FILE(GLOB examples\_SRCS "\*.cpp")

set(EIGEN\_SYCL ON)

list(APPEND CMAKE\_EXE\_LINKER\_FLAGS -pthread)

if(EIGEN\_SYCL\_TRISYCL)

set(CMAKE\_CXX\_STANDARD 17)

else(EIGEN\_SYCL\_TRISYCL)

if(MSVC)

# Set the host and device compilers C++ standard to C++14. On Windows setting this to C++11

# can cause issues with the ComputeCpp device compiler parsing Visual Studio Headers.

set(CMAKE\_CXX\_STANDARD 14)

list(APPEND COMPUTECPP\_USER\_FLAGS -DWIN32)

else()

set(CMAKE\_CXX\_STANDARD 11)

list(APPEND COMPUTECPP\_USER\_FLAGS -Wall)

endif()

# The following flags are not supported by Clang and can cause warnings

# if used with -Werror so they are removed here.

if(COMPUTECPP\_USE\_COMPILER\_DRIVER)

set(CMAKE\_CXX\_COMPILER ${ComputeCpp\_DEVICE\_COMPILER\_EXECUTABLE})

string(REPLACE "-Wlogical-op" "" CMAKE\_CXX\_FLAGS ${CMAKE\_CXX\_FLAGS})

string(REPLACE "-Wno-psabi" "" CMAKE\_CXX\_FLAGS ${CMAKE\_CXX\_FLAGS})

string(REPLACE "-ansi" "" CMAKE\_CXX\_FLAGS ${CMAKE\_CXX\_FLAGS})

endif()

list(APPEND COMPUTECPP\_USER\_FLAGS

-DEIGEN\_NO\_ASSERTION\_CHECKING=1

-no-serial-memop

-Xclang

-cl-mad-enable)

endif(EIGEN\_SYCL\_TRISYCL)

FOREACH(example\_src ${examples\_SRCS})

GET\_FILENAME\_COMPONENT(example ${example\_src} NAME\_WE)

ei\_add\_test\_internal(${example} example\_${example})

ADD\_DEPENDENCIES(unsupported\_examples example\_${example})

ENDFOREACH(example\_src)

set(EIGEN\_SYCL OFF)