***Ram organization :***

Each slot is 256 bits.-

-Ram has 1024 slots

|  |  |  |
| --- | --- | --- |
| Notes | Contents | Address |
| - | Number of layers | 0 |
| For each layer  (loop) | Sx, In.Depth, Out.Depth ,Sx.out, Sx.in, L1.Decay, L2.Decay, Conv/Pool, Number of filters | 1 |
| For each layer  If Conv | Biases Depth, Biases Weight | 2 |
| For each layer  If Conv  \*2 | Filters | 3:n |
| - | In size | M-1 |
| Original Image | Image Rows | M:M+27 |
| And yes they will overwrite the original image | Needed for calculations | M::M+419 |
| - | Num.inputs | M+420 |
| - | 10 biases | M+421 |
| \*1 | Neurons and their weights | M+422 : K |

--The slot of neurons and its 10 weights : (\*1)

|  |  |
| --- | --- |
| Weights | Neuron |
| 160 bits each 16 bits contains one weight | 96 bits (from bit 0 to bit 95) |

Note that we will use only 16 bits to represent one weight.

Notes :

-Filters are N\*N usually N is odd.

-Max Number of filters for each layer is 8

-Max filter size is 5\*5

-Size of filter : 00 for 1\*1 , 01 for 3\*3 , 10 for 5\*5

--The slot (\*2):

|  |
| --- |
| Filter contents |
| Divided into 10 bits slots each represents the value at this position in the filter |

Notes:

-We represent each value in the filter with only 10 bits.

-Bits starts from 0

-Max needed slots are 25 (for the 5\*5 filters)

**Image encoding and Compression:**

- **Input**: image gray level 28\*28.

- **output**: compressed binary text file.

- the image is gray level (0-255) so we needed 8 bits to define the color of the pixel.

- as the data bus is 16 bits, we used the other 8 bits to know how many pixels are similar next to each other.

- we begin calculating the pixels from top left row to top right row then calculate again second top left row to second top right row and so on.

- the compression is done on the byte level.

- the output file each row in it contains 16 bits 0’s and 1’s rows.

**CNN Json encoding: -**

- **Input**: CNN json file.

- **Output**: two files : biString = each row contains 16 bit and each row represents value for each object in the json file.

: idString = identify the meaning of each row in the output biString.

- we take from json file.

1) convolution layer (sx, in\_depth, out\_depth, out\_sx, l1\_decay\_mul, l2\_decay\_mul, pad, layer name, biases {depth, weight ‘w’}, filters weights).

2) pooling layer (sx, in\_depth, out\_depth, out\_sx,in\_sx ”convlayer out\_sx”, pad, layer name).

3) fully connected layer (num\_inputs, biases{weight ‘w’}, filters weights).

- we add also (num of layers, filters count for convolution layer,in\_sx in pool layer which represents out\_sx from convolution layer).

***Design modules for I/O :***

**1-The load enable circuit :**

**-Job :**

To inform the chip that it is in the loading mode as it generates a signal load that remains one during the whole loading .

**- The circuit design :**

**2-The Process enable circuit :** **-Job :** To inform the chip that it is in the Processing mode as it generates a signal process that remains one during the whole processing and this signal is sent to the modules of CNN and FC .

**- The circuit design :**

**3-LDONE Circuit:**

**-Job:**

Generates a load done PULSE to be sent to the cpu.(ORed with the PDone to generate Done )

**- The circuit design :**

NOTE: The upcoming 3 circuits combined are considered as DMA

**4-Address generator :**

**-Job:**

This circuit tells the address to be used by the the ram to read or write. It allows to choose between the next address and an outer address decided by the process modules.

**- The circuit design :**

**5-Ram eanble reading and writing Circuit:**

**-Job :**

This circuit tells if we are to read from or write to the ram.

**- The circuit design :**

**6- Ram control:**

**Job :**

This file contains the 2 MDR registers 1 as input and the other as output as the ram can read and write at the same time. The file also contains the ram of the chip.

**- The circuit design :**

***The I/O module' black box :***

***Determined scenario of the I/O chip :***

1-The cpu MUST send a reset signal to put the chip into idle mode, when ever the reset is set the deevice stops all of its work and return to the starting point.

2-The cpu synchronize the device by sending a clock signal.

3-Once the reset signal is removed the chip is removed the chip will be read to work but it will remain idle till recieving an interrupt.

4-On recieving the interrupt load enable circuit and process enable circuit starts to work to inform the device to enter the loading mode or processing mode (here it will enter the loading image mode).

5-The address generator will indicate the starting address of storing the image (0)

6-The cpu will start sending the compressed image on the 16 bit bus and each clock

Data is decompressed and once we have a whole row of the image ready ,it is stored at input MDR.

7-When the whole row of the image is ready and decompressed it is loaded into ram and then the address is incremented.

8-the previous loading steps are repeated tell the end of the image

9-when the data on data bus are equal to (0000 0000 1010 0001) , the image is loaded and the chip sends a Done pulse to the cpu.

10-Cpu will interupt the chip and send the CNN contents and they will be stored in the ram by the same manner explained above.

11-The chip will send a Done signal to the cpu by the same way.

12-The cpu will raise the process signal but the device will remain idle till recieving the cpu interrupt.

13-On recieving the interrupt the process enable circuit will decide to start processing.

14-The process enable signal is sent to the other modules to start processing.

15-On finishing the processing a Done signal will be sent to the I/O to send it to the CPU and the result is stored at the ram and then I/O sends it to cpu.

***Work load :***

|  |  |  |  |
| --- | --- | --- | --- |
| ***Mohamed Abdel-Aziz*** | ***Mahmoud Youssri*** | ***Khaled Sabry*** | ***Ahmed Salah*** |
| -Load Enable Circuit.  -Process Enable Circuit.  -LDone Circuit.  -DMA  -Address Generator Circuit.  -Ram Enable Reading And Writing Circuit.  -Ram Control Circuit.  -Ram.  -Interfacing with other modules. | -The Decompression Circuit. | -Compression program.  -Studying the jason file.  -Ram Organization with other teams. | -The Decompression Circuit. |