Test Board User Guide: complement to the AER-VGA Converter report

Florian Bochud and Yann Baud
Institut fuer Neuroinformatik
ETH Zurich, Switzerland
e-mail: fbochud@ee.ethz.ch, yabaud@ee.ethz.ch

July 5, 2004

Contents

1 Board Map

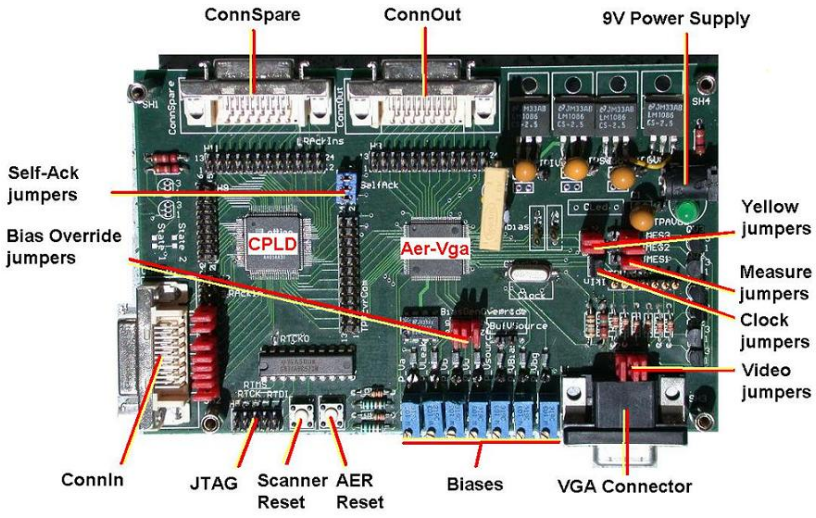

Figure 1: *Board view*

2 Quick Start

In order to use the board, follow these steps[1]:

1. Plug the 9V power supply
2. Connect the testboard to the monitor with a standard VGA cable.
3. Plug a neuromorphic chip to the input connector: use the *ConnIn* connector if using a simple burst-mode communication link or the *ConnSpare* connector if using a parallel word-serial communication link.
4. Connect the reset input of the CPLD (Header H9, pin 3) to a Gnd pin (H8, pin 1 for instance).
5. Set a jumper between pins 15 and 16 of header H9 (external delay).
6. Set the bias voltage on pin 2 of *buffVSource* header to approx. 2.1 Volts with a DC power supply.
7. Reset the scanner and AER receiver with the corresponding buttons.
8. Look at the nice picture!

[1]This supposes that the CPLD has been programmed, the jumpers are set correctly and the bias are set to plausible values (See next sections for detailed information)

3 Setup description

3.1 Connectors and Testpins

ConnIn and *ConnSpare* Connectors – from external chip to CPLD

Both connectors are used as input to the AER-VGA chip via the CPLD. In the current CPLD implementation, *ConnSpare* interfaces with the word-parallel link used at INI, while *ConnIn* interfaces with the simple burst-mode word-serial link from UPENN (see program description for further details). Table 1 shows the pin description with the corresponding CPLD pin and its bank number.

ConnOut Connector – from Aer-Vga to next board

This connector is used in multi-chip system for communication with the next cascaded chip. If not used, the cpld must provide a self-acknowledgement to get to the AER-VGA chip to work properly (See program description). Table 1 shows the pin assignment of the connector.

Testpins TPRcvrCom – from CPLD to Aer-Vga

This header allows the visualization of signals between the CPLD and the AER-VGA chip. The middle part of Table 2 describes the header's signals.

Testpins H8

This header drives the unused CPLD pins, which can then be used for testing or other functions. The corresponding CPLD pins are given in Table 2, left.

Testpins TPVideo

These test pins give the values of different video signals. The three output currents representing the colors can be measured on pin 1,2 and 3 when the measurement jumpers are set. The clock pin allows to check or feed in the clock of the AER-VGA chip scanner. Finally, the two right pins show the two synchronization signals of the video, HSYNC and VSYNC.

3.2 Jumpers setting

In Table 3 is a summary of the the different jumper setups. Default setups are written in bold. The following paragraphs explain more in details these setups.

Self-acknowledgement jumpers

Because the router expects an acknowledgement from the next chip in the multi-chip array, it needs to be self-acknowledged with the CPLD when used in single chip configuration. In this case, three jumpers must be set on the *SelfAck* header.

ConnIn			
H2		CPLD	
pin	signal	pin	bank
1	addr0	37	b0
2	addr1	36	b1
3	addr2	35	b2
4	addr3	34	b3
5	addr4	31	b4
6	addr5	30	b5
7	addr6	29	b6
8	addr7	287	b7
9	addr8	22	b8
10	ack	21	b9
11	ReqH	20	b10
12	ReqA	19	b11
13-24	Gnd	–	–

ConnSpare			
H11		CPLD	
pin	signal	pin	bank
1-12	Gnd	–	–
13	addrX0	87	d0
14	addrX1	86	d1
15	addrX2	85	d2
16	addrX3	84	d3
17	addrX4	81	d4
18	addrY0	80	d5
19	addrY1	79	d6
20	addrY2	78	d7
21	addrY3	72	d8
22	addrY4	71	d9
23	Req	70	d10
24	Ack	69	d11

ConnOut	
H8	
pin	signal
1-12	Gnd
13	addr0
14	addr1
15	addr2
16	addr3
17	addr4
18	addr5
19	addr6
20	addr7
21	addr8
22	ack
23	ReqH
24	ReqA

Table 1: ConnIn (H2), ConnSpare (H11) and ConnOut (H8), pin description

H9		CPLD	
pin	signal	pin	bank
1	testpin1	11	a15
2	testpin2	10	a14
3	testpin3	9	a13
4	testpin4	8	a12
5	testpin5	6	a11
6	testpin6	5	a10
7	testpin6	4	a9
8	testpin8	3	a8
9	testpin9	100	a7
10	testpin10	99	a6
11	testpin11	98	a5
12	testpin12	97	a4
13	testpin13	94	a3
14	testpin14	93	a2
15	testpin15	92	a1
16	testpin16	91	a0

TPRcvrCom		CPLD	
pin	signal	pin	bank
1-12	Gnd	–	–
13	addr0	41	c0
14	addr1	42	c1
15	addr2	43	c2
16	addr3	44	c3
17	addr4	47	c4
18	addr5	48	c5
19	addr6	49	c6
20	addr7	50	c7
21	addr8	53	c8
22	ack	54	c9
23	ReqH	55	c10
24	ReqA	56	c11

TPVideo	
pin	signal
1	Blue (mes)
2	Green (mes)
3	Red (mes)
4	Clock_out
5	Clock_in
6	HSync
7	VSync

Table 2: H8, TPRcvrCom and TPVideo Headers, pin description

Bias Jumpers	
ON	Manual Bias enabled
OFF	Bias generator enabled

Self-ack Jumpers	
ON	self-ack enabled
OFF	self-ack disabled

Clock Jumper	
ON	Board clock enabled
OFF	Board clock disabled

Video Jumpers	
ON	color signal to monitor
OFF	color signal disabled

Yellow Jumpers	
ON	Yellow enabled
OFF	Yellow disabled

Measure Jumpers	
Left	RGB to meas.testpins
Right	RGB to VGA monitor

Table 3: Jumper Settings, bold is default

Bias Jumpers

Set the jumpers *BiasGenOverride* to override any bias provided by the bias generator.

Clock Jumper

Set the jumper *TPClkIn* to use the clock circuit on the PCB. It is possible to generate a clock signal from a function generator; it suffices to leave the pins in *TPClkIn* open – i.e. the pin ClkOut is floating – and the clock signal is fed into ClkIn through its test pin on *TPVideo* header.

Video Jumpers

The three colors red, green and blue can be observed individually on the monitor and are assigned an enable jumper on header *J2*.

Measurements Jumpers

The PCB allows the measurerment of the RGB video currents before they get amplified in the source-follower circuit. This is done by setting the three jumpers JMESX to the left. The currents are converted into voltages with three resistors *RMX*. The voltages can be measured on test pins in header *TPVideo*. For normal operation of the video amplifier, keep the jumpers to the right

Yellow Jumpers

To enable yellow, the jumpers on header *J3*, associated with both its green and red components, must be set..

3.3 Bias settings

Signal	Value Min	Default	Max	Units
Va	0.5	1.5	1.6	[V]
VLeak	2.2	2.37	2.45	[V]
Vb	1.6	1.7	2.3	[V]
Vw	$Va + 0.35$	$Va + 0.39$	$Va + 0.45$	[V]
VSource	1.8	1.9	2.2	[V]
VBiasP	1.8	1.87	2	[V]
VBg	2.2	2.4	2.5	[V]

Table 4: *Typical bias voltages*

Signal	Active	Description
$\overline{Req}$	low	Request from neuro-chip
$\overline{Ack}$	low	Acknowledgement to neuro-chip
$AddrX$	high	X-Address from neuro-chip
$AddrY$	high	Y-Address from neuro-chip
$YReq$	high	Request Y to AER-VGA chip (*li* in program)
$\overline{XReq}$	low	Request X to AER-VGA chip (*lj* in program)
Ack	high-low	Ack from AER-VGA chip (*lo* in program)
$Addr$	high	X and Y address to AER-VGA chip
$s1$	high	internal signal for computation

Table 5: Signal description

4 Interfacing AER protocols with the CPLD

There are many different ways to use the Address-Event-Representation (AER) as communication protocol. The two most common are the word-parallel AER, where the X- and Y-addresses are communicated simultaneously in parallel and the burst-mode word-serial AER, where one Y-address is possibly followed on the same link by several X-addresses. The AER-VGA chip uses an extension of this last protocol for multi-chip networking, called burst-mode word-serial AER with chip address extension.

The testboard has a CPLD that allow the AER-VGA to be compatible with other version of AER communication, especially the word-parallel version used at INI and the simple burst-mode used at UPENN. To convert from word-parallel to multi-chip burst-mode, a first conversion to normal burst-mode is necessary. Figure 2 and 3 illustrates these two conversions. In Table 5 is a short description of the signals. Note that the signal *s1* is an internal signal specifying when the Ack signal gets low-active (i.e. right after the Y-address has been transmitted)

4.1 Program structure

The CPLD code is written in ABEL-HDL, a very simple language using exclusively *and* (&), *or*(#) and *inverse* (!) operations. The chipAddress.abl programm is divided in two parts: a declaration part, which allocate each pin of the CPLD and an equation part, which describe the combinatorial behavior of the CPLD. Equations again are divided in three blocks : the self-acknowledgement for the AER-VGA chip, the word-parallel to simple burst conversion and the simple burst to multi-chip burst conversion.

4.2 Equations, where do they come from?

These conversions are done in standard combinatorial logic. The arrows in Figures 2 and 3 shows the desired flow in the handshake and the causality of

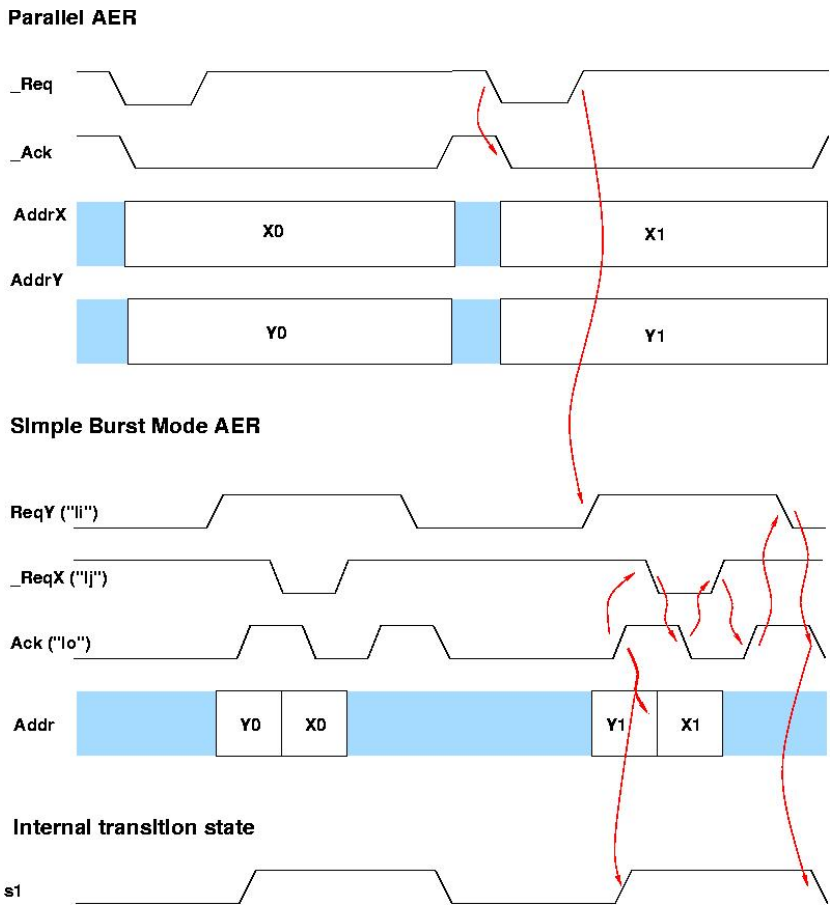

Figure 2: *Word-parallel to word serial conversion*

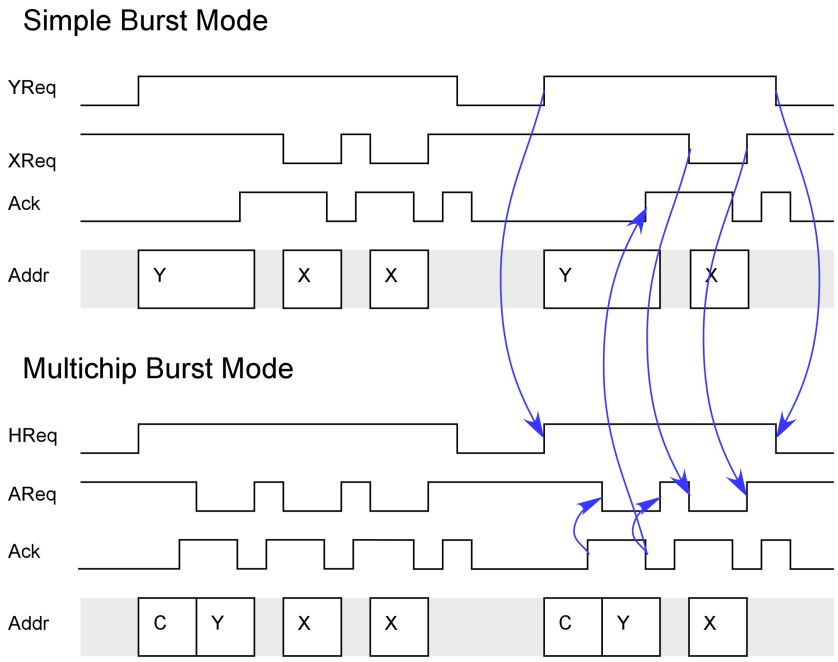

Figure 3: *Simple to multi-chip burst mode conversion*

Input				Output		
BMReqY	BMReqX	BMAck	s1	BMReqY	BMReqX	s1
1	1,dc	0	0	1	1	0
1	1,dc	1	0	1	0	1
1	0/1	0	1	1	1	1
1	1	1	1	0	1	1
1	0	1	1	1	0	1
0	1,dc	0	0/1	0	1	0
0	1,dc	1	1,dc	0	1	1

Table 6: Truth table for parallel word-serial to normal burst-mode
Suffix "BM" stands for "Burst-Mode"

each signal. The truth table in Table 6 describes the conversion from word-parallel to normal burst-mode. Simplifying this table for each output gives the equations in the program file.

For the normal burst-mode to multi-chip burst-mode conversion, please contact Paul Merolla (pmerolla@seas.upenn.edu) who is the author of the code. The conversion happens with a prelimary conversion to 4 wires self-time protocol (request y, request x, ack y, ack x), then by adding the chip address and finally with a reconverion to 3 wires protocol (request y, request x and ack).

4.3 How to program the CPLD?

This section describes the different steps to follow in order to compile and load an ABEL program onto the board CPLD.

1. Open ispLEVER Project Navigator[2] from the "Lattice semiconductor" directory and create a new project. Select "Schematic/ABEL" as Project type. A Project icon appears in the menu.

2. Configure the device (right mouse button, "Select new device"). Choose the corresponding CPLD device from the list (ispMACH 4000, LC4064B or LC4256B, 7.5 ns, TQFP 100, commercial).

3. Import or create a new description program (ABEL file) to this project (right mouse button, Import or New))

4. If necessary, use a text editor to edit the ABEL file. It should assign the necessary pins and describes the CPLD behavior with equations. Refer to ABEL reference guide or start from a previous program (for example: DiplomaThesis/PCB/CPLD/chipAddress.abl from the AER-VGA report).

[2]ispLEVER can be downloaded from www.latticesemi.com and a free 6 months licence is also available from the webpage

5. Compile your program: in the ispLEVER Project Navigator, select the project on the lhs and click on fit design in the list on the rhs. Create a jedec-file if necessary. This file will be loaded into the CPLD in the next steps.

6. Open ispVM System[3] from lattice semiconductor directory and add a new device (Edit/Add device). Set the properties of the device (ispMACH 4000, LC4064B or LC4256B, 7.5 ns, TQFP 100, load type: "erase, program, verify", jedec file from ispLEVER).

7. Don't forget to plug the JTag cable on the testboard... You can build the cable yourself simply by refering to the online documentation of ISP Download Cables from Lattice semiconductors (www.latticesemi.com). The input buffer are already available on the board, only the connections are required. Note that a download cable is available at the lab.

8. Press the *GO* button of the toolbar and wait that the file is loaded. Be careful! IspVM only loads a new 'jedec' file when the status is reset, i.e. when the status is equal to N/A. In order to do so, reload the "jedec"-file or reset with the right mouse button the load type to "erase, program, verify".

9. Have fun, the CPLD is loaded.

5 Miscellaneous

5.1 Known problems for the PCB board

This section lists the major problems we encountered with the PCB during the testing:

- The IC LM6262 we used for buffering V_{Source} is not working properly. This might because of the missing capacitor at its power supply pin.
- Bypass capacitors are missing at each power supply pin of the chip.
- The footprint used for the voltage regulators are wrong.
- Add labels to test pins

[3] ispVM can be downloaded from www.latticesemi.com for free