NeuroStack Configuration And API Library

Version 1.7
Revised 06/12/2017

Contents

1 Getting Started with NeuroStack

At factory default, the board is ready to use as a single board, powered by the same USB port as the one used for communication. For instruction regarding the board installation, please refer to the first chapter of the NeuroStack Hardware Manual.

1.1 Installation on PC running Windows

- Connect the NeuroStack to the USB connector of your PC.
- Go to Settings/Device Manager
- If the NeuroStack is properly recognized, it will appear as a USB device named “USB Serial Converter” with “FTDI” as the manufacturer.
- Otherwise it will appear as Other devices/USB<->Serial Converter.
    - Right click and select Properties, then select the Driver tab and click the Install Driver button.
    - If not found automatically, go visit http://www.ftdichip.com/Drivers/D2XX.htm to download the driver compatible with your hardware.
- Test the proper connection to the board and access the NeuroMem CM1K chip with the NeuroStack Console application described below

Troubleshooting:

- The NeuroStack uses an FTDI USB chip commonly used in many USB devices. In case of trouble detecting the board, try unplugging the other USB devices to detect conflicts.
- Verify the jumper and dip switch settings described below

1.2 NeuroStack Console Utility

The NeuroStack Console software is compatible with a single board or a stack of boards and allows you to verify the proper connection to the board and access the NeuroMem CM1K chip and modules programmed in the FPGA through a simple Register Transfer Level protocol. These modules and registers are described later in this manual. For detailed information about the NeuroStack Console interface, refer to the NeuroStack Console User’s Manual.

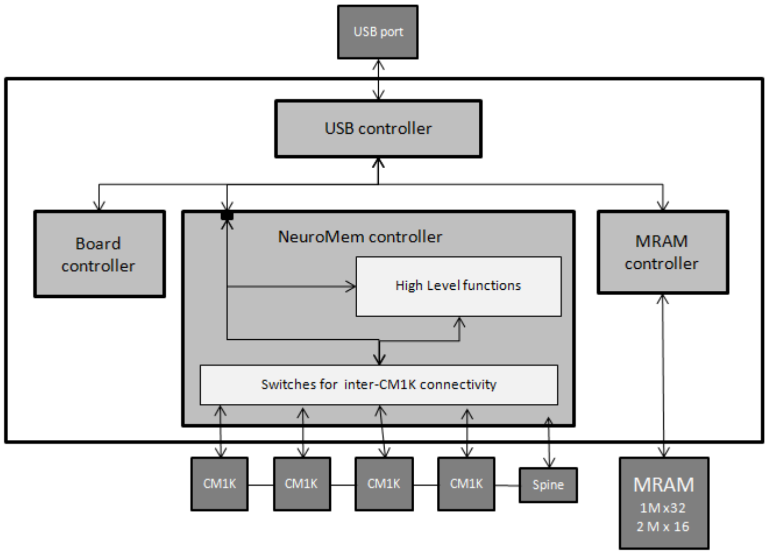
If you are using a stack of NeuroStack, the application will automatically size the neural network and let you verify that all boards are accounted for (4096 neurons per board). Note that this operation which runs at the launch of the program may takes a few seconds.

The modules and their registers are described in the following chapters. For example, reading the register 6 (MINIF) of module 1 (CM1K) must return the value 2 at factory settings.

1.3 Factory Default Configuration

At factory settings, the board is configure to be used as a single board, powered via its USB port and programmed with a simple protocol to read and write data to registers or addresses of modules instantiated in the FPGA. Refer to the NeuroStack_Hardware_Manual for details about the switches and jumpers configurations.

The only module instantiated at first is the controller of the chain of 4 CM1K chips residing on the board.



1.4 Address Mapping

The default modules instantiated in the board are accessible through the following 32-bit address map:

Address Range	**Module=** Address[30-24]	Functionality defined by registers = Address[23:8]
0x01000000 0x0100001F	NeuroMem 0x01 (d01)	Access to the registers of the CM1K chips whether limited to the chain of 4 chips on a single board or a chain of N*4 chips on a stack of N boards.
0x02000000 0x0200001F	Board settings 0x02 (d02)	Access to the settings of the board connected to the host including but not limited to a version number, a clock counter, etc.
0x03000000 0x03FFFFFF	MRAM 0x03 (d03)	Access to the bank of MRAM of the board connected to the host.
0x04000000 0x0400001F	Recognition Engine 0x04 (d04)	Access to the recognition engine programmed in the FPGA.

2 USB communication protocol

The USB protocol is based on a 10-bytes control command as follows:

• Byte0	Reserved
• Byte 1-2-3-4	R/W bit + 31-bit address
• Byte 5-6-7	24-bit Data length
• Byte 8-9	Minimum first 2 bytes of data
• Byte +	Remaining data (up to Data Length)

2.1 Write protocol

Reserved	Address[31:0]			Data length[23:0]	Data
0x00	Bit 31=1	Module[6:0]	Register[24:0]	Size of the input array **expressed in words**	Input array
1 byte	1 byte		3 bytes	3 bytes	Data length *2 bytes

Examples:

Write the 16-bit value 0x33AA to the MAXIF (register 7) of the CM1K (module 1)

0x00 81 00 00 07 00 00 01 33 AA

Write a vector of 4 consecutive byte values 9,8,7,6 to the COMP (register 1) of the CM1K (module 1)

0x00 81 00 00 01 00 00 02 09 08 07 06

2.2 Read protocol

Reserved	Address[31:0]			Data length[23:0]	Data
0x00	Bit 31=0	Module[6:0]	Register[24:0]	Size of the output array **expressed in words**	Output array
1 byte	1 byte		3 bytes	3 bytes	Data length *2 bytes

Examples:

Read the 16-bit value of the MINIF (register 6) of the CM1K (module 1)

0x00 01 00 00 06 00 00 01

Data is returned into 2 bytes or a word. Unless the MINIF register has been written, its default value is 2.

Read 8 consecutive byte values of the RESULTS (register 2) of the RECO_LOGIC (module 4)

0x00 04 00 00 02 00 00 04

Data is returned into 8 bytes.

2.3 Warning about USB

The NeuroStack USB interface is practical to connect to a host, but can be seriously detrimental to the overall speed performance of the board if the number of R/W transactions is not carefully optimized. A serious limitation of the USB standard is the minimum length of its 'Microframe' which is equal to 125us. For more information, refer to http://www.usbmadesimple.co.uk/ums_6.htm. Furthermore, some benchmarks have revealed that the FTDI USB chip can deliver 8000 Write per second, but only 1000 Read per second and this regardless of the size of the packet.

3 Module #1: NeuroMem controller

The NeuroMem controller is the module 0x01. It transmits and receives data to and from the chain of CM1K chips residing on the board and possibly extending to the CM1K chips of additional boards stacked on top of the master board.

3.1 The control Bus

The bus connecting all the CM1K chips in parallel is defined by the following lines:

DS	Data strobe line
RW_	Read/Write line (default is Read with RW_=1)
REG	5 bit register address
DATA	16-bit register data
RDY	Ready control line mixing the ready output signal of all the neurons in the chain and indicating that the neurons are all ready to execute a new command
ID_	Control line mixing the ready output signal of all the neurons in the chain and indicating that neurons have identified the last vector and that these neurons are all in agreement for its classification.
UNC_	Control line mixing the ready output signal of all the neurons in the chain and indicating that neurons have identified the last vector but that these neurons are in disagreement with its classification. This line is an in/out line because used as an input during the execution of certain Write register.

All the neurons of the CM1K chips sample a new command on the positive edge of the system clock and pull down their RDY line for the duration of its execution. Upon completion, the RDY line is pulled back up on the positive edge of the system clock.

A Write command (DS, RW_=0, REG, DATA) must be stable on the positive edge of the system clock and released before the next positive edge of the system clock.

A Read command (DS, RW_=1, REG) must be stable on the positive edge of the system clock and released before the next positive edge of the system clock. DATA is stable when the RDY control line is pulled high.

The DATA bus default status must be 0xFFFF to ensure that the pull-ups are not pulled down which would result in a heating of the CM1K chips.

For more information on the control lines, refer to the manual of the CM1K chip.

3.2 CM1K Registers

Access to the chain of CM1K is made through the module 1.

The following table describes the 15 registers controlling the entire behavior of the neurons. For a detailed description of the neuron's behavior and their interactions, please refer to the CM1K Hardware Manual and the NeuroMem Technology Reference Guide.

3.2.1 Neuron registers:

	Description	Addr 8-bit	Normal mode	SR mode	16-bit default
NSR	Network Status Register	0x0D	RW	W	0x0000
GCR	Global Control Register	0x0B	RW		0x0001
MINIF	Minimum Influence Field	0x06	RW	RW	0x0002
MAXIF	Maximum Influence Field	0x07	RW		0x4000
NCR	Neuron Context Register	0x00		RW	0x0001
COMP	Component	0x01	W	RW	0x0000
LCOMP	Last Component	0x02	W		0x0000
INDEXCOMP	Component index	0x03	W	W	0x0000
DIST	Distance register	0x03	R	R	0xFFFF
CAT	Category register	0x04	RW	RW	0xFFFF
AIF	Active Influence Field	0x05		RW	0x4000
NID	Neuron Identifier	0x0A	R	R	0x0000
POWERSAVE	PowerSave	0x0E	W		n/a
FORGET	Forget	0x0F	W		n/a
NCOUNT	Count of committed neurons	0x0F	R	R	0x0000
RESETCHAIN	Points to the first neuron	0x0C		W	n/a

3.2.2 Reco logic registers:

	Description	Addr 8-bit	Access	Data 16-bit Default
TOP	Left corner of the ROI in pixels	0x11	R/W	200
LEFT	Top corner of the ROI in pixels	0x12	R/W	120
WIDTH	Nominal width of the ROI in pixels	0x13	R/W	340
HEIGHT	Nominal height of the ROI in pixels	0x14	R/W	220
BWIDTH	Width of a primitive block in pixels	0x15	R/W	20
BHEIGHT	Height of a primitive block in pixels	0x16	R/W	20
ROIINIT	Reset the ROI to default	0x1F	W	0

4 Module #2: Hardware Settings

Access to the board settings is made through the module 2.

Register	Description	Addr 8-bit	Operation	Data 16-bit/ Default
HW_REV	Revision number of the board, including its USB protocol. Note that the recognition engine programmed under the Reco Logic module is stored in another register number 0 of the module 4.	0x01	R	0x0000
CLK_CNT	Clock counter Write : resets its value to 0 Read: number of clock cycles spent at execution of the USB commands since the last reset	0x02	RW	0x0000

Example:
USB_Write(1,7,400) //Write MAXIF
USB_Read(2,2)=19 clock cycles

USB_Read(1,6) //Read MINIF
USB_Read(2,2)= 20 clock cycles

5 Module #3: MRAM

Access to the MRAM is made through the module 3.

The memory can be read easily using Read and Write command with a datalength expressed in number of bytes you wish to retrieve starting from the position ADDR[23:0] expressed on word. The default ADDR is 0x000000.

Example:
USB_Write 0x03000010 → write data to the MRAM (module 4) starting at the ADDR=0x000010
USB_Read 0x03005000 → read data to the MRAM (module 4) starting at the ADDR=0x005000

6 Module #4: Recognition Logic

Access to the Recognition Logic is made through the module 4.

The advantage of using operations programmed in the Recognition Logic module #4 is that they integrate sequences of Read and Write RTL commands executed internally and not interrupted nor delayed by USB microframes.

Register	Description	Addr 8-bit	Operation	Data 16-bit/ Default
VERSION	Version number of the recognition logic module. 0=bare minimum, no module 4 1= vector recognition with automatic K readout Refer to next chapter on configuration files for information on additional Recognition Logic features or versions.	0x00	R	0x0001
RECO	Write: Recognize the input vector of length of up to 256 bytes. The responses of the top K firing neurons are accumulated in a Result buffer. Read: Read the Length of the Result buffer in words.	0x01	W R	0x0000

RESULTS	Read: output the Result buffer with the following format
	
	Format for a recognized vector:
	Up to **K** series of
	{
	- Distance value (16-bit value),
	-Category value (16-bit value)
	-Identifier value { 0x00, 24-bit value}
	}
	Ended with 0xFFFF, delimiter between consecutive RECO
	
	Format for a non recognized vector:
	Byte 1-2= 0xFFFF
	
	Write: Clear the Result buffer
		0x02	R
			
			W
				0x0000

K	Write the value K or maximum number of responses to read if applicable Read the value K	0x03	W R	0x0001

6.1 Examples

Clear the Result buffer and recognize a vector [01 02 03 04 05 06] in KNN mode with K=2

Commands	USB protocol
Write RESULT, 0 Write CM_NSR, 32 Write K, 2 Write RECO vector	0x01 0x84000002 0x000001 0x0000 0x01 0x8100000D 0x000001 0x0020 0x01 0x84000003 0x000001 0x0002 0x01 0x84000001 0x000003 0x01020 0x0304 0x0506

Read the Result buffer

Commands	USB protocol
Read RECO Size the output buffer to N Read RESULT buffer	0x01 0x04000001 Returns N= the size of Results or 2*4words +1words, or 9 0x01 0x04000003 N Returns the series Dist1 Cat1 NCR1 NID1 DIST2 CAT2 NCR2 NID2 0xFFFF

Recognize a vector [07 08 09 10] in RBF mode with K=5

Commands	USB protocol
Write CM_NSR, 0 Write K, 5 Write RECO vector	0x01 0x8100000D 0x000001 0x0000 0x01 0x84000003 0x000001 0x0005 0x01 0x84000001 0x000002 0x0708 0x0910

Read the Result buffer

Commands	USB protocol
Read RECO Size the output buffer to M Read RESULT buffer	0x01 0x04000001 Returns M 0x01 0x04000003 M Returns the first series Dist1 Cat1 NCR1 NID1 DIST2 CAT2 NCR2 NID2 0xFFFF Followed by the second series of 4 words if only 1 neuron fires Dist3 Cat3 NCR3 NID3 0xFFFF

The NeuroStack Diagnostic Utility supplied with the board features some tests using the Reco Logic more. For more information, refer to the manual of the NeuroStack Diagnostic Utility which describes a script on how to use the reco logic in RBF or KNN mode.

7 Speed Performances

The NeuroStack USB interface is practical to connect to a host, but can be seriously detrimental to the overall speed performance of the board if the number of R/W transactions is not carefully optimized. A serious limitation of the USB standard is the minimum length of its 'Microframe' which is equal to 125us. For more information, refer to http://www.usbmadesimple.co.uk/ums_6.htm.

Programming high-end functions inside the FPGA circumvents this limitations by reducing the number of I/Os between the FPGA and the host. The default firmware implements basic functions for vector learning and recognition but General Vision has a library of additional functions for text, signal and image analytics. Custom functions can be developed on demand and IP source code licensed. Please inquire at info@general-vision.com if you are interested.

This chapter reports the latency of common operations such as Learn_Vector and Recognize_Vector, along with the number of these operations executed per second.

These timings include the transfer of the vector packets between the host and the NeuroStack board via high-speed USB. Knowing that only 8 USB microframes or packet transfer can be processed per millisecond, the highest number of Recognize_Vector operations per second is already limited to 8000. This limitation comes from the USB standard which uses a 'Microframe' which is 125us long. For more information, refer to http://www.usbmadesimple.co.uk/ums_6.htm.

The advantage of using operations programmed in the Recognition Logic module #4 is that they integrate sequences of Read and Write RTL commands executed internally and not interrupted nor delayed by USB microframes.

Table #1: Timings using the NeuroMem Controller Module#1

Operations are programmed as a series of Read/Write instructions sent to the NeuroStack through its USB port. Depending on the content of the vectors to recognize and the results required by the application, the processing rate ranges from 125 to 500 vector recognition per second. Note that the limitation of the USB frame rate is such that the length of the vectors has no impact on the timings. The good thing is that the number of neurons committed on the NeuroStack or a stack of NeuroStack has no impact on the timings.

The following timings were executed on an Intel Core i3 CPU running at 2.5 Ghz.

Module1	components	ms	op/sec
Single W to CM1K		0.24	4167
Single R to CM1K		1.00	1000
Write neuron (WCOMP+WCAT in SR mode)	256	0.49	2034
Read neuron (RCOMP+RCAT in SR mode)	256	2.01	497
VectorLearn_RTL	256	0.73	1370
VectorReco_RTL KNN=1	256	2.15	465
VectorReco_RTL KNN=5	256	3.12	95

Table #2: Timings using Recognition Logic Module#4

Depending on the content of the vectors to recognize and the results required by the application (Distance, Category, Neuron Identifier), the processing rate ranges between 3800-4500 vectors per second including the data transfer to and from the host.

The following timings were executed on an Intel Core i3 CPU running at 2.5 Ghz.

Module4	**Components**	**ms**	**op/sec**
VectorReco_ FPGArev1 KNN=1	256	1.01	993*
VectorReco_ FPGArev1 KNN=5	256	1.02	977*

(*) can be improved by at least a factor 2 by accumulating results of multiple recognition and reading them in a single packet. The limitation is that the packet cannot exceed 16K bytes which comes down to a maximum of 1600 recognitions with K=1, or 888 recognitions with K=2, etc.

8 FPGA Configuration Files

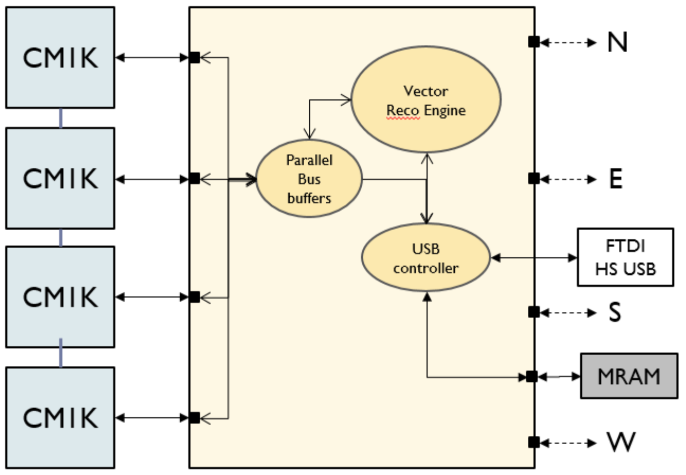
The FPGA of the NeuroStack can be programmed using Lattice's Diamond software which is available from the Lattice website for both Windows and Linux. Once downloaded and installed, it can be used with either a free license or a subscription license. This chapter gives examples of use models for the NeuroStack board.

8.1 Default configuration

FPGA programmers who wish to start an application using the solid common features of the General Vision firmware can order a license of our FPGA default source code and build on it. This is not mandatory but can save a lot of development time.

Description:

- Same file works for a master or slave board as long as the dip switch SW1 is setup properly
- The USB controller is enabled on a single board or on the master board of a stack.
- The four CM1K chips of a board are daisy-chained to form a chain of 4096 neurons which can itself be inserted at any position in a stack
- Optimized vector recognition based on the RBF or KNN classifier (version 1, default)

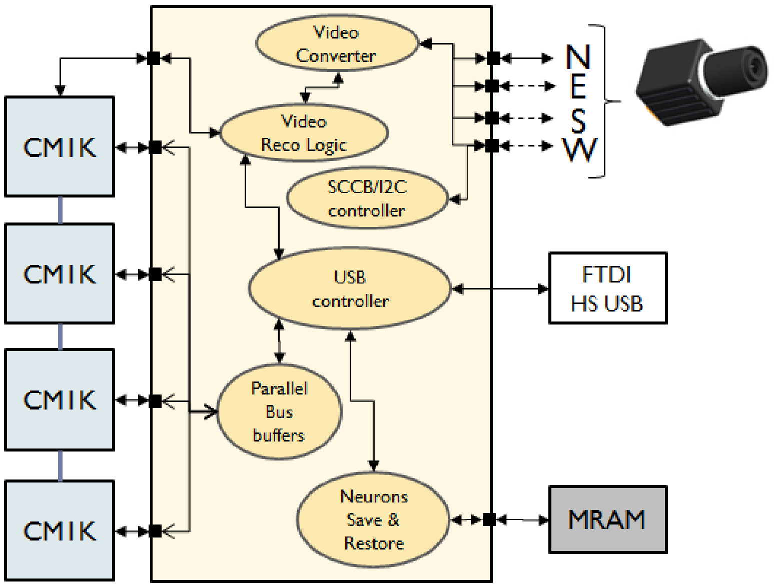


8.2 Examples of other configurations

The following paragraphs are just examples of applications which can be developed thanks to the novel architecture of the NeuroStack and its high scalability.

8.2.1 Real-time video monitoring

- The lines of the four cardinal connectors can be used as an interface to the video bus of camera (LVSD, Camera link, VGA, etc.)
- A video converter can sample and convert the signal in a format compatible with the digital input bus of the CM1K chip: V_CLK, L_Valid, F_Valid, V_Data. Refer to the CM1K manual for a description of the expected signals (chapter 4 and paragraph 6.4)
- Depending on the selected sensor, its settings such as Gain, Shutter, AGC, External Trigger, can be controlled through a serial protocol such as I2C, SPI, SCCB.
- The Video Reco Logic simply connects the output of the Video Converter to the digital input bus of the 1st CM1K chip of the chain. The RECO_EN and VI_EN lines must also be pulled up in order for the recognition stage to run and bit 0 of the RSR register must be set to 1. Beyond this setup, the controller simply samples the DATA bus of the CM1K upon the rise of its CAT_VALID pulse and formats or pushes it As IS to the USB controller for output.

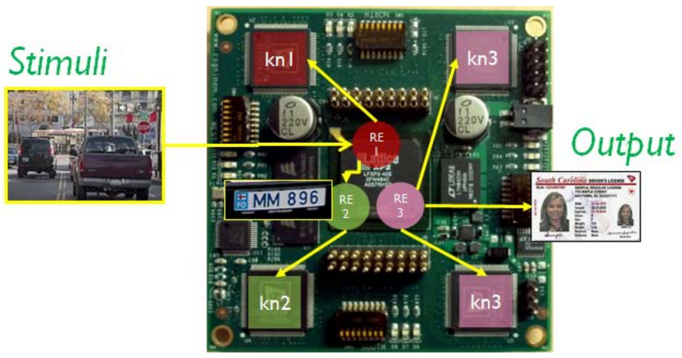


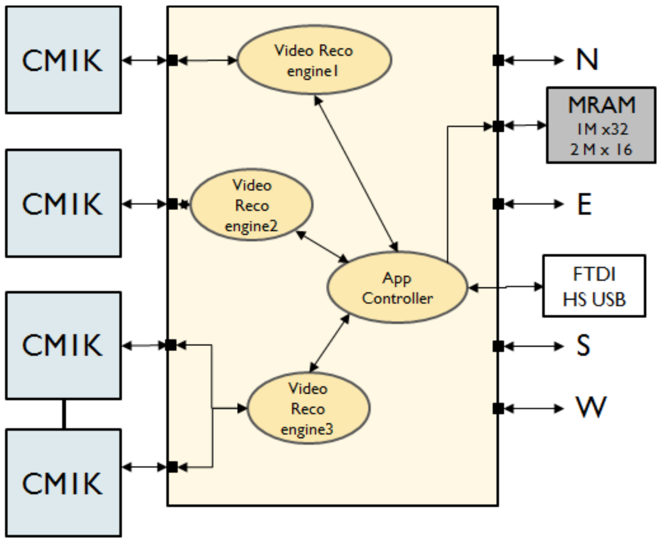
How to test that the video frame is properly captured?

- Option #1: Save the pixel values to MRAM and later retrieve the resulting byte array over USB for display as a 2D array with a size equal to frame width times frame height.
- Option #2: Set the Region Of Interest (ROI) of the CM1K to a 16x16 area. After each fall of the frame valid signal write the value k to the Category register, displace the ROI in a raster pattern with 16 pixels increment and increment k. When the ROI has reached the position of the last patch of 16x16 in the video frame, stop the continuous recognition by writing bit 0 of the RSR register to 0. From the PC host, execute the ReadNeurons function to retrieve the content of the neurons and re-assemble their models into an image.

8.2.2 Cascade image classifiers

- Live video is digitized and saved to memory
- A first recognition engine using 1 CM1K chip detects the location of license plates in the image
- A second recognition engine using 1 CM1K chip recognizes the characters inside the license plate frames
- A third recognition engine using 2 CM1K chips recognize the owner of the license plate. This last network is expanded by stacking NeuroStack boards and thus allowing to access a large database of license plate number.

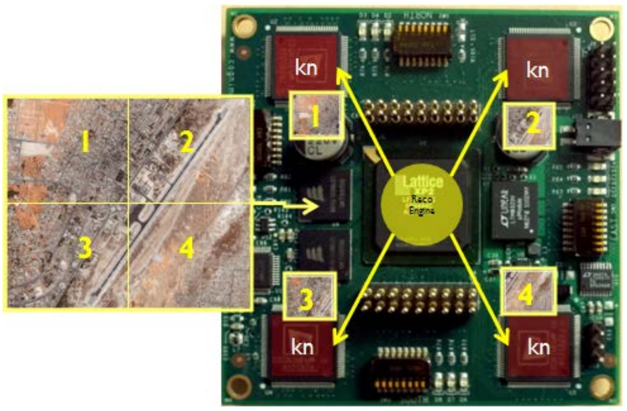




8.2.3 Distributed parallel image analytics

Process images N times faster by distributing the recognition to multiple CM1K chips.

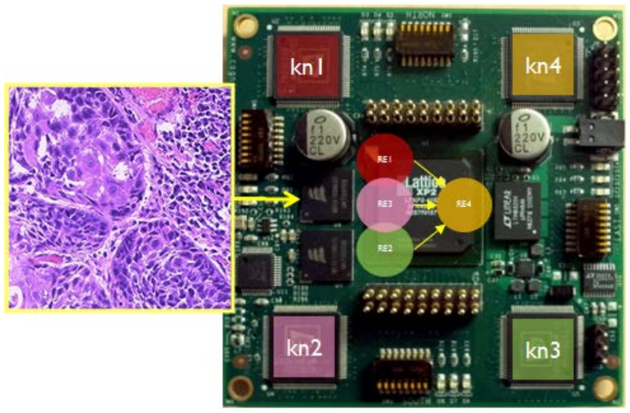
The four quadrants of a high resolution image are recognized in parallel by four neural networks (of 1024 neurons or more) loaded with the same knowledge. The latter can be simple and just intended to recognize edges or simple objects, or it can be more complex and composed of neurons assigned to different contexts to build a decision based on multiple features.



8.2.4 Parallel and sequential multiple-experts

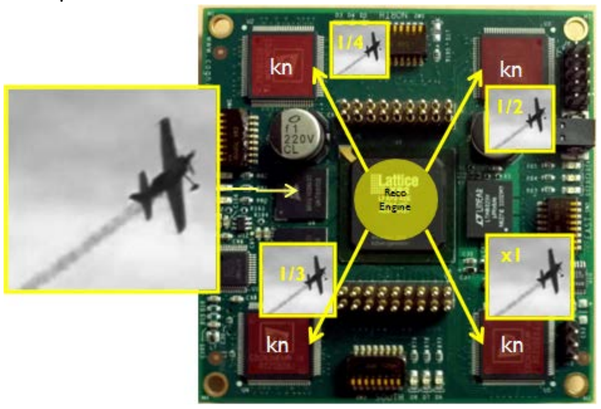
Example:

Expert in color (RE1+kn1)
Expert in texture (RE2+kn2)
Expert in shapes (RE3+kn3)
Expert in cell biology (RE4+kn4)



8.2.5 Distributed multi-scale image recognition

Example:

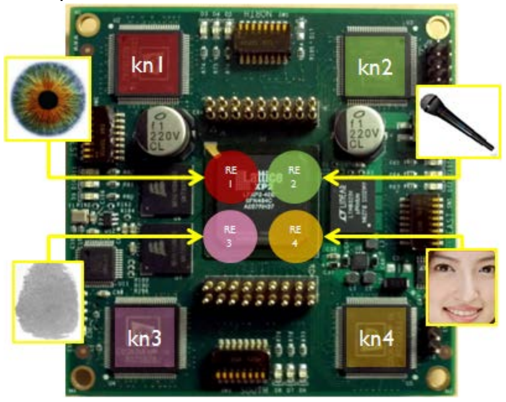


The image is recognized in parallel at four different scales:
Expert at Scale1 (RE1+kn)
Expert at Scale2 (RE2+kn)
Expert at Scale3 (RE3+kn)
Expert at Scale4 (RE4+kn)
The same knowledge is loaded in the four CM1Ks.

8.2.6 Sensor Fusion

Multiple sensor inputs (video, sound, accelerometer) for composite recognition.

Example:



Robust recognition of a person based on its iris and fingerprint and if the person has authorized access to the "appliance" detection of its emotions based on its facial expression and tone of voice.

Expert in voice (RE1+kn1)
Expert in face expression (RE2+kn2)
Expert in fingerprint (RE3+kn3)
Expert in iris(RE4+kn4)

8.2.7 Data Mining

Recognize and classify of vectors against large datasets or knowledge bases.

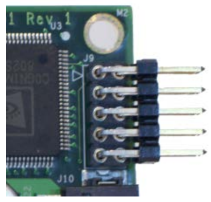
Text analytics
Search/Retrieval of biometric feature vectors (signature of fingerprint, iris, EEG, Eigen face vectors)
Object classification (SIFT, codebook, etc)

9 Programming the FPGA

The FPGA of the NeuroStack can be programmed using Lattice's Diamond software which is available from the Lattice website and a Lattice USB JTAG programmer.

9.1 JTAG connection

- Connect NeuroStack to its power supply, whether through a USB port or an external supply.
- Connect the output leads of your programming cable to the corresponding pins of the JTAG connector (J9).

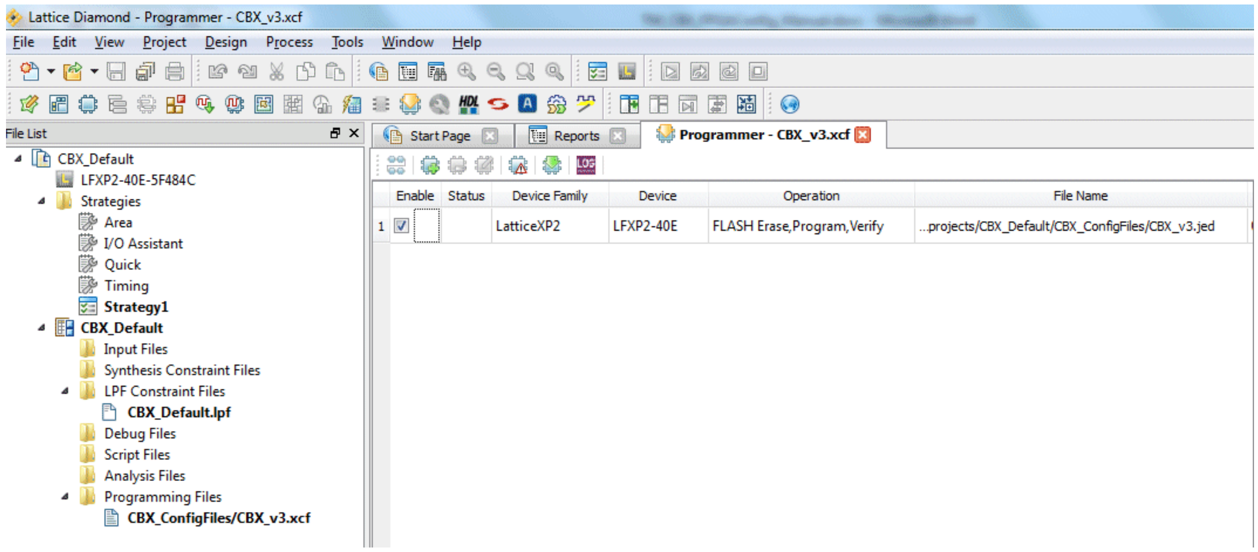


Pin #	Pin Name	Pin Name	Pin #
1	JTAG_TDI	Vcc	2
3	JTAG_TDO	n/a	4
5	JTAG_TCK	n/a	6
7	JTAG_ TMS	n/a	8
9	JTAG_TRST	Ground	10

9.2 Programming with the Diamond Programmer Tool

The files necessary to program a new configuration include a *.jed file and a *.xcf file. They are supplied in the folder FPGA_Firmware\CBX_Default.

Open the Diamond application and select the project saved in this folder and labeled CBX_Default.ldf.



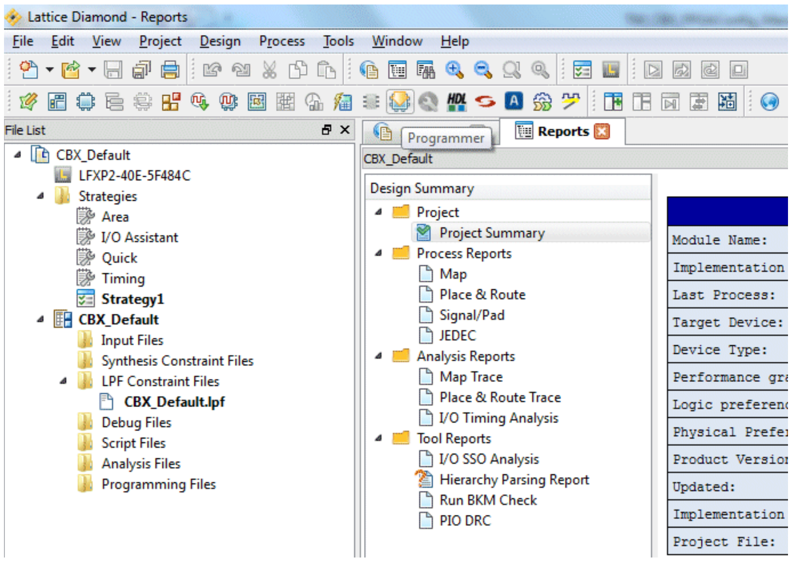
	Enable	Status	Device Family	Device	Operation	File Name
1	✓		LatticeXP2	LFXP2-40E	FLASH Erase,Program,Verify	...projects/CBX_Default/CBX_ConfigFiles/CBX_v3.jed

Click at the Program icon ( ) and wait until the operation is reported as successful.

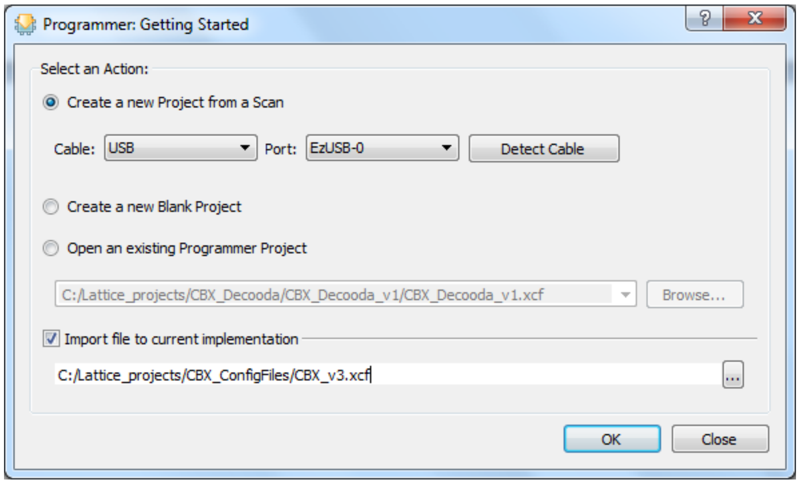
If the project does not load properly, the following definition steps might be necessary through a series of pop-up menus:

The model of FPGA on the NeuroStack board is an XP2 packaged in a BGA 484 balls.
The synthesis tool is SynplifyPro

Click the Programmer icon in the toolbar or select it through the Tools menu:



Fill the next panel by clicking the “Detect Cable” button and selecting the file CBX_vX.xcf in the folder CBX_ConfigFiles.


Programmer: Getting Started

Select an Action:

[✓ Create a new Project from a Scan]

Cable: USB   Port: EzUSB-0   Detect Cable

Create a new Blank Project

Open an existing Programmer Project

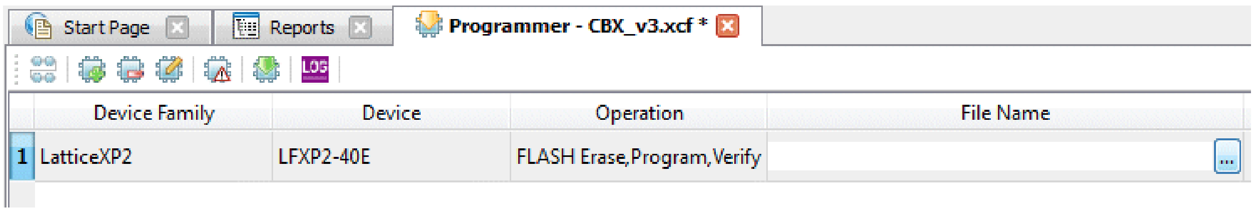
C:/Lattice_projects/CBX_Decooda/CBX_Decooda_v1/CBX_Decooda_v1.xcf   Browse...

[✓ Import file to current implementation]

C:/Lattice_projects/CBX_ConfigFiles/CBX_v3.xcf

OK   Close

The last step is the selection of the associated jed file which is also stored in the CBX_ConfigFiles folder.



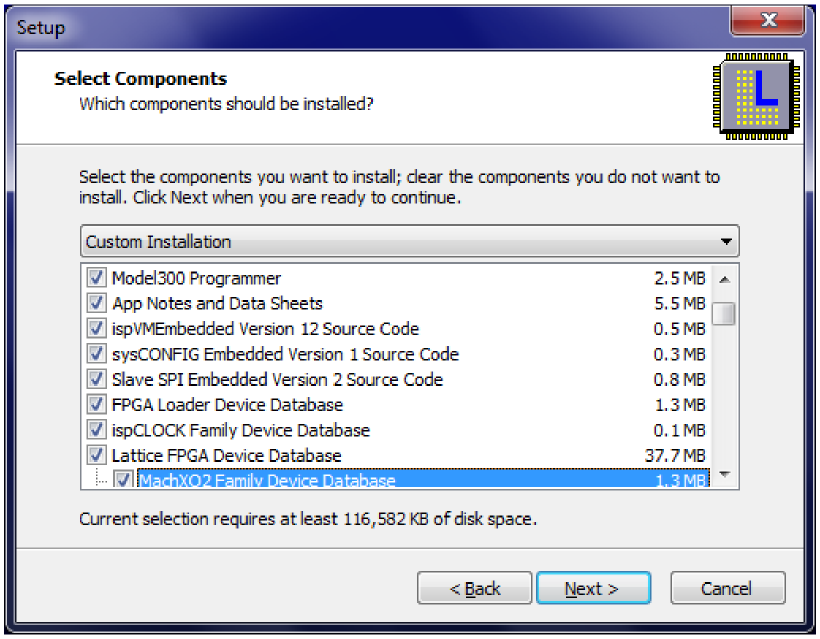
	Device Family	Device	Operation	File Name
1	LatticeXP2	LFXP2-40E	FLASH Erase,Program,Verify	

Finally click at the Program icon ( ) and wait until the operation is reported as successful.

9.3 Programming with ispVM System Software

The files necessary to program a new configuration include a *.jed file and a *.xcf file.

Install the Lattice ispVM software located on the CD in the folder BringUp_firmware/Ispvm_v18. For latest versions and more information you can refer to http://www.latticesemi.com/products/designsoftware/ispvmsystem/index.cfm



Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Custom Installation

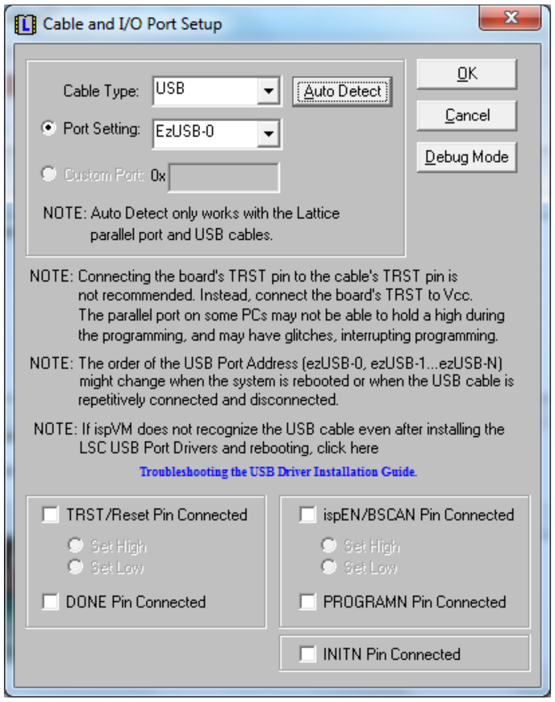
☑ Model300 Programmer	2.5 MB
☑ App Notes and Data Sheets	5.5 MB
☑ ispVMEmbedded Version 12 Source Code	0.5 MB
☑ sysCONFIG Embedded Version 1 Source Code	0.3 MB
☑ Slave SPI Embedded Version 2 Source Code	0.8 MB
☑ FPGA Loader Device Database	1.3 MB
☑ ispCLOCK Family Device Database	0.1 MB
☑ Lattice FPGA Device Database	37.7 MB
☑ MachXO2 Family Device Database	1.3 MB

Current selection requires at least 116,582 KB of disk space.

Remark: In the list above, you can exclude the installation of the components listed after the Lattice FPGA Devices such as CPLD, ORCA, ispPAC, Digital Interconnect and SPLD devices.

During the installation, you will be prompted to choose the driver to install. Choose depending on the model of your programming cable (USC PC if you are using the Lattice programming cable model HW-USBN-2A). For more information you can refer to http://www.latticesemi.com/products/developmenthardware/programmingcables.cfm?source=topnav

Launch the program ispVM System application. Under Options, select the Cable and I/O Port setup menu and click auto-detect. Click OK when the USB is found.


Cable and I/O Port Setup

Cable Type: USB

Auto Detect

(●) Port Setting: EzUSB-0

( ) Custom Port: 0x

NOTE: Auto Detect only works with the Lattice parallel port and USB cables.

NOTE: Connecting the board's TRST pin to the cable's TRST pin is not recommended. Instead, connect the board's TRST to Vcc. The parallel port on some PCs may not be able to hold a high during the programming, and may have glitches, interrupting programming.

NOTE: The order of the USB Port Address (ezUSB-0, ezUSB-1...ezUSB-N) might change when the system is rebooted or when the USB cable is repetitively connected and disconnected.

NOTE: If ispVM does not recognize the USB cable even after installing the LSC USB Port Drivers and rebooting, click here

Troubleshooting the USB Driver Installation Guide.

[ ] TRST/Reset Pin Connected
( ) Set High
( ) Set Low

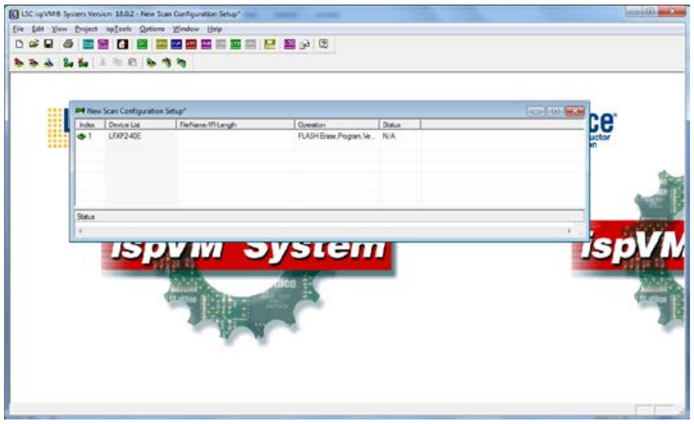
[ ] DONE Pin Connected

[ ] ispEN/BSCAN Pin Connected
( ) Set High
( ) Set Low

[ ] PROGRAMN Pin Connected

[ ] INITN Pin Connected

Under ispTools, select Scan chain. The Device List should report one LFXP2-40E device which is the model of FPGA on the NeuroStack board.

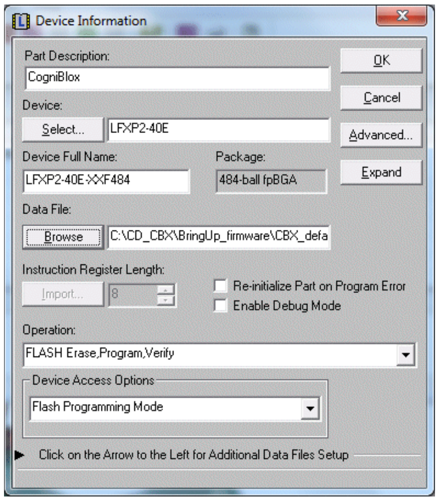

LSC ispVM® System Version 18.0.2 - New Scan Configuration Setup*

New Scan Configuration Setup*

Index	Device List	FileName/IR Length	Operation	Status
1	LFXP2-40E		FLASH Erase,Program,Ve...	N/A

Status

Double click on the line LFXP2-40E to display the Device Information.
Edit a Part Description.
Under the Data File button, select Browse and find the location of the file CBX_default.jed.


Device Information

Part Description: CogniBlox

Device: LFXP2-40E

Device Full Name: LFXP2-40E-XXF484

Package: 484-ball fpBGA

Data File: C:\CD_CBX\BringUp_firmware\CBX_defa

Instruction Register Length: 8

☐ Re-initialize Part on Program Error

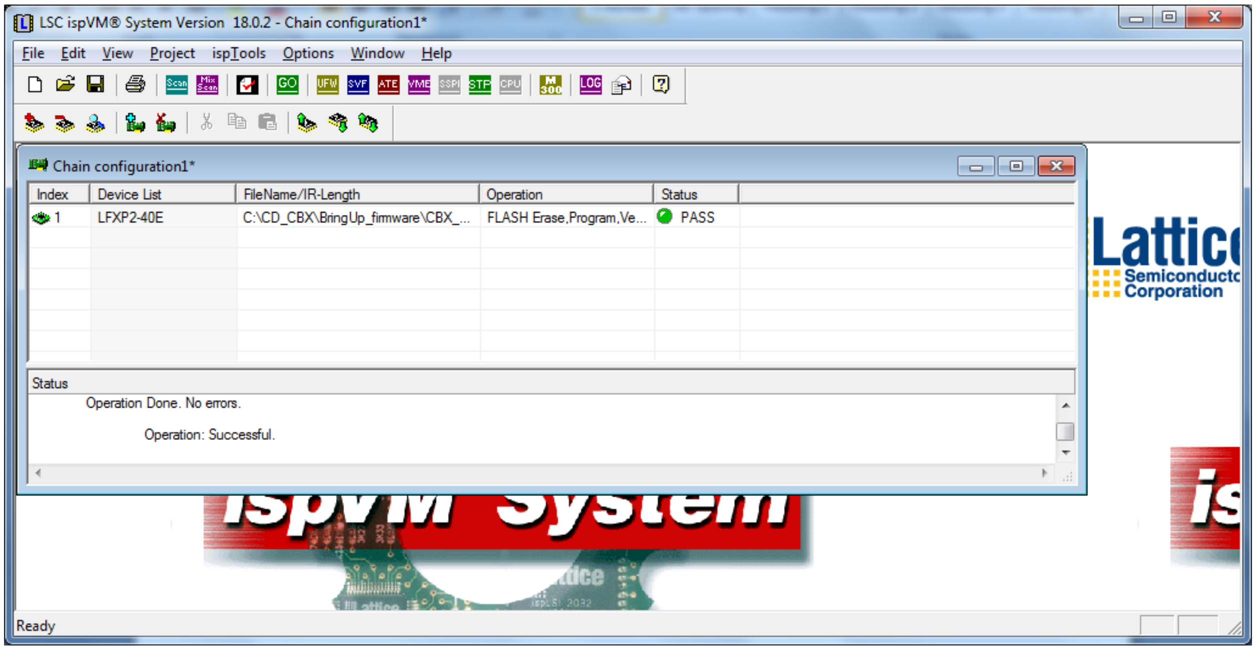
☐ Enable Debug Mode

Operation: FLASH Erase,Program,Verify

Device Access Options: Flash Programming Mode

Click on the Arrow to the Left for Additional Data Files Setup

Click OK and wait under the panel closes.
Click the Go icon to program the FPGA and wait until the process passes. This command can also be found under the Project menu. This takes approximately 30 seconds and the Status should report Pass.


LSC ispVM® System Version 18.0.2 - Chain configuration1*

Chain configuration1*

Index	Device List	FileName/IR-Length	Operation	Status
1	LFXP2-40E	C:\CD_CBX\BringUp_firmware\CBX_...	FLASH Erase,Program,Ve...	PASS

Status

```
        Operation Done. No errors.

                Operation: Successful.
```

Ready