

Set Up an Epiphany Cluster

This article is meant to guide the user in recreating an Epiphany level cluster using multiple Parallella boards. It covers the steps in setting up the environments, manufacturing the required PCB connectors, and running a basic parallel software on it.

Introduction

Epiphany has a 1GB memory space, and every epiphany holds 4MB of this space (32 KB of which are actually shared memory and addressable). Theoretically this leads to the conclusion that it might be possible to connect 256 Epiphany boards together (4096 cores). While this might be possible in the future, it is far from the case with 2015 Parallella boards. These boards only have E-link connectors for the north and south ports of the Epiphany, which narrows the possible cluster to one 'column' of Parallella boards (16 boards). This number is narrowed by half if taking into account that all lower half of the memory space is used by the Parallella operating system as the FPGA logic and SDRAM's address space. So for a simple application as is demonstrated here it is possible to use up to 8 Parallella boards in a column (128 cores).

This article does not discuss using SDRAM space for Parallella boards, even though it is definitely possible. The demonstration uses 2015.1 ESDK with 7020 ZYNQ and a corresponding FPGA build version. An attempt to apply it to other ESDK versions or builds might lead to driver mismatches, and would probably require some adjustments to be made.

Setting up the environment

- Download this disk image : <https://github.com/parallella/pubuntu/releases/tag/pubuntu-14.04-esdk.2015.1-20150130>
Make sure you choose the file that matches your board's ZYNQ version (7010 or 7020).
- Write the image into a micro-USB card. You may use this software to do this:
<https://sourceforge.net/projects/win32diskimage/>
- Connect to your Parallella using SSH. You may follow this Parallella tutorial (page 3):
http://www.adapteva.com/wp-content/uploads/Parallella-Quick-Start-Guide_rev3.pdf
I recommend using this software for your SSH connection:
<http://mobaxterm.mobatek.net/download-home-edition.html>

Connecting multiple Parallella Boards

You may connect two or more boards via their north and south e-link connectors in one of two ways, both described below. The main advantage of the Porcupine connection is how comfortable and configurable it is, while the main disadvantage is it using large, slow and unreliable jumper cables. The main advantage of using my design is how small and fast it is, while the main disadvantage is the necessity of independently manufactured PCB.



Author: Aviv Burshtein

Instructor: Oz Shmueli

Academic supervisor: Prof. Tsahi Birk

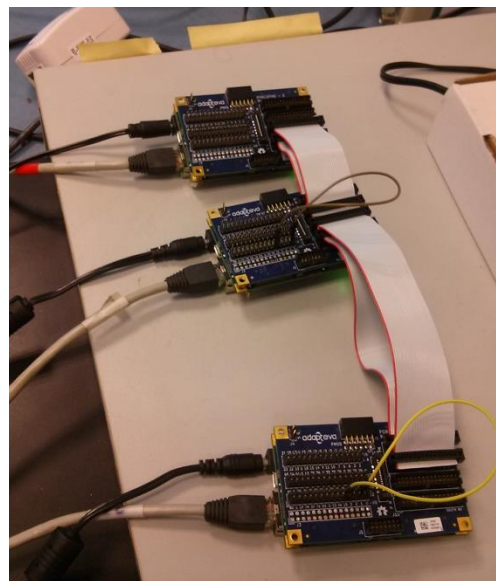
1. Use Porcupine boards, and make your own connectors:

- Order a porcupine board for each Parallella: <http://www.digikey.com/product-detail/en/ACC1600-01/1554-1003-ND/5048176>
- Order flat cables and connectors: <http://www.digikey.ca/product-search/en?keywords=MC30L-5-ND> and get some jumper cables from the nearest electronics shop.
- Order flat cable connectors, two for each board you wish to connect: <http://www.digikey.ca/product-search/en?keywords=WM14330-ND>
- Connect both ends of the cable to the connectors you purchased:

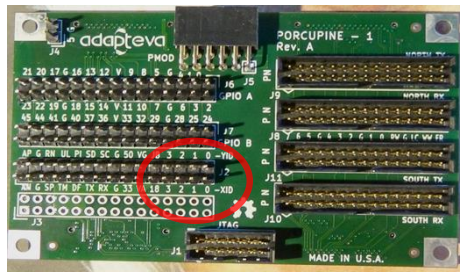


You must make two of these for every pair of Parallella boards.

- Connect Parallella boards as follows:



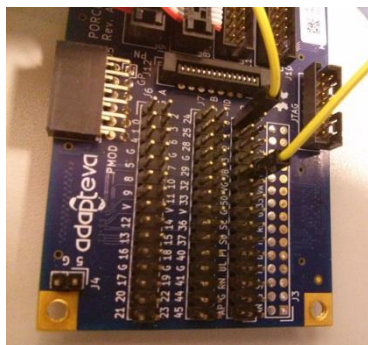
- Hardwire core-id:
The core-id can be reconfigured by hardwiring the PEC_POWER connector.
The pins representing the core-id are indicated in this image:



The core-id vector is 6 bits for Yid (columns) and 6 bits for Xid (rows). The last two bits of both are always 00 for the upper left core in each board (row and column are multiples of 4) and therefore 8 bits are enough to represent the address. The default core-id coordinates are (32,8), which translate to 808 in hexadecimal representation, and are represented as follows:

Yid 3	Yid 2	Yid 1	Yid 0	Xid 3	Xid 2	Xid 1	Xid 0
0	1	0	0	0	1	0	0

In order to set the core-id, jumpers must be used to connect GND or 1.8 Volts outputs to the correct coordinates. For example, coordinates (36,8) would translate to 908 in hex representation, and would be wired like this:



2. Produce PCB connectors and jumper sockets using my design:

- Download the connector design files from [here](#).
- Send to reproduction in a fab. I recommend: <https://www.itead.cc/>
- Order connectors: <http://www.toby.co.uk/content/catalogue/products.aspx?series=BTH-xxx-01-x-D-A-xx>
Make sure you order some spares for hardwiring core-ids.
- Solder the connectors to the PCB:

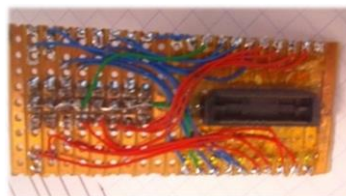


Academic supervisor: Prof. Tsahi Birk

-
- A photograph showing three Raspberry Pi 4 Model B boards connected in a chain. Each board is connected to the next via a USB-C to Ethernet adapter. The first board on the left has a black Ethernet cable plugged into its Ethernet port. The second board in the middle has a green Ethernet cable plugged into its Ethernet port. The third board on the right has a black Ethernet cable plugged into its Ethernet port. The boards are connected to a power source via USB-C cables. The background is a plain white surface.

-
- The diagram shows the PEC_POWER header and its connection to the jumper pin. The header is a 2x12 pin connector with the following pin numbers and labels:
- Top row: 312, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
 - Bottom row: 311, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0
- The connections are as follows:
- Pin 11 is connected to V100.
 - Pin 12 is connected to V101.
 - Pin 13 is connected to V102.
 - Pin 14 is connected to V103.
 - Pin 15 is connected to X100.
 - Pin 16 is connected to X101.
 - Pin 17 is connected to X102.
 - Pin 18 is connected to X103.
 - Pin 19 is connected to X104.
 - Pin 20 is connected to X105.
 - Pin 21 is connected to X106.
 - Pin 22 is connected to X107.
 - Pin 311 is connected to X108.
 - Pin 312 is connected to X109.
 - Pin 1 is connected to X110.
 - Pin 2 is connected to X111.
 - Pin 3 is connected to X112.
 - Pin 4 is connected to X113.
 - Pin 5 is connected to X114.
 - Pin 6 is connected to X115.
 - Pin 7 is connected to X116.
 - Pin 8 is connected to X117.
 - Pin 9 is connected to X118.
 - Pin 10 is connected to X119.
 - Pin 11 is connected to X120.
 - Pin 12 is connected to X121.
 - Pin 13 is connected to X122.
 - Pin 14 is connected to X123.
 - Pin 15 is connected to X124.
 - Pin 16 is connected to X125.
 - Pin 17 is connected to X126.
 - Pin 18 is connected to X127.
 - Pin 19 is connected to X128.
 - Pin 20 is connected to X129.
 - Pin 21 is connected to X130.
 - Pin 22 is connected to X131.
 - Pin 311 is connected to X132.
 - Pin 312 is connected to X133.
 - Pin 1 is connected to X134.
 - Pin 2 is connected to X135.
 - Pin 3 is connected to X136.
 - Pin 4 is connected to X137.
 - Pin 5 is connected to X138.
 - Pin 6 is connected to X139.
 - Pin 7 is connected to X140.
 - Pin 8 is connected to X141.
 - Pin 9 is connected to X142.
 - Pin 10 is connected to X143.
 - Pin 11 is connected to X144.
 - Pin 12 is connected to X145.
 - Pin 13 is connected to X146.
 - Pin 14 is connected to X147.
 - Pin 15 is connected to X148.
 - Pin 16 is connected to X149.
 - Pin 17 is connected to X150.
 - Pin 18 is connected to X151.
 - Pin 19 is connected to X152.
 - Pin 20 is connected to X153.
 - Pin 21 is connected to X154.
 - Pin 22 is connected to X155.
 - Pin 311 is connected to X156.
 - Pin 312 is connected to X157.
 - Pin 1 is connected to X158.
 - Pin 2 is connected to X159.
 - Pin 3 is connected to X160.
 - Pin 4 is connected to X161.
 - Pin 5 is connected to X162.
 - Pin 6 is connected to X163.
 - Pin 7 is connected to X164.
 - Pin 8 is connected to X165.
 - Pin 9 is connected to X166.
 - Pin 10 is connected to X167.
 - Pin 11 is connected to X168.
 - Pin 12 is connected to X169.
 - Pin 13 is connected to X170.
 - Pin 14 is connected to X171.
 - Pin 15 is connected to X172.
 - Pin 16 is connected to X173.
 - Pin 17 is connected to X174.
 - Pin 18 is connected to X175.
 - Pin 19 is connected to X176.
 - Pin 20 is connected to X177.
 - Pin 21 is connected to X178.
 - Pin 22 is connected to X179.
 - Pin 311 is connected to X180.
 - Pin 312 is connected to X181.
 - Pin 1 is connected to X182.
 - Pin 2 is connected to X183.
 - Pin 3 is connected to X184.
 - Pin 4 is connected to X185.
 - Pin 5 is connected to X186.
 - Pin 6 is connected to X187.
 - Pin 7 is connected to X188.
 - Pin 8 is connected to X189.
 - Pin 9 is connected to X190.
 - Pin 10 is connected to X191.
 - Pin 11 is connected to X192.
 - Pin 12 is connected to X193.
 - Pin 13 is connected to X194.
 - Pin 14 is connected to X195.
 - Pin 15 is connected to X196.
 - Pin 16 is connected to X197.
 - Pin 17 is connected to X198.
 - Pin 18 is connected to X199.
 - Pin 19 is connected to X200.
 - Pin 20 is connected to X201.
 - Pin 21 is connected to X202.
 - Pin 22 is connected to X203.
 - Pin 311 is connected to X204.
 - Pin 312 is connected to X205.
 - Pin 1 is connected to X206.
 - Pin 2 is connected to X207.
 - Pin 3 is connected to X208.
 - Pin 4 is connected to X209.
 - Pin 5 is connected to X210.
 - Pin 6 is connected to X211.
 - Pin 7 is connected to X212.
 - Pin 8 is connected to X213.
 - Pin 9 is connected to X214.
 - Pin 10 is connected to X215.
 - Pin 11 is connected to X216.
 - Pin 12 is connected to X217.
 - Pin 13 is connected to X218.
 - Pin 14 is connected to X219.
 - Pin 15 is connected to X220.
 - Pin 16 is connected to X221.
 - Pin 17 is connected to X222.
 - Pin 18 is connected to X223.
 - Pin 19 is connected to X224.
 - Pin 20 is connected to X225.
 - Pin 21 is connected to X226.
 - Pin 22 is connected to X227.
 - Pin 311 is connected to X228.
 - Pin 312 is connected to X229.
 - Pin 1 is connected to X230.
 - Pin 2 is connected to X231.
 - Pin 3 is connected to X232.
 - Pin 4 is connected to X233.
 - Pin 5 is connected to X234.
 - Pin 6 is connected to X235.
 - Pin 7 is connected to X236.
 - Pin 8 is connected to X237.
 - Pin 9 is connected to X238.
 - Pin 10 is connected to X239.
 - Pin 11 is connected to X240.
 - Pin 12 is connected to X241.
 - Pin 13 is connected to X242.
 - Pin 14 is connected to X243.
 - Pin 15 is connected to X244.
 - Pin 16 is connected to X245.
 - Pin 17 is connected to X246.
 - Pin 18 is connected to X247.
 - Pin 19 is connected to X248.
 - Pin 20 is connected to X249.
 - Pin 21 is connected to X250.
 - Pin 22 is connected to X251.
 - Pin 311 is connected to X252.
 - Pin 312 is connected to X253.
 - Pin 1 is connected to X254.
 - Pin 2 is connected to X255.
 - Pin 3 is connected to X256.
 - Pin 4 is connected to X257.
 - Pin 5 is connected to X258.
 - Pin 6 is connected to X259.
 - Pin 7 is connected to X260.
 - Pin 8 is connected to X261.
 - Pin 9 is connected to X262.
 - Pin 10 is connected to X263.
 - Pin 11 is connected to X264.
 - Pin 12 is connected to X265.
 - Pin 13 is connected to X266.
 - Pin 14 is connected to X267.
 - Pin 15 is connected to X268.
 - Pin 16 is connected to X269.
 - Pin 17 is connected to X270.
 - Pin 18 is connected to X271.
 - Pin 19 is connected to X272.
 - Pin 20 is connected to X273.
 - Pin 21 is connected to X274.
 - Pin 22 is connected to X275.
 - Pin 311 is connected to X276.
 - Pin 312 is connected to X277.
 - Pin 1 is connected to X278.
 - Pin 2 is connected to X279.
 - Pin 3 is connected to X280.
 - Pin 4 is connected to X281.
 - Pin 5 is connected to X282.
 - Pin 6 is connected to X283.
 - Pin 7 is connected to X284.
 - Pin 8 is connected to X285.
 - Pin 9 is connected to X286.
 - Pin 10 is connected to X287.
 - Pin 11 is connected to X288.
 - Pin 12 is connected to X289.
 - Pin 13 is connected to X290.
 - Pin 14 is connected to X291.
 - Pin 15 is connected to X292.
 - Pin 16 is connected to X293.
 - Pin 17 is connected to X294.
 - Pin 18 is connected to X295.
 - Pin 19 is connected to X296.
 - Pin 20 is connected to X297.
 - Pin 21 is connected to X298.
 - Pin 22 is connected to X299.
 - Pin 311 is connected to X300.
 - Pin 312 is connected to X301.
 - Pin 1 is connected to X302.
 - Pin 2 is connected to X303.
 - Pin 3 is connected to X304.
 - Pin 4 is connected to X305.
 - Pin 5 is connected to X306.
 - Pin 6 is connected to X307.
 - Pin 7 is connected to X308.
 - Pin 8 is connected to X309.
 - Pin 9 is connected to X310.
 - Pin 10 is connected to X311.
 - Pin 11 is connected to X312.
 - Pin 12 is connected to X313.
 - Pin 13 is connected to X314.
 - Pin 14 is connected to X315.
 - Pin 15 is connected to X316.
 - Pin 16 is connected to X317.
 - Pin 17 is connected to X318.
 - Pin 18 is connected to X319.
 - Pin 19 is connected to X320.
 - Pin 20 is connected to X321.
 - Pin 21 is connected to X322.
 - Pin 22 is connected to X323.
 - Pin 311 is connected to X324.
 - Pin 312 is connected to X325.
 - Pin 1 is connected to X326.
 - Pin 2 is connected to X327.
 - Pin 3 is connected to X328.
 - Pin 4 is connected to X329.

A photograph of a green printed circuit board (PCB) with a gold-plated edge connector and a blue component.



- If anyone could pick this up and add a PCB design for this I would gladly add it to the files here!
- When connecting the boards connect the correct pins for the core-id according to the explanation in the previous section.

Set up and run example code

The code uses the 2015 ESDK interrupts library to prove basic communication between the different Parallella boards. The communication is in the Epiphany level, meaning there is no read-write connection from a ZYNQ to a distant epiphany, and communication between OS and distant Epiphany boards is done via mailboxes in the local Epiphany board.

- Download the code folder from this [link](#) and unzip it.
- Copy the folder to your Parallella machines.
- Set up your HDF file using the edit_hdf script:

From the first board (with a single slave board):

```
Sudo ./edit_hdf.bash master 1
```

Here the second argument represents the number of slave boards in the system.

From a single slave board:

```
Sudo ./edit_hdf.bash slave 1
```

In this example the second argument represents the current slave board number.

Note that if you want to use more than two boards you must change the *maxcorenum* macro in the files *int-test.c* and *e-int-test.master.c* to match the amount of cores you need, and use edit_hdf accordingly.

For example, if you wish to use three boards in total (1 master and 2 slaves), you should set *maxcorenum* to 48, type "*Sudo ./edit_hdf.bash slave 1*" in the first slave Parallella, and type "*Sudo ./edit_hdf.bash slave 2*" in the second slave Parallella. In the master Parallella type "*Sudo ./edit_hdf.bash master 2*".

- Run *./build.sh* from the slave directory on all slave boards and run it from the master directory on the master board.
- Run *./run.sh* from the slave directory on all slave boards and then run it from the master directory on the master board.

The output should be the text in each board's space in the mailbox (within the first core's memory space).

Note that it might be wise to set up each core on its own at first and run "hello world" on it with the new core-id configuration.

Key Design Decisions and Barriers

Core-id

To my knowledge Adepteve plans to add core-id configuration support using FPGA register and next generation Epiphany Config-Registers in the future. As these are not yet available I decided to hardwire the core-id. A possible alternative would be to change FPGA design and set the core-id.

E-Link Enabling

The `reset_system()` function in 2015.1 ESDK locks the communication via E-Link connections. I therefore had to write a similar function with a slight difference- it doesn't lock the E-Link connection. I called it "reset_connected_systems" and I believe adding it to the next ESDK might be a good idea. This solution was made possible by the help of peter ([Parallella forum profile](#)).

User Interrupts

The 2015 ESDK added extraordinary support of user interrupt functions that allow synchronized communication between the cores. The interrupts based code is based on Yaniv Sapir's great interrupts code example. This is a good solution, but I believe it must be wrapped in more convenient user interface in order to scale.

Known Issues

- Loose connectors: The connected boards must be aligned well and not bend the connector in order for the system to work. This is difficult to maintain and uncomfortable. Possible solutions could be adding bolt holes to the design or using a mounting surface.

Next Steps

- Add a bolt to the connector design.
- Create a PCB design for the core-id setting connector.
- Add a library to support ZYNQ to distant communication via mailboxes, without dealing with space allocation and interrupts
- Use GPIO and FPGA routing configuration to close loop for read-write connection between from ZYNQ to remote Epiphanies.

Useful links

[Project Repository](#)

Relevant forum discussions:

[Epiphany Mesh1](#)

[Epiphany Mesh2](#)

Guides and Starting:

[Setup tutorial](#)



Author: Aviv Burshtein

Instructor: Oz Shmueli

Academic supervisor: Prof. Tsahi Birk

Code repository and design files:

[Our Repository](#)



Parallel Systems Lab
המעבדה למערכות מקביליות

