

Building a Super Computer

from open source
by

Peter Sanderson

2016

Proof Of Concept

Contents

- A proof of concept super computer was created in 2015 (Supercomputer.io)
- Used [Epiphany-III](#) 16-core processor (32 GFLOPS with 8Gbps interface)
 - note early Parallella tests did not use the 8Gbps [oh](#) i/f
- Can start small with little cost (<£100) and grow as required
- Not restricted to [MPMD](#) architecture

Changing Environment

Contents

- Existing technologies ([MPI](#), [OpenCL](#), [OpenMP](#)) are maturing
- The advocates for competing tools are coming together [OpenHPC](#)
- But the environment is changing. See for example
 - [Simon McIntosh-Smith's recent presentation](#)
 - [M. Mitchell Waldrop's recent essay The chips are down for Moore's law](#)
- New approaches are required

New Software Stack

New Software Stack

Contents

- The Epiphany chip has been proven with many of the available HPC tools:
 - [MPI / OpenCL - COPRTHR SDK](#)
 - [OpenMP - OMPi](#)
 - [OpenSHMEM - James Ross, David Richie](#)
 - Even [SLURM](#) for cluster management - see [forum post](#)
- The [Parallel Architectures Library](#) plan to rewrite maths, blas, dsp and fft libraries from the ground up
 - aim to be portable and run on multiple device types
 - what the [PAL](#) libraries lack in industry wide support they make up

for in boldness of purpose

Parallel Architectures Library

Building the System

Contents

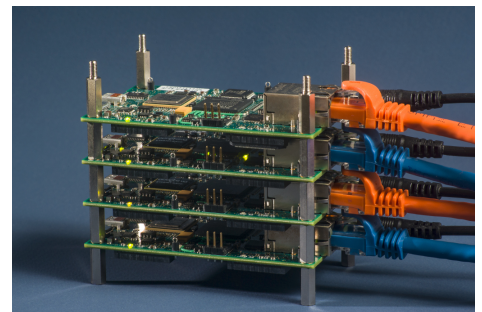
- [System Requirements](#) - for [Parallella](#) based system
- [Building Open Hardware fpga](#) - [8Gbps link](#) to [Epiphany-III](#)
- [Parallella Yocto Build](#) - taylor made distribution using [Yocto](#)
 - [meta-exotic](#) - generic build of non-native code
- [Testing the System](#) - rapid testing cycle: automation possible

System Requirements

System Requirements

Building the System

- Need to keep up to date with the latest
 - interrupt driven [oh](#) elink i/f
 - new update kernel drivers and latest eSDK or [PAL](#) libraries



- Easy method of distributing the software to the cluster
- Would like easy extension of fpga, kernel and software
- Would like to build Epiphany software on the build machine

Future Work

Contents

In no particular order:

- Built in support for PAL libraries
- Cluster management tool like SLURM
- Update of meta-exotic for gcc 5.x tools and adding gdb
 - also prove with another processor (RISK-V)
- North / south direct Epiphany connection
- Updating the various repositories that make up <https://github.com/peteasa/parallella/wiki> takes time

Contributors or sponsors for this work are always welcome!

@paracpg #parapg on Twitter

Peter on GitHub

The End

Additional Material

Contents

Architectures

Using Multiple Cores

SC.References

Online copy at <https://peteasa.github.io/parapg/parapg.html>

SC.References

Additional Material

Andreas Olofsson

Andreas Olofsson, Tomas Nordström, Zain Ul-Abdin "Kickstarting high-performance energy-efficient manycore architectures with Epiphany" 2014 48th Asilomar Conference on Signals, Systems and Computers

David A. Richie

David A. Richie and James A. Ross "[OpenCL](#) + [OpenSHMEM](#) Hybrid Programming Model for the Adapteva Epiphany Architecture", [OpenSHMEM](#) 2016, Third workshop on [OpenSHMEM](#).

Elias Kouskoumvekakis

["RISC-V port to Parallella"](#), Google Summer of Code 2016

M. Mitchell Waldrop

article: "The chips are down for Moore's law", Nature weekly journal of science, 2016

Michael J Flynn

"Some Computer Organizations and Their Effectiveness", IEEE Transactions on Computers. Vol. c-21, No.9, September 1972

Simon McIntosh-Smith

slides: "It's the end of the world as we know it ...", University of Bristol, HPC Research Group, 2015

wikipedia

various articles