AppNote 8 - Interfacing AER devices to SpiNNaker using an FPGA

SpiNNaker Group, School of Computer Science, University of Manchester

Luis A. Plana - 03 Apr 2014 - Version 1.1

Introduction

This application note describes the implementation of an interface between Address-Event Representation (AER) devices and the SpiNNaker system using an FPGA.

AER interface

The Address Event Representation (AER) interface is an asynchronous interface that implements a 4-phase handshake protocol with active-low request ($\overline{REQ}$) and acknowledge ($\overline{ACK}$) signals. Events are transmitted in parallel 16-bit words. The logic levels 0 and 1 are represented by voltages of 0V and 5V respectively, and most devices will operate correctly if driven by 3.3V signals. Additional information can be found in [1].

SpiNNaker link interface

The SpiNNaker interface is also an asynchronous interface but implements a 2-phase handshake protocol. This protocol, also known as Non-Return-to-Zero (NRZ), uses signal transitions, as opposed to levels, to represent data. Events are transmitted as 40-bit packets which are sent serially in 4-bit flits starting with the least significant bits. The logic levels 0 and 1 are represented by voltages of 0V and 1.8V respectively and will tolerate being driven by 2.5V and 3.3V signals. Most 2.5V devices will operate correctly when driven by SpiNNaker signals. Additional information can be found in [2].

Mapping AER Events to SpiNNaker Multicast Packets

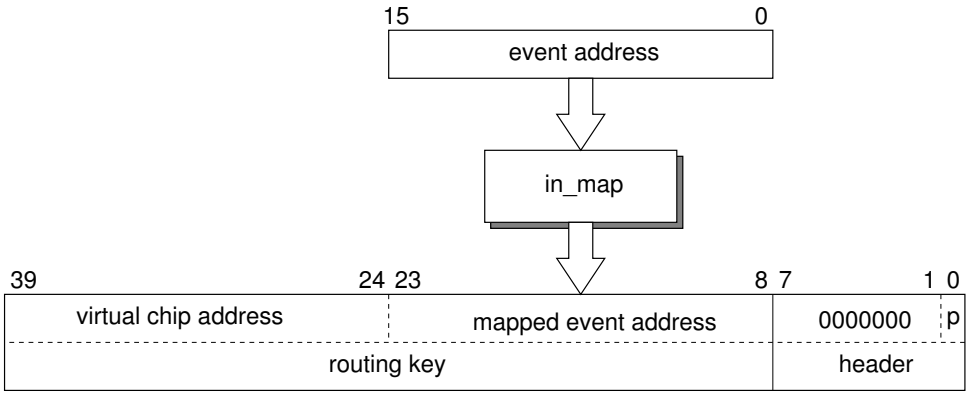

Figure 1: Mapping events to SpiNNaker packets.

As mentioned above, events in SpiNNaker are transmitted as 40-bit packets. Packets consist of a short 8-bit header, which includes a parity bit (bit[0]), and a 32-bit routing key. The routing

key is used by the SpiNNaker routers to deliver the packets to the correct destinations. As is the case for the AER devices, the routing key represents the origin of the event and not its destination or destinations.

In order to map an AER event to a SpiNNaker packet routing key, the AER device is treated as if it was a SpiNNaker chip generating events and is assigned a "virtual" chip address. This address must not be shared with existing chips in the SpiNNaker system. Once the address is selected, a packet can be constructed as shown in Figure 1. The 16-bit event is mapped according to the characteristics of the AER device and the way it is treated in the software.

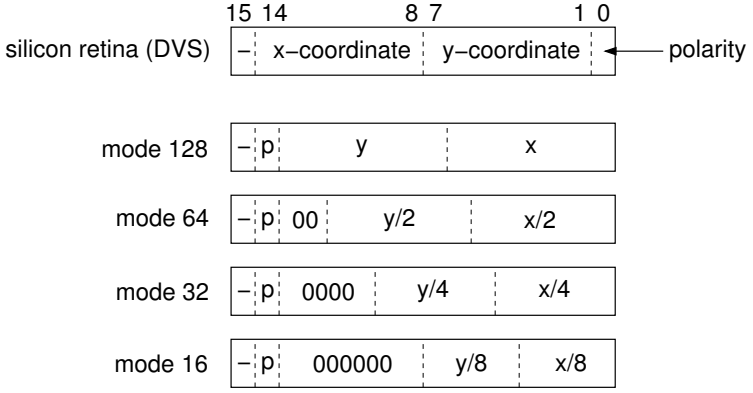

Figure 2: Silicon retina mappings.

Figure 2 shows several mappings used in the example presented below. The figure shows, at the top, the event sent by an AER silicon retina with 128 x 128 pixels. Four different mappings are shown below it, corresponding to 4 different sampling options: no under-sampling (128 x 128 pixel resolution), four neighbouring pixels grouped together (64 x 64 pixel resolution), groups of 16 pixels (32 x 32 pixel resolution) and, finally, groups of 64 pixels (16 x 16 pixel resolution).

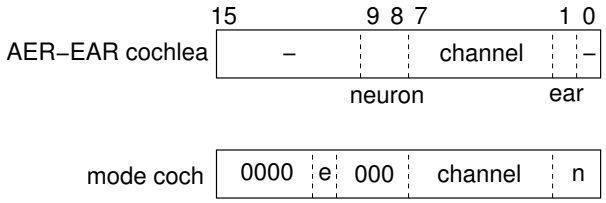

Figure 3: AER-EAR cochlea mapping.

Figure 3 shows an AER-EAR cochlea event and how it is mapped into a SpiNNaker package. The cochlea has 2 ears with 64 channels associated with each ear. Every channel contains 4 neurons. The retina and cochlea mappings have been chosen to simplify the construction of neuron populations and their placement on SpiNNaker cores.

Mapping SpiNNaker Multicast Packets to AER Events

As indicated earlier, SpiNNaker events are transported as 40-bit SpiNNaker multicast packets. The SpiNNaker packet routing key, which is used to deliver the packet to its destination, represents the origin of the event and not its destination. This key usually consists of two parts: the address of the SpiNNaker chip that generated the event and an event identifier, as shown in

Figure 4. The 16-bit event identifier should be mapped according to the characteristics of the AER device and the way it is treated in the software. The most common mapping is a direct map.

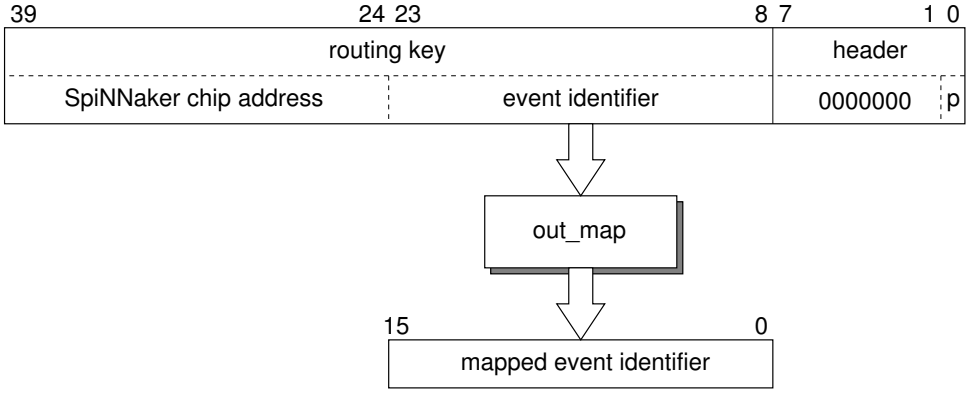

Figure 4: Mapping SpiNNaker packets to events.

Example Implementation using a Xilinx FPGA

The mappings shown above are used in a SpiNNaker–AER interface implemented on a Xilinx Spartan6 FPGA, using a RaggedStone2 board [3]. The board has two banks of I/O pins, one set at 3.3V and the other at 2.5V. The 3.3V bank is used to connect to the AER devices and the 2.5V bank is used to connect to the SpiNNaker board.

Figure 5 shows a block diagram of the implementation. The basic block in the AER-to-SpiNNaker path is the *in_mapper*, which implements the mappings shown in Figures 2 and 3 above. The constructed 40-bit SpiNNaker packet is passed to the *SpiNN driver* module to be serialised and sent through a SpiNNaker link.

The SpiNNaker-to-AER path consists of a *SpiNN receiver* module that deserialises incoming SpiNNaker packets and passes them to the *outmapper* which delivers the mapped event identifier to the AER device.

It is important to note the use of synchronisers for the incoming handshake signals in both interfaces. These asynchronous signals must be synchronised to the FPGA clock to avoid metastability. In this case, simple 2-flop synchronisers are used.

device	function
sw1	mode and virtual address selection button
sw2	reset button
led2	activity indicator (flashes continuously)
led3	reset indicator
led4	dump mode indicator
7-seg display	mode and virtual chip address indicator (optional address indicated by decimal point on)

Table 1: RaggedStone2 board user interface.

A *mode select* module allows the selection of the operating mode by the user, which receives visual information through a 7-segment display, driven by the *display driver* module. The func-

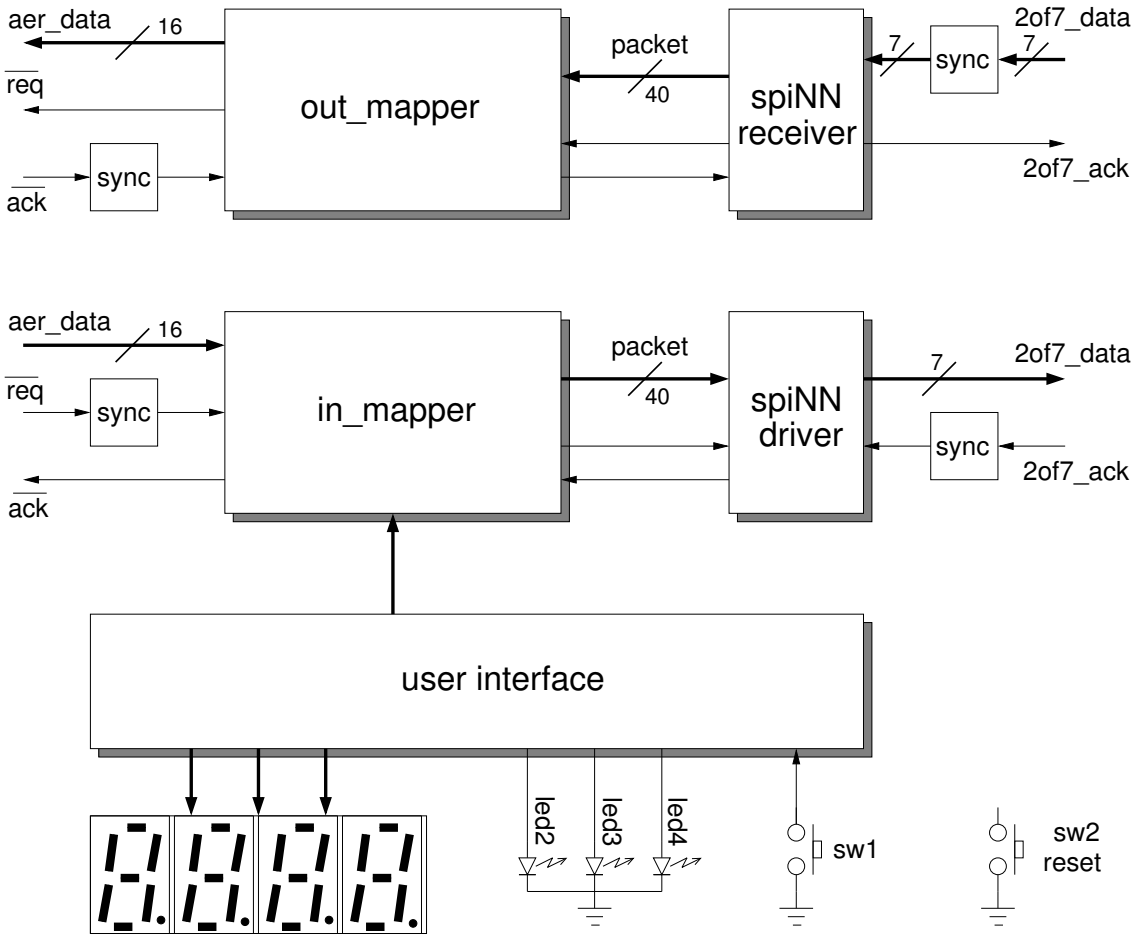

Figure 5: Interface Block Diagram.

tions of the buttons, leds and displays are shown in Table 1.

selection		virtual chip address
default	(x-chip = 2, y-chip = 0)	0x0200
optional	(x-chip = 254, y-chip = 254)	0xfefe

Table 2: Virtual chip address selection.

As mentioned above, the AER device is assigned a "virtual chip address" as part of the mapping process. The board is configured to provide a *default* address, shown in Table 2. This address works correctly with 4-node SpiNN-3 boards but not with 48-node SpiNN-4 ones. The user can select an *optional* address, also shown in Table 2, which is suitable for both types of board but requires the use of *proxy* populations in the neural network.

The different operating modes available are listed in Table 3. Each mode is a combination of a device, a resolution (if available) and a virtual address. There is also a *direct map* mode which passes the 16-bit AER event directly to the packet, without any mapping. The user can step through the different modes by repeatedly pushing the mode select button (*sw1*).

mode	display	description
def. retina 128x128	128	default address & retina with 128 x 128 pixels
def. retina 64x64	64	default address & retina sub-sampled to 64 x 64 pixels
def. retina 32x32	32	default address & retina sub-sampled to 32 x 32 pixels
def. retina 16x16	16	default address & retina sub-sampled to 16 x 16 pixels
def. cochlea	coch	default address & cochlea: 2 ears/64 channels/4 neurons
def. direct map	0000	default address & event passed through without mapping
alt. retina 128x128	.128	optional address & retina with 128 x 128 pixels
alt. retina 64x64	. 64	optional address & retina sub-sampled to 64 x 64 pixels
alt. retina 32x32	. 32	optional address & retina sub-sampled to 32 x 32 pixels
alt. retina 16x16	. 16	optional address & retina sub-sampled to 16 x 16 pixels
alt. cochlea	c.och	optional address & cochlea: 2 ears/64 channels/4 neurons
alt. direct map	0.000	optional address & event passed through without mapping

Table 3: Operating modes.

If at any point the SpiNNaker interface *deadlocks*, i.e., stops accepting packets, the board goes into *dump mode*. In this mode, indicated by *led4*, the board continues to accept events from the AER device but dumps them. The SpiNNaker system may need to be reset and re-booted.

Implementation Code

The interface was implemented using Verilog. The code is available for download from the SpiNNaker wiki.

References

[1] P. Häfliger, "CAVIAR Hardware Interface Standards", Version 2.0, Deliverable D WP7.1b, online: http://www2.imse-cnm.csic.es/caviar/download/ConsortiumStandards.pdf, 2003.

[2] S. Temple, "AppNote 7 - SpiNNaker Links", Version 1.0,
online: http://solem.cs.man.ac.uk/documentation/spinn-app-7.pdf, 2012.

[3] Enterpoint Ltd., "RaggedStone2 User Manual", Issue 1.02,
online: http://www.enterpoint.co.uk/raggedstone/RaggedStone_User_Manual_Issue_1_02.pdf,
2010.

Change log:

- *1.00 - 17apr13 - lap* - initial release - comments to *luis.plana@manchester.ac.uk*
- *1.10 - 03apr14 - lap* - bidirectional interface - comments to *luis.plana@manchester.ac.uk*