

CAD

Computer Assignment 2 - Phase 2

810197464

محمد امین باقرشاهی

810197640

سینا نگارنده

مقدمه

در فاز قبلی پروژه، ساختار یک نورون منفرد از قبل طراحی شده را پیاده سازی کردیم. در این فاز می خواهیم با پیاده سازی یک (Multi-Layer Perceptron) بر روی مجموعه داده MNIST عملیات classification را انجام دهیم. در این پروژه ساختار model و عملیات train از قبل انجام داده شده است و اطلاعات و داده های مورد نیاز داده شده اند.

ابتدا نورون منفرد که در قسمت قبلی پیاده سازی شده است را بررسی می کنیم و سپس آنها را یک module در نظر گرفته و در سطح بالاتر از آنها استفاده می کنیم.

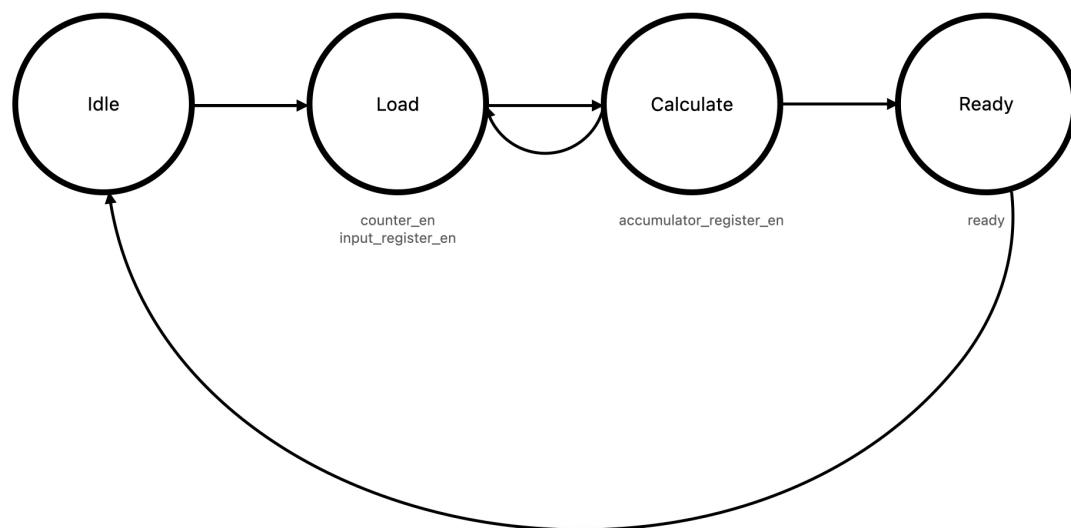
ساختار نورون واحد

طراحی نورون به دو قسمت control unit و data path تقسیم می شود.

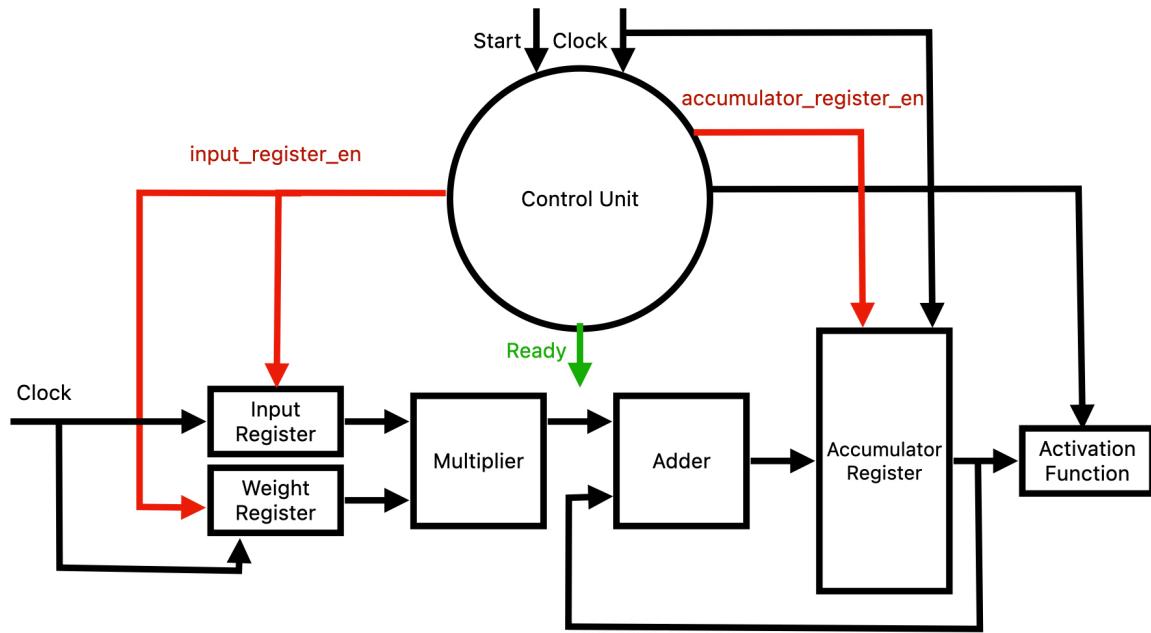
هر نورون دو ورودی می گیرد، برداری از ورودی ها و وزن ها که هر عنصر آنها هشت بیت دارد. هفت بیت اول نشان دهنده مقدار و اندازه عدد و MSB نشان دهنده علامت عدد است. در داخل نورون ضرب داخلی این بردارها را انجام می شود و سپس خروجی را به activation function داده می شود و مقدار خروجی بدست می آید.

8-bit sign-magnitude format	
Sign	Magnitude
7	6-0

قرار گرفته اند که ساختار نورون را نشان می دهند.



FSM of the controller unit



Detailed block diagram

همچنین از تابع ReLU به عنوان activation function در نورون استفاده شده است. تابع ReLU به صورت زیر عمل می‌کند.

$$ReLU(x) = \max(0, x)$$

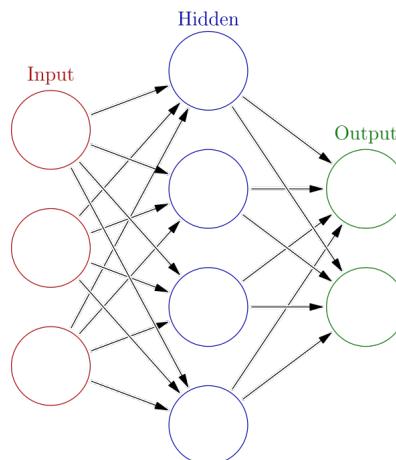
ورودی که به داده می‌شود ضرب داخلی بردار ورودی‌ها و وزن‌ها است که با بایاس جمع شده است.

$$z_i^l = \sum_i w_{ij}^l \times a_i^{l-1} + b_{ij}^l$$

$$a_i^l = ReLU(z_i^l)$$

ساختار MLP

ساختار کلی یک MLP به صورت زیر است و از سه لایه input، hidden و output تشکیل شده است. در این فاز از assignment به تعداد 62 ورودی به عنوان feature-ها داده شده است. و بر اساس ساختار داده شده در این assignment تنها از یک لایه hidden با 20 نورون تشکیل شده است. و به دلیل اینکه می‌خواهیم بر روی مجموعه داده MNIST عملیات classification را انجام دهیم یعنی تشخیص دهیم در عکس داده شده به عنوان ورودی کدام رقم نوشته شده است و 10 رقم داریم پس باید 10 نورون داشته باشیم.



General MLP Structure

اما برای طراحی این MLP بر اساس خواسته assignment تنها می‌توانیم از 10 نورون استفاده کنیم. بنابراین برای انجام classification باید 3 بار محاسبات انجام دهیم یک بار برای 10 نورون اول لایه hidden و بار دیگر برای 10 نورون دوم لایه hidden و بار سوم برای لایه output. وزن‌ها در دو فایل یکی

برای بین لایه ورودی و hidden و بین لایه output به ابعاد (20, 62) و بین لایه hidden و output به ابعاد (10, 20) داده شده است. برای پیاده‌سازی دو قسمت Data Path و Control Unit را باید طراحی و پیاده‌سازی کنیم.

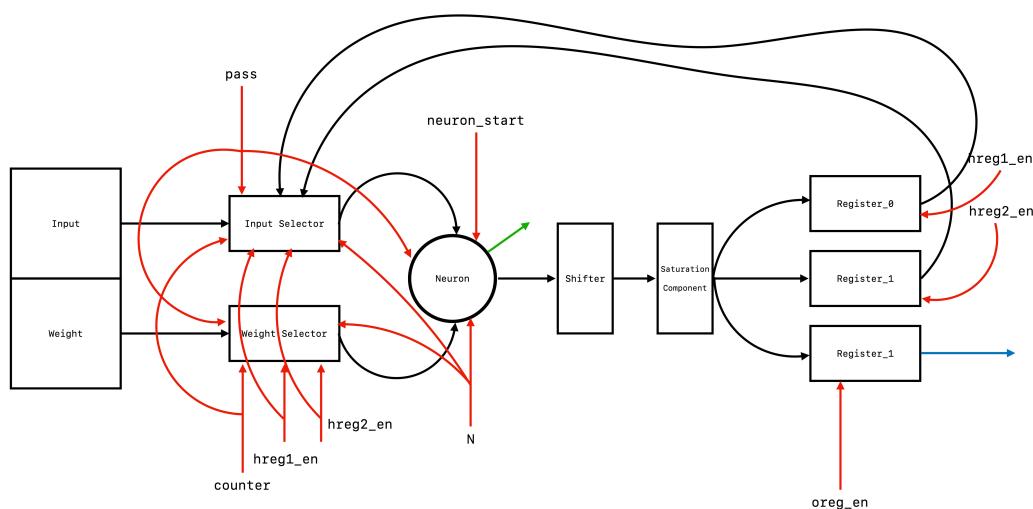
طراحی Data Path

ابتدا بررسی می‌کنیم که Data Path از چه قسمت‌هایی تشکیل شده است.

Data Path:

- Artificial Neuron Module
- Shifter Module
- Saturation Module
- Register

با توجه به توضیحات بالا از نورون در سطح بالا استفاده کرده و به بررسی داخل آن نمی‌پردازیم. در هر بار استفاده از نورون خروجی آن باید 9 بیت شیفت داده شوند و بعد خروجی آن را از saturation (out) و یا hidden می‌دهیم. و در آخر در register ذخیره می‌شوند. خروجی بر اساس اینکه در کدام لایه (hidden یا output) قرار داریم و اینکه اگر در لایه hidden هستیم در دو نورون اول هستیم یا دوم یک سیگنال enable تنها برای یکی از register-ها فعال می‌شود و مقدار خروجی ذخیره می‌شود. برای ورودی هر بار بر اساس اینکه در pass چندم قرار داریم ورودی از یکی از register-ها یا ورودی‌های داده شده که همراه با یاس نیز است انتخاب می‌شود. در زمانی که با یاس انتخاب می‌شود وزن را یک می‌دهیم. و هر بار ورودی‌های نورون داده می‌شود و خروجی مورد نیاز بدست می‌آید. در شکل پایین یکی از ده نورون آمده است.



A neuron inside MLP Data Path

در قسمت قبل که نحوه انتخاب ورودی‌های نورون توضیح داده شد، اگر به صورت رفتاری کد آن را در یک always پیاده سازی کنیم به صورت زیر می‌شود.

```

genvar q;
generate
    for (q = 0; q < 10; q = q + 1) begin

        // input
        always @(pass, counter, test_data, bh, N, hreg1, hreg2, bo, q) begin
            if (pass == 2'b00) begin
                if (counter != 16'b1) data_input[q] = test_data[N - counter];
                else data_input[q] = bh[q + pass * 10];
            end
            else if (pass == 2'b01) begin
                if (counter != 16'b1) data_input[q] = test_data[N - counter];
                else data_input[q] = bh[q + pass * 10];
            end if (pass == 2'b10) begin
                if (counter == 16'b1) data_input[q] = bo[q];
                else begin
                    if (N - counter < 10) data_input[q] = hreg1[N - counter];
                    else data_input[q] = hreg2[N - counter - 10];
                end
            end
        end
    end

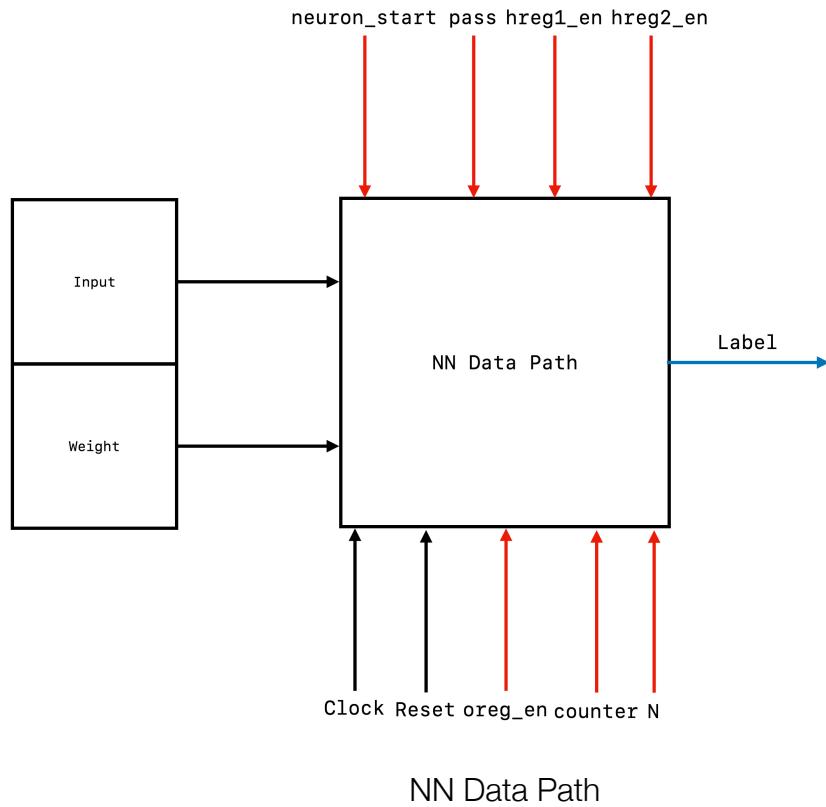
    // weight
    always @(pass, counter, test_data, bh, N, hreg1, hreg2, bo, q) begin
        if (pass == 2'b00) begin
            if (counter != 16'b1) weight_input[q] = wh[(N - counter) + q * (N - 1)];
            else weight_input[q] = 8'b01111111;
        end
        else if (pass == 2'b01) begin
            if (counter != 16'b1) weight_input[q] = wh[(N - counter) + q * (N - 1) + 620];
            else weight_input[q] = 8'b01111111;
        end if (pass == 2'b10) begin
            if (counter != 16'b1) weight_input[q] = wo[(N - counter) + (q * (N - 1))];
            else weight_input[q] = 8'b01111111;
        end
    end
end
endgenerate

```

در نهایت با قرار دادن ده نورون کنار هم می‌توانیم data path را تشکیل دهیم که اگر از سطح بالاتر نگاه کنیم به شکل زیر می‌شود.

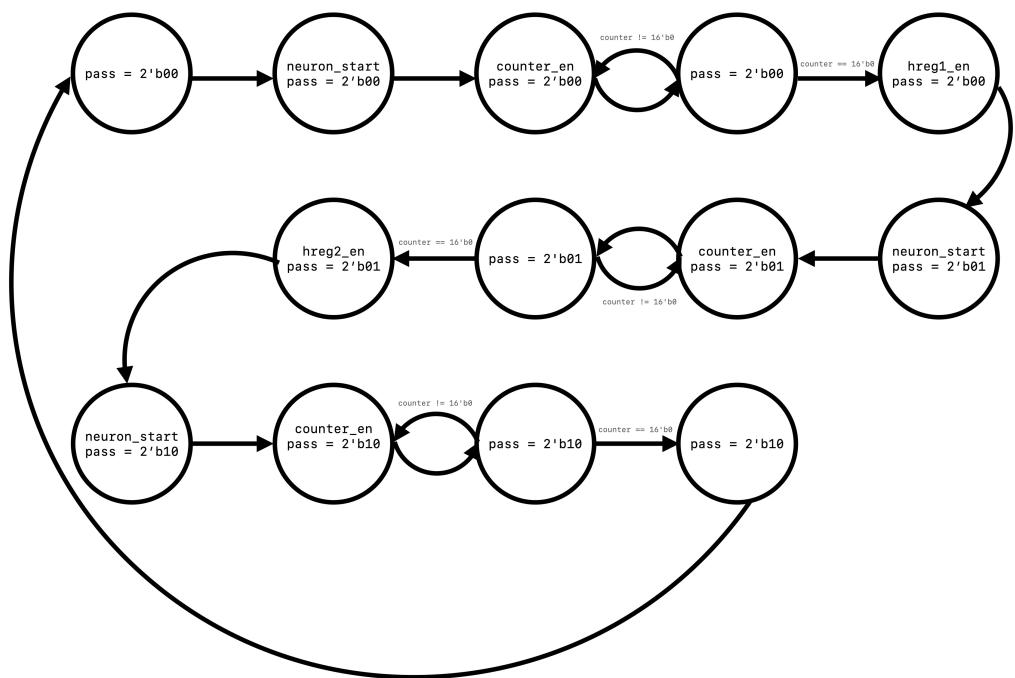
سیگنال‌هایی که با خط قرمز وارد شده‌اند از سمت Control Unit می‌باشند و درنهایت سیگنال آبی رنگ سیگنال خروجی است که با max گرفتن از سیگنال‌های نهایی index نورون را برمی‌گرداند، که رقمی است که model پیش‌بینی کرده است.

با توجه به این که بررسی جزئیات data path پرداختیم شکل شماتیک آن را بررسی کنیم.



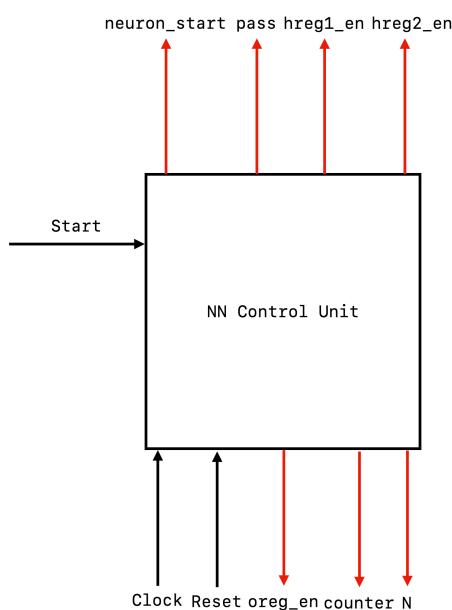
طراحی Control Unit

ابتدا به بررسی FSM واحد کنترل می‌پردازیم. می‌توانیم state-ها را به سه قسمت تبدیل کنیم. قسمت اول گرفتن اطلاعات، پردازش و ذخیره کردن خروجی برای 10 نورون اول لایه hidden، و به همین ترتیب برای 10 نورون دوم لایه hidden و لایه output. این سه قسمت در هر state با فرستادن سیگنال pass مشخص می‌شود. زمانی که ورودی‌های نورون آماده شود سیگنال neuron_start فعال می‌شود و نورون شروع به کار می‌کند. که در قسمت بالا اتفاقاتی که در درون نورون می‌افتد مشخص شده است. سپس در دو state بعدی ورودی‌های مناسب نورون داده می‌شود. و درنهایت زمانی که قسمت پردازی نورون تمام می‌شود و یک سیگنال برای ذخیره کردن خروجی در یکی از سه register داده می‌شود. و این برای دو قسمت دیگر نیز باید طی شود. در شکل بعدی FSM واحد کنترل داده شده است.



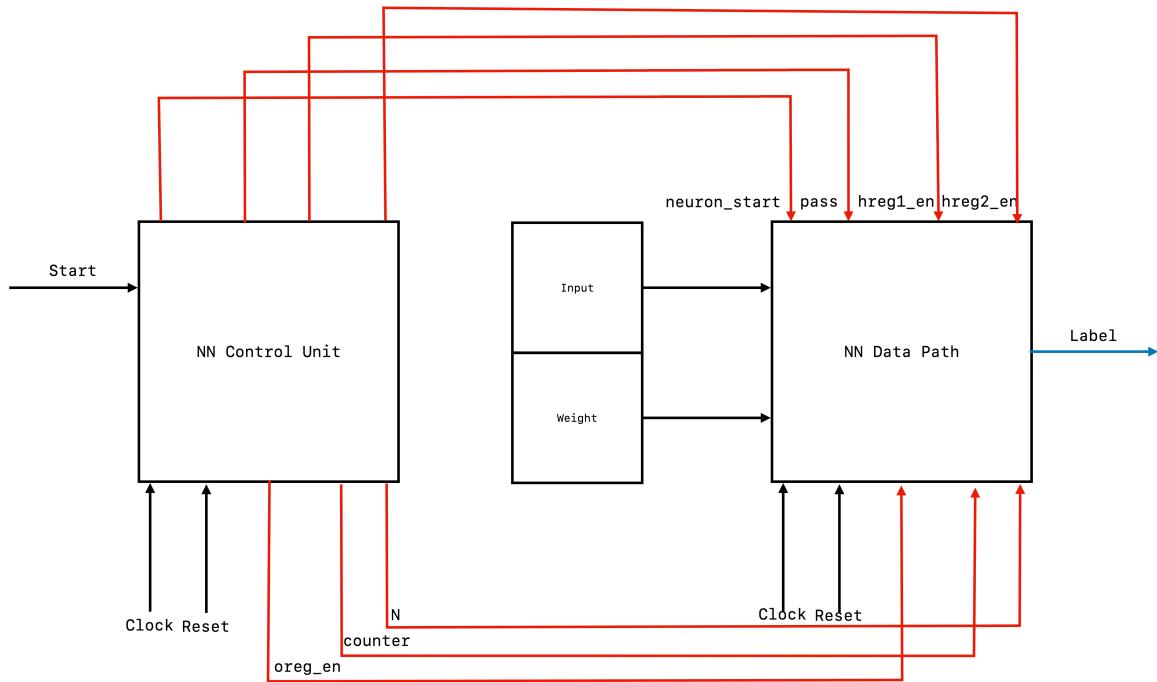
MLP Control Unit FSM

با توجه به این که بررسی FSM پرداختیم شکل شماتیک واحد کنترل را نیز بررسی کنیم.



MLP Control Unit

پس از بررسی control unit و data path برای MLP با وصل کردن این دو قسمت پیاده سازی به پایان می‌رسد.



MLP Schematic Design

بررسی نتیجه testbench

در پس از گرفتن خروجی MLP باید به میزان دقت (accuracy) را بدست آوریم. می‌دانیم دقت از فرمول زیر محاسبه می‌شود.

$$Accuracy = \frac{Correct\ Detected}{Total}$$



پس از پیاده‌سازی مدل و اجرای آن به نتیجه که در شکل مشخص شده است رسیدیم که میزان Correct Detected است.

بنابراین داریم

$$Accuracy = \frac{627}{650} = 0.9026666667 \approx \% 90$$

که نتیجه رضایت بخشی است.

در ادامه مساحت و همچنین تعداد LUTs و DSP blocks و flip flops و ... مشخص شده است.

Device Utilization Summary			
Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	872	93,120	1%
Number used as Flip Flops	580		
Number used as Latches	292		
Number used as Latch-thrus	0		
Number used as AND/OR logics	0		
Number of Slice LUTs	3,066	46,560	6%
Number used as logic	2,719	46,560	5%
Number using O6 output only	1,455		
Number using O5 output only	725		
Number using O5 and O6	539		
Number used as ROM	0		
Number used as Memory	256	16,720	1%
Number used as Dual Port RAM	256		
Number using O6 output only	256		
Number using O5 output only	0		
Number using O5 and O6	0		
Number used as Single Port RAM	0		
Number used as Shift Register	0		

Number used exclusively as route-thrus	91		
Number with same-slice register load	0		
Number with same-slice carry load	91		
Number with other load	0		
Number of occupied Slices	1,025	11,640	8%
Number of LUT Flip Flop pairs used	3,347		
Number with an unused Flip Flop	2,485	3,347	74%
Number with an unused LUT	281	3,347	8%
Number of fully used LUT-FF pairs	581	3,347	17%
Number of unique control sets	48		
Number of slice register sites lost to control set restrictions	144	93,120	1%
Number of bonded IOBs	28	240	11%
Number of RAMB36E1/FIFO36E1s	0	156	0%
Number of RAMB18E1/FIFO18E1s	0	312	0%
Number of BUFG/BUFGCTRLs	2	32	6%
Number used as BUFGs	2		
Number used as BUFGCTRLs	0		
Number of ILOGICE1/ISERDESE1s	0	360	0%
Number of OLOGICE1/OSERDESE1s	0	360	0%
Number of BSCANs	0	4	0%

Number of BUFHCEs	0	72	0%
Number of BUFIODQSS	0	36	0%
Number of BUFRs	0	18	0%
Number of CAPTUREs	0	1	0%
Number of DSP48E1s	11	288	3%
Number of EFUSE_USRs	0	1	0%
Number of FRAME_ECCs	0	1	0%
Number of GTXE1s	0	8	0%
Number of IBUFDS_GTXE1s	0	6	0%
Number of ICAPs	0	2	0%
Number of IDELAYCTRLs	0	9	0%
Number of IODELAYE1s	0	360	0%
Number of MMCM_ADVs	0	6	0%
Number of PCIE_2_0s	0	1	0%
Number of STARTUPs	1	1	100%
Number of SYSMONs	0	1	0%
Number of TEMAC_SINGLES	0	1	0%
Average Fanout of Non-Clock Nets	3.35		

Power Consumption of different parts of MLP

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Device	Virtex6	On-Chip	Power (W)	Used	Available	Utilization (%)		Supply	Summary	Total	Dynamic	Quiescent	
Family	Virtex6	Clocks	0.000	13	--	--		Source	Voltage	Current (A)	Current (A)	Current (A)	
Part	xc6vcx7t	Logic	0.000	3066	46560	7		Vccint	1.000	0.619	0.000	0.619	
Package	ff484	Signals	0.000	3662	--	--		Vccaux	2.500	0.045	0.000	0.045	
Temp Grade	Commercial	DSPs	0.000	11	288	4		Vcco25	2.500	0.001	0.000	0.001	
Process	Typical	IOs	0.000	28	240	12		MGTAVcc	1.000	0.303	0.000	0.303	
Speed Grade	-2	Leakage	1.293					MGTAVt	1.200	0.213	0.000	0.213	
		Total	1.293										
Environment								Supply	Power (W)	Total	Dynamic	Quiescent	
Ambient Temp (C)	50.0									1.293	0.000	1.293	
Use custom TJA?	No												
Custom TJA (C/W)	NA												
Airflow (LFM)	250												
Heat Sink	Medium Profile												
Custom TSA (C/W)	NA												
Board Selection	Medium (10"x10")												
# of Board Layers	8 to 11												
Custom TJB (C/W)	NA												
Board Temperature (C)	NA												
The Power Analysis is up to date.													
(*) Place mouse over the asterisk for more detailed BRAM utilization.													
Table View													

مقدار performance of MLP

$$\left\{ \begin{array}{l} \text{performance} = \frac{1}{\text{Max Delay}} \\ \text{Total REAL to Xst completion : } 686.00 \end{array} \right. \rightarrow \text{performance} = \frac{1}{\text{Max Delay}} = \frac{1}{686.00} = 0.001457725948$$

Clock to Setup on destination clock clk	
Source Clock	Src:Rise Src:Fall Src:Rise Src:Fall
	Dest:Rise Dest:Rise Dest:Fall Dest:Fall
NNController_0/Mram_n01014	1.546
NNDatapath_0/[0].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/[1].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/[2].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/[3].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/[4].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/[5].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/[6].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/[7].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/[8].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/[9].Neuron_0/datapath_0/Register21_0/out_20	1.109
NNDatapath_0/pass[1]_pass[1]_OR_113_o	0.951
clk	5.267

Total REAL time to Xst completion: 686.00 secs
 Total CPU time to Xst completion: 685.87 secs

```

Clock to Setup on destination clock NNController_0/Mram_n01014
-----+-----+-----+-----+
      | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock  |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
-----+-----+-----+-----+
clk          |           |       2.656|           |
-----+-----+-----+-----+


Clock to Setup on destination clock NNDatapath_0/Mmult_n1449_Madd_cy<4>
-----+-----+-----+-----+
      | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock  |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
-----+-----+-----+-----+
NNController_0/Mram_n01014|           |     4.124|           |
clk          |     2.675|           |           |
-----+-----+-----+-----+


Clock to Setup on destination clock NNDatapath_0/[0].Neuron_0/datapath_0/Register21_0/out_20
-----+-----+-----+-----+
      | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock  |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
-----+-----+-----+-----+
clk          |     1.150|           |           |
-----+-----+-----+-----+

```