EE287 Semester Project
Fall 2018
NN3
A simple Neural Network calculation Engine

This semester, you are designing a very simple neural network engine. The engine works with an external three (3) ported memory. The memory has two read ports, and one write port. The engine takes data one sample at a time on an input interface, and when it has collected the required amount of data, it calculates a result, and pushes the result to an output interface. There is some allowed overlap between input, calculation, and output. The design is required to run at 333Mhz.

The interface has the following signals:

Name	**Direction**	**Bits**	**Function**
clk	In	1	Positive clock
rst	In	1	Active High reset
sel	In	1	Read or write to memory
RW	In	1	0=Read, 1=Write operation to memory (qualified by sel)
addr	In	13	Address on memory interface
din	In	32	Memory data from test bench to device
dout	Out	32	Memory data from device to test bench
pushA	In	1	Calculation data push signal to device
stopA	Out	1	Stop signal to test bench
dataA	In	32	Sample data to test bench 8.24 format
firstA	In	1	This is the first sample in a set of inputs
lastA	In	1	This is the last sample in a set of inputs
pushB	Out	1	Result push from device
stopB	In	1	A stop signal from test bench to result portion of device
dataB	Out	32	Resulting data associated with pushB
firstB	Out	1	The first data of a result
lastB	Out	1	The last data of a result
mwadr	Out	13	Memory module address for writing
mwrite	Out	1	Memory write signal

Name	Direction	Bits	Function
mr0	Out	13	Memory read address (port 0)
mr1	Out	13	Memory read address (port 1)
mwdata	Out	32	Data associated with mwrite and mwadr
mrdata0	In	32	Memory data associated with mr0 read
mrdata1	In	32	Memory data associated with mr1 read

The engine waits for memory to be configured, and a set of data to be stored. (pushA and lastA). It then performs a neural calculation, and starts data output. After it goes back and wait for input data to arrive.

The neural operations are described in the memory. Location 0x0000 contains a control register. This register has 3 fields:

```
reg [12:0] input_offset ;
reg [12:0] output_offset ;
reg [5:0] number_outputs ;
```

The input_offset is the location in memory where the first value will be stored. Each subsequent data in goes into the next memory locations. The output_offset is where the output data will be for pushing using pushB. The number_outputs tells the output engine how many data values to push on the output interface.

A list of neuron descriptions begins at location 1. Each description is 2 32 bit words. The two words (64 bits) contain the following data fields:

```
reg [8:0] Reserved ;          // Not used
reg [3:0] neuron_opcode ;     // What to do with the summed data
reg neuron_last ;             // The last neuron in the calculation. Start output after this
reg generating_output ;       // This will generate output data. Wait here if the output machine is busy
reg input_processed ;         // All input have been processed. The input machine can accept more data
reg [7:0] number_inputs ;     // The number of inputs for this neuron calculation
reg [12:0] input_offset ;     // The starting address of calculation data values
reg [12:0] weight_offset ;    // The starting address of calculation weight values
reg [12:0] output_offset ;    // The location in memory to store the result
```

A neuron calculation will not begin until all the input set is received. After that, the first step in calculation is to compute:

$$\sum D_i W_i$$

This is done by reading each data item, and weight using the addresses in the neuron description. D is 32 bits (8.24), and W is 32 bits (8.24). Both are signed. The multiplication will result in a value which is 64 bits or 16.48 signed. This is summed to be 16.48, or 64 bits.

After the multiplication and summing, The neuron opcode is applied.

Opcode 0

Clipped to 32 bits.
Data is scaled by 24 bits, and the result limited to 32 bits. Values outside the range are limited to 32'h7fff_fff0 positive, and -(32'h7fff_fff0) are set to the range limits.

Opcode 1

Sign result.
the output is +1 is the sum is non negative (32'h0100_0000)
the output is -1 if the sum is negative -(32'h0100_0000)

Opcode 2

range +/- 1
the output is scaled by 24 bits, and limited to +/- 1 (32'h0100_0000)

This result is written to memory, and the next calculation is started.

The design is high speed. There must be as much overlap processing as possible. You are only allowed 1 cycle of overhead per each neuron processed once input is available.