

POWER Block Course

Assignment 3: AltiVec programming

(Submission deadline: 04.11.2016, 23:59 CET)

General Rules

The assignment solutions have to be submitted at:

<https://www.dcl.hpi.uni-potsdam.de/powersubmit/>

Our automated submission system is intended to give you feedback about the validity of your file upload. A submission is considered as accepted if the following rules are fulfilled:

- You did not miss the deadline.
- Your file upload can be decompressed with a zip / tar decompression tool.
- Your program runs without expecting any kind of keyboard input or GUI interaction.

If something is wrong with your submission, you will be informed via email (console output, error code). Re-uploads of corrected solutions are possible until the deadline.

All tasks must be submitted accordingly in order to pass the assignment.

Available Test Machines

BE	AIX 7.2	GCC 5.4.0	aix02.fsoc.hpi.uni-potsdam.de	192.168.42.56
LE	Ubuntu 16.04	XLC 13.1.1	ubp8le-01.fsoc.hpi.uni-potsdam.de	192.168.42.250

Assignment 3

The fourth assignment covers the vector processing units of the POWER8 processor. In the plenum we showed you the basics of the AltiVec extensions and its C interface. In this exercise you will practice their usage in your own applications on a more complex example.

Your final submission must compile and work properly on little-endian and big-endian systems, running Ubuntu or AIX.

Task 3.1: AltiVec Colorfilter

Revisit the bitmap reading program from Exercise 1.3. Implement the following three tasks using AltiVec instructions. The alpha channel must not be changed in any of them.

- a) create the negative of the input (flip black/white, blue/yellow etc)



b) create a grayscale image



c) make only those pixels grayscale for which red is not the dominant color



Submit a single tar.gz archive containing all source code files and your Makefile.

For simplicity you may assume that all input images have RGBA channels and a pixel count divisible by four.

Extra points

Make sure the pixel data is aligned at a 16 byte boundary. Use optimized load/store operations and measure the effects using `perf` on Linux.

Add your measurement results as well as 2-3 sentences of insight as a plain *.txt file to your submission archive.