

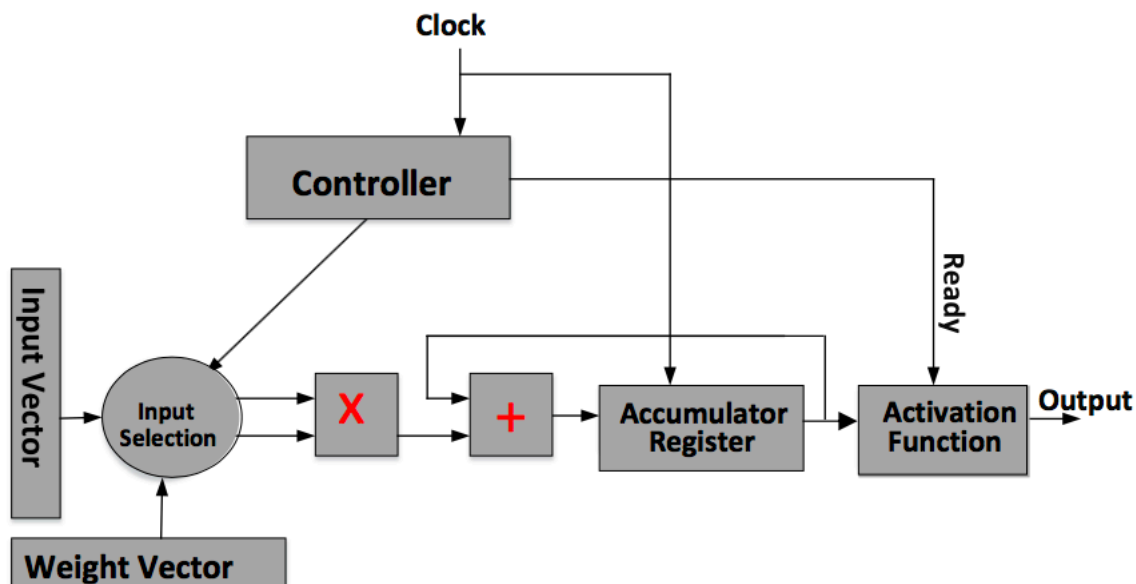
گزارش کار

تمرین کامپیوتری سوم

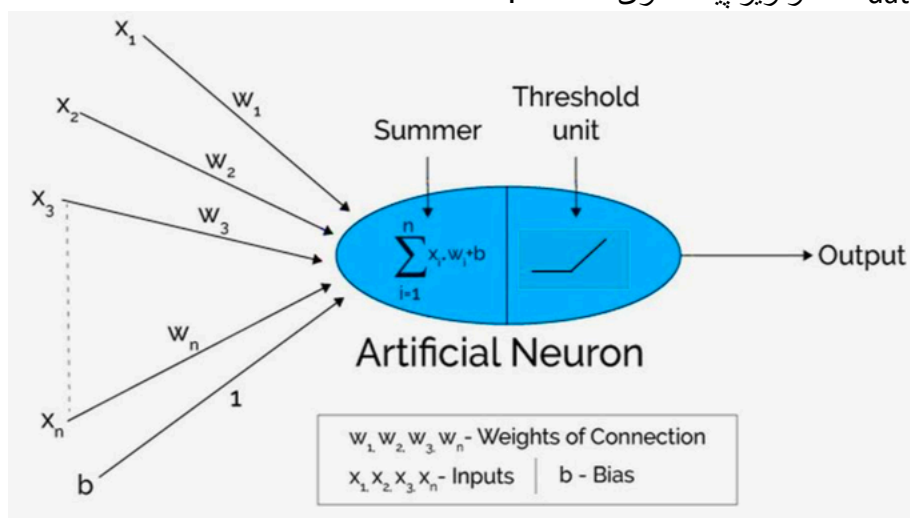
تهیه کنندگان: ملیکه احقاقی، سید علی طباطبایی، کیمیا شریفی

چکیده

در این پروژه هدف پیاده سازی ساختار داخلی یک نورون عصبی به زبان VHDL در فضای Xilinx-Spartan3 است. ساختار داخلی این نورون از دو بخش sequential و combinational به نام neuron_dp و neuron_controller تشکیل شده است:



در قسمت datapath ساختار زیر پیاده سازی شده است:

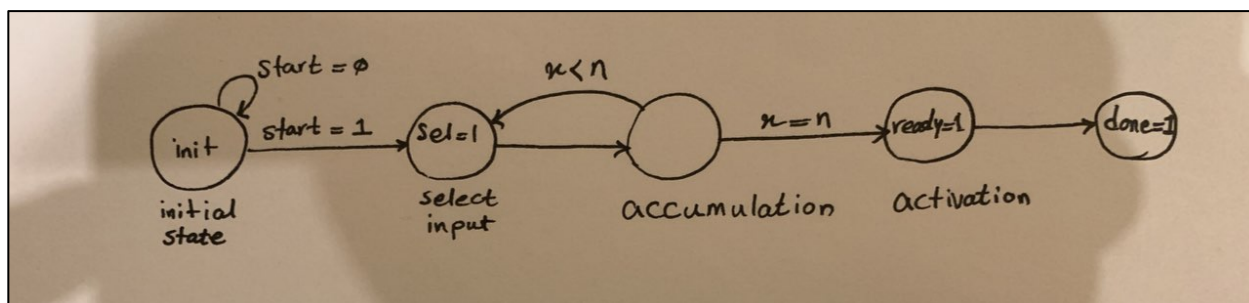


توجه شود که در یک نورون دو بردار ورودی به عنوان بردار ورودی ها و وزن ها داده می شود. بعد از ضرب دوی دوی متناظر از اینها حاصل ضرب ها در یک سیگما جمع شده و پس از عبور از تابع فعال ساز reluz خروجی به شکل زیر به دست خواهد آمد:

$$\text{Activation Function: } f(\text{Output}) = \begin{cases} \text{Output} & \text{Output} > 0.5 \\ 0 & \text{Output} \leq 0.5 \end{cases}$$

برای پیاده سازی این بخش سه واحد اصلی شامل input_selection، MAC و activation_function پیاده سازی شده است که جزئیات مربوط به آن ها در ادامه آمده است.

جهت پیاده سازی بخش کنترلر نیز از FSM زیر استفاده شده است و فرآیند عملکرد نوروں به state های زیر تقسیم شده است:



تنظیمات ISE:

- Family : Spartan 3
- Device : XC3S400
- Package : PQ208
- Speed : -4

واحد کنترلر:

تا زمانی که سیگنال start بیاورد در state اولیه باقی می ماند و سپس با یک شدن سیگنال start وارد مرحله select input خواهد شد که این جا سیگنال خروجی sel فعال می شود و به واحد input_selection در datapath اجازه ی انتخاب ورودی جدید می دهد. و بدون شرطی وارد مرحله ی accumulation می گردد و Id در این مرحله فعال می شود. (توجه شود که در این بخش فرض شده عمل ضرب و جمع متوالی در یک کلاک پایان می یابد) در ادامه چک می شود که آیا تمام ورودی ها و وزن ها دیده شده اند یا خیر. بدین جهت یک upcounter در واحد datapath پیاده سازی شده است که با خواندن هر ورودی یکی افزوده می شود. اگر این عدد یعنی x از n که تعداد ورودی ها است بیشتر شود خرابی نهایی وارد activation_function می شود. بدین جهت سیگنال ready را فعال میکند. در نهایت با فرض این که کار بخش activation در یک کلاک به پایان می رسد در state نهایی خروجی done فعال می شود.

واحد datapath:

input_selection-

در این بخش جهت پارامتری کردن کد از روش unconstraint استفاده شده است به این صورت که در صورت نیاز به طول بردار ها از attribute به صورت a'length استفاده شده است. به عنوان ورودی دو std_logic_vector به صورت یک بعدی داده شده است که هر بار که sel فعال شود به اندازه ی طول ورودی و وزن از بردار یک بعدی می خواند و به x یک واحد اضافه می نماید:

```

sel_in <= inputs((((1 + to_integer(unsigned(count))) * 16 - 1) downto
(to_integer(unsigned(count)) * 16));

```

```
sel_w <= weights(((1 + to_integer(unsigned(count))) * 16 - 1) downto
(to_integer(unsigned(count)) * 16));
```

- multiplication

۱۶ بیت اول حاصل ضرب به عنوان خروجی داده خواهد شد. توجه شود که ضرب علامت دار با کمک casting به فرم زیر انجام شده است:

```
tmp <= signed(a_in) * signed(b_in);
```

- adder

در این بخش نیز دو عدد را به صورت علامتدار sign extend می نماییم و حاصل جمع signed را محاسبه می کنیم. توجه شود که بیت شانزدهم overflow است.

```
tmp <= signed( in1(in1'left) & in1) + signed( in2(in2'left) & in2 );
add_ov <= tmp(n);
add_result <= std_logic_vector(tmp(n-1 downto 0));
```

- accumulation

در واقع یک رجیستر است که در لبه ی بالارونده ورودی را روی خروجی می گذارد که ورودی است حاصل سیگما تا کنون است.

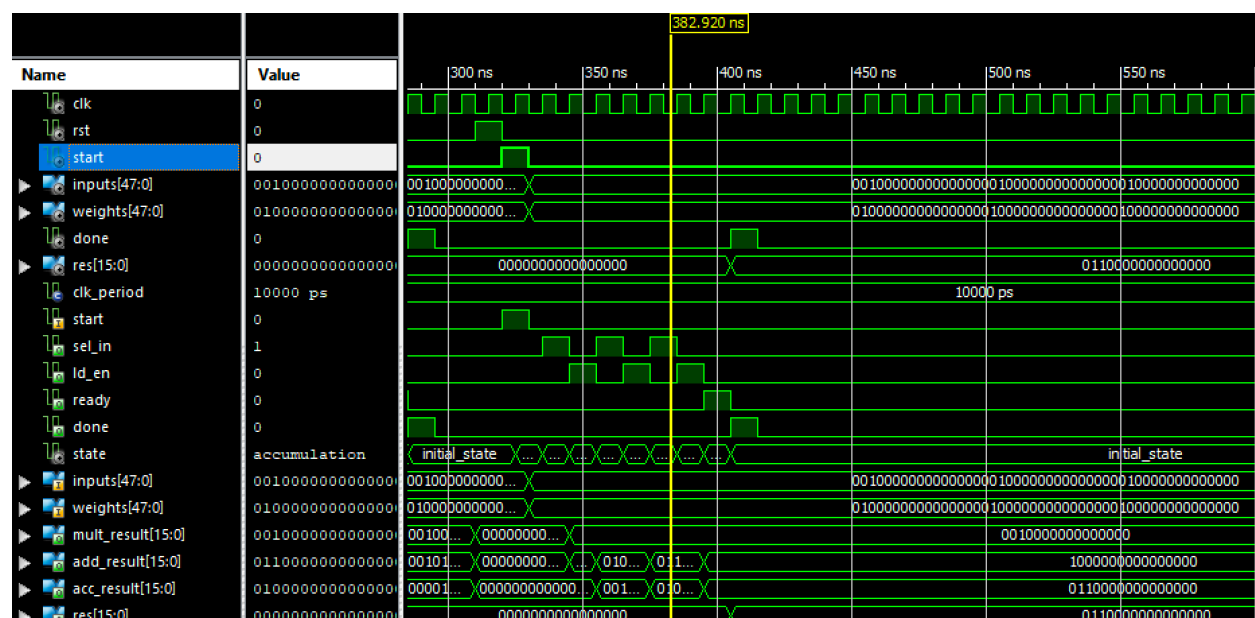
- activation

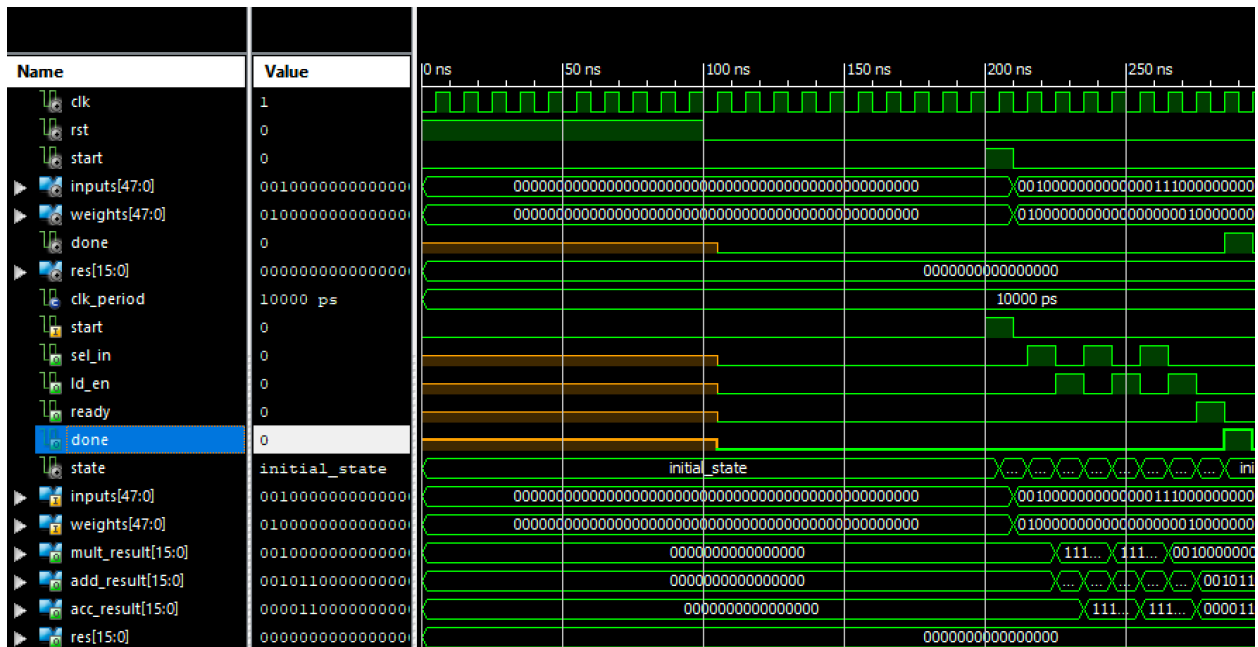
جهت پیاده سازی تابع فعالساز از relu استفاده شده است که در چکیده معرفی شده است. ورودی ها در سیستم ما signed fixed point به صورت 16 بیتی در نظر گرفته شده است که در این صورت نمایش 0.5 به فرم زیر خواهد بود:

```
signal const: signed(15 downto 0) := ("001" & (12 downto 0 => '0'));
```

نتایج تست

خروجی ها در دو حالت کوچکتر از 0.5 و بزرگتر از 0.5 بررسی شده است. در حالت اول $w = [-0.25, 0.125, 1]$ و $in = [1, -0.5, 0.5]$ و همچنین در حالت دوم برابر با $w = [1, 1, 1]$ و $in = [0.5, 0.5, 0.5]$ است. خروجی simulator در ادامه قابل مشاهده است:





neuron_dp Project Status (12/01/2018 - 22:24:15)					
Project File:	CA3.xise	Parser Errors:	No Errors		
Module Name:	neuron	Implementation State:	Synthesized		
Target Device:	xc3s400-4pq208	Errors:	No Errors		
Product Version:	ISE 14.7	Warnings:	7 Warnings (7 new)		
Design Goal:	Balanced	Routing Results:			
Design Strategy:	Xilinx Default (unlocked)	Timing Constraints:			
Environment:	System Settings	Final Timing Score:			

Device Utilization Summary (estimated values)					[+]
Logic Utilization	Used	Available	Utilization		
Number of Slices	91	3584	2%		
Number of Slice Flip Flops	89	7168	1%		
Number of 4 input LUTs	165	7168	2%		
Number of bonded IOBs	116	141	82%		
Number of MULT18X18s	1	16	6%		
Number of GCLKs	1	8	12%		

Detailed Reports						[+]
Report Name	Status	Generated	Errors	Warnings	Infos	
Synthesis Report	Current	Sat Dec 1 22:24:13 2018	0	7 Warnings (7 new)	0	
Translation Report						
Map Report						
Place and Route Report						
Power Report						
Post-PAR Static Timing Report						
Bitgen Report						

Secondary Reports			[+]
Report Name	Status	Generated	
ISIM Simulator Log	Out of Date	Sat Dec 1 22:09:55 2018	

Date Generated: 12/01/2018 - 22:24:15