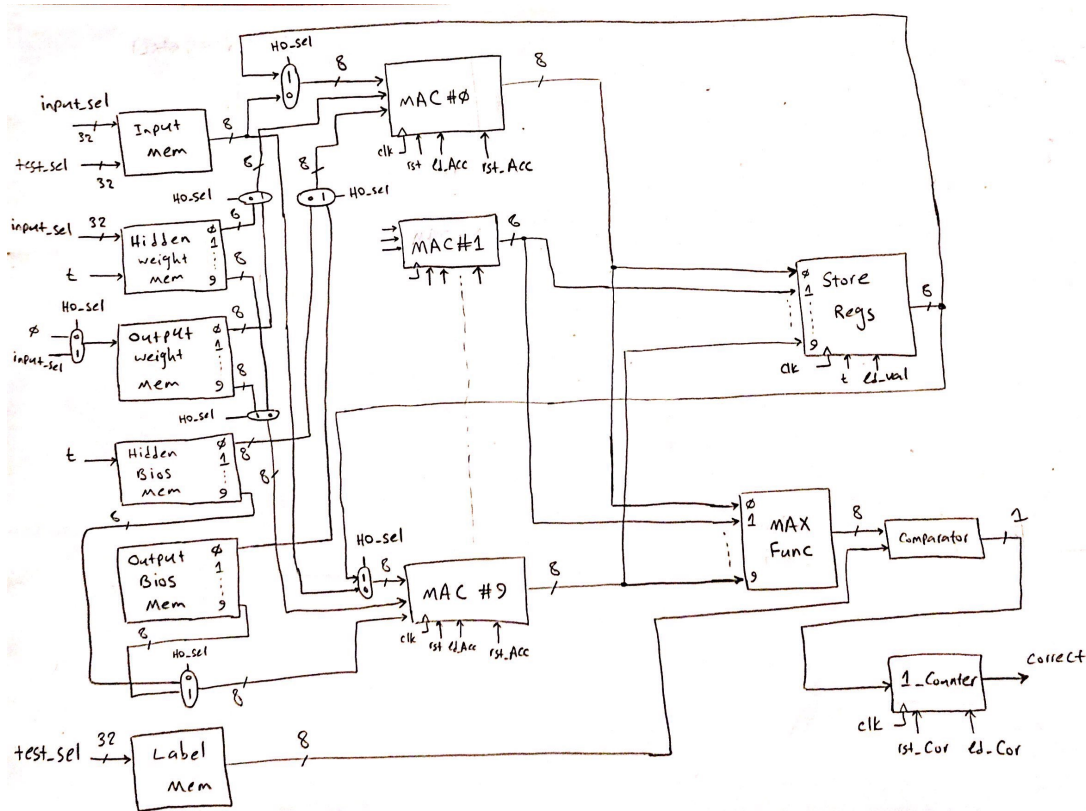
به طور کلی شبکه مصنوعی عصبی طراحی شده به این صورت کار می‌کند که ابتدا با استفاده از ۱۰ ماژول MAC موجود، خروجی ۱۰ سلول عصبی ابتدایی از hidden layer را به دست آورده و سپس آن‌ها را در ۱۰ رجیستر ذخیره می‌کند. سپس خروجی‌های ۱۰ سلول عصبی دیگر از hidden layer را به دست آورده و در ۱۰ رجیستر دیگر ذخیره می‌کند. حال با استفاده از ۲۰ مقدار ذخیره شده در رجیستر ها و ۱۰ ماژول MAC موجود، خروجی ۱۰ سلول عصبی داخل قسمت Output layer را به دست می‌آوریم. سپس با استفاده از ماژول MAX\_Func از بین ۱۰ خروجی تولید شده، label که مقدار متناظر با آن، بیشتر از بقیه است را به دست می‌آوریم. با استفاده از یک مقایسه کننده، خروجی به دست آمده را با label داده شده مقایسه می‌کنیم. در صورت برابر بودن آن‌ها با هم، به مقدار خروجی یا correct\_count یک واحد اضافه می‌کنیم. در نهایت برای به دست آوردن دقت شبکه عصبی پیاده سازی شده کافی است مقدار correct\_count را بر تعداد تست‌های داده شده تقسیم کنیم، که با توجه به خروجی به دست آمده از شبکه عصبی پیاده سازی شده خواهیم داشت:

$$Accuracy = \frac{correct\_count}{test\_cnt} = \frac{671}{750} = 89.46\%$$

در ادامه تصویری از ماژول MAC استفاده شده در طراحی شبکه عصبی و ساختار داخلی آن قرار داده شده است:



تصویری از Datapath شبکه عصبی طراحی شده:



همان طور که در تصویر مشخص است، در Datapath از ۱۰ ماژول MAC، تعدادی Memory برای ورودی ها، وزن ها و مقادیر bias، ۲۰ رجیستر داخل واحد Store Regs، یک ماژول MAXFunc که در نهایت label با بیشترین مقدار خروجی را به دست می آورد، مقایسه کننده، شمارنده تعداد label های صحیح به دست آمده و تعدادی MUX تشکیل شده است. در ماژول Store Regs، می توان با فعال کردن سیگنال Id\_val عملیات نوشتن در رجیستر های داخل آن را انجام داد و با استفاده از سیگنال t میتوان رجیستر های ۰ تا ۹ (اگر  $t = 0$ ) و یا رجیستر های ۱۰ تا ۱۹ (اگر  $t = 1$ ) را به عنوان رجیستر های مقصد، انتخاب کرد.

در ماژول Input Mem، ۲ سیگنال ورودی input\_sel (برای مشخص کردن شماره ورودی) و test\_sel (برای مشخص کردن شماره تست) وجود دارد و با توجه به این دو سیگنال، ورودی مورد نظر روی خروجی ماژول قرار می گیرد. در ماژول Hidden weight Mem، ۲ سیگنال ورودی input\_sel (برای مشخص کردن شماره ورودی) و t (برای مشخص شدن این که ۱۰ سلول ابتدایی hidden layer را نیاز داریم یا ۱۰ سلول دوم) وجود دارد که توسط این دو سیگنال ۱۰ وزن متناظر با شماره ورودی و گروه سلول ها در hidden layer روی ۱۰ پورت خروجی قرار می گیرد.

در مازول Output Weight Mem، توسط سیگنال ورودی input\_sel، وزن های متناظر با این شماره ورودی و مربوط به ۱۰ سلول داخل output layer روی ۱۰ پورت خروجی قرار میگیرد.

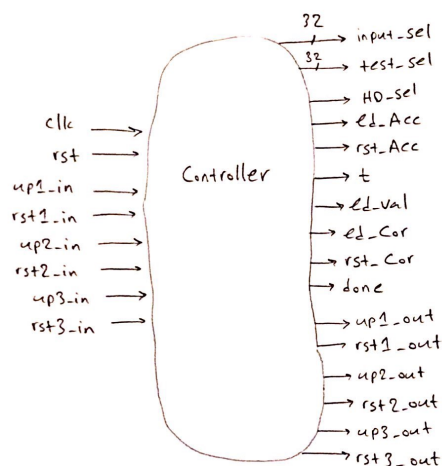
در مازول Hidden Bios Mem، توسط سیگنال ورودی t، مقدار Bios متناظر با ۱۰ سلول اول یا ۱۰ سلول دوم داخل hidden layer، روی ۱۰ پورت خروجی قرار میگیرد.

در مازول Output Bios Mem، همواره روی ۱۰ پورت خروجی، مقدار Bios متناظر با ۱۰ سلول داخل output layer قرار میگیرد.

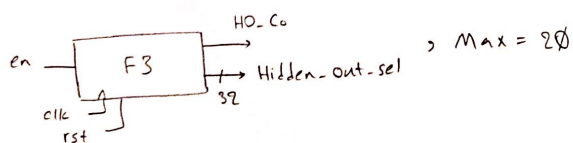
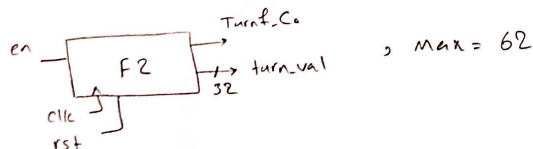
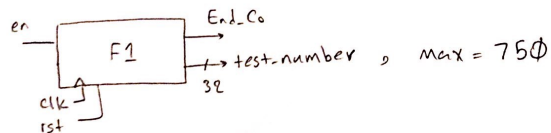
در مازول Label Mem، توسط سیگنال test\_sel، label متناظر با test\_sel روی پورت خروجی قرار میگیرد.

حال تصویری از شبکه عصبی طراحی شده قرار می دهیم:

ورودی و خروجی های کنترلر:

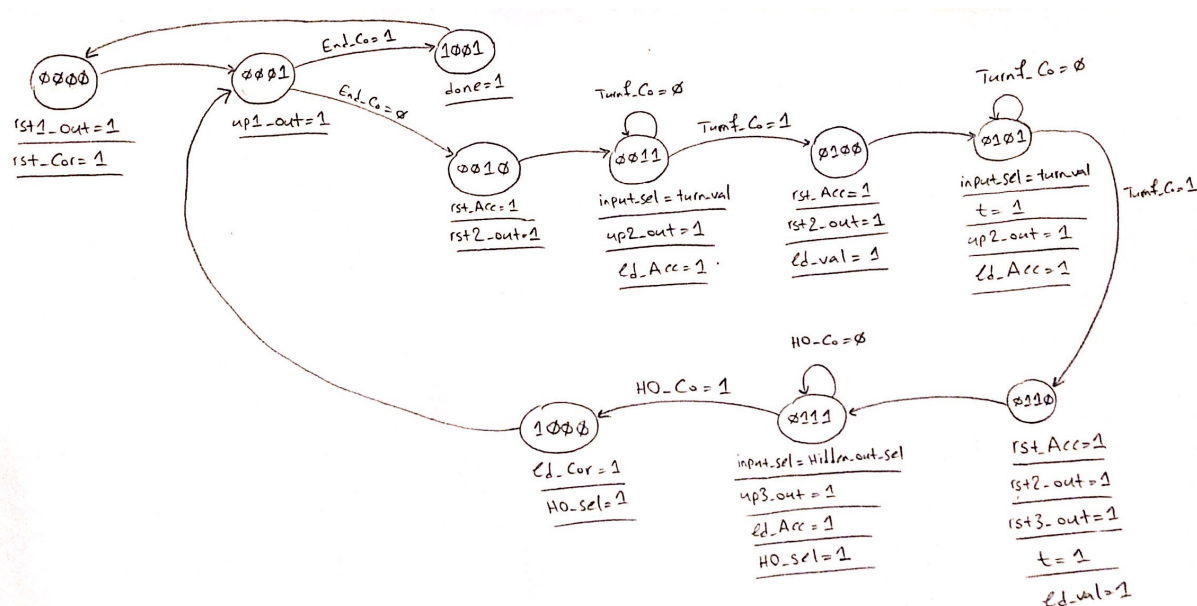


داخل کنترلر، ۳ Counter و مقدار دکر به شکل زیرند:



در داخل Controller، ۳ تا counter وجود دارد که برای مشخص کردن شماره تست، شماره ورودی برای سلول های hidden layer، شماره ورودی برای سلول های output layer استفاده می شود. مقدار ماکسیمم مقدار هر کدام از این counter ها نیز در تصویر نوشته شده است. همچنین سیگنال های ورودی و خروجی Controller نیز در تصویر آورده شده است.

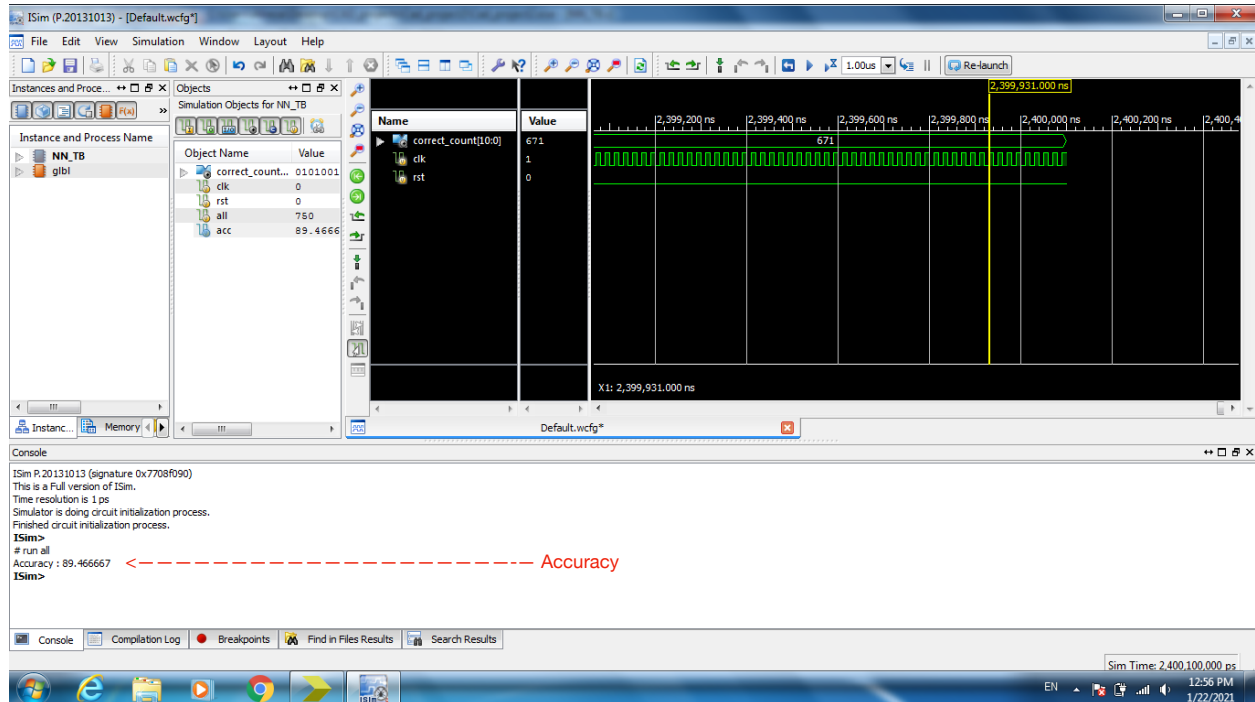
تصویر مربوط به FSM متناظر با Controller:



همان طور که در تصویر مشخص است، FSM طراحی شده دارای ۱۰ استیت است. از 0000 تا 1001.

## نتایج شبیه سازی و سنتز:

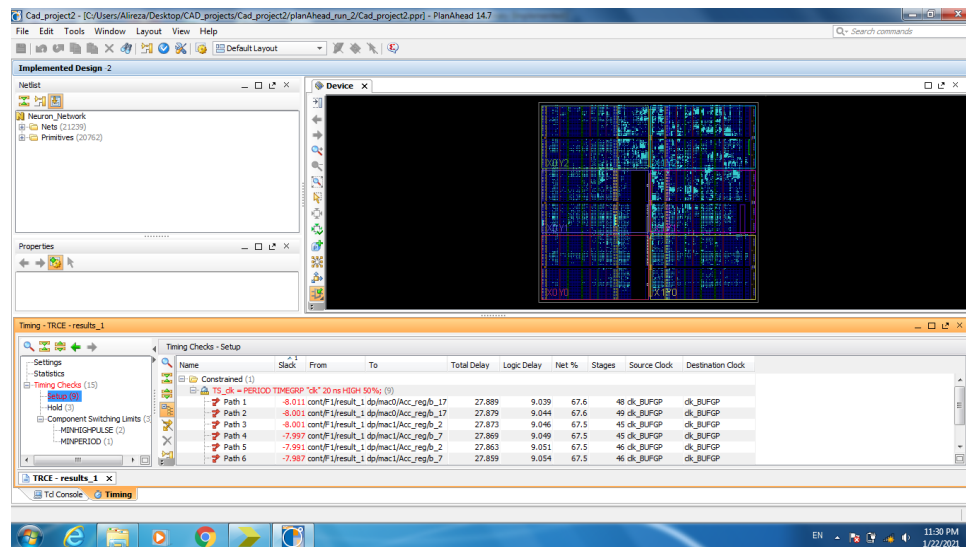
تصویر خروجی حاصل از شبکه عصبی:



میزان utilization و تعداد LUT ها، DSP Block ها، Flip Flop ها و ... :

| Device Utilization Summary             |        |           |             |
|--|--------|-----------|-------------|
| Slice Logic Utilization                | Used   | Available | Utilization |
| Number of Slice Registers              | 910    | 93,120    | 1%          |
| Number used as Flip Flops              | 471    |           |             |
| Number used as Latches                 | 0      |           |             |
| Number used as Latch-thrus             | 0      |           |             |
| Number used as AND/OR logics           | 439    |           |             |
| Number of Slice LUTs                   | 15,381 | 46,560    | 33%         |
| Number used as logic                   | 13,099 | 46,560    | 28%         |
| Number using O6 output only            | 11,424 |           |             |
| Number using O5 output only            | 261    |           |             |
| Number using O5 and O6                 | 1,414  |           |             |
| Number used as ROM                     | 0      |           |             |
| Number used as Memory                  | 1,960  | 16,720    | 11%         |
| Number used as Dual Port RAM           | 1,960  |           |             |
| Number using O6 output only            | 1,920  |           |             |
| Number using O5 output only            | 0      |           |             |
| Number using O5 and O6                 | 40     |           |             |
| Number used as Single Port RAM         | 0      |           |             |
| Number used as Shift Register          | 0      |           |             |
| Number used exclusively as route-thrus | 322    |           |             |
| Number with same-slice register load   | 0      |           |             |

میزان Performance:



$$Performance = \frac{1}{max\_delay} = \frac{1}{27.889ns} = 35856430.85$$

میزان توان مصرفی بخش های مختلف:

