```
SHELL := /bin/bash

INCD = /home/young/MyWork/inc
LIBD = /home/young/MyWork/lib
EXED = /home/young/MyWork/exe


BurkDir = /home/young/MyWork/2.cordic_cpp/Burkadt
GHDLDir = /home/young/MyWork/7.cordic_accuracy/IF.GHDL


VPATH = Class.Core:Class.Figures:Class.GPData:Class.Angles:TestBench


.SUFFIXES : .o .cpp .c

.cpp.o :
	g++ -c -Wall -g -I${INCD} $<

.c.o :
	g++ -c -Wall -g -I${INCD} $<


#-----------------------------------------------------------------
# Classes into libraries
#-----------------------------------------------------------------
LIB_Core    = -lcordic-core   \
              -lcordic-burk   \
              -lcordic-ghdl   \

LIB_Angles  = -lcordic-angles \

LIB_Figures = -lcordic-figures \

LIB_GPData  = -lcordic-gpdata \


LIB_All     = ${LIB_Angles} ${LIB_Core} ${LIB_Figures} ${LIB_GPData}


#-----------------------------------------------------------------
Core :
	cd Class.Core; make -f Core.make all

Angles :
	cd Class.Angles; make -f Angles.make all

Figures :
	cd Class.Figures; make -f Figures.make all
```

```
GPData :
	cd Class.GPData; make -f GPData.make all


#-------------------------------------------------------------
# Testbenches
#-------------------------------------------------------------
Core_tb.o : Core_tb.cpp Core_tb.hpp
	g++ -c -Wall -g -I${INCD} Class.Core/Core_tb.cpp

Core_tb : Core Core_tb.o
	 g++ -o $@ Core_tb.o -L${LIBD} ${LIB_Core} -lm

run_core : Core_tb
	cp ${EXED}/cordic_vtb .
	./Core_tb
	rm ./cordic_vtb


#-------------------------------------------------------------
Angles_tb.o : Angles_tb.cpp Angles_tb.hpp
	g++ -c -Wall -g -I${INCD} Class.Angles/Angles_tb.cpp

Angles_tb : Core Angles Figures GPData Angles_tb.o
	 g++ -o $@ Angles_tb.o -L${LIBD} ${LIB_All}  -lm

run_angles : Angles_tb
	./Angles_tb


#-------------------------------------------------------------
cordic_tb01.o : cordic_tb01.cpp cordic_tb01.hpp
	g++ -c -Wall -g -I${INCD} TestBench/cordic_tb01.cpp

cordic_tb01 : Core Figures GPData Angles cordic_tb01.o
	 g++ -o $@ cordic_tb01.o -L${LIBD} ${LIB_All}  -lm

run_tb01 : cordic_tb01
	./cordic_tb01


#-------------------------------------------------------------
cordic_tb02.o : cordic_tb02.cpp cordic_tb02.hpp
	g++ -c -Wall -g -I${INCD} TestBench/cordic_tb02.cpp

cordic_tb02 : Core Figures GPData Angles cordic_tb02.o
	 g++ -o $@ cordic_tb02.o -L${LIBD} ${LIB_All} -lm

run_tb02 : cordic_tb02
	./cordic_tb02
```

```
#-------------------------------------------------------------
cordic_tb03.o : cordic_tb03.cpp cordic_tb03.hpp
        g++ -c -Wall -g -I${INCD} TestBench/cordic_tb03.cpp

cordic_tb03 : Core Figures GPData Angles cordic_tb03.o
         g++ -o $@ cordic_tb03.o -L${LIBD} ${LIB_All} -lm

run_tb03 : cordic_tb03
        ./cordic_tb03



# for f in fig_*; do if [ -f $$f ]; then mv fig_*.* DIR.FIG; fi done
# for f in *.o; do if [ -f $$f ]; then rm *.o; fi done
#-------------------------------------------------------------
QuadTree.o : QuadTree.cpp
         g++ -c -Wall -g -I${INCD} QuadTree.cpp

QuadTree_tb.o : QuadTree_tb.cpp
         g++ -c -Wall -g -I${INCD} QuadTree_tb.cpp

QuadTree_tb : QuadTree.o QuadTree_tb.o
         g++ QuadTree_tb.o QuadTree.o -o QuadTree_tb -lm



#-------------------------------------------------------------
# Move Figures and Latex Files to DIR.FIG
#-------------------------------------------------------------
move_fig :
        for f in fig_*; do if [ -f $$f ]; then mv fig_*.* DIR.FIG; fi done
        for f in *.eps; do if [ -f $$f ]; then mv *.eps   DIR.FIG; fi done
        for f in *.o; do if [ -f $$f ]; then rm *.o; fi done


EXES = Angles_tb cordic_tb01 cordic_tb02

#-------------------------------------------------------------
#print_all : print_angles print_tb01 print_tb02 print_quad
#       more makefile > print.Makefile
#       more ${SRCCo}  > print.Core


#-------------------------------------------------------------
# Tar
#-------------------------------------------------------------
tar_Angles :
        cd Class.Angles;   make -f Angles.make   tar
```

```
tar_Core :
        cd Class.Core;     make -f Core.make     tar

tar_Figures :
        cd Class.Figures;  make -f Figures.make  tar

tar_GPData :
        cd Class.GPData;   make -f GPData.make   tar

TARS =  Class.Angles/Angles.tar \
        Class.Core/Core.tar       \
        Class.Figures/Figures.tar \
        Class.GPData/GPData.tar

tar_all : tar_Angles tar_Core tar_Figures tar_GPData
        tar cvf cordic.tar ${TARS}


#-------------------------------------------------------------
# Print
#-------------------------------------------------------------
print_make :
        /bin/more makefile > makefile.print
        /bin/more Class.Angles/Angles.make   >> makefile.print
        /bin/more Class.Core/Core.make       >> makefile.print
        /bin/more Class.Figures/Figures.make >> makefile.print
        /bin/more Class.GPData/GPData.make   >> makefile.print

print_Angles :
        cd Class.Angles;   make -f Angles.make   print

print_Core :
        cd Class.Core;     make -f Core.make     print

print_Figures :
        cd Class.Figures;  make -f Figures.make  print

print_GPData :
        cd Class.GPData;   make -f GPData.make   print


print_all : print_Angles print_Core print_Figures print_GPData
        /bin/more ./Class.Angles/Angles.print   >  cordic.print
        /bin/more ./Class.Core/Core.print       >> cordic.print
        /bin/more ./Class.Figures/Figures.print >> cordic.print
        /bin/more ./Class.GPData/GPData.print   >> cordic.print


print_tb01 :
```

```
	/bin/more makefile ${SRC01} > print.tb01
print_tb02 :
	/bin/more makefile cordic_tb02.cpp Core.cpp Core.hpp > print.tb02
print_quad :
	/bin/more QuadTree.hpp QuadTree.cpp QuadTree_tb.cpp > print.quadtree


#-------------------------------------------------------------
# Clean
#-------------------------------------------------------------
clean_obj:
	\rm -f *.o

clean_fig:
	\rm -f fig_*.*

clean : clean_obj clean_fig
	\rm -f *.o *~ *# *.UA0
	\rm -f ${EXES}
	\rm -f *.emf *.eps eg*.pdf eg*.tex *.dat *.gp *.tmp
	\rm -f angle*.dat
	\rm -f cordic_tb*
	\rm -f cordic_vtb ip_fifo op_fifo

clean_all: clean
	cd Class.Angles;   make -f Angles.make   clean
	cd Class.Core;     make -f Core.make     clean
	cd Class.Figures;  make -f Figures.make  clean
	cd Class.GPData;   make -f GPData.make   clean
	cd ${BurkDir};     make clean
	cd ${GHDLDir};     make clean

	# cd Class.QuadTree; make -f QuadTree.make clean



#-------------------------------------------------------------
# copy include files    ${INC} into the directory ${INCD}
# copy library files    ${LIB} into the directory ${LIBD}
# copy executable files ${EXE} into the directory ${EXED}
# include files in ${INCS} directories to compile this module
#-------------------------------------------------------------
INCD = /home/young/MyWork/inc
LIBD = /home/young/MyWork/lib
EXED = /home/young/MyWork/exe

VPATH = ../Class.Core:../Class.Figures:../Class.GPData

INCS = -I../Class.Core    \
       -I../Class.Figures \
```

```
       -I../Class.GPData  \


.SUFFIXES : .o .cpp .c

.cpp.o :
        g++ -c -Wall -g ${INCS} $<

.c.o :
        g++ -c -Wall g ${INCS} $<


#-------------------------------------------------------------
# Classes
#-------------------------------------------------------------
SRC  = Angles.cpp   Angles.hpp                    \
       Angles.1.b1.plot_angle_tree.cpp            \
       Angles.1.b2.plot_circle_angle.cpp          \
       Angles.1.b3.plot_line_angle.cpp            \
       Angles.1.b4.plot_quantization.cpp          \
       Angles.2.t1.calc_tscale_statistics.cpp     \
       Angles.2.t2.plot_tscale_statistics.cpp     \
       Angles.2.t3.plot_tscale_residual_angles.cpp \
       Angles.3.u1.calc_uscale_statistics.cpp     \
       Angles.3.u2.plot_uscale_statistics.cpp     \
       Angles.3.u3.plot_uscale_residual_angles.cpp \
       Angles.3.u4.plot_uscale_histogram.cpp      \
       Angles.a.compute_angle_arrays.cpp          \

OBJ  = Angles.o                                   \
       Angles.1.b1.plot_angle_tree.o             \
       Angles.1.b2.plot_circle_angle.o           \
       Angles.1.b3.plot_line_angle.o             \
       Angles.1.b4.plot_quantization.o           \
       Angles.2.t1.calc_tscale_statistics.o      \
       Angles.2.t2.plot_tscale_statistics.o      \
       Angles.2.t3.plot_tscale_residual_angles.o  \
       Angles.3.u1.calc_uscale_statistics.o      \
       Angles.3.u2.plot_uscale_statistics.o      \
       Angles.3.u3.plot_uscale_residual_angles.o  \
       Angles.3.u4.plot_uscale_histogram.o       \
       Angles.a.compute_angle_arrays.o           \


INC = Angles.hpp                        \

LIB = libcordic-angles.a                \

EXE = Angles_tb                         \
```

```
#-------------------------------------------------------------
Angles.o : ${SRC}
        g++ -c -Wall -g ${INCS} Angles.cpp

#-------------------------------------------------------------
all : ${OBJ}
#       ar -rcs libcordic-angles.a ${OBJ}
        ar -cvq libcordic-angles.a ${OBJ}
        \cp -f ${INC} ${INCD}
        \cp -f ${LIB} ${LIBD}
        \rm -f ${OBJ}


print : Angles.make ${SRC}
        /bin/more $? > Angles.print

tar : Angles.make ${SRC}
        tar cvf Angles.tar $?

clean :
        \rm -f *.o *~ *#
#-------------------------------------------------------------
# copy include files    ${INC} into the directory ${INCD}
# copy library files    ${LIB} into the directory ${LIBD}
# copy executable files ${EXE} into the directory ${EXED}
# include files in ${INCS} directories to compile this module
#-------------------------------------------------------------
INCD = /home/young/MyWork/inc
LIBD = /home/young/MyWork/lib
EXED = /home/young/MyWork/exe


BurkDir = /home/young/MyWork/2.cordic_cpp/Burkadt
GHDLDir = /home/young/MyWork/7.cordic_accuracy/IF.GHDL

VPATH = ${BurkDir}:${GHDLDir}        \

INCS =  -I${BurkDir} -I${GHDLDir}    \

.SUFFIXES : .o .cpp .c

.cpp.o :
        g++ -c -Wall -g ${INCS} $<

.c.o :
        g++ -c -Wall -g ${INCS} $<


#-------------------------------------------------------------
```

```
# Classes
#------------------------------------------------------------
SRC = Core.hpp                          \
      Core.cpp                          \
      Core.1.fptr1.cordic_org.cpp       \
      Core.1.fptr2.cordic_burk.cpp      \
      Core.1.fptr3.cordic_vhdl.cpp      \
      Core.2.wrap1.cordic_stat.cpp      \
      Core.2.wrap2.cordic_break.cpp     \

OBJ = Core.o                            \
      Core.1.fptr1.cordic_org.o         \
      Core.1.fptr2.cordic_burk.o        \
      Core.1.fptr3.cordic_vhdl.o        \
      Core.2.wrap1.cordic_stat.o        \
      Core.2.wrap2.cordic_break.o       \


INC = Core.hpp                          \

LIB = libcordic-core.a                  \

EXE = Core_tb                           \



#------------------------------------------------------------
Core.o : cordic-burk cordic-ghdl ${SRC}
        g++ -c -Wall -g ${INCS} Core.cpp

cordic-burk :
        cd ${BurkDir}; make all;
        # cordic-burk library to ${LIBD}

cordic-ghdl :
        cd ${GHDLDir}; make all;
        # cordic-ghdl library to ${LIBD}
        # cordic_vtb executable to ${EXED}


#------------------------------------------------------------
all : ${OBJ} cordic-burk cordic-ghdl
#       ar -rcs libcordic-core.a cordic_core.o
        ar -cvq libcordic-core.a ${OBJ}
        \cp -f ${LIB} ${LIBD}
        \cp -f ${INC} ${INCD}
        \rm -f ${OBJ}

print : Core.make Core_tb.cpp ${SRC}
        /bin/more $? > Core.print
```

```
tar : Core.make Core_tb.cpp ${SRC}
        tar cvf Core.tar $?

clean :
        \rm -f *.o *~ *#




#-------------------------------------------------------------
# copy include files    ${INC} into the directory ${INCD}
# copy library files    ${LIB} into the directory ${LIBD}
# copy executable files ${EXE} into the directory ${EXED}
# include files in ${INCS} directories to compile this module
#-------------------------------------------------------------
INCD = /home/young/MyWork/inc
LIBD = /home/young/MyWork/lib
EXED = /home/young/MyWork/exe

VPATH =                     \

INCS =                      \

.SUFFIXES : .o .cpp .c

.cpp.o :
        g++ -c -Wall -g ${INCS} $<

.c.o :
        g++ -c -Wall -g ${INCS} $<


#-------------------------------------------------------------
# Classes
#-------------------------------------------------------------
SRC = Figures.cpp                       \
      Figures.hpp                       \

OBJ = Figures.o                         \

INC = Figures.hpp                       \

LIB = libcordic-figures.a               \

EXE =                                   \
```

```
#-------------------------------------------------------------
Figures.o : ${SRC}
        g++ -c -Wall -g  Figures.cpp


all : ${OBJ}
#       ar -rcs libcordic-figures.a ${OBJ}
        ar -cvq libcordic-figures.a ${OBJ}
        \cp -f ${LIB} ${LIBD}
        \cp -f ${INC} ${INCD}
        \rm -f ${OBJ}

print : Figures.make ${SRC}
        /bin/more $? > Figures.print

tar : Figures.make ${SRC}
        tar cvf Figures.tar $?

clean :
        \rm -f *.o *~ *#

#-------------------------------------------------------------
# copy include files    ${INC} into the directory ${INCD}
# copy library files    ${LIB} into the directory ${LIBD}
# copy executable files ${EXE} into the directory ${EXED}
# include files in ${INCS} directories to compile this module
#-------------------------------------------------------------
INCD = /home/young/MyWork/inc
LIBD = /home/young/MyWork/lib
EXED = /home/young/MyWork/exe

VPATH = ../Class.Angles

INCS = -I../Class.Angles    \

.SUFFIXES : .o .cpp .c

.cpp.o :
        g++ -c -Wall -g ${INCS} $<

.c.o :
        g++ -c -Wall -g ${INCS} $<


#-------------------------------------------------------------
# Classes
#-------------------------------------------------------------
SRC = GPData.cpp                        \
      GPData.hpp                        \
```

```
OBJ = GPData.o                          \

INC = GPData.hpp                        \

LIB = libcordic-gpdata.a                \

EXE =                                   \


#-------------------------------------------------------------
GPData.o : ${SRC} Angles.hpp
        g++ -c -Wall -g ${INCS} GPData.cpp


all : ${OBJ}
#       ar -rcs libcordic-gpdata.a ${OBJ}
        ar -cvq libcordic-gpdata.a ${OBJ}
        \cp -f ${LIB} ${LIBD}
        \cp -f ${INC} ${INCD}
        \rm -f ${OBJ}

print : GPData.make ${SRC}
        /bin/more $? > GPData.print

tar : GPData.make ${SRC}
        tar cvf GPData.tar $?

clean :
        \rm -f *.o *~ *#
```