

Intimate mixing of analogue and digital signals in a field-programmable mixed-signal array with lopsided logic

Simeon A. Bamford^{1,2}, Massimiliano Giulioni²

Abstract—A field-programmable device has been developed, specialised for neural signal processing and neural modelling applications. The device combines analogue and digital functions, yet unlike other designs for Field-Programmable Mixed-signal Arrays (FPMA), there is no separation between the analogue and digital domains. To allow analogue values to act directly as inputs to digital blocks, all digital circuitry has limited crowbar current. The method of limiting yields lopsided logic thresholds. Two uses of this are demonstrated: a gate which detects digital saturation, and a D-type flip flop which is insensitive to clock slew rate.

I. INTRODUCTION

The ReNaChip project aims to create an implantable device which can replace the function of a cerebellar circuit by learning timed associations in a form of classical conditioning. This requires a pathway of signal processing from recording electrodes through detection of relevant stimulus events and a model of neural learning, to creating a response at a stimulation electrode. A Field-Programmable Analog Array (FPAA) has been designed which is capable of carrying out the event detection and neural modelling parts of this pathway, based on Switched Capacitor (SC) circuitry. The basic approach has been explained and argued for in [1]. Briefly, FPAA's are VLSI arrays of components that can be reconfigured in order to carry out analogue computations. Choosing a field-programmable approach for this project allowed a chip to be fabricated before having to commit on low level implementation details such as the precise form of a high-level neural model. The device uses a combination of capacitors which can be configured to act as switched capacitors, and amplifiers, in a manner similar to [2], to implement signal filters as well as the elements of neural models such as integration, decays and thresholds. Using an SC approach allows precise control of timings which may vary over several orders of magnitude to implement both multi-unit signals and neural models. In general the domain of neural signal processing and neural modelling may be an appropriate fit to the capabilities of this form of FPAA, due to the relatively low speeds of operation and relative low accuracy required for many functions, and due to the need for low-power operation especially for implantable devices.

This work was supported by ReNaChip, an FP7 funded project. 1: Laboratory for Synthetic Perceptive, Emotive and Cognitive Systems, Universitat Pompeu Fabra, Barcelona; 2: Complex Systems Modelling Group, Istituto Superiore di Sanità, Rome. Paolo Del Giudice made organisational contributions. e-mail: simeon.bamford@iss.infn.it.

Thus the basic design of the device is an array of (a) Configurable Switched Capacitors (CSC), and (b) amplifiers, with reconfigurable interconnect. Several components of the computation required for this project benefit from explicitly digital functions. For example the neural model (which is described in [3]) assumes that the learning of the timing of a conditioned response is effected by synapses from parallel fibres to purkinje cells. The direction of synaptic plasticity depends on the timed convergence of direct and modulatory inputs on synapses; such a decision can be implemented with a logical AND gate. Beyond this, a combined synaptic weight value needs to be stored, with a certain analogue depth. Long term storage of synaptic weights is a long-standing problem in neuromorphic VLSI; long-term potentiation and depression modify synaptic efficacies over periods of hours or longer, whereas analogue values stored on capacitors can be stable only for seconds. Thus attempts to store synaptic weights for realistic periods either use floating gates or else some form of digital storage, whether arrays of long-term bistable elements [4], or storage as binary values with Analogue-to-Digital and Digital-to-Analogue (AD/DA) converters [5]. Including reconfigurable digital circuitry allows the possibility of building binary-valued weights.

Thus the two elements of the reconfigurable array noted above are supplemented by Configurable Logic Blocks (CLB), whose design will be explained in section II. Thus the device is a Field Programmable Mixed-signal Array (FPMA). Typical FPMA designs assume separate analogue and digital domains with an interface between them including dedicated AD/DA [6] (one design in which AD/DA converters could be constructed where needed from reprogrammable resources is [7]). In this design however, the digital and analogue signals are mixed in the same configurable interconnect such that any component can receive either analogue or digital signals as inputs. One problem this causes is that if an analogue signal acts as input to a digital gate then if the input is not saturated at the level of either the high or low power rail but rather somewhere in between, a large current can flow through the gate. In section II will be explained the general approach that has been taken to resolving this problem, which involves limiting the current through digital gates. The method chosen delivers a lopsided form of logic, which can be utilised. Two such uses are a gate which detects digital saturation and a D-type flip flop which is insensitive to the slew rate of its clock. Results will be presented for these designs in section III.

II. DESIGN

Configurable array: Analogue filtering is performed with combinations of amplifiers, resistors and capacitors. In the SC technique, a resistance is emulated by switching the terminals of capacitors (as this is a standard technique, it will not be explained here). Thus in an SC solution a minimal requirement for blocks is (a) CSCs (capable of acting as either capacitors or resistors), and (b) amplifiers. The CSC is described in [1] whereas the amplifier is based on a standard topology; neither will be described here. The intended speeds of operation of the SC circuits vary over several orders of magnitude from $<1\text{Hz}$ to $\approx 100\text{kHz}$; accordingly, groups of amplifiers can be independently biased with currents over several orders of magnitude in order to provide sufficient drive such that settling times in given SC circuits are respected without wasting power where it is not required.

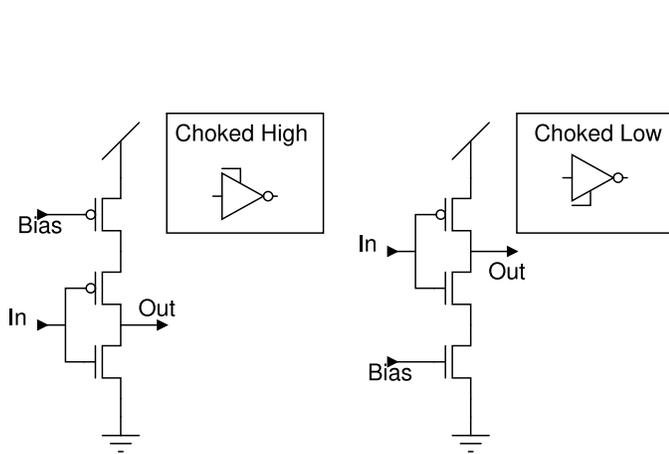
Regarding digital circuitry, an amplifier can be used in open-loop configuration such that, unless its inputs are close to each other in voltage, the output will saturate at one of the power rails, thus creating a digital decision. This can be used for some simple glue logic functions, for example, reversing the polarity of a decay if a level exceeds a threshold. However the desirability of explicitly digital circuitry including combinatorial logic was argued for in section I. Thus a third type of block, a Configurable Logic Block (CLB), completes the suite of available resources; this will be described below. In search of a simple flexible design, the CLBs have been placed in the same array of configurable interconnect as the amplifiers and CSCs, such that any block can act as an input to any other block. For example an amplifier in open-loop configuration implementing a threshold function can act as an input to a CLB.

Choking the crowbar current: As the system is intended to support processes happening on multiple timescales, it is not known *a priori* what the slew rate of an input may be. This causes a specific problem for digital gates. If for example an inverter has an input which is not saturated at the level of either the high or low power rail but rather somewhere in between, a large current can flow through the inverter, since both the NMOS and the PMOS will be switched on (called the crowbar current). In digital design, the standard approach to reducing power loss due to crowbar current is to ensure that inputs make fast transitions between the power rails, but this is not appropriate here. [8] solved this problem in a specific neural model by adding positive feedback to the input to make an intermediate input voltage unstable, but this approach would not be applicable where it is not known *a priori* what the purpose of the circuit is. Noting that the intended application domain does not require high speed digital circuitry, the general approach which has been taken in this design is to limit (or “choke”) the current through the inverter, so that there is only a small power consumption when the input is at an intermediate voltage. Thus in the same way that the current of groups of amplifiers can be independently set to define their speed of operation, the current that flows through the digital gates of the CLB is likewise programmable,

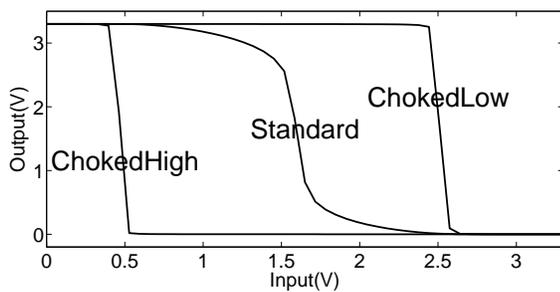
also defining their intended speed of operation.

A simple way to limit the current through an inverter is to put another transistor in series, gated at a fixed voltage to provide a low V_{gs} , such that I_{ds} is low. Figure 1a shows two ways of doing this, with a PMOS and an NMOS. Let’s call these solutions a high choke and a low choke respectively. This has two implications for performance. Firstly (figure 1b) the voltage at which the output switches from high to low is approximately centred for a standard inverter (where the size of the PMOS relative to the NMOS has been adjusted to achieve a centred switching voltage); however for an inverter with a high choke the switching voltage is low and for a low-choked inverter the switching voltage is high. Thus these inverters have unbalanced switching voltages, yet for properly saturated digital inputs they deliver properly saturated digital outputs. Secondly (figure 1b) a choked-high inverter can sink a high current to drive its output down quickly but can only source a low current to drive its output up slowly; the opposite is true for a choked-low inverter. Figure 1c demonstrates the difference in crowbar current, with the choked inverters set to operate around 50nA , but the standard inverter free to pass up to $\approx 60\mu\text{A}$. With saturated inputs all inverters pass $<1\text{pA}$.

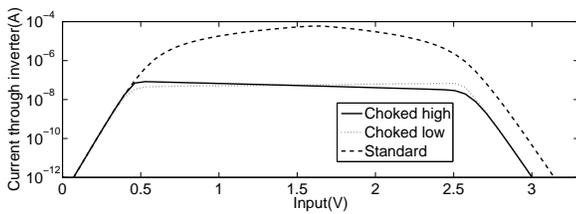
Configurable Logic Block : A basic CLB for an FPGA can consist of a Look-Up Table (LUT) and a D-type Flip-Flop (DFF), though many embellishments of this basic design are possible [9]. The n inputs to the CLB are used to decide between 2^n outputs of the LUT, and that output can then pass either directly to the output of the CLB or can pass to the input of the flip-flop, to be registered and output upon an incoming clock event. With a minimum of 2 inputs, any logic function can be constructed with a combination of these blocks. Larger numbers of inputs can allow easier construction of complex logic functions and faster performance. Figure 2a shows the basic CLB design. Figure 2b shows the adaptation of the LUT using choked inverters. The outputs of SRAM cells are inverted. The outputs of these inverters can directly become the output of the CLB, and so they are all choked with the same polarity (all choked low). If outputs were of different polarities then there would be the possibility that a choked-high inverter would try to deliver a strong logical low against a choked-low inverter trying to deliver a strong logical high, thus allowing a large current to pass through the unchoked branches of the two inverters. This could happen both within the reconfigurable interconnect, or within the LUT itself in the case of an input at intermediate voltage holding both transmission gates (T-gate) of a multiplexer (MUX) open. The two inputs go to inverters of opposite polarity, and this allows the different logic thresholds available to be utilised, as described below. Following the inverter at each input is another inverter, to create the complementary control signals for the MUXes; the following inverters have the opposite polarity from their predecessors, allowing an input change in one direction to cause a fast change in output selection, whereas a change in the other direction will yield a slow change in selection. This can permit design decisions about the relative priority of signals, though space does not allow a demonstration of this usage.



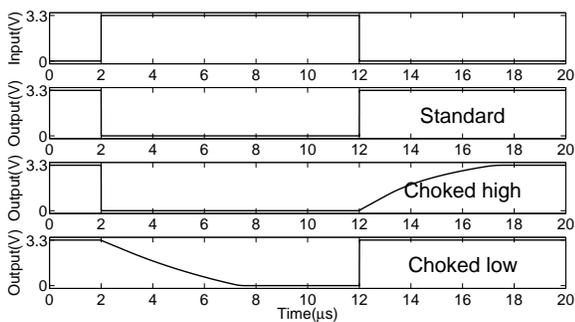
(a)



(b)



(c)



(d)

Figure 1. Choked inverters. (a) Circuits and proposed symbols for choked inverters. (b-d) Simulated outputs (using spectre, excluding parasitics) for choked inverters with minimum-sized transistors vs standard inverter with a minimum-sized NMOS and with the PMOS widened to centre the switching voltage. Choking transistors were biased to deliver a current of 50nA. All inverters had 100fF output load. (b-c) DC output voltage and current respectively for sweep of input voltages. (d) Transient output voltages for sharp 10us input pulse.

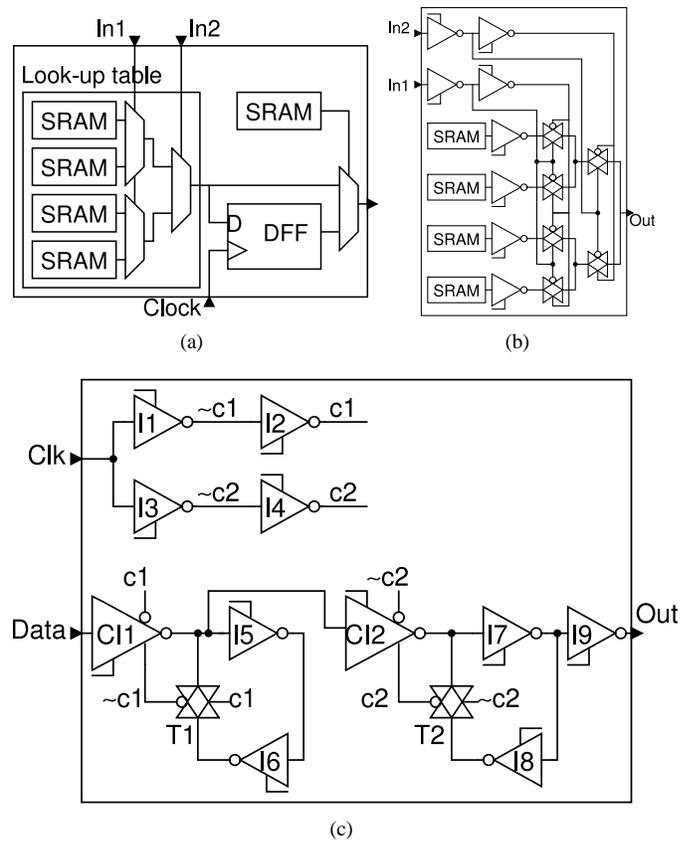


Figure 2. Configurable Logic Block. (a) Basic design. (b) LUT, using choked inverters; MUXes implemented with pairs of T-gates. (c) DFF using choked inverters; CI1-2 are choked clocked inverters - clocked inverters with the addition of an extra transistor in the same way as for choked inverters and with the polarity indicated by the same modifications of the symbol.

D-type flip-flop: Figure 2c shows the design of the DFF, which has the useful property of insensitivity to clock slew rate. This means that a shift register chain can be created where the rising edge of the clock can occur arbitrarily slowly without allowing data to fall through the chain. It works as follows. The incoming clock is an input to both I1 and I3, choked high and low respectively. For a rising edge with finite slew rate, $\sim c1$ falls before $\sim c2$ because of the differing switching thresholds of I1 and I3, and $c1$ rises before $c2$. When the input falls, $c2$ falls before $c1$. Thus a 2-phase cycle of rising and falling input is converted into a 4-phase cycle: when $c1$ rises the master latch (T1 & I5-6) becomes isolated from the input (by CI1); later when $c2$ rises, the state of master latch is passed through CI2 to the slave latch (T2 & I7-18); the new state passes to the output of I9, (choked low, consistently with the outputs of the LUT); when $c2$ falls again the slave latch is isolated from the master latch; finally, when $c1$ falls the master latch becomes once again driven by the input. This system requires only that the clock pulse lasts for a certain minimum period, which is defined by the biases controlling the choking transistors. Other designs of DFF exist which are insensitive to clock slew rate, e.g. [10] [11], however neither of these are designed for low crowbar current, and they also rely on transistor sizing, whereas this design uses all minimum-sized transistors.

Logic level detector: The resulting CLB performs straightforwardly as a programmable logic gate if the inputs are saturated digital signals. If the inputs are intermediate it is possible for the CLB to output an intermediate result, although the extra open-loop gain provided by the inverters at the inputs makes this unlikely. One use of the CLB can be to detect whether a voltage is in fact properly saturated at one of the power rails. To do this it is necessary only to program the LUT table to perform the XNOR function and then wire a single input signal to both inputs. then if the input is above the level at which the choked-low input switches, or below the level at which the choked-high input switches, then the output will be high. If however the input is intermediate then the output will be low since the input will be recognised as high by the choked-high input but not by the choked-low input (see figure 3b). This can be useful for checking the validity of digital inputs where necessary.

III. RESULTS

Simulated results are shown in figure 3, which demonstrate the clock-slew insensitive performance of the DFF, and the performance of a 2-input CLB as a logic level detector.

A chip has been designed in Austria Microsystems C35B4 process and is currently in fabrication. The configurable blocks are laid out in an island-style topology [12]. 340 blocks of the various types fit inside an array of 1.58×2.59 mm. The CLB has three inputs (2 choked-high and 1 choked-low), contains a DFF which allows asynchronous reset and has other adaptations for asynchronous logic which cannot be described here.

IV. CONCLUSIONS

An FPMA has been designed and is being fabricated, in which analogue and digital components exist in the same array and use the same reconfigurable interconnect so that there is no separation between the analogue and digital domains. In order to allow analogue values to act directly as inputs to digital blocks, all digital gates are choked in the path from the output to one of the power rails to limit crowbar current. Consequently the gates have lopsided logic thresholds and are unbalanced in their ability to source and sink current. These features have been utilised to create a gate which detects digital saturation and a D-type flip flop which is insensitive to the slew rate of its clock.

REFERENCES

- [1] S. Bamford and M. Giulioni, "Towards a field programmable analogue array for neural signal processing and neural modelling," in *BIOCAS*, 2010 - submitted - <note for reviewers, this can be evaluated at: <http://www.sim.me.uk/neural/2010BIOCAS1.pdf>>.
- [2] E. Lee and W. Hui, "A novel switched-capacitor based field-programmable analog array architecture," *Analog Integrated Circuits and Signal Processing*, vol. 17, pp. 35–50, 1998.
- [3] C. Hofstotter, M. Mintz, and P. Verschure, "The cerebellum in action: a simulation and robotics study," *European Journal of Neuroscience*, vol. 16, pp. 1361–1376, 2002.

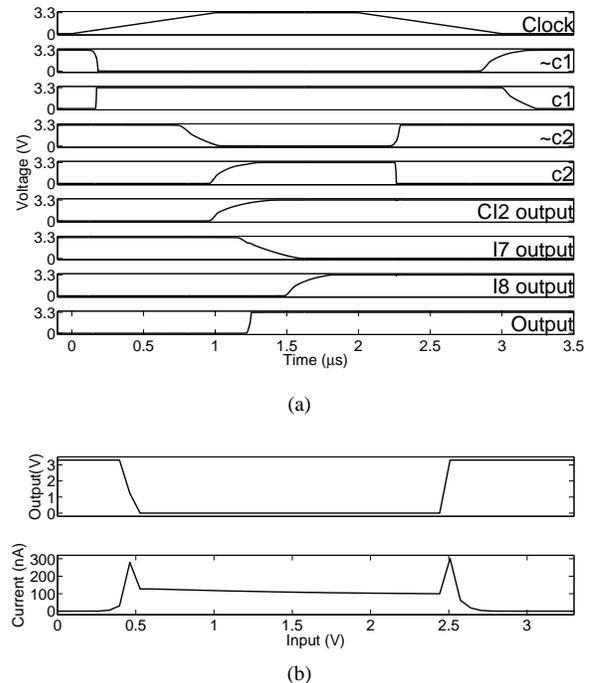


Figure 3. Results simulated using Spectre, without extracted parasitics. (a) DFF: For the choking transistors, the same 50nA biases were used as in figure 1; the DFF was initialised with a low output value; the input was held high; the output had no load; the clock was ramped up over a period of $1\mu\text{s}$, held high for $1\mu\text{s}$, and then ramped down over $1\mu\text{s}$; c1 and its complement changed state shortly after the beginning of the ramp, whereas c2 didn't change until near the end; c1 rose quickly and fell slowly whereas c2 rose slowly and fell quickly; the new state, already present in the master latch, passed through Cl2 immediately following the rise of c2; I7 then fell and allowed the output to rise to its new level. finally I8 rose, stabilising the slave latch; thereafter it was safe for the clock to fall. (b) CLB configured as saturation detector: A 2-input CLB was programmed with the XNOR function and biased with the same 50nA biases as above, and both of its inputs were swept with the same input voltage between the rails (0-3.3V); the output was high except for the central range of voltages; the current consumption is low at the edges but around 100-130nA in the central range, mainly due to the input inverters passing their limited internal currents.

- [4] D. Badoni, M. Giulioni, and V. Dante, "An aVLSI recurrent network of spiking neurons with reconfigurable and plastic synapses," in *ISCAS*, 2006.
- [5] J. Schemmel, J. Fieres, and K. Meier, "Wafer-scale integration of analog neural networks," in *International Joint Conference on Neural Networks (IJCNN)*, 2008.
- [6] T. Giurma and A. Ebenal, "Programmable hardware and the new analog capacity," in *Systems, Second International Conference on*, pp. 19–24, 2007.
- [7] P. Chow, P. Chow, and P. Gulak, "A field-programmable mixed-analog-digital array," in *Field-Programmable Gate Arrays, Proceedings of the Third International ACM Symposium on*, 1995.
- [8] G. Indiveri, "A low power adaptive integrate-and-fire neuron circuit," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, pp. 820–823, 2003.
- [9] S. Brown and J. Rose, "FPGA and CPLD architectures: a tutorial," *Design & Test of Computers, IEEE*, vol. 13S, no. 2, pp. 42–57, 1996.
- [10] C. Mead and T. Delbruck, "Scanners for visualizing activity of analog VLSI circuitry," *Analog Integrated Circuits and Signal Processing*, vol. 1, pp. 93–106, 1991.
- [11] R. Pasqualini, "Flipflop that can tolerate arbitrarily slow clock edges," 2007 US Patent 7265599.
- [12] I. Kuon, R. Tessier, and J. Rose, "FPGA architecture: Survey and challenges," *Foundations and Trends in Electronic Design Automation*, vol. 2, no. 2, pp. 135–253, 2007.