ISCI, Volume *19*

Supplemental Information

A Neuromorphic Prosthesis
to Restore Communication
in Neuronal Networks

Stefano Buccelli, Yannick Bornat, Ilaria Colombi, Matthieu Ambroise, Laura Martines, Valentina Pasquale, Marta Bisio, Jacopo Tessadori, Przemysław Nowak, Filippo Grassia, Alberto Averna, Mariateresa Tedesco, Paolo Bonifazi, Francesco Difato, Paolo Massobrio, Timothée Levi, and Michela Chiappalone

Supplemental Information

Supplemental Figures

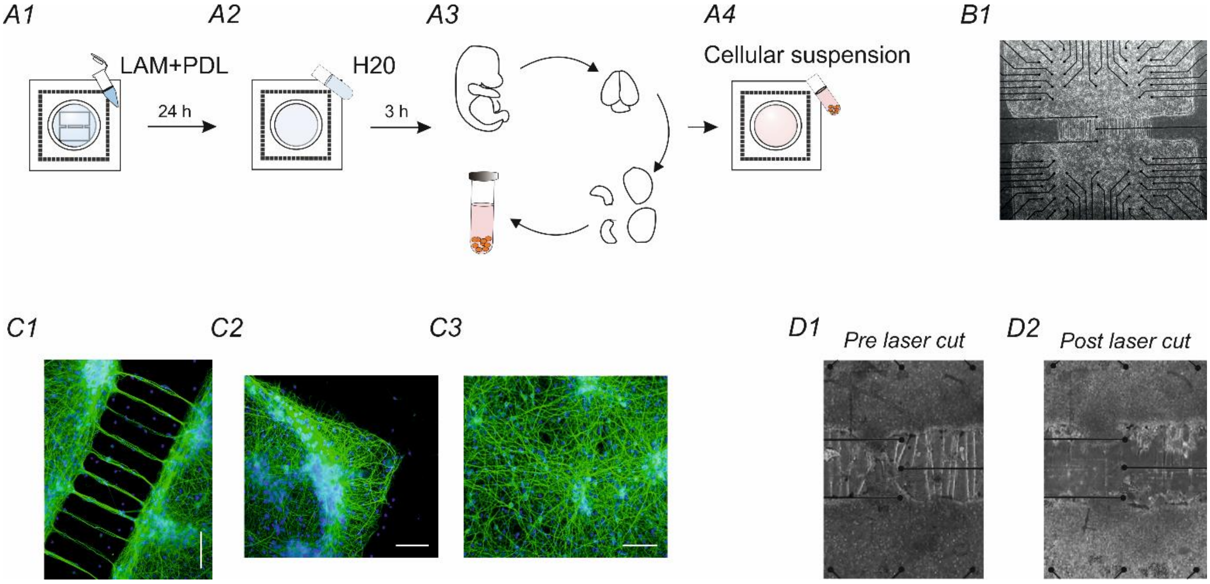


Figure S1. Bimodular neuronal network preparation steps and morphological characterization, Related to Figure 2. A1 MEA surface coating using poly-D-lysine (PDL) and laminin (LAM) to promote cell adhesion. **A2** Three consecutive surface rinsing steps using sterilized water and 3 hours of waiting before plating cells. **A3** Dissection and dissociation procedures of E18 rat neocortex followed by resuspension of the resulting tissue in Neurobasal medium (see 'Methods'). **A4** Cellular solution plating onto a 60-channel MEA surface. **B1** A 60-channel MEA (Multichannel Systems MCS, Reutlingen, Germany) with square layout (4Q) plated with a bimodular pattern. **C1** Immunofluorescence micrograph of MAP2 of a representative bimodular culture on a coverslip at DIV 25 with magnification of the connections between modules. **C2**, **C3** Immunofluorescence micrographs of MAP2 of a representative uniform culture and of one corner on a coverslip at DIV 25. Scale bar = 200 µm. **D1** Magnification of the connections between the two modules before performing the laser cut. **D2** Magnification of the cut connections between the two modules.

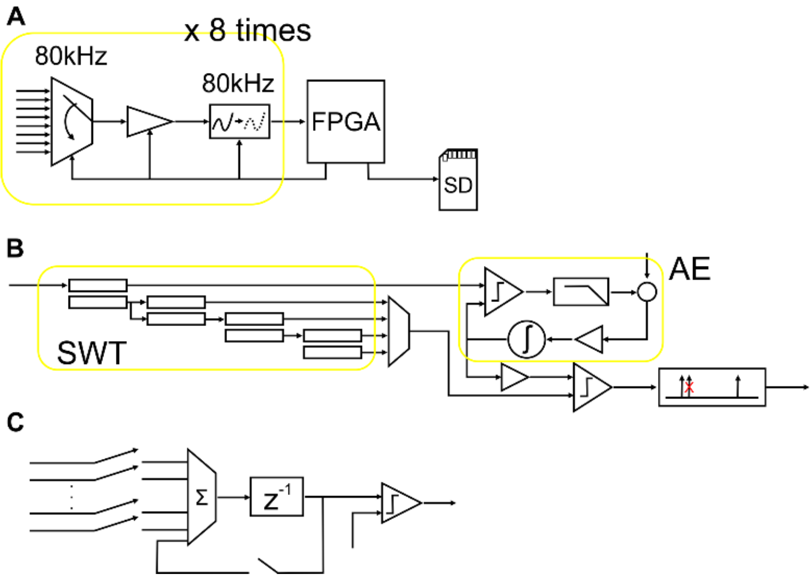


Figure S2. Schematic view of the hardware computation architecture, Related to Figure 1. A, Integrated circuit data flow on the device (Multiplexer: ADG1408, amplifier: LTC6912, analogue to digital converter: AD7686, FPGA: XC6SLX150). Each flow makes it possible to sample 8 channels at 10 kHz (80 kHz multiplexing), eight equivalent structures are connected to the FPGA to sample 64 inputs, and 60 of these inputs are connected to the MEA. **B**, Hardware spike detection architecture embedded in the FPGA. SWT represents the wavelet-based signal enhancement (stationary wavelet transform). The SWT consists of a series of FIR filters. The first two filters (on the left) are on the 8th order, and the following filters are on the 16th order with 8 null coefficients. AE represents the signal amplitude estimator used to set the final threshold. The loop estimates the standard deviation as the value above 15.9% of samples. The comparator and lowpass filter measure the ratio of samples above the estimated standard deviation; then, the target value is subtracted to identify the relative error. The regulator is composed of an amplifier and integrator to update the estimation of the standard deviation. After the final threshold, an event filter drops each second event too close to the previous one. **C**, Hardware network burst detector, incoming sources are events produced by the spike detector or the ISI-based burst detector, and the event count is stored in the accumulator (z^-1) and cleared according a timer (the accumulator is not actually cleared, but its value is updated without considering the number of accumulated events thus far, which results in data clearance). The number of events counted since the beginning of the time window is compared with the user-defined threshold to decide whether the network is in a burst state.

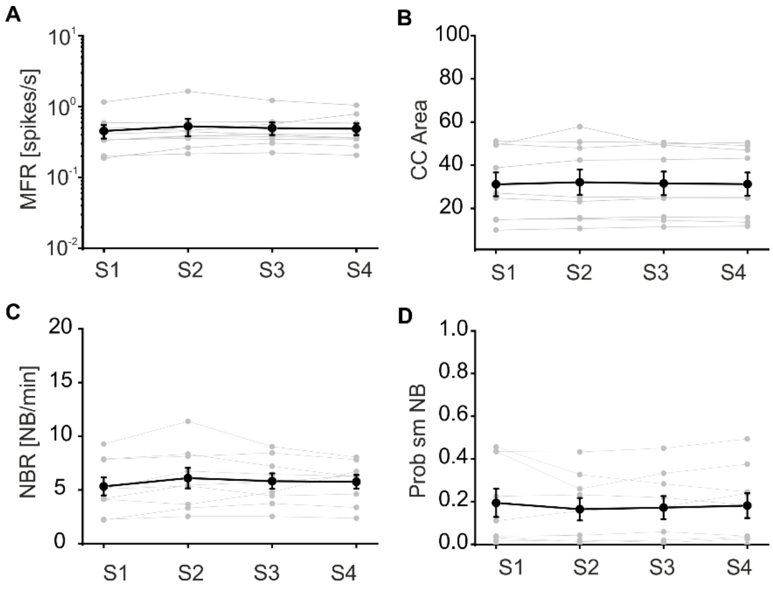


Figure S3. Control no lesion, Related to Figure 2. A, Mean firing rate (MFR) was stable for all control with no lesion experiments for all experimental phases (from S1 to S4). No significant difference was found (Friedman’s repeated measures analysis of variance on ranks; n=9; p=0.04, DF=3, Chi-square= 8,333 but no significant difference between rank sums was found by the Tukey test). **B**, Cross-correlation (CC) area (obtained integrating the CC function of the collapsed spike trains from module 1 and 2 in a range of ± 500 ms) was stable for all control with no lesion experiments for all experimental phases (from S1 to S4). No significant difference was found (Friedman’s repeated measures analysis of variance on ranks; n=9; p=0.833, DF=3, Chi-square= 0.867). **C**, Network burst rate (NBR) was stable for all control with no lesion experiments for all experimental phases (from S1 to S4). No significant difference was found (one-way repeated measures analysis of variance. n=9; p=0.308, DF=3, F= 1.267). **D**, Probability of single-module network burst (Prob smNB) was stable for all control with no lesion experiments for all experimental phases (from S1 to S4). No significant difference was found (Friedman’s repeated measures analysis of variance on ranks; n=9; p=0.789, DF=3, Chi-square= 1.050).

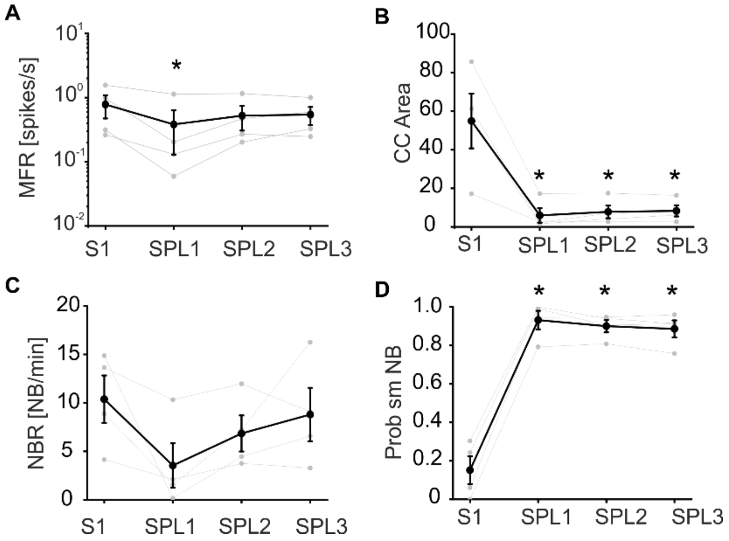


Figure S4. Control with lesion, Related to Figure 2. A, Mean firing rate (MFR) was almost stable for all control with lesion experiments. The only significant difference was found between the first spontaneous phase (S1) before the lesion and the first spontaneous phase post lesion (SPL1). Starting from the second hour post lesion, no significant difference was found between SPL2 and S1 and SPL3 and S1 (one-way repeated measures analysis of variance. n=4; p=0.037, DF=3, F= 4,386; all pairwise multiple comparison procedures (Bonferroni t-test): S1 vs. SPL1: p=0.036; S1 vs. SPL2: p=0,282; S1 vs. SPL3: p=0.376; SPL3 vs. SPL1: p=1; SPL3 vs. SPL2: p=1; SPL2 vs. SPL1: p=1). **B**, Cross-correlation (CC) area (obtained integrating the CC function of the collapsed spike trains from modules 1 and 2 in a range of ± 500 ms) collapsed following laser ablation. We found significant differences between all phases post lesion and the pre-lesion phase (one-way repeated measures analysis of variance. n=4; p<0.001, DF=3, F= 16,555; All Pairwise Multiple Comparison Procedures (Bonferroni t-test): S1 vs. SPL1: p=0.001; S1 vs. SPL2: p=0.002; S1 vs. SPL3: p=0.002; SPL3 vs. SPL1: p=1; SPL3 vs. SPL2: p=1; SPL2 vs. SPL1: p=1). **C**, Network burst rate (NBR) was stable for all control with lesion experiments (from S1 to SPL3). No significant difference was found (one-way repeated measures analysis of variance. n=4; p=0.142, DF=3, F= 2.331). **d**, Probability of single-module network burst (Prob smNB) before the lesion (S1) was on average close to 0.2, meaning that the majority of NBs involved both modules. Following the lesion, the probability increased to an average value higher than 0.85, meaning that the large majority of NBs involved only one module or the other. We found significant differences between SPL1 and S1 (one-way repeated measures analysis of variance. n=4; p<0.001, DF=3, F= 109.911; all pairwise multiple comparison procedures (Bonferroni t-test): SPL1 vs. S1: p<0.001; SPL1 vs. SPL3: p=1; SPL1 vs. SPL2: p=1; SPL2 vs. S1: p<0.001; SPL2 vs. SPL3: p=1; SPL3 vs. S1: p<0.001).

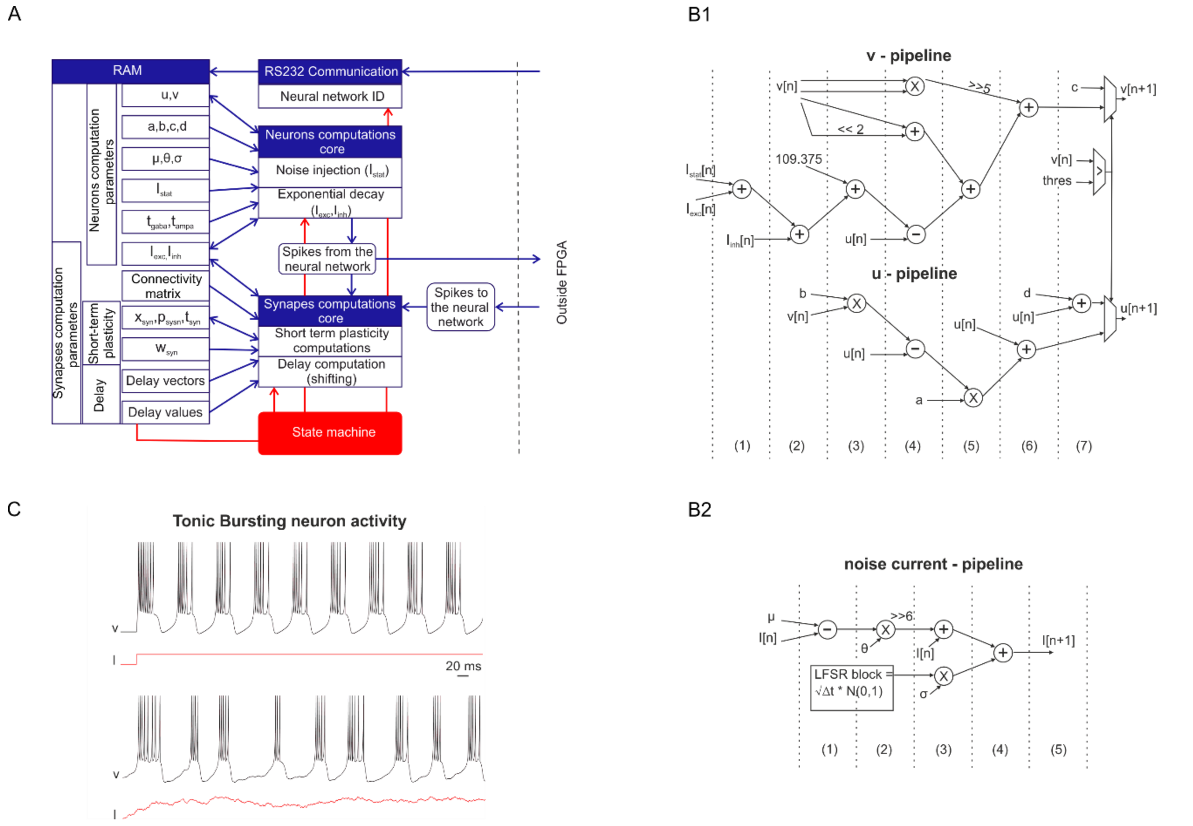


Figure S5. Hardware system organization, pipeline implementation and synaptic noise results, Related to Figure 4. A, Organization of the SNN system with two computation cores, the neuron and synapse. A state machine allows communication between the different blocks. The RAM stacks all the parameters and updates all values every 1 ms. RS232 communication allows communication with a computer to send the initial neural network topology. Spikes to the neural network block are for hybrid experiments where all detected spikes from the neuron culture are sent to the SNN via external synapses. **B1, B2,** Architecture of ‘v’, and ‘w’ and ‘I’ pipelines for digital implementation. The computation cycles are separated by dotted lines. Each of these dotted lines represents a rising clock edge on which the result of each operation is saved. For the sake of clarity, we have not shown the sequences of the flip-flops. ‘u’ and ‘v’ need 7 clock cycles (140 ns), and ‘I’ needs 5 clock cycles (100 ns). The synaptic noise current ‘I’ is modelled using an Ornstein-Uhlenbeck process, which makes the neuron implementation more biologically plausible, as shown in C. The parameter μ represents the equilibrium or mean value for the process. We set this value equal to 0 with a bias current value necessary for tonic bursting activity without noise. For the other parameters, σ represents the degree of volatility around the mean value caused by shocks, and θ represents the rate by which these shocks dissipate and the variable reverts towards the mean. The stationary variance depends on that parameter. Therefore, in our case, we can set the stationary (long-term) variance with σ and θ parameters (neuron bursting activity) of 35 and 1, respectively. **C,** SNN output of the tonic bursting neuron in which the parameters are set to obtain tonic bursting activity with and without synaptic noise. The synaptic noise allows for better biomimetic dynamics.

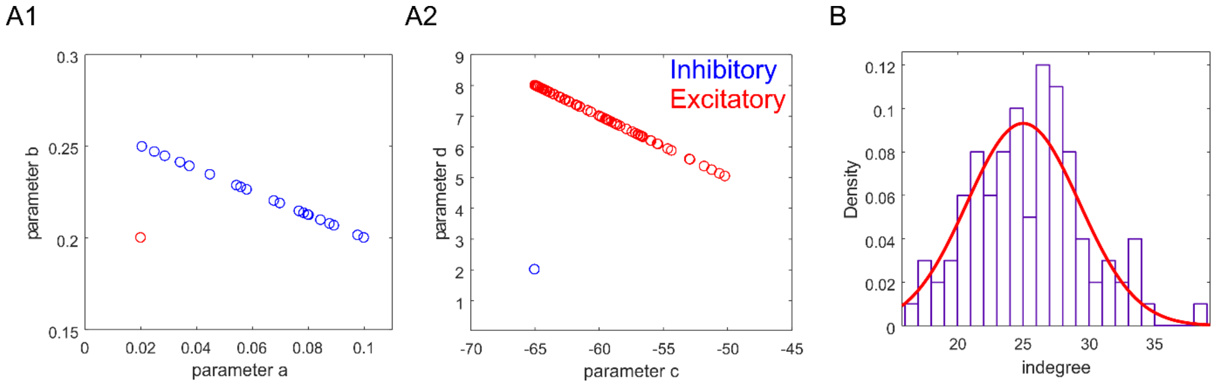


Figure S6. SNN model parameters, Related to Figure 4. A1, Parameters 'a' and 'b' of the Izhikevich model (see Methods) for excitatory (red circle) and inhibitory (blue circles) neurons. All excitatory neurons have the same parameter a=0.02 and b=0.2. All inhibitory neurons have different parameters a (from 0.02 to 0.1) and b (from 0.2 to 0.25). **A2**, Parameters 'c' and 'd' of the Izhikevich model for excitatory (red circles) and inhibitory (blue circles) neurons. All inhibitory neurons have the same parameter c=-65 and d=2. All excitatory neurons have different parameters c (from -64.97 to -50.17) and d (from 5.04 to 7.99). **B,** Distribution of the indegree (i.e., the number of pre-synaptic neurons) with a Gaussian distribution superimposed (red curve). Mean value = 25, standard deviation = 4.3, R-Square = 0.806 (fit converged, Chi-square tolerance value of 1e-9 was reached).

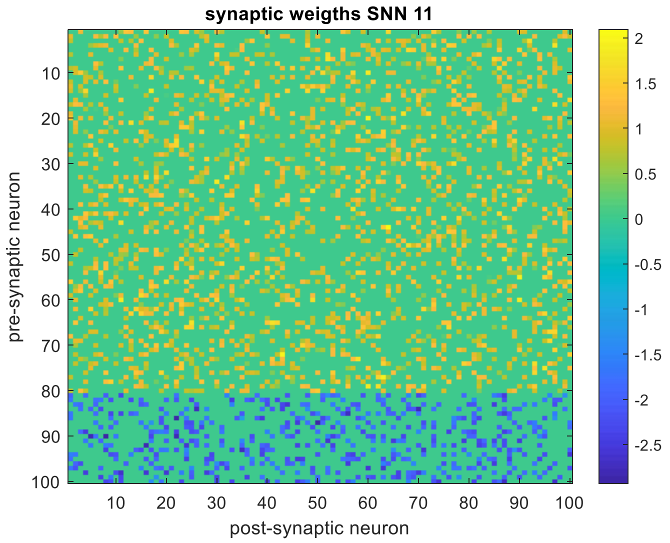
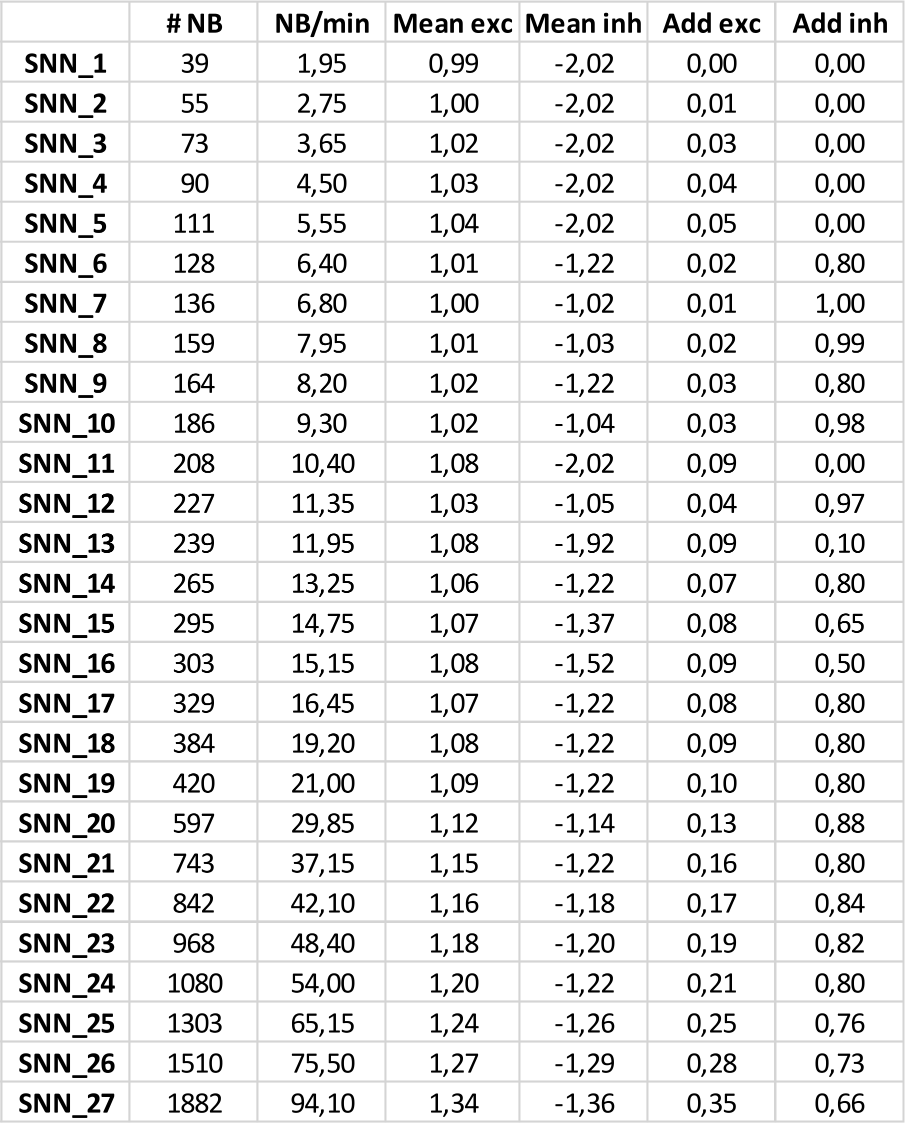


Figure S7. SNN connectivity matrix, Related to Figure 4. The matrix represents the connectivity matrix of one of the 27 SNNs (SNN 11) belonging to the SNN library. Pre-synaptic neurons are on the y-axis and post-synaptic neurons are on the x-axis. Each element of the matrix represents the synaptic weight between the pre- and post-synaptic neuron. If the pre-synaptic neuron is excitatory (i.e. one of the first 80 neurons), the synaptic weight will be positive. Conversely, if the pre-synaptic neuron is inhibitory (i.e. one of the last 20 neurons), the synaptic weight will be negative. We set the outdegree (i.e., the number of post-synaptic neurons) to 25 for all neurons in the network, while the indegree (i.e., the number of pre-synaptic neurons) followed a normal distribution with a mean value of 25 and a standard deviation of 4.3 (Figure S6, R-Square=0.806).

Supplemental Tables



	# NB	NB/min	Mean exc	Mean inh	Add exc	Add inh
SNN_1	39	1,95	0,99	-2,02	0,00	0,00
SNN_2	55	2,75	1,00	-2,02	0,01	0,00
SNN_3	73	3,65	1,02	-2,02	0,03	0,00
SNN_4	90	4,50	1,03	-2,02	0,04	0,00
SNN_5	111	5,55	1,04	-2,02	0,05	0,00
SNN_6	128	6,40	1,01	-1,22	0,02	0,80
SNN_7	136	6,80	1,00	-1,02	0,01	1,00
SNN_8	159	7,95	1,01	-1,03	0,02	0,99
SNN_9	164	8,20	1,02	-1,22	0,03	0,80
SNN_10	186	9,30	1,02	-1,04	0,03	0,98
SNN_11	208	10,40	1,08	-2,02	0,09	0,00
SNN_12	227	11,35	1,03	-1,05	0,04	0,97
SNN_13	239	11,95	1,08	-1,92	0,09	0,10
SNN_14	265	13,25	1,06	-1,22	0,07	0,80
SNN_15	295	14,75	1,07	-1,37	0,08	0,65
SNN_16	303	15,15	1,08	-1,52	0,09	0,50
SNN_17	329	16,45	1,07	-1,22	0,08	0,80
SNN_18	384	19,20	1,08	-1,22	0,09	0,80
SNN_19	420	21,00	1,09	-1,22	0,10	0,80
SNN_20	597	29,85	1,12	-1,14	0,13	0,88
SNN_21	743	37,15	1,15	-1,22	0,16	0,80
SNN_22	842	42,10	1,16	-1,18	0,17	0,84
SNN_23	968	48,40	1,18	-1,20	0,19	0,82
SNN_24	1080	54,00	1,20	-1,22	0,21	0,80
SNN_25	1303	65,15	1,24	-1,26	0,25	0,76
SNN_26	1510	75,50	1,27	-1,29	0,28	0,73
SNN_27	1882	94,10	1,34	-1,36	0,35	0,66

Table S1. Library of SNN, Related to Figure 4. SNN database was created to cover a wide range of NBRs from 1.95 NB/minute (SNN 1) to 94.1 NB/minute (SNN 27). To obtain such variability, we kept the same connectivity matrix with an average degree of 25% for all networks. The main parameters that we tuned to obtain such variability were the mean synaptic excitatory and inhibitory weights. Starting with SNN 1, we added a variable number to each synapse, maintaining separated excitatory (add exc) and inhibitory synapses (add inh). This SNN database can provide a wide range of activities and allows for a better choice of SNN for HBB experiments.

	bit registers			LUTs			RAM blocks			multipliers		
	numb.	Project	Device	numb.	Project	Device	numb.	Project	Device	numb.	Project	Device
Signal processing	4753	31%	3%	6274	31%	7%	11	5%	4%	3	7%	2%
SNN	3433	23%	2%	5747	29%	6%	82	39%	31%	42	93%	23%
Environment	6993	46%	4%	7996	40%	9%	118	56%	44%	0	0%	0%
Total	15179	100%	8%	20017	100%	22%	211	100%	79%	45	100%	25%

Table S2. Hardware resource usage for the different functions of the hardware board, **Related to Figure 4.** The signal processing part includes filters, spike detection, and burst detection (for events from electrodes and from the SNN). The SNN is considered a part by itself, and the environment part includes the necessary engineering to interface the system to the real world. This part includes analogue front-end management, computer interface, VGA display management and SD-card storage management. Bit registers (referenced as flip-flops in digital computing) are used to synchronously store local values or module states. Look-Up tables (LUTs) are asynchronous devices used to implement arbitrary 6-input logic gates. Random Access Memory (RAM) blocks are mostly used to store indexed values (e.g., channel samples for filters, and waiting queues). Multipliers are hard-coded embedded multiplier circuits that are available to increase both performance and density since multiplications are resource-demanding functions in hardware. The high majority of RAM blocks are required by the environment for VGA display and SD-card data buffer. The SNN part uses RAM blocks extensively to address the individual parameters of neurons and synapses. The SNN also requires many hardware multipliers because of the original mathematical formalism. Resources for signal processing are more focused on bit registers and LUTs because of pipeline structures that implement computations structures close to the device that stores data. The environmental cost in terms of resources is heavy, but this cost is constant and should not increase if implementing new signal processing features.

Transparent Methods

Bimodular confinement

The polymeric structure for the physical confinement of neuronal cultures on conventional MEAs has been realized in polydimethylsiloxane (PDMS) by soft lithography and by using a photolithographically defined EPON SU-8 master on a silicon substrate. PDMS is an elastomer widely used for biomedical applications because of its high temperature, chemical and oxidation resistance; biocompatibility; transparency; and permeability to gases (Mata et al. 2005). Additionally, PDMS is an electrical insulator and nontoxic. These features also render PDMS particularly suited to the culture of primary neurons (Taylor and Jeon 2010). In addition, PDMS can be easily micro-structured by soft lithography, thus obtaining several low-cost replicas from a single master (Habibey et al. 2015; Weibel et al. 2007; Whitesides et al. 2001). In our case, replicas were produced for single use even if they could in principle be sterilized and re-used. In particular, PDMS stencil fabrication consisted of the following consecutive steps: i) structure design; ii) in silico wafer fabrication through a photolithographic technique; and iii) fabrication of PDMS replicas (for details, see (Bisio et al. 2014)). The developed modular neuronal networks were composed of two 3x2 mm modules that were interconnected by 25 channels 390 µm long and 10 µm wide with an inter-channel distance of 50 µm. Once the PDMS masks were ready, they were positioned on the MEA substrates to include all the electrodes within the two modules. Next, the MEAs underwent the coating procedure for promoting cell adhesion, and the following day, the PDMS mask was removed immediately before the plating procedure, thus allowing neurons to slowly move only towards the promoting adhesion areas.

Neuronal preparation

Dissociated neuronal cultures were prepared from the neocortex of 18-day-old embryonic rats (pregnant Sprague-Dawley female rats were obtained from Charles River Laboratories Italia, Milano, Italy). All experimental procedures and animal care were conducted in conformity with institutional guidelines in accordance with the European legislation (European Communities Directive of 24 November 1986, 86/609/EEC) and with the NIH Guide for the Care and Use of Laboratory Animals. For our experiments, we used dissociated neurons arranged on a bimodular layout (cf. 'Bimodular confinement' paragraph for details) grown on MEAs. Culture preparation was performed as in (Bisio et al. 2014; Colombi et al. 2016). First, we coated MEAs overnight with poly-D-lysine and laminin to promote cell adhesion (Figure S1 A1). We washed the MEA devices at least 3 times with sterilized water before plating (Figure S1 A2). The neocortex of 4–5 embryos was dissected from the brain and dissociated first by enzymatic digestion in trypsin solution 0.125% (25–30 minutes at 37°C) and subsequently by mechanical dissociation with a fire-polished pipette. The resulting tissue was resuspended in Neurobasal medium supplemented with 2% B-27, 1% Glutamax-I, 1% Pen-Strep solution and 10% Foetal Bovine Serum (Invitrogen, Carlsbad, CA) at a final concentration of 500 cells/ul (Figure S1 A3). Cells were then plated onto 60-channel MEAs and maintained with 1 ml nutrient medium (i.e., serum-free Neurobasal medium supplemented with B27 and Glutamax-I). Then, cells were placed in a humidified incubator with an atmosphere of 5% CO_2−95% air at 37°C. Half of the medium was changed weekly.

Immunofluorescence staining and image analysis

The cultures on coverslips were fixed with 4% paraformaldehyde in phosphate-buffered saline (PBS) for 30 minutes. After permeabilization with 0.1% Triton X-100 in PBS for 10 minutes four times, the cultures were incubated with PBS containing 5% goat serum and 0.1% Triton X-100 for 1 hour. The permeabilized cultures were incubated with primary antibodies (anti-microtubule associated protein 2 [MAP2] mouse IgG; 1:100; Sigma-Aldrich) in PBS containing 5% goat serum overnight at 4°C and were rinsed with PBS for 10 minutes four times. Then, the cultures were incubated with a secondary antibody (Alexa Fluor 488-labelled anti-mouse IgG; Molecular Probes) in PBS containing 5% goat serum for 2 hours at room temperature and rinsed four times. The coverslips were removed from 12-well plates and mounted on glass slides with mounting media containing DAPI for nuclear staining. Fluorescence images were captured using a fluorescence microscope (Nikon eclips80). More details on the procedure can be found in (Ito et al. 2010).

Focal lesion procedure and laser setup

The laser dissection system (Difato et al. 2011) allowed for focal ablation of the sample in a three-dimensional confined volume due to the sub-nanosecond pulsed UVA laser source, which required delivering very low average power, thus confining the material breakdown to the focus spot. The setup was configured in an upright optical layout to allow optical surgery of neuronal networks plated on thick and non-transparent support such as the MEA chip. Therefore, simultaneously monitoring network activity using the MEA device and fluorescence calcium imaging during optical dissection of the connection of neuronal assembly was possible. The system was equipped with a custom micro-incubator (Aviv et al. 2013) maintaining the physiological parameters of the neuronal cultures (pH, osmolarity and temperature) to carry out long-term network activity recording before and after laser injury (Soloperto et al. 2016).

Micro-electrode array recordings

MEAs (Multi Channel Systems, MCS, Reutlingen, Germany) consisted of 60 TiN/SiN planar round electrodes (electrode diameter: 30 μm; inter-electrode distance: 200–500 μm). One recording electrode was replaced by a larger ground electrode. Each electrode provided information on the activity of the neural network in its immediate area. The array design using our bimodular mask allowed us to confine the growth of cells in two macro areas hosting 28 electrodes each (i.e., the ground electrode and the three central electrodes were excluded). A microwire connected each microelectrode of the MEA to a different channel of a dedicated amplifying system with a gain of 1100. The signals from the biological neural network (BNN) were amplified by a commercial system (MEA1060-Inv-BC amplification system, Multi Channel Systems, MCS, Reutlingen, Germany).

To reduce the thermal stress of the cells during the experiment, we maintained MEAs at 37°C with a controlled thermostat (MCS) and covered them with a custom PDMS cap to avoid evaporation and prevent changes in osmolarity (Blau et al. 2009). Additionally, we used a custom chamber to maintain a controlled atmosphere (i.e., gas flow of 5% CO_2 and 95% O_2 + N_2) during the entire recording time (Frega et al. 2012).

Electrical stimulation

All electrical stimuli to BNN were triggered by our FPGA-based board (by a commercial stimulator, STG 4002, Multi Channel Systems, MCS, Reutlingen, Germany). The basic electrical stimulus was a biphasic voltage pulse, 300 µs in the half-length phase and 750 mV of half-amplitude (in accordance with the literature (Wagenaar et al. 2004)). The necessary condition to choose the stimulating electrode was its capability to evoke a clear response on that module, meaning a network burst (NB) involving the entire module. In the pre-lesion condition, the stimulation evoked a response in both modules. In the post-lesion condition, the stimulation evoked a network burst confined to a single module.

Experimental protocols and databases

The general protocol included at least 30 minutes of no recording immediately after the culture was moved from the incubator to the amplifier to ensure the stability of the network in recording conditions. For control experiments without lesions, the four hours were recorded continuously. In controls with lesions, after the first hour of recording, we took the culture from the amplifier system, and we moved it to a separate room for laser ablation. The entire procedure usually took 20 minutes. After the lesion, the culture was moved back to the amplifier system where we immediately started the recording. The total number of experiments performed with these protocols was 9 for controls without lesions (26.8 ± 0.8 days *in vitro*, DIV) and 4 for controls with lesions (26 ± 0.6 DIV). For the bidirectional bridging (BB) protocol and hybrid bidirectional bridging (HBB), we performed similar protocols. We recorded 20 minutes of spontaneous activity before the lesion. Next, we performed laser ablation as described above and then waited for two hours to achieve stable activity in both modules, as shown by the results of control experiments (Figure 2). Then, we recorded 20 minutes of spontaneous activity after the lesion. To choose the best parameters that allowed us to reliably detect NBs in both modules, we performed offline NB detection. Next, we set these detection parameters on the FPGA with a custom-made MATLAB code and performed a 20 minute session of BB or HBB. The final step consisted of recording 20 minutes of spontaneous activity. The total number of experiments performed with these protocols was 9 for BB (26 ± 0.8 Days *in vitro*, DIV) and 7 for HBB (26 ± 1.5 DIV). In one HBB experiment, we did not record the last spontaneous phase because of a technical problem.

Online processing

Real-time spike detection

The custom experimental system retrieved analogue data from MEA1060-Inv-BC pre-amplifiers through a standard MCS connector. Analogue input signals were packed in subgroups of 8 signals. For each subgroup, the signals were connected to the input of an analogue multiplexer that switched from one signal to the next one at a frequency of 80 kHz (each 12.5 µs) as shown in Figure S2 A. Then, the output signals were amplified. The amplification gain ranged from 1 (no amplification) to 100. Multiplexed signals were finally converted to digital with 16-bit accuracy at a frequency of 80 kHz to provide one sample each time the multiplexer switched to a new input. Then, each of the 8 inputs of each multiplexer was sampled at a frequency of 10 kHz. The system relied on the antialiasing filters embedded in the pre-amplifier.

The spike detection module of the neuroprosthetic device was based on a threshold principle. A detection system was available for each input channel of the system. The first step of the detection was to increase the signal to noise ratio (SNR) of the input signal. The second step was to determine the appropriate threshold to perform the detection; this part of the processing is illustrated in Figure S2 B. To emphasize the amplitude of spike shapes and improve SNR, we used the wavelet decomposition technique – SWT (Quotb et al. 2012). This technique consisted of applying two orthogonal filters (one highpass and one lowpass) tailored to the characteristic shape of a spike.

Because of the impedance variation phenomenon, the input gain was expected to be different from one electrode to another and from one experiment to another. Therefore, we needed automatically set thresholds to avoid setting this parameter for each input channel individually. We determined these thresholds as a multiple of the standard deviation (σ) of the first detail level from the SWT outputs. The module that computed σ was referenced as an amplitude estimator (AE) in Figure S2. The correspondence factor to determine the threshold was set system-wide by the experimenter depending on the experimental conditions. To compute σ, we used a continuous evaluation to avoid characteristic steps that occurred when performing computation on windows. To reduce computation resources, we digitized a method from analogue computing (Harrison 2003). With the hypothesis that the signal distribution was constant, we knew the proportion of samples below and above the standard deviation, and the system used this property to determine σ with a regulation loop. The signal distribution may change from experiment to experiment, but the estimated σ will always be relevant to the signal amplitude. Then, variations are compensated by adjusting the multiplier that determines the final threshold. Harrison (Harrison 2003) showed that in addition to its computation efficiency, this method was much more immune to accidental phenomena such as spike residues or stimulation artefacts.

As previously stated, the threshold was applied to the chosen output detail level of the SWT. The final step was to filter glitches that may come from noise around the threshold value and to change a spike-length pulse to a single event. This step was performed by a state machine that as far as the threshold had been crossed, provided a detected spike event and remained silent during a user-configured refractory period. Then, the output event was available as a spike event for the remaining computation modules of the system.

Addressing the signal to noise ratio (SNR) with the stationary wavelet transform (SWT)

As stated in the main text, the SWT technique we used is composed of a set of orthogonal filters (one highpass, one lowpass). The output of the highpass filter (detail level) was used as an output of the decomposition. The output of the lowpass filter (approximation level) was downsampled, and the filters were applied to provide the next detail and approximation outputs. The process is repeated as long as necessary. From the theoretical point of view, this method split the noise among the different detail outputs but restricted the spike contribution to one or two outputs, which improved the SNR for these specific outputs. The drawback is that the downsampling generated non-stationary behaviours and resulted in sub-optimal detection when the spike time corresponded to dropped samples. To avoid this phenomenon, we chose to replace signal downsampling by filter upscaling (transforming an 8th order filter to a 16th order filter containing 8 interleaved null coefficients). With this technique, stationary wavelet transform (SWT)(Pesquet et al. 1996), we kept the best performance for all samples with a price of computational requirements twice as high, and memory requirements 4 times higher. This change in requirements was not an issue since we used hardware computing, and we added the computing resources according to our needs. Although our hardware was able to compute 8th order mother wavelets (highpass/lowpass filter couples), experience showed that the Haar mother wavelet (2nd order) was sufficient (Quotb et al. 2012). An empirical study showed that the third detail level provides the best performance for detection. As the first detailed output was expected to receive very few contributions from the spikes, it was used to determine the comparison threshold for spike detection.

Real-time network burst detection

The network burst detector (NBD) was based on the number of action potentials. Although this module was usable on a single channel, it was intended to gather events from multiple channels at the network scale as the preliminary experiments raised the necessity. Therefore, this module was not part of the computation flow build for each channel. There were 16 NBDs. Each NBD received events produced by all the incoming channels. Whether the input channels contributed to any of the 16 NBDs was user-defined. During a predefined time window, events on the selected channels were counted. At the end of the time window, the number of events was compared to a user-defined threshold to determine if an activity burst was occurring or not, and the counter was reset. The detector constantly kept track whether a burst was pending or not. Depending on user choice, each NBD could be independently configured to produce a single event when a burst started, a single event when a burst stopped, one event at each end of computing window detecting a burst, or continuously while the detector was in burst mode. The NBD could receive any event coming from the inputs on the system, including ISI-based burst detection events. It was possible to weight the event contributions such that an event coming from an ISI-based detected burst might have a much higher weight than an action potential on a single channel. The 16 NBDs worked on 1 kHz sampled signals because the application they fed was sampled at that frequency. Input events (10 kHz) were combined with a logical OR by group of 10 to produce the downsampling. There was no risk of losing an event because of the behaviour of neural cells that were unable to produce two action potentials within 1 ms.

An offline version of the NBD module was written in MATLAB to expedite the choice of parameters for reliable NB detection during experiments. By running the offline NBD with a combination of different windows in the range 1-50 ms and different thresholds in the range 1-250, we looked for a combination of parameters that could reliably detect NBs.

Closed-loop implementation

To close the loop, we chose to deliver a stimulation either to the BNN or the SNN based on the detection of a network burst (NB). The time to propagate the NB event depended on the source on which it is computed (BNN or SNN) and the stimulation destination (BNN or SNN). To follow the computation rhythm of the SNN, all events were processed at a frequency of 1kHz (so an event might have to wait for 900 µs until the next step occurs).

NBs were computed within 37.4 µs (including spike detection) and were available for processing 1.3 µs later. Therefore, the BNN to SNN latency was 38.7 µs. After a computation step of the SNN, the stimulator was eventually activated after a delay of 3.9 µs. As we measured the stimulator delay to be 85 µs, the SNN to BNN latency was 88.9 µs.

The BNN to BNN latency was composed of the BNN to SNN latency, the time to wait for the next computation step (*Tw*), and the BNN to SNN latency. *Tw* was time dependant, values were equally distributed between 61.3 µs and 961.3 µs with a mean value of 511.3 µs.

Spiking neural network

The biomimetic SNN is a neuromorphic system with a most detailed level of analogy with the nervous system. The SNN is a network of silicon neurons connected via excitatory and inhibitory silicon synapses and plasticity rules. The characteristics of the systems for bio-hybrid experiments are compatible with the time constants of the real biological systems, complex neuronal models and plasticity that reproduce spatio-temporal patterns of activation. The hardware real-time implementation needs a low-resource neuron model.

Neuron model

The (Izhikevich 2003) IZH model is composed of two equations (1) and (2) in which the state variables ‘*v*’ represents the membrane potential of a neuron and ‘*u*’ represents the regeneration of the membrane potential, which takes into account the activation of ionic currents K^+ and the inactivation of ionic currents Na^+. I_{Izh} describes the input current from other neurons. The integration time stamp *dt* of the IZH model is 1 ms.

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I_{Izh} \tag{1}$$

$$\frac{du}{dt} = a(bv - u) \tag{2}$$

with the after-spike resetting conditions:

$$if\ v \geq 30mV\ \rightarrow \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \tag{3}$$

In this model, parameter *a* describes the time scale of the regeneration variable *u(t)*. Lower values of *a* indicate slower regeneration. Parameter *b* describes the sensitivity of the regeneration variable influencing the under-threshold fluctuations of the membrane potential. Parameters *c* and *d* are the reset values of the membrane voltage and regeneration variable after an action potential, respectively. The IZH model can reproduce the behaviour of all known cortical neurons by changing only 4 parameters (*a*, *b*, *c* and *d*), see Figure S6.

The detailed model and its implementation in digital hardware (Ambroise et al. 2013) are described below (see SNN implementation on FPGA).

Synapse

Synaptic transmission includes both excitatory and inhibitory models. In particular, the proposed model takes into account AMPA (i.e., an excitatory neurotransmitter) and GABA (i.e., an inhibitory neurotransmitter). Depolarization or hyperpolarization are represented by a positive or negative contribution to synaptic currents I_{syn}. Following the effects of AMPA and GABA, all excitatory or inhibitory synaptic currents tend to zero out, exponentially decreasing (Ben-Ari et al. 1997). These two synaptic currents obey the same law of exponential decay τ_{syn} (3 ms for excitatory decay and 10 ms for inhibitory decay). According to the literature (Izhikevich 2004), whenever a pre-synaptic neuron emits a peak, a synaptic weight (W_{syn}) is added to the synaptic current of the post-synaptic neuron.

To connect the BNN to the SNN, we added external synapses to the SNN. Each neuron of the SNN was connected to one external synapse which can be tuned to be excitatory or inhibitory. Each event detected in the BNN could be used to stimulate the SNN through these external synapses. The SNN stimulation was performed through current stimulation via the external synapses. The stimulation duration depended on the type of synapses (excitatory or inhibitory) and the amplitude depended on the weight applied to the synapses. In our case study, excitatory synapses with 3ms exponential decay and with a weight of 9 are used for connection with BNN. We stimulated a maximum of 20 neurons to avoid over stimulation of the SNN.

Short-term synaptic plasticity model

To improve the biological behaviour of the synaptic model, we added short-term synaptic plasticity, which modified the synaptic weight according to the activity of pre-synaptic neurons.

Short-term plasticity is a biological phenomenon that modifies and regulates connection weights as a function of network activity. The model used in this work was proposed by (Izhikevich and Edelman 2008). This model is called "short-term plasticity" because the facilitation as well as the depression of a synapse is reabsorbed when no action potential has been emitted during a time constant. The implementation of the short-term plasticity and synapses are described below (parameters*:* x_{syn}*, P* and t_{syn}*,* see *SNN implementation on FPGA*).

Synaptic noise and axonal delay

To enable spontaneous activity and the activity of our network, we added stochasticity by a source of current noise for each neuron in the neuron model. We used the Ornstein-Uhlenbeck (OU) process (with parameters: mean value = 0; degree of volatility = 35; dissipation rate = 1), which is a suitable model for modelling synaptic noise in a neural network (Rudolph and Destexhe 2005). The OU process Xt is a prototype of a noisy relaxation process and an example of a Gaussian process that has bounded variance and a stationary probability distribution. The process is stationary, Gaussian and Markovian. It satisfies the following stochastic differential equation:

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t \qquad (4)$$

Where θ > 0, μ and σ > 0 are the parameters, and Wt is the Wiener process.

The parameter μ represents the equilibrium or the mean value of the process. The stationary variance is given by:

$$var(X_t) = \frac{\sigma^2}{2\theta} \qquad (5)$$

This form of current may represent an approximation to that resulting from the random opening and closing of ion channels on a neuron’s surface or to randomly occurring synaptic input currents with exponential decay (Tuckwell et al. 2002).

Figure S5 C represents the dynamic behaviour of the neuron model (tonic bursting neuron) family in which the parameters are set for tonic bursting activity. This figure represents the same neuron in which we add the stochastic input currents. We noticed that the current noise source caused variability in the timing of action potentials. The digital implementation of this OU process is described in (Grassia et al. 2016) (see also SNN implementation on FPGA).

The synapse computation core also manages the axonal delay. The axonal delay is the phenomenon according to which an action potential emitted at time t arrives at time t + t1 to the post-synaptic neuron. In our implementation, the axonal delay was implemented by a 50-bit delay (shift register) vector, a D value and a multiplication factor. We stored the action potential (1 bit) in this vector delay at the position indicated by the value D. We can have a delay from 0 to 49 ms if the multiplication factor is equal to 1. We can increase this time by tuning this factor. The axonal delay makes it possible to describe a superposition of a network of neurons in 2D and consequently design pyramidal or 3D networks.

Neural network architecture

Our architecture of the neural network was based on blocks of RAM (storage of the parameters necessary for the definition of a network), two computing cores (a neuron and synapse), a block to manage the state machine and addresses and an "RS232 Communication" block (which will allow us to configure/modify the parameters of our system from a configuration file). Figure S5 A shows the interaction between these blocks. To simplify the figure, we did not represent the clock signal, but the entire architecture is synchronous. The limitation of the number of neurons is due to the following factors: 1) the clock frequency used to maximize the number of neurons in parallel and 2) the size of the RAM, which allows the storage of the parameters, especially the synapses. The resources used are presented in Table S2.

One of the advantages of our architecture was that each neuron and synapse were independent. A network configuration file was sent to the FPGA to select the connections and set the number of neurons, synapses, and used options. We could also define multiple networks with different connectivity between networks and within each network.

Network configuration

To add flexibility to our system, we added the ability to communicate with the FPGA via serial links to send the configuration, topology, and neural network options. Several options were possible depending on the desired application as follows: axonal delay, short-term plasticity, and synaptic noise. Then, we chose the family of neurons (excitatory, inhibitory), the percentage of connectivity between neurons, the inhibitory/excitatory ratio k, synaptic weight distributions, and the amplitude of the OU noise. From a library of configuration files, we could select the desired network for the given application and particularly in terms of frequency of NB. Figure 4 describes SNN library processing. Table S1 shows some metrics relative to the bursting features of the implemented artificial networks as well as the mean values of the excitatory and inhibitory weights.

The SNN library has been populated with network implementations sharing a fixed number of neurons (i.e. 100), the same topology (i.e. random connectivity) but with different synaptic weights (cf. Figure S7), necessary to reproduce different bursting behaviours.

Depending on the FPGA board, the maximum number of neurons varies. For instance, in a Spartan 6 FPGA board, we implemented 512 neurons, 66048 synapses with synaptic noise, axonal delay and synaptic plasticity. Table S2 describes the resources used. This biomimetic digital SNN worked in real-time, was tuneable and reproduced complex biological neural networks due to the biophysically neuron model, synapses, plasticity, axonal delay and synaptic noise.

SNN implementation on FPGA

Digital implementation of the Izhikevich model

The mathematical equations must be modified to implement the IZH model in digital hardware. To make the IZH neural network more biorealistic, we split the current I_{Izh} into the following currents: I_{stat}, I_{exc}, and I_{inh}. I_{stat} is the biasing current, I_{exc} is the positive contribution due to excitatory synapses and I_{inh} is the negative contribution of inhibitory synapses.

We also used the methodology developed at the University of Hopkins in the United States(Cassidy and Andreou 2008) by multiplying the equation of the membrane voltage by 0.78125 such that all parameters are decomposable with powers of 2, which allows easier digital implementation.

$$\frac{dv}{dt} = {}^{1}\!/_{32}\, v^2 + 4v + 109.375 - u + I_{bias} + I_{exc} + I_{inh}$$
$$\frac{du}{dt} = a.(bv - u) \qquad (6)$$

And $\frac{dv}{dt} = \frac{v[n+1] - v[n]}{\Delta t}$ knowing that the computation time of the IZH model is 1 ms (Δt = 1):

$$v[n+1] = {}^{1}\!/_{32}\, v[n]^2 + 5v[n] + 109.375 - u[n] + I_{bias}[n] + I_{exc}[n] + I_{inh}[n]$$
$$u[n+1] = u[n] + a.(b.v[n] - u[n]) \qquad (7)$$

Using a calculation pipeline, we can compute different neurons in parallel. All model parameters are stored in RAM. Figure S5 B1 describes the calculation pipelines for *v* and *u*.

Digital implementation of synapse and synaptic plasticity

Synapse model: When W_{syn} is positive, the synapse is excitatory, and when W_{syn} is negative, the synapse is inhibitory. The current I_{exc} is always positive, and the current I_{inh} is always negative. However, these two synaptic currents obey the same law of exponential decay τ_{syn}. I_{syn}, the synaptic current, is written:

$$I_{syn}(t) = -\tau_{syn}.I'_{syn}(t) = -\tau_{syn}.\frac{I_{syn}(t+T) - I_{syn}(t)}{T} \qquad (8)$$

$$I_{syn}(t+T) = \left(1 - \frac{T}{\tau_{syn}}\right).I_{syn}(t) \qquad (9)$$

Where computation time *T* is 1 ms:

$$I_{syn}(t+1) = \left(1 - \frac{1}{\tau_{syn}}\right) . I_{syn}(t) \quad (10)$$

In digital implementation:

$$I_{syn}[n+1] = I_{syn}[n] - \frac{1}{\tau_{syn}} . I_{syn}[n] \quad (11)$$

Short-term synaptic plasticity model: This model is defined by 3 parameters:

- a scalar factor x_{syn}, which indicates the state of the synapse (depression or facilitation) and computes the synaptic weight.
- a percentage P, which will be multiplied by the factor x_{syn} after each emission of a pre-synaptic action potential. If this percentage is larger than 1, this synapse will describe a short-term facilitation. Otherwise, if this percentage is less than 1, this synapse will describe a short-term depression. In facilitation (or depression), the value added to the stimulation current of a post-synaptic neuron will increase (or decrease) with each emission of action potential.
- a time constant t_{syn} of exponential decay (or growth) in facilitation (or depression).
  When a pre-synaptic spike occurs:

$$I_{syn}[n+1] = I_{syn}[n] - \frac{1}{\tau_{syn}} \cdot I_{syn}[n] + W_{syn}[n] \quad (12)$$

Where W_{syn}, the synaptic weight, is defined by:

Ws [n] = xsyn [n] Wsyn (13)

xsyn [n+1] = P.xsyn[n] (14)

Parameters W_{syn}, x_{syn}, P, τ_{syn} are stacked into RAM.

Synaptic noise implementation:

The digital implementation of this noise (equation (4)) in the stimulation current of the neuron is given by:

$$I[n+1] = I[n] + \theta(\mu - I[n])\Delta t + \sigma \Delta W[n] \quad (15)$$

Where n is the iteration step, and *Δt = T/N* is the time step after the partition of the interval *[0, T]* into *N* equal subintervals of width. Note that the random variables *ΔW[n]* are independent and identically distributed normal random variables with expected value zero and variance *Δt*, thus *ΔW[n]~N(0, Δt)* = √*Δt * N(0,1)*.

Figure S5C represents the dynamic behaviour of the neuron model in which the parameters are set for tonic bursting activity.

Offline data analysis

Offline data analysis was performed by custom scripts developed in MATLAB (MathWorks, Natick, MA, USA).

Preprocessing

To characterize the activity level of BNNs, we used the percentage of active channels and the mean firing rate (MFR). To compute the former, we considered active electrodes only those presenting a firing rate higher than 0.01 spikes per second (spikes/s), while the latter was defined as the mean number of spikes per second, computed over the total recording time, of the active channels. The low threshold guaranteed excluding only those electrodes that were not covered by cells or with very few spikes, keeping all the others (Bisio et al. 2014). In stimulation phases, the activity of stimulated channels was deleted.

Cross-correlation

To quantify the level of synchronization among multi-unit recordings, we applied correlation analysis (Chiappalone et al. 2007) to the spike trains. Cross-correlograms were built according to the method of the activity pairs described by (Eytan et al. 2004) as follows: given two trains (i.e., *X* and *Y*), we counted the number of events in the Y train within a time frame around the *X* event of ±T (T set at 500 ms), using bins of amplitude Δt (set at 1 ms). The CC function $C_{xy}(\tau)$ was obtained by a normalization procedure, according to the following formula (Pasquale et al. 2008):

$$C_{xy}(T) = \frac{1}{\sqrt{N_x N_y}} \sum_{t_i=\left(T-\frac{\Delta T}{2}\right)}^{\left(T+\frac{\Delta T}{2}\right)} X(t_s - t_i) \tag{16}$$

where ts indicates the timing of an event in the X train, Nx and Ny are the total number of events in the X and Y trains, respectively, and Δτ is the bin size. Equation (1) yields the symmetry between $C_{xy}(\tau)$ and $C_{yx}(\tau)$ (i.e., $C_{xy}(\tau)$=$C_{yx}(-\tau)$) (Eytan et al. 2004).

To evaluate the amount of correlation between two modules, we performed the computation between the collapsed spike trains for each module. A collapsed spike train is the result of a logic OR operation between the spike trains recorded on all the electrodes belonging to the same module. For control experiments and BB experiments, the modules were modules of the BNN. In the HBB protocol, we collapsed the activity of the entire SNN to a single spike train. We applied a locally weighted linear regression to smooth the curve. The area of the CC was the integral of the curve in the range from -500 to 500 ms.

Offline network burst detection

The offline NBD was performed using a software implementation of the NBD algorithm running online on the neuromorphic device, called the “fixed window accumulator”. Similar to the online version, the algorithm summed all the spikes fired from the neurons of the networks during a given time interval, and it detected an NB if this sum overcame a threshold. Each NB was highlighted by the generation of an event. The input parameters were the window time (i.e., time interval within which the spikes were summed), the threshold (i.e., the number of spikes to overcome to identify an NB event) and the source (i.e., the channels on which the analysis was performed).

Depending on the specific protocol we wanted to analyse, the source could be the whole BNN (in controls without lesions or pre-lesion spontaneous activity), a part of it (for example, to compute the NBR in only one module or to exclude stimulated electrodes), the SNN (for example, to characterize the SNN) or a hybrid network formed by both the SNN and whole BNN or part of it (in HBB). The offline version was also useful during the experimental protocols to set the best parameters (window time and threshold) to obtain a reliable NB detection and to choose the most suitable SNN, referring to NBR, for the HBB experiment.

Probability single-module network burst

To quantify the confinement of the electrophysiological activity in a single module, we used the single-module NB probability parameter. This parameter represented the probability that NBs were composed of spikes belonging to a single module (i.e., one of the modules of the BNN or the SNN). This parameter was computed by counting the number of spikes that belonged to each burst, followed by the percentage of spikes that belonged to each module. If this percentage was higher than 85%, we assumed that the burst was generated in a single module. Setting 85% as the threshold percentage allowed us to consider the possibility of having random spikes in the other module at the same time as the NB. To perform this analysis, we modified the NBD by adding start and stop thresholds. For each NB identified, the algorithm looked for an empty time window (i.e., a time window in which the number of spikes was equal or lower than the threshold) before and after the window in which an NB was detected to identify the first and last spikes of that event as the first spike in the first window and the last spike in the last window belonging to the NB, respectively. We set the start and stop thresholds to zero and 5, respectively, for BNN and SNN (the SNN threshold was higher because of the presence of fast spiking neurons).

Statistics

Data within the text is expressed as the mean ± standard error of the mean (SEM), unless otherwise specified. Statistical tests were employed to assess significant differences among different experimental conditions. The normal distribution of experimental data was tested using the Shapiro-Wilk normality test. We performed one-way repeated measures ANOVA to compare data from the same group at different time points. When ANOVA gave a significant ($p < 0.05$) result, the post hoc Bonferroni test was employed to assess differences between all phases. When normality was rejected, we used Friedman’s repeated measures ANOVA on the ranks test to compare data from the same group at different time points. When ANOVA gave a significant ($p < 0.05$) result, the post hoc Tukey test was employed to assess differences between all phases. To compare data from two different populations (Figure 2), we performed the Mann-Whitney test. Statistical analysis was carried out by using OriginPro (OriginLab Corporation, Northampton, MA, USA) and Sigma Stat (Systat Software Inc., San Jose, CA, USA).

Data and Software Availability

The data sets of the experiments presented in our manuscript are available in the Mendeley Data repository. The accession number for the experimental data reported in this paper is the following:

doi: 10.17632/rgbgvshs52.2

The FPGA SNN is available upon request to Timotheé Levi, and it will be available in the website of the authors (Timothée Levi, http://timothee.levi.free.fr). Spike computation related modules are publicly available at http://yannick.bornat.fr.

The above accession numbers and links will be accessible to readers upon publication of our manuscript.

Upon reasonable request, data and information on the reported results can be asked to the corresponding authors.

Supplemental References

Ambroise, M., Levi, T., Bornat, Y., Saighi, S., 2013. Biorealistic spiking neural network on FPGA. 2013 47th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6. IEEE, Baltimore, MD, USA.

Aviv, M.S., Pesce, M., Tilve, S., Chieregatti, E., Zalevsky, Z., Difato, F., 2013. Motility flow and growth-cone navigation analysis during *in vitro* neuronal development by long-term bright-field imaging. J. Biomed. Opt. 18(11), 111415.

Ben-Ari, Y., Khazipov, R., Leinekugel, X., Caillard, O., Gaiarsa, J., 1997. GABAA, NMDA and AMPA receptors: a developmentally regulated 'ménage à trois'. Trends Neurosci. 20(11), 523–529.

Blau, A., Neumann, T., Ziegler, C., Benfenati, F., 2009. Replica-moulded polydimethylsiloxane culture vessel lids attenuate osmotic drift in long-term cell cultures. Journal of biosciences 34(1), 59–69.

Cassidy, A., Andreou, A.G., 2008. Dynamical digital silicon neurons. Biomedical Circuits and Systems Conference, 2008. BioCAS 2008., pp. 289–292. IEEE, Baltimore, MD, USA.

Chiappalone, M., Vato, A., Berdondini, L., Koudelka-Hep, M., Martinoia, S., 2007. Network dynamics and synchronous activity in cultured cortical neurons. Int. J. Neural Syst. 17(2), 87–103.

Colombi, I., Tinarelli, F., Pasquale, V., Tucci, V., Chiappalone, M., 2016. A simplified *in vitro* experimental model encompasses the essential features of sleep. Front. Neurosci. 10, 315.

Difato, F., Schibalsky, L., Benfenati, F., Blau, A., 2011. Integration of optical manipulation and electrophysiological tools to modulate and record activity in neural networks. Int. J. Optomechatronics 5(3), 191–216.

Eytan, D., Minerbi, A., Ziv, N., Marom, S., 2004. Dopamine-induced dispersion of correlations between action potentials in networks of cortical neurons. J. Neurophysiology 92(3), 1817–1824.

Frega, M., Pasquale, V., Tedesco, M., Marcoli, M., Contestabile, A., Nanni, M., Bonzano, L., Maura, G., Chiappalone, M., 2012. Cortical cultures coupled to micro-electrode arrays: a novel approach to perform *in vitro* excitotoxicity testing. Neurotoxicology and teratology 34(1), 116–127.

Harrison, R.R., 2003. A low-power integrated circuit for adaptive detection of action potentials in noisy signals. Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439). IEEE, Cancun, Mexico.

Ito, D., Tamate, H., Nagayama, M., Uchida, T., Kudoh, S., Gohara, K., 2010. Minimum neuron density for synchronized bursts in a rat cortical culture on multi-electrode arrays. Neuroscience 171(1), 50-61.

Mata, A., Fleischman, A.J., Roy, S., 2005. Characterization of polydimethylsiloxane (PDMS) properties for biomedical micro/nanosystems. Biomedical microdevices 7(4), 281–293.

Pasquale, V., Massobrio, P., Bologna, L.L., Chiappalone, M., Martinoia, S., 2008. Self-organization and neuronal avalanches in networks of dissociated cortical neurons. Neuroscience 153(4), 1354-1369.

Pesquet, J.-C., Krim, H., Carfantan, H., 1996. Time-invariant orthonormal wavelet representations. IEEE Trans. Signal Process. 44(8), 1964–1970.

Quotb, A., Bornat, Y., Raoux, M., Lang, J., Renaud, S., 2012. NeuroBetaMed: A re-configurable wavelet-based event detection circuit for in vitro biological signals. IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1532 - 1535. IEEE, Seoul, South Korea.

Rudolph, M., Destexhe, A., 2005. An extended analytic expression for the membrane potential distribution of conductance-based synaptic noise. Neural Comput. 17(11), 2301–2315.

Taylor, A.M., Jeon, N.L., 2010. Micro-scale and microfluidic devices for neurobiology. Curr. Opin. Neurobiol. 20(5), 640–647.

Tuckwell, H.C., Wan, F.Y., Rospars, J.-P., 2002. A spatial stochastic neuronal model with Ornstein–Uhlenbeck input current. Biol. Cybern. 86(2), 137–145.

Wagenaar, D.A., Pine, J., Potter, S.M., 2004. Effective parameters for stimulation of dissociated cultures using multi-electrode arrays. J. Neurosci. Methods 138(1-2), 27–37.

Weibel, D.B., Diluzio, W.R., Whitesides, G.M., 2007. Microfabrication meets microbiology. Nature reviews. Microbiology 5(3), 209–218.

Whitesides, G.M., Ostuni, E., Takayama, S., Jiang, X., Ingber, D.E., 2001. Soft lithography in biology and biochemistry. Annual review of biomedical engineering 3, 335–373.