Communications Infrastructure for Neuromorphic Analog VLSI Systems

Rodney Douglas and Misha Mahowald and Adrian Whatley

August 1, 1998

Institute of Neuroinformatics, ETH/UNIZ, Gloriastr 32, 8006 Zürich.

Correspondence: Prof. Rodney Douglas
Institute of Neuroinformatics
Gloriastr 32
CH-8006 Zürich
Switzerland

Telephone: +41 1 257 2661
Fax: +41 1 257 6983
Email: `rjd@neuroinf.ethz.ch`

1 Abstract

Models of the peripheral sensory structures of organisms such as retinas and cochleas have been successfully embodied in single neuromorphic analog Very Large Scale Integration (aVLSI) chips. However, the amount of computation that can be performed by a single chip is limited. The construction of large neuromorphic systems requires a multi-chip communication framework optimized for neuromorphic aVLSI designs. One such framework is an asynchronous multiplexing communication network based on address event data representation (AER). In AER, analog signals from the neurons are encoded by pulse frequency modulation. These pulses are abstractly represented on a communication bus by the address of the neuron that generated it. However, it is still the timing of the address-event that communicates analog information. The multiplexing used by the communication framework attempts to take advantage of the greater speed of silicon technology over biological neurons to compensate for more limited physical connectivity. The AER provides a large degree of flexibility for routing digital signals to arbitrary physical locations.

2 Introduction

Neuromorphic engineering applies the computational principles discovered in biological organisms to those tasks that biological systems perform easily, but which have proved difficult to do using traditional engineering techniques. These problems include visual navigation, auditory localization, olfaction, recognition, compliant limb control, and locomotion.

The computational principles that biology employs are still only incompletely understood. For this reason, neuromorphic engineering is closely linked to biological research, particularly research in computational neuroscience. The greatest successes of neuromorphic aVLSI to date have been in the emulation of sensory transduction and processing of the kind performed by biological retinas and cochleas. The sensory periphery is a logical place to begin an analog neuromorphic system, since the light impinging onto a retina or sound waves entering the cochlea are all continuous analog signals. Furthermore, these structures are easily accessible to neurobiologists and their purpose is obvious, at least in the general sense, so a great deal is known about their biology.

Models of peripheral sensory structures such as retinas and cochleas have been successfully embodied in single neuromorphic analog Very Large Scale Integration (aVLSI) chips . However, the amount of computation that can be performed on a single chip is limited by silicon area, and external access to the inputs and outputs of the computation. The challenge of the moment is to overcome these limitations by developing multi-chip systems, and now the first multi-chip neuromorphic systems are being constructed. Typically, these systems use modifications of previously designed sensory chips as input devices to multi-chip processing systems. Multi-chip systems require a communication framework optimized for neuromorphic aVLSI designs as a foundation for their interaction. In this chapter we discuss some of the issues that must be considered in designing the communication frameworks.

3 Single Chip Networks

The stimuli impinging on an organism are generally continuous analog signals. The extraction of behaviourally relevant information from this sensory data is a natural starting point for neuromorphic aVLSI design. Fortunately, the processing performed at the periphery is quite simple. The processing is stereotyped, and usually involves reporting the state of a given receptor with reference to a local average over similar receptors. This regularity implies that peripheral sensors are usually composed of many similar elements, laid out in regular spatial arrays of one or two dimensions. A high proportion of the interactions that they have with other elements are confined to near neighbours. These arrangements simplify considerably an artificial implementation. For example, a 'silicon retina' that emulates the outer plexiform layer of the mudpuppy was one of the earliest attempts to use aVLSI technology to emulate the principles of biological sensory processing. The silicon retina includes circuits that represent the photoreceptor, horizontal cell, and bipolar cell layers of the retina. Unlike conventional CCD cameras, the silicon retina responds to contrast rather than absolute luminance and so is able to report comparable detail in shaded and bright areas of the same scene. Comparison of a particular receptor response against a spatially weighted average of its neighbours has the effect of suppressing redundant features of the image, while enhancing the novel ones . This simple but useful collective processing depends only on a regular connectivity with near neighbours. These designs are relatively cheap in terms of silicon area by contrast with more general circuits that also require longer range connections.

Quite large circuits with only local connectivity can be implemented on a single silicon die. More central neural structures like the neocortex which compute more complex functions require denser and more general connection patterns. In silicon, this usually means that the computational resources of many chips must be integrated into a single system. An interchip communication that can bind the system is critical to the development of large scale neuromorphic aVLSI systems, which must inevitably distribute their functionality over multiple chips. The development of a common method for interchip communication will enable the growing community of aVLSI researchers to build systems using chips designed in different laboratories, and so accelerate the maturation of complex systems in aVLSI.

4 Inter-chip Communication

The high degree of connectivity and the strict real-time performance required by neural processing systems demands a different approach to the problem of interchip communication from that used in traditional digital computers. Typically, digital computers have relatively few processing elements. The variables which they must process are stored in remote memory accessed via a common communication bus, and they are shipped serially across the bus, to and from the processor, as the algorithm proceeds. By contrast, in analog VLSI systems the many computations are implemented locally and distributed spatially over the entire surface of the silicon. Variables are voltages on capacitors, or currents in wires in the vicinity of the computational structures. The computations occur in parallel, and are densely distributed in space. This organization leads to a hard problem of spatial and temporal communication between the many computational nodes, particularly if the nodes are distributed across different chips.

Designers of hardware artificial neural networks have adopted several methods for communication between nodes on the same chip, and across different chips (Fig. 1) . Each method has advantages and costs, and the correct choice depends on which factors are most crucial to the application.

< Figure 1 >

4.1 Hard-wired Connections

This method provides direct, hard-wired connections between computational nodes that can be used to transmit continuous signals, or events such as pulses or action potentials. The wired connections can be permanent, or reconfigurable . Reconfigurable connections are preferred for most research applications, because neuromorphic systems are still largely experimental and so complete and exact connections cannot always be specified at the time that the chip is fabricated. In general, it is impractical to build permanent connections for anything but the most well-defined functions because design and fabrication is quite expensive, and the development and fabrication cycle is typically a few months.

One of the most literal approaches to interconnecting silicon neurons has been adopted by Mueller. They use a direct physical connection between nodes on different chips via cross-bar switches. (A cross-bar switch to connect N channels consists of a square $N \times N$ array of individually controlled switches, each of which can connect input on one channel to output on another. Contention can of course arise if multiple input channels need to connect to the same output channel, this must be resolved in some way.) The major advantages of this approach are that it allows continuous-time communication between nodes, and the switching arrays provide flexible connectivity that can be programmed digitally by a host computer. The system is able to handle larger connectivities than are possible on a single chip because the dendrites of a single artificial neuron can be extended over multiple chips. The disadvantage of this approach is that it requires many chips to model even a small number of neurons. At present the number of artificial neurons on each output chip is limited to less than 1000 by the number of input and output pins available. However, new methods are being developed that will allow direct connections between the surfaces of adjacent chips ('chip-stacking'), permitting construction of quasi three-dimensional circuits.

4.2 Virtual Connections

Since technological limitations preclude completely connected systems, some form of multiplexing of a limited number of channels must be used to compensate for low wiring density. Fortunately, the speed advantage of VLSI technology over biological neurons can be exploited in this context. VLSI signal propagation times (and hence notional clock rates) can be about $10^5 - 10^6$ times faster than those of biology. But neuromorphic systems are intended to interact with the real world and so their dynamics should evolve concurrently with external events. Slowing their behaviour to speeds comparable with biological neurons, makes the large excess bandwidth available for communication by time-multiplexing connection paths. This communication strategy attempts to achieve in time what the axonal arborizations achieve in space.

The multiplexing approach can be applied in a number of ways. A single channel can connect all chips together using a packet-switching approach in which packets of one or more words can be routed along a single bus by virtue of the address the packets contain. (In Address-Event Representation, packets may consist of only an address, see section 5.2.) More elaborate interconnection systems could be composed out of multiple, possibly circuit-switching channels in which separate point to point source to destination links (cicuits) are temporarily established, e.g. by means of cross-bar switches. Given packet switched channels, which may themselves be subject to circuit switching, a variety of network topologies can be established, for example linear, star, and ring topologies, two, three or higher dimensional meshes such as hypercubes, or any kind of hybrid of these topologies that might be desired.

The physical channel could be implemented on a parallel or bit-serial bus. Bit-serial communication is very popular in the digital industry because (amongst other reasons) the address space is easily scaleable. The number of bits that can be carried over a bit-serial bus can be increased merely by increasing the word length being used. Hardware changes at every node are not necessarily required, as they are if the number of bits carried on a parallel bus is to be increased. A disadvantage of bit-serial compared with parallel buses is the greater handshaking complexity. For this reason, aVLSI communication systems using the multiplexing approach have so far used parallel buses.

5 Multiplexing Pulses

5.1 Polling

Traditional time-division multiplexing schemes are serial access. Each computational node is polled in a fixed sequence, and its output is sent off chip. Each time slot is allocated to a particular node, and the receiving node must be synchronized with the sending device so that the identity of the source node is preserved. Most multiplexing schemes rely on a global clock to perform this synchronization. The timing signals from this clock may become skewed if the participating chips are too far from one another.

The choice of multiplexing technique depends on how the neural elements in the system encode information. In systems with analog valued outputs these outputs can be multiplexed onto a single wire by some form of sequential polling. This approach is particularly useful for interacting with video equipment, which is designed to work with analog-valued image frames composed of sequentially polled (scanned) pixels . However, analog data transfer is difficult between chips, in part because the analog data are easily perturbed by noise resulting from the multiplexing. Even more important are variations in the parameters of fabrication on different wafers, across a single wafer, and even across a single chip. Parameter mismatch causes different chips to have disparate interpretations of analog voltages. Some of these difficulties are avoided by transmitting digitally encoded signals which do not depend upon an analog reference voltage.

Both synchronous and asynchronous techniques have been used to time multiplex digital amplitude data. Digital signal transmission can be extremely fast because it does not require the settling time necessary for an analog amplifier. Furthermore, digital signals are noise resistant and are independent of variations in fabrication parameters. On the other hand, synchronous transmission of multiple bits of information has the drawback that synchronous switching of many elements causes noise (cross-talk) which can affect the performance of analog systems.

The temporal efficiency of a traditional, sequentially scanned data multiplexing system is easy to evaluate because it is an exclusive property of the machine, and not a property of the data. For this we make the assumption that the data occur at random within a frame scan period, so the average error introduced by waiting to read a particular data item is one half of the frame scan time. The scan time increases linearly with the number of elements in the array.

Murray and Tarassenko use an interesting variant of polling, in which the signal transmitted is a digital pulse whose width represents the interval between action potentials, and thus the state of activation, of the source neuron. This is a form of pulse width modulation (PWM). The neurons on the transmitting chip are polled sequentially. At a polling access, a digital circuit converts the last interval between neural pulses into a proportionally wide digital pulse that is transmitted to the receiving chip. The receiver converts the pulse width back to an analogue voltage by using the received pulse to discharge a pre-charged capacitor through a constant current sink. This analogue voltage is demultiplexed onto a storage node of the target neuron on the receiver chip, which regenerates the pulse stream of the source neuron.

The polling sequence is synchronized between the chips by using an asynchronous handshake pair of signals (request to send, and ready to receive). A time-out circuit deals with over-long intervals that occur when the source neuron activation is low. Unfortunately, in typical neural networks, only a small fraction of the neurons are active at any time. Thus the pulse stream polling method spends a disproportionate amount of time waiting for a time-out on those neurons. Moreover, the handshake used to synchronize polling limits transmission to single sender - single receiver connections, whereas practical neuromorphic systems require multiple sender - multiple receiver communications.

5.2 Address-Event Representation

The interchip-communication protocol that we have developed is an asynchronous digital multiplexing technique that uses the *address-event representation* (AER). It has the characteristics of event-driven pulse-frequency modulation in which the address of the node which is the source of an event is broadcast during the pulse, to all computational nodes within a defined region. The nodes decode the broadcast neuronal addresses.

Like neuronal action potentials, events in this system are stereotyped digital events, and the interval between events is analog. Each digital event is a digital word representing the identity of the neuron that is generating an action potential. It is placed on the common communications bus, which is effectively a universal axon, at the time the action potential is generated. Thus, information is encoded in the temporal pattern of events.

The savings in the number of wires required for communication between neurons is due to the replacement of N axonal fibres, with one active at a time, by $(1 + \log_2 N)$ wires, which are simultaneously active. However, in a real nerve bundle, several axons may be simultaneously active. We can accommodate this condition by making the event duration very short (of the order of 1 μs) compared with the width of neural action potentials (of the order of 1 ms). These short-duration events are less likely to overlap. Since, as in a real neuron, the maximum firing rate of a node is limited, even if events from several nodes did occur synchronously, they could be arranged such that they occurred in close succession with little loss of information.

The address-event representation is illustrated in Fig. 2. The neurons in the sender array generate a temporal sequence of digital amplitude events to encode their output — a representation conceptually equivalent to a train of action potentials. Each neuron is associated with a digital address that uniquely identifies it. Whenever a neuron signals an event, the encoding circuitry broadcasts that neuron's address on the interchip data bus. The nodes have a refractory time that limits the frequency at which they can issue events and, like their biological counterparts, only a small fraction of the silicon neurons embedded in a network are generating action potentials at any time. The inter-event interval at a neuron is much longer than is the time required to broadcast the neuron's address. Therefore, events from many neurons can be multiplexed on the same bus. The receiver interprets the broadcast of the address as an event that corresponds to the occurrence of an action potential from the neuron identified by that address. In a neuronal implementation, the receiving nodes are typically synapses. Those synapses that are 'connected' to the source neuron detect that their source neuron has generated an action potential, and they initiate a synaptic input on the dendrite to which they are attached.

$<$ Figure 2 $>$

Neuronal events can be broadcast and removed from the data bus at frequencies of about one megahertz. Therefore, about one thousand address-events (AE) can be transmitted in the time it takes one neuron to complete a single one millisecond action potential. If, say, 10% of neurons discharge at 100 spikes per second, a single bus could support a population of about 10^5 neurons. The advantage of AER over traditional multiplexing such as scanners relies upon the sparse representations of data across the neural population.

Of course, many more than 10^5 neurons will be required to emulate even one cortical area. Fortunately, the limitation imposed by the bandwidth of the AER bus is not as bad as it may seem. The brain has a similar bandwidth constraint in space. If every axon from every neuron were as long as the dimensions of the brain, the brain would increase exponentially in size as the number of neurons increased. The brain avoids this fate by adopting a mostly local wiring strategy in which the average number of axons emanating from a small region decreases at least as the inverse square of their length . If the action potential traffic on the AER bus were similarly confined to a local population of neurons, the same bus configuration could repeat in space in an overlapping fashion, and so serve a huge total number of neurons.

AER bears a close resemblance to the action-potential representation that is the common coinage of communication in the nervous system. The advantages of using a digital-amplitude representation in biological systems and in silicon systems may be similar. In an event-based communication scheme, the amplitude of a signal is represented by the number and times of events. Time is the same everywhere in the system, and number is an abstract quantity that is also the same everywhere in the system. For example, a source signal may be at maximum value when generating 200 events per second. However, the actual voltage value to which this maximum corresponds may be specified independently for each unit in the system. Such an abstract encoding is useful because the source analog value is difficult to transmit reliably if the ground potential is not the same everywhere. In the nervous system the ground potential is the local potential of the extracellular fluid and is not uniform in different areas of the brain. This lack of a common ground is like transistor mismatch, which can be modelled to first order as offset voltages on transistor gates. In addition to reducing the influence of static offset, the problem of dynamic noise on the axonal wires is less significant if the signal they transmit is a fully restored digital pulse.

Although we have chosen to transmit only the digital-address event, it is possible in principle to use the address-event representation to transmit explicitly analog signals. In such a system, the address would be transmitted along with one or more analog values associated with the identified neuron. The neuron would make a request to transmit the analog data when a control signal indicated that a threshold had been passed.

The AER protocol has a number of advantages over sequential multiplexing. It preserves the temporal order of events taking place in the neural array as much as possible. Events are transmitted as they occur. AER has better temporal accuracy than standard sequential multiplexing because the bandwidth of the communication channel is devoted to the transmission of significant signals only. For example, an AER silicon retina generates address-events only at regions of the image where there is spatial or temporal change. In areas of uniform illumination, the pixels do not generate any action potentials, and so they do not contribute to the communication load. Finally, the AER scheme has the advantage that the mappings between source neurons and recipient synapses can be easily configured by inserting digital circuitry to map the output addresses to different lists of synaptic codes.

6 Address-Event Systems

6.1 Generating the Address-Events

The first stage in the AER system is the detection of events on the source chip and then transferring them onto the communication bus. Polling methods can be used to detect the events. This method is attractive with just a few thousand neurons, but polling is cumbersome for large numbers of neurons, for the reasons pointed out in section 5. The alternative approach is to service only those nodes that are generating events . This method is more efficient than polling if the nodes are sparsely active. However AER neuromorphic systems, like neurons, generate stochastically independent action potentials. Therefore the problem now becomes arbitration for service when collisions occur between nodes that are simultaneously active. Arbitration can guarantee safe event transmission but adds complexity to the chip design and time to the event cycle.

6.2 Arbitration

Mortara and Vittoz have proposed an unarbitered system in which colliding events collide in such a way that the resulting event is meaningless and is simply ignored. This method simplifies chip design considerably, provided some data loss can be tolerated. Accepting the loss is a reasonable strategy for asynchronous neuronal networks. The loss of a single event from a slightly active neuron is insignificant, and the loss of a single event from a strongly active neuron is also not significant because it will soon be followed by further events from the same source. Provided that the active neurons do not discharge synchronously and that the bus load is not too high, a reasonable sample of events will be transmitted successfully. If the activity load rises above about ten percent of the bus bandwidth, data loss by collisions becomes unacceptable. The unarbitered system is particularly useful where each sender and receiver is connected by a dedicated bus, so that there is only one sending chip on the bus. This requirement limits interchip connectivity to the number of buses available per chip. The decision whether to have a bus that can accept multiple senders and receivers is therefore critical to the final size and configuration of the system.

Mahowald and others, for instance Lazzaro, avoid collisions by handshaking and arbitration within a chip, but at the system level we use a broadcast method that relaxes the handshaking requirement such that senders do not have to wait to receive an acknowledgement from remote elements in a system. The fully arbitered approach maximises the average throughput, with a significantly higher utilization of the communication bus than the unarbitered system. However, the arbitration costs both time, and the space required to implement the additional logic circuitry. Fortunately, our arbiters scale well, in terms of both area and speed, as the size of the neuron array is increased.

The basic one-dimensional arbiter is a binary tree of simple arbiter elements (Fig. 3). For a linear array of size N, where N is a power of two, the total number of arbiter elements required is $N-1$. The entire arbiter occupies only a thin strip along one edge of the array. In the two-dimensional array case, arbitration is performed separately in the x and y directions, so an $N \times N$ array requires only $2(N-1)$ arbiter elements which are arranged along two edges of the array. Each arbiter element receives two input request lines from the lower level, and sends a single request line to the next higher level of the tree. Each element receives a single select line from above, and sends two select lines to the lower level. The job of each element is to choose one of the two incoming request signals, and to pass along the select from above to the chosen request. If the select is not received from above, then neither of the incoming requests is selected. Starting from a completely initialized state, the time required to complete the arbitration is determined by the amount of time required for a request to propagate to the top level of the tree and for the select to propagate back down. Arbitration occurs in parallel at each level of the tree, so the delay through the arbiter increases only slowly with the size of the array. The total delay through the arbiter is proportional to $\log_2(N)$.

$<$ Figure 3 $>$

Lande and co-workers take the middle path between full and no arbitration. They use partial arbitration of the nodes on the sender chip. Their method tries to balance timing errors against the number of collisions in order to preserve the significant statistical properties of the spike coding. This stochastic approach is based on the following principles. When collisions occur that are too coincident for the arbiter, they are immediately discarded by the colliding source nodes and no re-transmission is attempted. Otherwise the competing events are placed in a queue for transmission off-chip and are 'aged' while waiting for access to the communication bus. If they have not been transmitted by some maximum age, they too are discarded.

7 Event Timing and Delays

The main drawback of the time-multiplexing approach is the unavoidable timing uncertainty and delay in activating target synapses, introduced by arbitration, queuing, mapping, and fan-out. The experimental neurophysiological literature remains divided on the nature of the coding used for neuronal processing. One view is that information is encoded in the exact times of spike occurrence, but the resolution of this timing is not known. Psychophysical studies of visual and auditory processing suggest that submillisecond precision may be required. There is at present no physiolgical evidence for such precision at the level of single spike processing, so the psychophysical observations may depend on a population mechanism. The alternative view of coding is that neuronal information is encoded in the discharge rate of neurons, and that the rate is measured on a time scale of about ten milliseconds. On balance, in designing practical neuromorphic systems today, one should aim for precision of at least the order of one millisecond.

A second problem is the implementation of transmission delays between the source and destination nodes. In real nervous systems these delays are about 1–5ms; much longer than the transmission delays experienced by events traversing an AE bus. There are several strategies for implementing these delays. The simplest is not to deal with it at all, and assume that delays through the transmission process will be negligible. If there is concern about over-long delays resulting from dense collision arbitrations, then the delayed spikes can be neglected.

One practical method of implementing delays is to attach a second digital word to the address of the AE, and use it to hold a future-time stamp. The couplet is then transmitted on the AE bus as usual. A queue processor at the destination activates the target synapses at or near the exact time specified by the future-time stamp. Alternatively, the source event can be transmitted alone, and a constant delay can be added to every destination event when it arrives on the target chip or its associated mapper. A variant of this strategy can be used to make the delay dependent on the number of domains that the event crosses on route to its target. For example, DeWeerth uses fixed delays attached to his source events that accumulate at each segment of a structured multi-chip system. As events propagate along the segments of the system, the delay of the event is determined by its address and it is put in a time-sorted queue so that it emerges at the appropriate time.

None of these methods fully compensates for delays in arbitration times in the communication network. The only way to do this is to time-stamp the source event at the soonest possible moment, before any arbitration delays have accumulated. At that point, the event can be transmitted as quickly as possible so that the transmission delays in the system are likely to be smaller than the desired delay. The event can then be sent to its final destination at the correct time, irrespective of transmission delays in the system.

8 Routing

Neuromorphic systems are composed of distributed computational nodes. Some nodes only send events, others only receive events, and yet others do both. Often nodes of the same type are clustered on single chips. For example, retina chips are senders only, silicon neurons are senders and receivers, and motor drivers are typically receivers. In addition to the computational chips, the system might include a second class of chips that do not contribute directly to neuromorphic computation, but simply route events between the senders and the receivers. If these routing chips were programmable, then new networks could be constructed with the same basic chips. Of course, the function of the routing chip could be assimilated into the sender, or the receiver, or be distributed across both.

Time-multiplexing systems often use some variant of source event to destination event mapping to emulate the projective field of the the source neuron. The fan-out of this projection in biological neural networks can be very large, as much as 1:10,000 for some cortical neurons . It is impractical to build neuromorphic systems with such large fan-out at present. Values of a few hundred are the best that can be expected at present. Fortunately, there are many important problems that could be solved with networks having this degree of fan-out.

The number of nodes that can be supported by an AE bus (AEB) is limited (at present) to about 10,000. However, the number of neurons in an entire system can be much larger, because most of the connectivity of neurons in cortex is local, and so different AEBs can support adjacent populations of neurons in a manner analogous to the cortex. Neither brains nor neuromorphic systems require all-to-all connection of their computational nodes. Although any one node receives many inputs and distributes many outputs, these connections are a small fraction of the total number of computational nodes in the system. This raises the possibility of dividing the system into regions of relatively densely interconnected nodes, which we call domains.

Domains have a much greater number of connections between members of the domain than either efferent or afferent connections. In practical terms a domain comprises all the hardware necessary for activating the synapses within its borders. The number of nodes within each domain is limited by the maximum timing delay that can be tolerated for a particular application and the rate at which synapses can be activated by the domain hardware. A source event reaches the distant domain by passing through the intervening domains. The probability of connection to a particular destination domain decreases with its displacement (number of domain crossings) from the source domain.

The domain concept also provides for a mixture of hardware and AER communication. Connections that fall largely within the domain, and especially on a particular chip, could be hardwired. The domain strategy supposes some prior knowledge about the connection patterns between neurons. Although we do not know the detailed connection patterns of neurons, there are some regions, such as cerebral cortex, for which some average connection properties are known. In these cases the experimentally observed spatial distribution of connections could be used to specify intra-domain and inter-domain connection patterns.

Experimental investigation of cortical connectivity can contribute strongly to neuromorphic design by correlating the connectivity patterns with computational function. For example, anatomical and physiological data suggest that there is a circuit typical of cortex that is the biological equivalent of an engineering circuit like an operational amplifier . It is a general purpose circuit that is able to subserve a multitude of functions given slight changes in its input or configuration.

8.1 Source to destination mapping

The encoding of the AE may change as it passes through the system. Typically, the source address of the event can be mapped to the destination address of the event, so providing the final routing of the events to the appropriate presynaptic terminals (for example). This translation, or mapping, may occur at any convenient point in the transmission process.

Elias and co-workers have adopted a strategy in which the address of the target node is broadcast to all nodes. In this system a mapping table is stored on the board with the receiving chips on it. Individual synapses have different morphological properties, so each synapse must be individually addressed, leading to a large lookup table.

Each source gets a block of memory containing eight synapse addresses and/or links to other destination blocks. Two or more source events can be linked to the same destination block if projective fields overlap. Another possible saving on fan-out can be realized by simultaneously activating the same synapse on different chips. Arbitrary mapping to individual synapses is very memory intensive and would make it difficult to condense the source to synapse mapping circuitry within a chip of hundreds of computational nodes.

Since the purpose of the address event bus is to transmit events generated by source neurons to the desired synaptic destinations, the natural form of the mapping is to list the projective field of neuron x as a list of synapses $s_1, s_2, s_3, ..., s_n$. This raises the question of whether synapses on a particular chip should be addressed individually, or whether the chips should distribute a particular input with internal wiring. These two possibilities are illustrated in Fig. 4. The merits of these two strategies depend on the kind of connectivity that is desired. If connectivity is local, as in a retinotopic structure (Fig. 4 a), or fully global, as in a fully connected neural network (Fig. 4 b), then it makes more sense to distribute the event on chip. If connectivity is extremely sparse (Fig. 4 c and d), then addressing individual synapses becomes a more attractive strategy.

< Figure 4 >

It seems appropriate because of fan-out, that close to the sender, events will be source encoded, and be mapped to the multiple destinations as late as possible in a system, to conserve bus bandwidth. We allow source events to be transmitted between chips and boards. The source events are mapped to a destination event when they arrive at the target chip or board. At present we use a microprocessor to map source addresses to destination addresses, and each analog chip interfaces with two buses: a source event bus and a synaptic activation bus. The mapper transfers events from one bus to another while performing source to destination mapping. With large synaptic fan-out, traffic on the destination bus becomes the main factor limiting system throughput. Finally, it would be more convenient to incorporate the fanout mapping function onto the destination chips themselves, but the presence of the microprocessor that currently performs this function does facilitate house keeping tasks such as setting up mapping tables and logging AE traffic.

< Figure 5 >

The problems of routing events through a large system are complex. The destination table of source events must be known at some single point, or at a distributed set of points in the system. It can be argued that in the nervous system, sources may know the approximate area to which to send their axons, but the detailed information is kept locally. Whether the rough information should be explicitly stored in the communication network or is implicit in the structure of the buses is an open question. By waiting to translate source events until they reach the destination board, we place the detailed connection information as close to the destination as possible. We use an arbitrary sized block of memory to store all of the locations that the source event must be translated to. This list may include addresses in other domains as well as local synapse addresses.

If the address of the event is permitted to change as it propagates through the system, then the change of address can be used to indicate the route that the address should take or has taken. For example, in DeWeerth's segmented system, a suffix is tacked onto the source address that is incremented as the address propagates. When the suffix is sufficiently large, the event is discarded. This strategy allows receptive fields to be specified in local coordinates, rather than as a global set of addresses. The receptive field is specified for example as "connect to the fifth neuron two segments away from here". This local coordinate frame allows the system to be extended indefinitely providing that the local coordinate frame is small enough to fit into the address space.

9 Synapses

The synaptic couplings between neurons determine the function of the network, and are the site of learning. Most VLSI synapse circuits have been developed for artificial neural networks and simply express a coupling weight, without concern for the dynamic behaviour of their biological counterparts . Artificial learning synapses are in an early stage of technical development.

VLSI synaptic circuits fall into two broad classes, those that store their coupling weights locally, and those that receive their weights together with each presynaptic input. Local weight storage facilitates local learning, which can occur in parallel at each synapse. The synaptic weight can be stored digitally , or in analog form as the charge on a capacitor. Digital storage can be more precise than analog storage, depending on the number of bits required, but even for moderate precision (about six bits), digital storage requires a larger silicon surface area than analog storage. Furthermore, digital arithmetic rules for synaptic update are cumbersome compared to those that can be implemented on analog systems . Analog storage may be volatile , in which case it must be refreshed periodically by some external source, or it may be a semi-permanent charge stored on a well insulated conductor called a floating gate . Semi-permanent storage is usually preferable to volatile storage, but methods for updating the state of the floating gate are not yet flexible enough to implement many common learning rules .

The number of synapses per real neuron is large ($5 - 10 \cdot 10^3$), and the provision of synapses is the limiting spatial consideration in neuromorphic design. The synapses of individual neurons can be distributed across dendrites on multiple chips , but this approach raises the problem of transmitting accurately between chips the analog voltages and currents at the compartment boundaries. Alternatively, entire silicon neurons can be fabricated on the same chip, so that only robust, action potential-like events need be transmitted between chips. In this case the synapses of the neuron are also limited to a single chip, and their number becomes a trade-off between space and biological verisimilitude.

A large development chip has an area of roughly 100 mm^2. Using available fabrication technology, a reasonable target for such a chip would be about 64 neurons, each having about 16 dendritic compartments and a total of about 128 synapses with realistic dynamics. Clearly, a larger ratio of presynaptic inputs to neurons is required. The effective number of synapses can be increased by using the same synaptic circuits for multiple inputs. (This is similar to delivering action potentials from different neurons into the same synaptic bouton.) The synaptic weight for each input to the synapse need not be the same: the weights can be multiplexed with each action potential. This is fairly easily achieved if the action potentials are already being routed by a multiplexing communication method such as the AER. The disadvantage of multiplexing the synaptic weights is that it makes local learning rules more difficult to implement, because the synaptic weight must be updated at some distant storage site.

The particular synaptic implementation that is chosen may restrict the kinds of learning that can occur. Some learning rules used in artificial neural networks need gradient sensing and continuous variables. These algorithms require such precision that it seems unlikely that they could be implemented in aVLSI. The result is a paradox: the learning is described in analog terms, but can only be executed on precise digital computers .

10 Conclusion

As larger neuromorphic aVLSI systems are constructed, they exceed what can be accommodated on one chip. Therefore the need arises for an inter-chip communications infrastructure. In biological systems, most neurons communicate by means of axons that carry information about their levels of activation in a digital pulse rate (or frequency) modulated (PRM/PFM) form in which each pulse is an action potential. In aVLSI systems, silicon neurons may communicate over wires that carry information about their levels of activation also in a digital PRM/PFM form (if synchronous polling is not used). When several such signals are multiplexed together onto a multi-bit digital bus, the pulses form an address-event representation, although different groups use different arbitration and multiplexing schemes and describe their PRM/PFM systems using different terms.

Communicating in the digital domain with multiplexed signals brings several advantages. It helps to solve the wiring problem: that is the problem faced by both biological and aVLSI systems of wiring up elements with a high degree of connectivity. Biological systems have the advantage of using space very efficiently due to their small scale computational and wiring elements, and their ability to make full use of three dimensions. They are also arranged to maximise locality of connectivity. For aVLSI systems, communicating in the digital domain brings the advantage of using time very efficiently, with signal transition times being several orders of magnitude smaller than in biological systems. Thus, an aVLSI system can achieve high connectivity by exploiting time multiplexing whereas biological systems achieve the same aim in space. Using digital encoding between chips brings immunity to certain types of noise. Working with digital signals also enables the application of readily available technology to the propagation, mapping, fan-out and other processing of the signals. This in turn can make systems easily reconfigurable, and can facilitate the division of neuromorphic systems into domains of high local connectivity with lower long-distance connectivity, again analogous to the situation in biological systems.

There remains the problem of delayed signals and uncertainties as to when a signal will arrive at its destination or destinations. Various strategies can be adopted, but all (except the do-nothing option) are likely to rely upon manipulations (time-stamping, queuing, etc.) that can be relatively easily performed on digital information. Once multiplexed signals arrive at their destinations, they must be demultiplexed and restored to the analog domain if further processing is to be carried out in aVLSI. The options available for the synapse circuits that perform this restoration on target silicon neurons are still somewhat cumbersome, but work is proceeding to improve these synapses both in terms of their suitability for the implementation of learning algorithms and in terms of their size.

A reliable and well worked out communications infrastructure for multi-chip neuromorphic aVLSI systems is a necessary prerequisite for investigating the behaviour of any but the simplest of such systems.

Acknowledgements: We thank Kwabena Boahen, Steve Deiss, John Elias, and the NSF Neuromorphic Workshop at Telluride for many helpful discussions. The preparation of this chapter was supported by the Swiss National Science Foundation SPP Grant and the Office of Naval Research.

References

Abbott, L., J. Varela, K. Sen, and S. Nelson (1997). Synaptic depression and cortical gain control. *Science 275*, 220–224.

Benson, R. G. and D. A. Kerns (1993). UV-activated conductances allow for multiple time scale learning. *IEEE Trans. Neural Networks 4*, 434–440.

Churcher, S., D. J. Baxter, A. Hamilton, A. F. Murray, and H. M. Reekie (1993). Generic analog neural computation - the EPSILON chip. In S. J. Hanson, J. D. Cowan, and C. L. Giles (Eds.), *Advances in Neural Information Processing Systems*, Volume 5, pp. 773–780. San Mateo, CA: Morgan Kaufmann.

Diorio, C., P. Hasler, B. Minch, and C. Mead (1996). A single-transistor silicon synapse. *IEEE Trans. Electron Devices 11*, 1972–1980.

Diorio, C., S. Mahajan, P. Hasler, B. Minch, and C. Mead (1995). A high-resolution non-volatile analog memory cell. In *IEEE International Conference on Circuits and Systems*, Volume 3, pp. 2233–2236.

Douglas, R., C. Koch, M. Mahowald, K. Martin, and H. Suarez (1995). Recurrent excitation in neocortical circuits. *Science 269*, 981–985.

Häfliger, P. and M. Mahowald (In press). Spike based normalizing Hebbian learning in an analog VLSI artificial neuron. *Analog Integrated Circuits and Signal Processing Special Issue on Learning on Silicon.*

Hopfield, J. (1990). The effectiveness of analogue 'neural network' hardware. *Network 1*, 27–40.

Lansner, J. A. and T. Lehmann (1993). An analog CMOS chip set for neural networks with arbitrary topologies. *IEEE Trans. Neural Net-*

works 4, 441–444.

Lazzaro, J., J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie (1993). Silicon auditory processors as computer peripherals. *IEEE Trans. Neural Networks 4*, 523–528.

Mahowald, M. (1994). *An Analog VLSI System for Stereoscopic Vision.* Boston: Kluwer.

Marienborg, J., T. Lande, Å. Abusland, and M. Høvin (1996). An analog approach to "neuromorphic" communication. In *Proc. IEEE Intl. Symposium on Circuits and Systems (ISCAS'96)*, Volume III, pp. 397–400.

Markram, H. and M. Tsodyks (1996). Redistribution of synaptic efficacy between cortical pyramidal neurons. *Nature 382*, 807–810.

Mead, C. (1990). Neuromorphic electronic systems. *Proc. IEEE 78*, 1629–1636.

Mitchison, G. (1992). Axonal trees and cortical architecture. *TINS 15*, 122–126.

Mortara, A., E. Vittoz, and P. Venier (1995). A communication scheme for analog VLSI perceptive systems. *IEEE J. Solid-State Circuits 30*, 660–669.

Reyneri, L. M. (1995). A performance analysis of pulse stream neural and fuzzy computing systems. *IEEE Trans. Circuits and Systems - II: Analog and Digital Signal Processing 42*(10), 642–660.

Säckinger, E., B. E. Boser, and L. D. Jackel (1992). A neurocomputer board based on the ANNA neural network chip. In J. E. Moody, S. J. Hanson, and R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems*, Volume 4, pp. 773–780. San Mateo, CA: Morgan Kaufmann.

Van der Spiegel, J., C. Donham, R. Etienne-Cummings, S. Fernando, P. Mueller, and D. Blackman (1994, June28-July2). Large scale analog neural computer with programmable architecture and programmable time constants for temporal pattern analysis. In *Proceedings of the International Conference on Neural Networks*, Volume III, Orlando, FL, pp. 1830–1835. IEEE.

Vittoz, E., H. Oguey, M. Maher, O. Nuys, E. Dijkstra, and M. Chevroulet (1991). Analog storage of adjustable synaptic weights. In U. Ramacher and U. Rueckert (Eds.), *VLSI Design of Neural Networks*, pp. 47–63. Boston: Kluwer.

General References

Boahen, K. (1996). A retinomorphic visual system. *IEEE Micro 16*, 30–39.

Douglas, R., M. Mahowald, and C. Mead (1995). Neuromorphic analog VLSI. *Ann. Rev. Neurosci. 18*, 255–281.

Douglas, R. and K. Martin (1998). Neocortex. In G. Shepherd (Ed.), *The synaptic organization of the brain* (4th ed.)., Chapter 12, pp. 459–509. New York: Oxford University Press, Oxford.

Mahowald, M. and C. Mead (1991, May). Silicon retina. *Sci. Amer. 264* (5), 76–82.

Mead, C. and M. Ismail (Eds.) (1989). *Analog VLSI Implementation of Neural Systems*. Boston: Kluwer.

Mead, C. A. (1989). *Analog VLSI and Neural Systems*. Reading, Massachusetts: Addison-Wesley.

Murray, A. and L. Tarassenko (1994). *Analogue Neural VLSI*. London: Chapman and Hall.

Figure 1: Taxonomy of interchip communication. Two important dimensions of interchip communication are shown, flexibility of connectivity running generally vertically, and the division between continuous, analog variable transmission on the left, and discrete, digital transmission on the right. Within Address-Event representation, which is a form of pulse rate, or pulse frequency modulation (PRM/PFM), a third dimension, the degree of arbitration is shown.

Figure 2: The address-event representation. Self-timed neurons on the sending chip generate trains of action potentials. The neurons request control of the bus when they generate action potentials and are selected to have their addresses encoded and transmitted by the multiplexing circuitry. A temporal stream of addresses passes between the sender chip and the receiver chip. This temporal stream is decoded by the receiver into trains of action potentials that reach their proper postsynaptic targets. The relative timing of the events is preserved.

Figure 3: An arbiter, a binary tree of two-input arbitration cells. Each arbitration cell (drawn as a trapezium) receives two request lines R_1 and R_2 from below. When it receives a request on one of these lines it transmits a request to the next higher level of the tree on R_{out}. When it receives an acknowledge signal from above on line S_{in}, it passes the signal down the tree on one of the two acknowledge lines S_1 or S_2. The request output from the top-most arbitration cell is connected directly back to its acknowledge input.

Figure 4: Mapping events to synapses. This figure illustrates the two possibilities of addressing synapses on a particular chip individually or of distributing a particular input with internal wiring. The merits of these two strategies depend on the kind of connectivity that is desired. If connectivity is local, as in a retinotopic structure (**a.**), or fully global, as in a fully connected neural network (**b.**), then it makes more sense to distribute events on chip. If connectivity is extremely sparse (**c.** sparsely connected network with each input mapped to its own synapse, or **d.** sparsely connected network with multiple inputs sharing the same synaptic terminal), then addressing individual synapses becomes a more attractive strategy. Long grey rectangles represent address decoders; grey squares represent neuromorphic elements (silicon neurons or pixels); grey circles represent synapses.

Figure 5: Fan-out. **a.** and **b.** illustrate how bandwidth is conserved if fan-out occurs as late as possible in a system (near to the destinations, **a.**) compared with the case in which fan-out occurs near the source, **b.**. Our current method of performing the fan-out using a microprocessor (μP) and our finally prefered method of performing the fan-out on the destination neuromorphic chip are shown in **c.** and **d.** respectively.

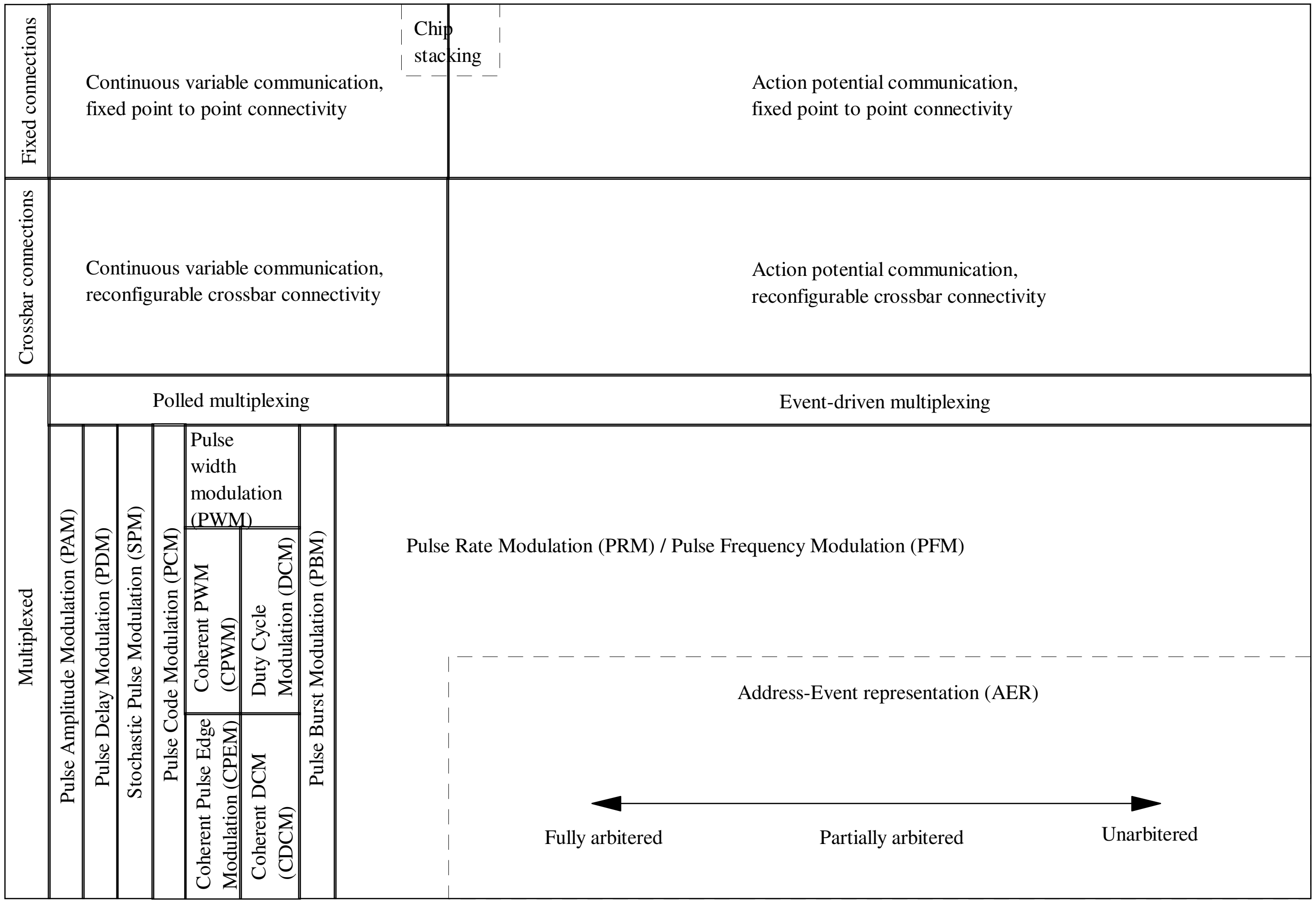
Fixed connections
Crossbar connections
Multiplexed
Continuous variable communication,
fixed point to point connectivity
Chip
stacking
Action potential communication,
fixed point to point connectivity
Continuous variable communication,
reconfigurable crossbar connectivity
Action potential communication,
reconfigurable crossbar connectivity
Polled multiplexing
Event-driven multiplexing
Pulse Amplitude Modulation (PAM)
Pulse Delay Modulation (PDM)
Stochastic Pulse Modulation (SPM)
Pulse Code Modulation (PCM)
Pulse width modulation (PWM)
Coherent PWM (CPWM)
Duty Cycle Modulation (DCM)
Coherent Pulse Edge Modulation (CPEM)
Coherent DCM (CDCM)
Pulse Burst Modulation (PBM)
Pulse Rate Modulation (PRM) / Pulse Frequency Modulation (PFM)
Address-Event representation (AER)
Fully arbitered
Partially arbitered
Unarbitered

Figure 1:

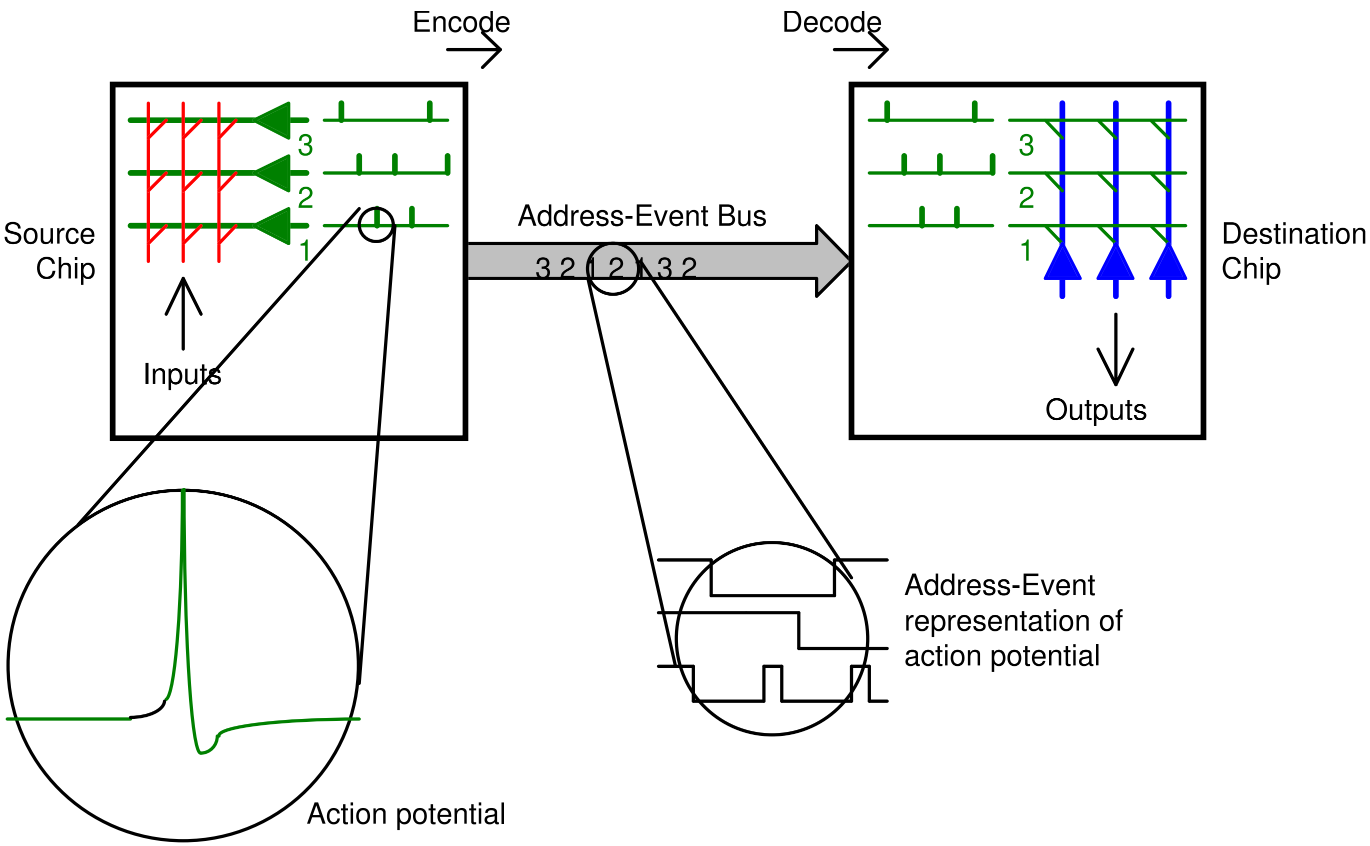
Encode
Decode
Source
Chip
Inputs
3
2
1
Address-Event Bus
3 2 1 2 1 3 2
3
2
1
Destination
Chip
Outputs
Action potential
Address-Event
representation of
action potential

Figure 2:

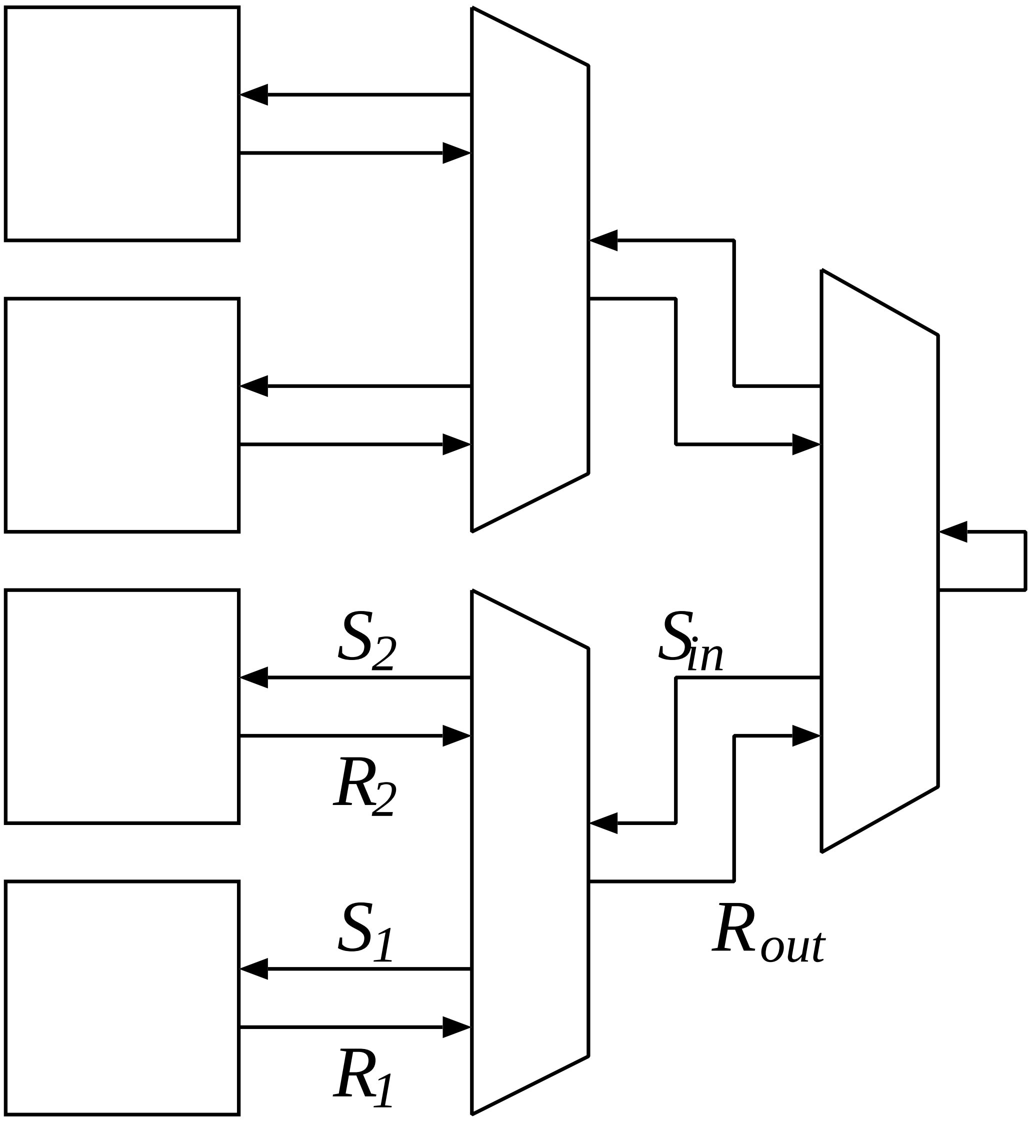

Figure 3:

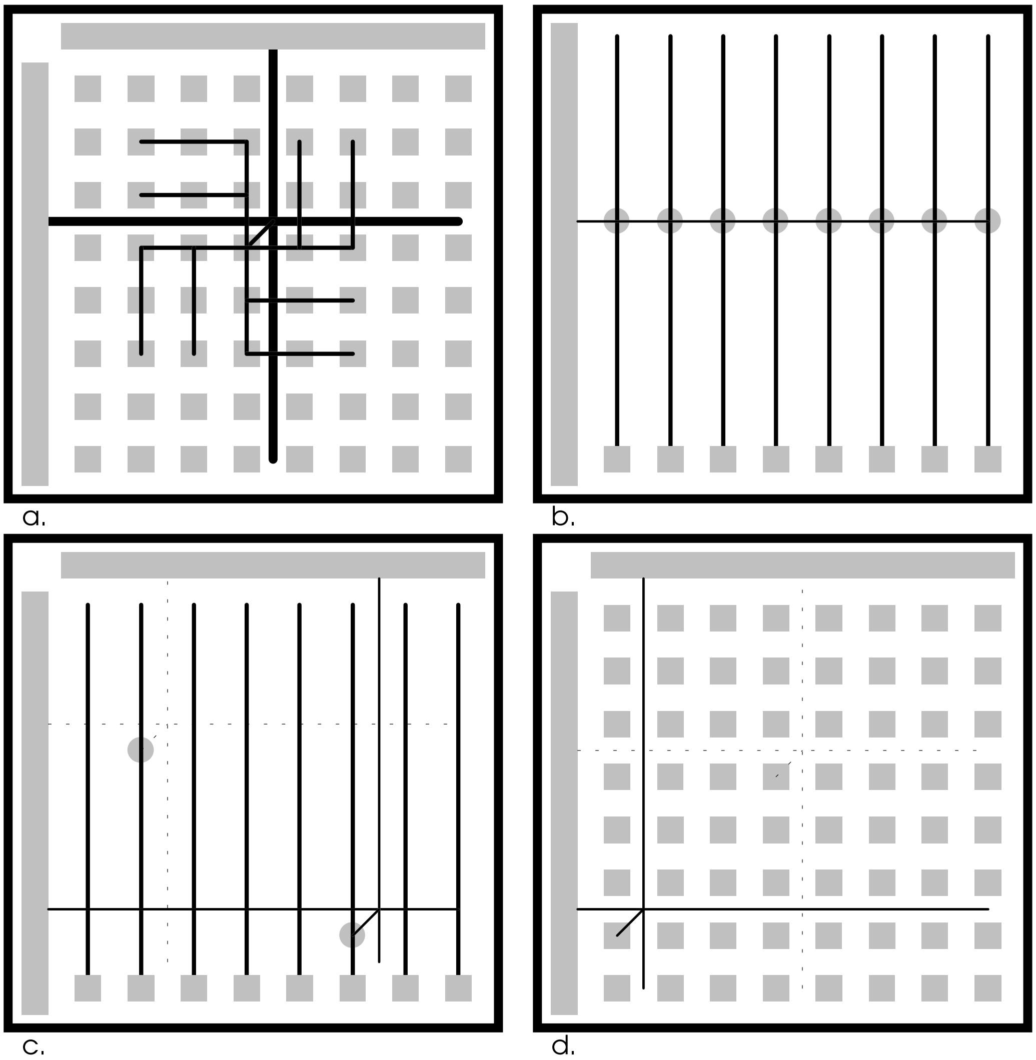

Figure 4:

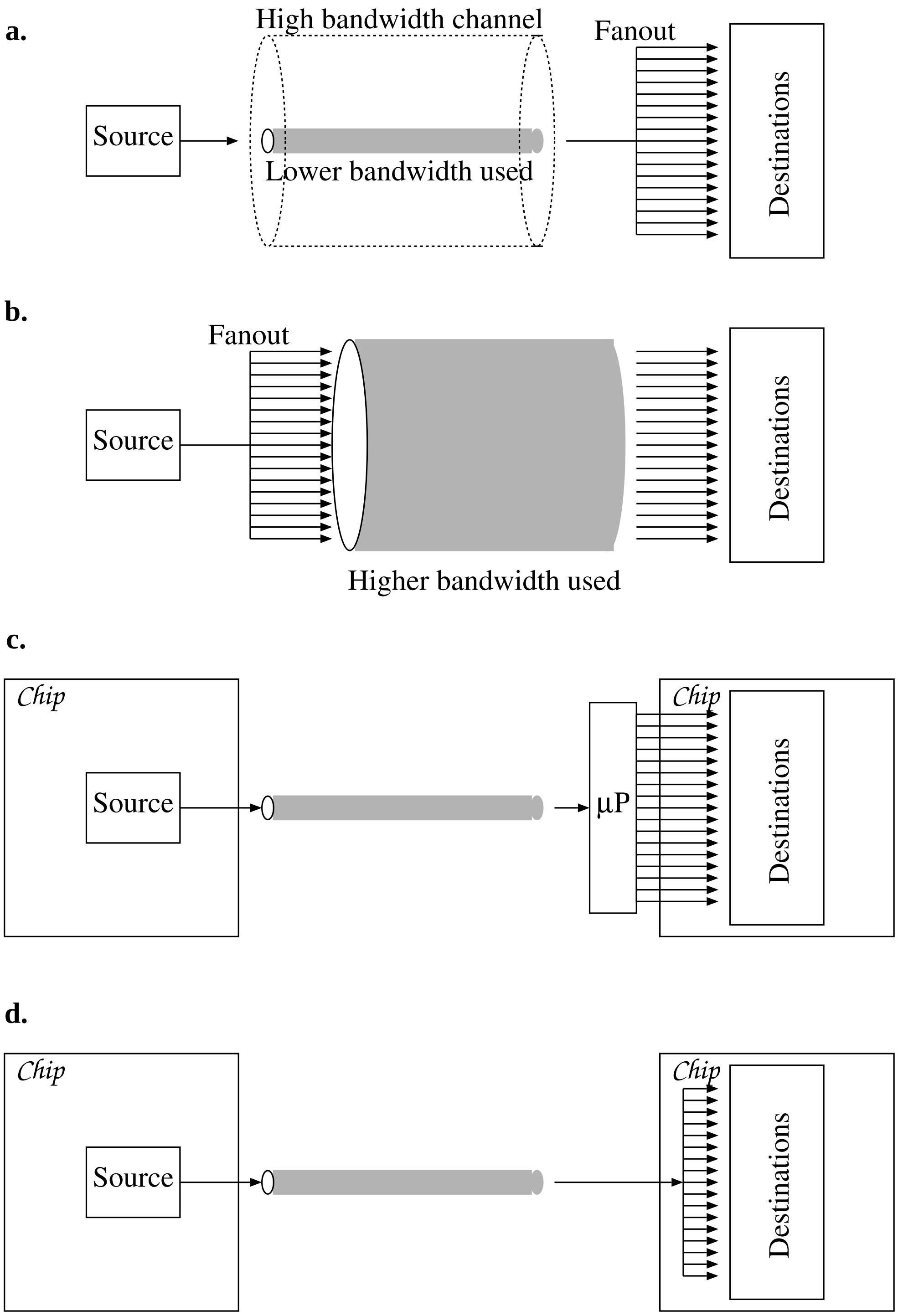
a.
High bandwidth channel
Fanout
Source
Lower bandwidth used
Destinations
b.
Fanout
Source
Destinations
Higher bandwidth used
c.
Chip
Source
μP
Chip
Destinations
d.
Chip
Source
Chip
Destinations

Figure 5: