Towards a parametrisable switch for Neuromorphic hardware

Internship report

Lea Kanzleiter

28th September 2018

Contents

1 Introduction **3**

2 Current implementation **4**

2.1 Structure 4

2.2 Data size 4

2.3 Disadvantages of the network 5

3 Improved implementation **6**

3.1 Structure 6

4 Analysis of the network **7**

4.1 Analysis plots 7

4.2 Comparison 7

4.2.1 Low input bandwidth 8

4.2.2 Optimum input bandwidth 9

4.2.3 High input bandwidth 10

5 Summary and outlook **12**

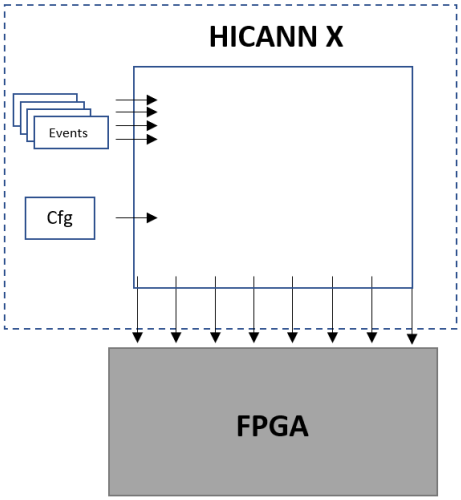


Figure 1: Simplified schematic of a part of the Hicann-X chip

1 Introduction

The image in figure 1 shows a simplified schematic of a part of the HICANN-X chip. 4 channels each connect to a part of the internal neuron blocks and produce spike events. Furthermore, a configuration bus transports slow control data via ARQ. Furthermore there are eight highspeed links from the chip to the FPGA for the communication of the Hicann-X. The events provided by the internal busses need to be distributed evenly to the highspeed links. Those are treated equally as they all link to the same FPGA. Therefore no routing is necessary.
Furthermore the improved network was analysed to quantitatively compare its efficiency to the existing network.

The aim of the project was to build a switch that connects the event busses and the configuration bus to the highspeed links while maintaining a high throughput and low latency. Toward that goal, a software framework was written in Python to visualise and analyse different parametrisations and loads of the switch.

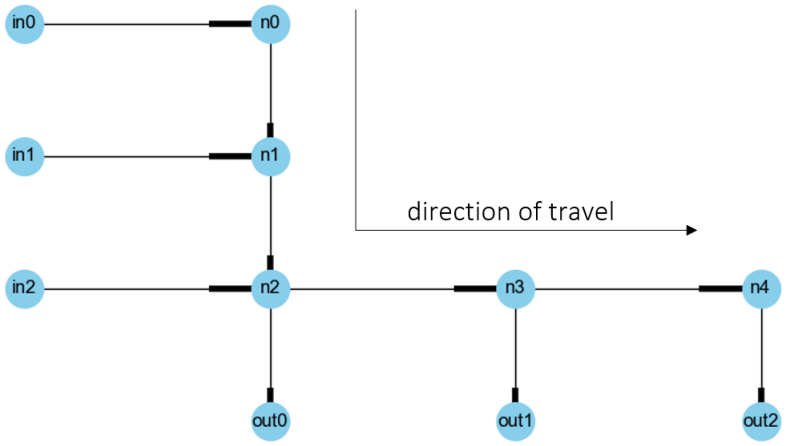


Figure 2: Simplified schematics of the currently implemented network

2 Current implementation

2.1 Structure

Currently, the internal busses of the Hicann-X which are represented by the inputs in figure 2 and the highspeed links which are represented by the outputs are connected via nodes in a chain. Each node works as a buffer and has a maximum of two incoming and two outgoing connections to its neighbors. Using those connections, the data from the inputs can be forwarded either vertically or horizontally through the network. For each node with two incoming connections, the data is received from the two links in an alternating order. This guarantees a fair distribution of data from all inputs. For the nodes that connect directly to the outputs, the priority is always the vertical connection. Only if the corresponding output is busy, the data will be forwarded horizontally to the next output.

2.2 Data size

To represent the length of the data produced at the inputs, each input is assigned a weight. The weights can be different for different inputs which is also the case at the internal busses of the Hicann-X. The weight determines how much time is needed for one output node to process a data package from a certain input. During that time the output is busy and cannot receive any more data. As each output can process a weight of w=1 per clock, the total bandwidth of the outgoing side of the network equals the number of links.

2.3 Disadvantages of the network

Observing the currently implemented network, there are a few problems concerning the equal use of all the highspeed links/outputs and concerning the throughput. In the network in figure 2 the connection between node n_1 and n_2 creates a bottleneck. All the data from inputs in_0 and in_1 need to pass through this connection to reach the outputs. If the output out_0 which is directly connected to n_2 is busy, the data will be forwarded horizontally in the next clock. As soon as the output out_0 switches to not being busy anymore, new data can be received. This causes that the maximum number of output nodes that are used equals the biggest weight of the data. That means the number of used outputs is limited as not all the output nodes can be reached, causing a stalling effect which leads to inputs not being able to insert their data into the network. The throughput in this case is very low, even though the output connections are not saturated.

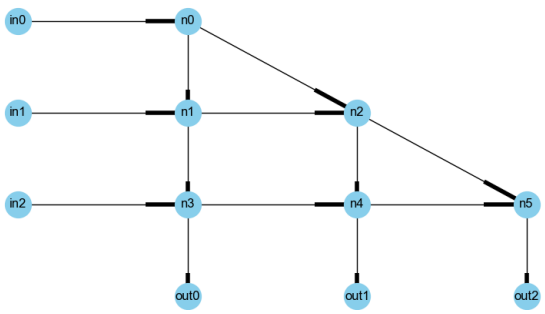


Figure 3: Simplified schematics of the improved network structure

3 Improved implementation

3.1 Structure

The improved version of the network is based on the same principle of the nodes as described above. The new network is parametrised by the number of inputs, outputs and nodes that should be used. For that, the nodes are added to the structure of the currently implemented version diagonally from the bottom left, so a triangular shape is formed as in figure 3. The additional connections allow the data to distribute evenly to all output nodes, so the bandwidth of the output links can be used fully. This also prevents the inputs from stalling and more data can be inserted in the network per clock. Especially for high input rates this means a big improvement in throughput.

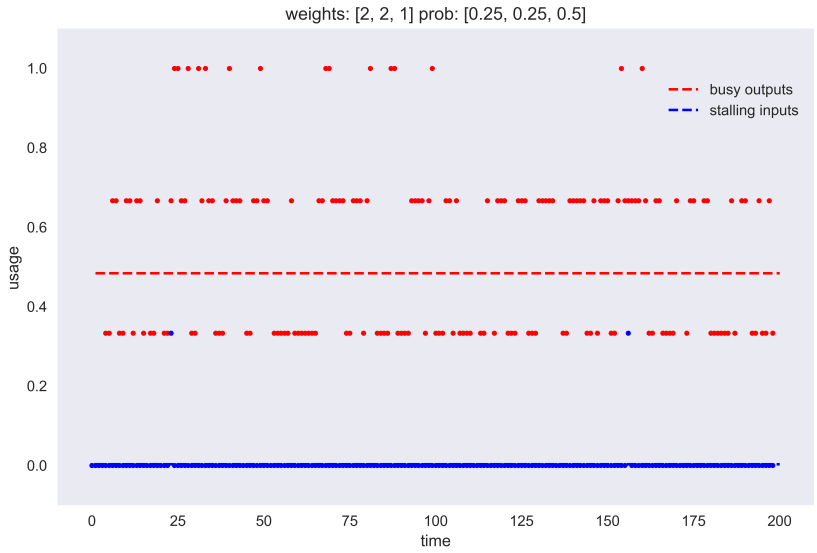


Figure 4: Example analysis plot for figure 3

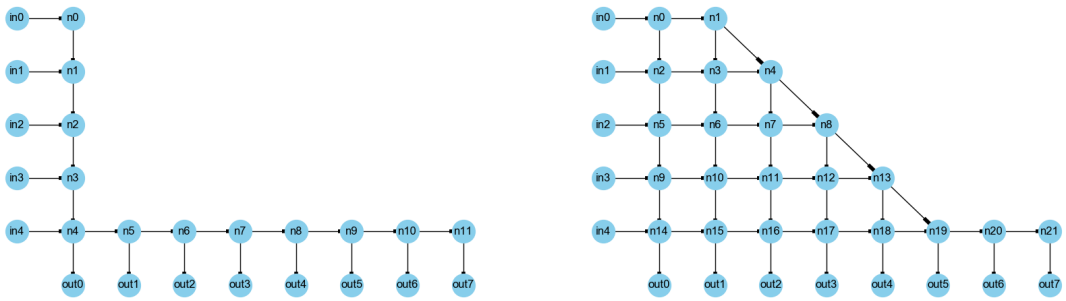
4 Analysis of the network

4.1 Analysis plots

To quantitatively analyse the networks and allow a comparison between the current solution and the improved version, the usage of the nodes was observed over time. First, the weights for each input were determined and for a duration of 200 clocks, data was randomly generated at the inputs with a certain probability. The weights and probabilities were chosen so that the total amount of data produced at the inputs was either smaller, equal or bigger than the total bandwidth at the output. For each time step, the relative number of busy outputs was calculated which is marked with a red dot in the plot in figure 4. Also the blue dots represent the relative number of busy inputs that cannot insert their data into the network. For both quantities a mean value was calculated as well, which is plotted as a dashed line. The aim for the improved system is to keep the blue mean value as low as possible and the red mean value as high as possible, so as many outputs as possible are used while as much data as possible is inserted into the network.

4.2 Comparison

To simulate the real situation on the Hicann-X, the time course was observed for the currently implemented and the improved network using five inputs and eight outputs.



(a) Current network (b) Improved network

Figure 5: Comparison of networks for the Hicann-X chip

The inputs represent the event busses and the configuration bus and the outputs repere-sent the highspeed links to the FPGA. For the improved version, 22 nodes were used to create the network as shown in figure 5b. In the following, three cases were simulated. In the first section, the mean input bandwidth N_{in} is smaller than the mean output bandwidth $N_{out} = 8$, in the second case N_{in} equals $N_{out} = 8$ and in the third case N_{in} is bigger than $N_{out} = 8$. The input bandwidth is calculated as follows:

$$N_{in} = \sum_{i=0}^{n_{in}-1} w_i \cdot p_i \tag{1}$$

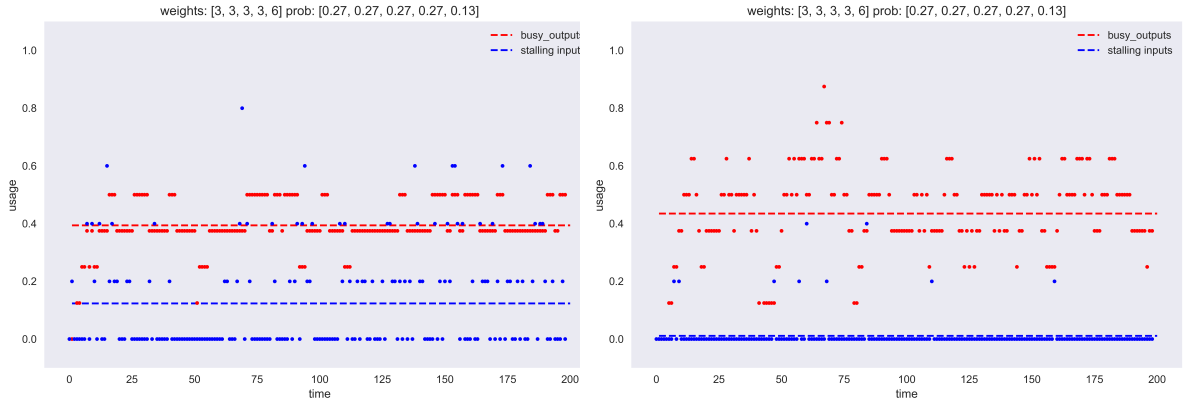
Here, w_i is the weight of input i, p_i the probability for input i and n_{in} the total number of inputs.

4.2.1 Low input bandwidth

For the experiment in this part, the mean input bandwidth N_{in} was lower than the output bandwidth with:

$$N_{in} = 4.02 < N_{out} = 8 \tag{2}$$

The plot in figure 6a for the current solution shows a lot of jitter in the relative number of stalling inputs. This is caused by the blocking connection between nodes n_3 and n_4, as described above. Also, the relative number of busy outputs is at 40%, which means less than half of the outputs is used for most of the time. In comparison to that, figure 6b shows a big improvement for the stalling inputs. The mean value for that is roughly at zero, as only 4.5% of the data points differ from zero which means that inputs could not insert their data into the network. This low mean value justifies the little gain for the mean value of busy output nodes. As the input bandwidth is very small, it is not

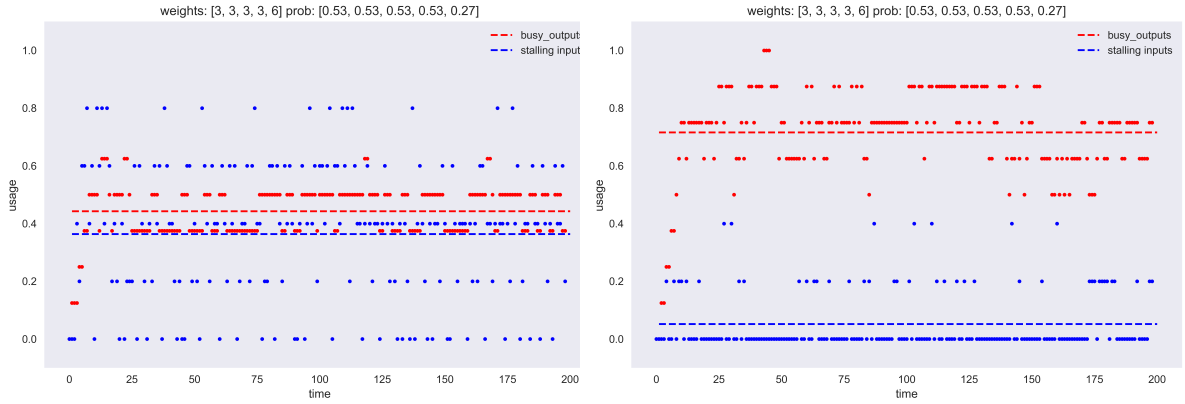


(a) Current network (b) Improved network

Figure 6: Time series for low input bandwidth

expected that all output nodes are constantly used, as there is not enough data inserted into the system per clock to satisfy all eight links.

4.2.2 Optimum input bandwidth



(a) Current network (b) Improved network

Figure 7: Time series for optimum input bandwidth

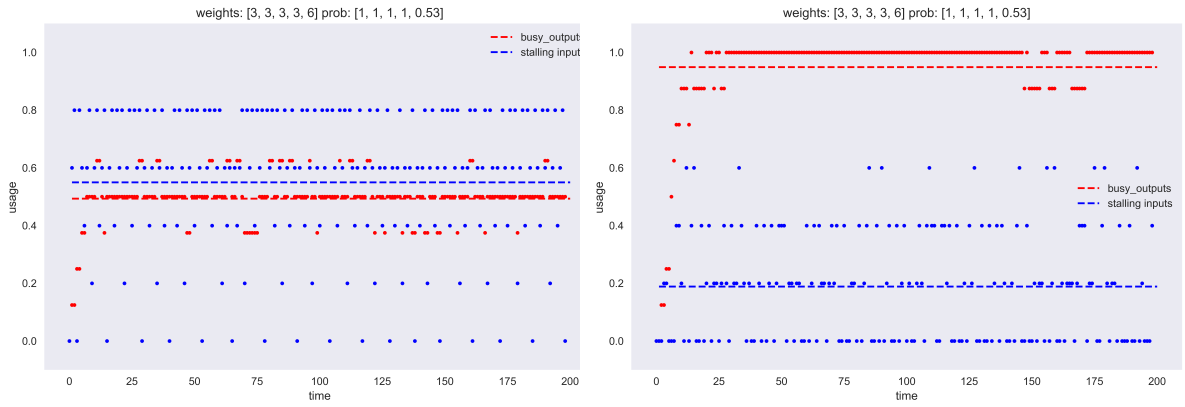
For the second comparison the input bandwidth was:

$$N_{in} = 7.98 \approx N_{out} = 8 \tag{3}$$

In the plot for the smaller network in figure 7a a huge jitter for the stalling inputs can be observed, where sometimes even 80% of the inputs are busy. The mean value for the

stalling inputs more than doubled compared to the low bandwidth, whereas the mean value for the busy outputs only increased a little. For the improved network in figure 7b the increase for the mean value of the inputs was not as dramatic, as for most of the time all of the inputs could insert their data into the network. On the other hand, the mean value for the busy outputs increased a lot. This shows the improvement as almost none of the inputs stall while a big percentage of outputs is used.

4.2.3 High input bandwidth



(a) Current network (b) Improved network

Figure 8: Time series for high input bandwidth

The last comparison is for an input bandwidth of:

$$N_{in} = 15.18 > N_{out} = 8 \tag{4}$$

The plot in figure 8a shows that only 9% of the experiment duration none of the inputs stalled, while most of the time 60% or 80% of the inputs stalled. The mean value also increased compared to the experiment with optimum input bandwidth. For the usage of the output nodes, the mean value again only increased a little, but is now also lower than the mean value for the inputs which should not happen. For the improved network on the other hand, the mean value for the outputs is now very close to 100%, which means that the full bandwidth is used most of the time. While this is the case, the jitter for the stalling inputs is fine, as there would be no free outputs to receive the data. So the increase for the mean input value is justified.
In general, the smaller network performs worse for an increase in input bandwidth. The mean value for stalling inputs increases a lot while for the mean output value there is

no big increase possible as explained above. For the improved version of the network, the increase in usage of the outputs is significant and the increase for the stalling inputs fits the expectations.

5 Summary and outlook

Overall, during the internship a new, improved solution for the connection of the internal busses of the Hicann-X chip to the highspeed links to the FPGA was successfully designed. In that process two network structures were tested and the more efficient one was presented in this report. The network is parametrisable, as the software designs a structure for an arbitrary number of inputs, outputs and nodes. For the visualisation and analysis of the solution, a software framework was built using the NetworkX library in Python. In the following Bachelor's thesis the network will be implemented in RTL. The aim is to keep it parametrisable to enable the usage for different purposes and maintain flexibility. Furthermore, the number of possible incoming or outgoing connections per node will be increased to be able to reduce the number of nodes that are necessary. Allowing connections between nodes in different layers of the network helps to reduce the latency and the occupied space on the FPGA.