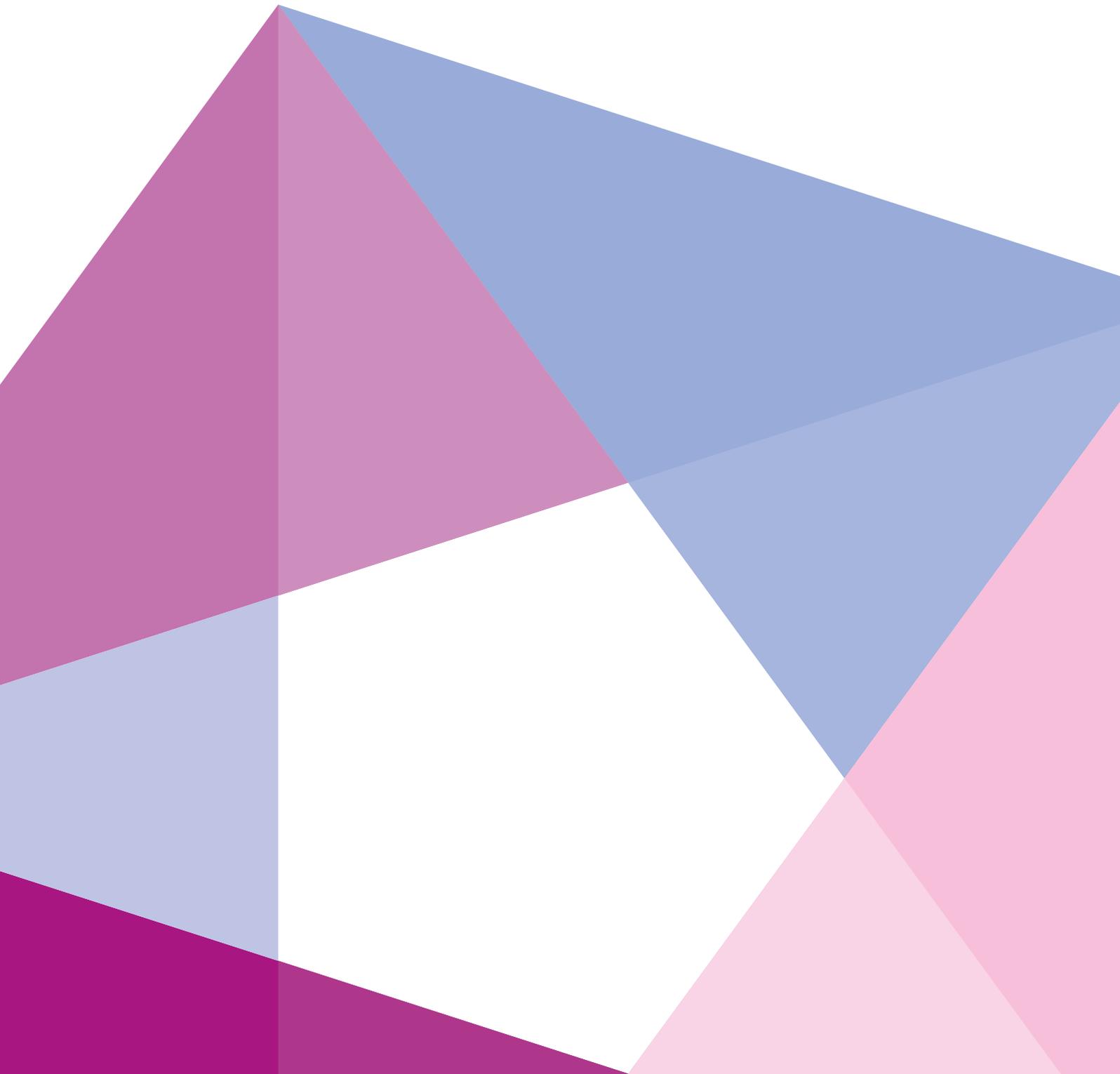




Using C++ on Cavium Octeon



Using C++ on Cavium Octeon

C++ programming greatly increases the structure and readability of your code. It supports object oriented programming and offers a lot of features not present in C. Cavium does not promote C++ programming because it comes at a price: performance. What is the use of designing a high speed packet processor when the programming on the processor is done with an inefficient language?

Ok, there is nothing faster than plain C, but if you use C++ wisely there will be almost no performance loss. Make sure that you know what you are doing and keep asking yourself: what is the impact of the C++ construction I want to use on the performance? Don't try to be too smart. If you keep this in mind, you will be ok. Having said this, let's move on to modifying the Octeon build environment.

Setup build environment

First we have to make sure that the source files are compiled with a C++-compiler instead of the C-compiler. For this we will make modifications to three files: *common.mk*, *application.mk* and *cvmx.mk*.

Open the *common.mk* file, which is in the *\$OCTEON_ROOT* directory, with your favorite editor, and search for the *CC* macro. This macro defines the compiler used when compiling C-files and is defined twice, one for simple executives and one for Linux builds. Replace the C-compiler with the C++-compiler by typing

```
ifeq (linux,$(findstring linux,$(OCTEON_TARGET)))
  CC = mips64-octeon-linux-gnu-g++
  AR = mips64-octeon-linux-gnu-ar
  LD = mips64-octeon-linux-gnu-ld
  STRIP = mips64-octeon-linux-gnu-strip
  OBJDUMP = mips64-octeon-linux-gnu-objdump
  NM = mips64-octeon-linux-gnu-nm
else
  CC = mipsisa64-octeon-elf-g++
  AR = mipsisa64-octeon-elf-ar
  LD = mipsisa64-octeon-elf-ld
  STRIP = mipsisa64-octeon-elf-strip
  OBJDUMP = mipsisa64-octeon-elf-objdump
  NM = mipsisa64-octeon-elf-nm
endif
```

Next, look at the beginning of the file where the *OCTEON_CFLAGS_GLOBAL_ADD* is added to *CFLAGS_GLOBAL*, and add the *-x c* option:

```
CFLAGS_GLOBAL += $(OCTEON_CFLAGS_GLOBAL_ADD) -x c
```

Adding "-x c" to *CFLAGS_GLOBAL* tells the c++-compiler to behave like a c-compiler.

Save *common.mk* and open *application.mk*. This one also resides in the *\$OCTEON_ROOT* directory. Add a compile rule for CPP-files just after the rule for the C-files

```
$(OBJ_DIR)/%.o: %.c
    $(COMPILE)

$(OBJ_DIR)/%.o: %.cpp
    $(subst -x c,, $(COMPILE))
```



When you are using SDK3.x instead of SDK2.x, one extra step is needed. Look for the rule where the linker is invoked and change it from:

```
$(CC) $(OBJS) $(LDFLAGS_PATH) $(LIBS_LIST) $(CFLAGS_GLOBAL) $(CFLAGS_LOCAL) $(LDFLAGS_GLOBAL) -o $@
```

to:

```
$(CC) $(OBJS) $(LDFLAGS_PATH) $(LIBS_LIST) $(subst -x c,,$(CFLAGS_GLOBAL)) $(CFLAGS_LOCAL) $(LDFLAGS_GLOBAL) -o $@
```

Save the *application.mk* file.

Next, modify the *cvmx-access-native.h* file located in the *OCTEON_ROOT/executive* directory.

Remove all instances of `__attribute__((__gnu_inline__))` eg.

```
static inline __attribute__((__gnu_inline__))
uint64_t __cvmx_prt_to_phys_internal__(void *ptr)
```

Modify it into

```
static inline
uint64_t __cvmx_prt_to_phys_internal__(void *ptr)
```

Important!

Be aware of the different linkage interface between C-files and C++-files. Calling C-functions from C++ will cause unresolved externals at linkage time because the bindings of the C-functions do not match the bindings of the C++-functions. To overcome this, declare the c-functions as *extern "C"*. Cavium mostly takes care of this in its header files because it has constructions at the start of its headers that are something like

```
#ifdef __cplusplus
extern "C" {
#endif
..header file contents..
#ifdef __cplusplus
}
#endif
```

However, not all Cavium packages are that 'sophisticated', so keep this in mind!

Example

We will use the passthrough example that comes with the SDK. First, using a terminal session or a file browser, make a copy of the passthrough example into your workspace. Use the following command

```
#cp -Rf $OCTEON_ROOT/examples/passthrough ~/workspace/passthrough
#cd ~/workspace/passthrough
```

Rename the *passthrough.c* file to *passthrough.cpp*, open the *passthrough.cpp* file for editing and add `extern "C"` around the include for the *cvmx-config-parse.h* header file

```
extern "C"
{
#include "cvmx-config-parse.h"
}
```



Next, add the `-Wno-write-strings` and `-fpermissive` parameters to the Makefile:

```
CFLAGS_LOCAL = -g -O2 -W -Wall -Wno-unused-parameter -Wno-write-strings -fpermissive
```

If you perform a clean build now, the passthrough example should compile and link without a problem. You are now ready to start programming with C++ and use those fancy classes of yours. Have fun!



Revision history

04-08-2015: Genesis

24-11-2015: Support for HFA in now included.



Powered by Phact

Sirius is a TCP/UDP/IP protocol stack and is Phact's solution of daily handling millions of TCP and SSL connections on OCTEON hardware. Sirius is available as source code or as a library build for your OCTEON platform. It runs as Simple Exec or on OCTEON-Linux.

Phact is a one stop shop for software and hardware solutions, support and consultancy and was founded in 2015 with 25 years of experience in the business.

How? Phact originated in VSN Systemen. VSN was founded in 1991. The last couple of years, the company grew and evolved tremendously. It was time to adjust the look and feel of the company to what it had become. That's why Phact was born.

We started developing telecom hardware and software for VoIP- and SS7-carriers in 1991. For them, our engineers developed stable and professional hardware and software solutions. Additionally, we provided our customers with support and management of their solutions.

Knowledge is a key part of our business. Therefore, we are engaged in close partnerships with universities and our professional suppliers.

Phact might still be very young, but we carry years of experience in software and hardware development, support, management and consulting.



Phact BV

Keizersveld 83, 5803AP, Venray, The Netherlands

T +31 (0) 478555000 F +31 (0) 478589563

Email: info@phact.nl

www.phact.nl

www.sirius-networks.com

