

MPPA Cheat Sheet 1/2

Remember : `k1-command --help` or `-h`
 HW : Hardware SIMU: simulator

HW Target:

see `$k1-jtag-runner --help`

HW: Cluster :

`$k1-jtag-runner --exec-file=Cluster0:clusterexec`

HW: I/O+Cluster :

`$k1-jtag-runner --multibinary
 output/bin/myapply.mpk --exec-
 multibin=IODDR0:exec_io`

HW : HOST(x86)+I/O+CLUSTER

`$./output/bin/hostexec ./output/bin/myapply.mpk`

K1 Simulator Target :

see `$k1-mppa --help`
 see `$k1-cluster --help`
 see `$k1-pciesim-runner --help`

SIMU: Cluster :

`$k1-cluster -- clusterexec`

SIMU: I/O+Cluster :

`$k1-mppa -- ./output/bin/myappli.mpk`

SIMU : HOST(x86)+I/O+CLUSTER

`$k1-pciesim-runner ./output/bin/host_sim
 ./output/bin/myappli.mpk`

HowToGDB :

`/usr/local/k1tools/doc/Manuals/
 HowToGDB/HowToGDB.pdf`

HW: I/O+Cluster :

`$k1-jtag-runner -D --multibinary
 ./output/bin/myappli.mpk --exec-
 multibin=IODDR0:ioexec --sym-
 file=Cluster5:clusterexec`

SIMU: I/O+Cluster :

`$k1-mppa -D128 -D5 -- ./output/bin/myappli.mpk`

GDB :

`$k1-gdb -ex "attach-mppa NUMPORT"
 (gdb)file output/build/clusterexec
 (gdb) ..
 (gdb)info thread`

HowToTrace :

`/usr/local/k1tools/doc/Manuals/
 HowToTrace/HowToTrace.pdf`

Activate trace user :

At compilation :
`cflags:= -DMPPA_TRACE_ENABLE`

Activate traces :

`$k1-multibin-util -f
 output/bin/low_pass_filter_multibin.mpk --enable-
 tracepoint "mytrace*" -a`
 (for one binary use `k1-trace-util` command)

Launch aquisition :

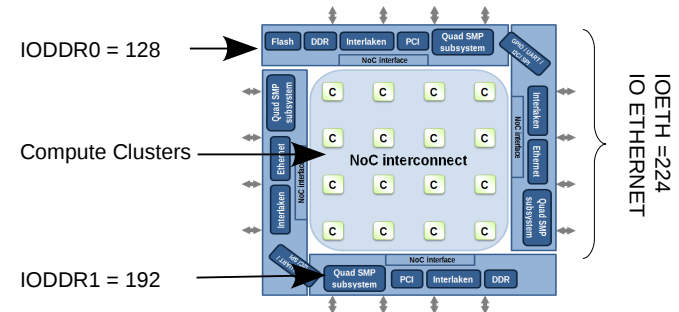
`$k1-trace-util --acquire=tracefile --lanes=0xff`

Launch your application (see TARGET HW)

At the end : split the result (ctr+c for aquisition)

Trace viewer :

`$k1-stv -hwtrace binary:tracefile.dump.1`



Compilation

see **KalrayMakefile**
`/usr/local/k1tools/doc/Manuals/
 KalrayMakefile/KalrayMakefile.pdf`
`k1-gcc --help=target`
 see **nodeos, rtems**
`/usr/local/k1tools/doc/Book/PosixProgramming.pdf`

Cluster :

Use `k1-gcc` as the compiler, with flag `-mos=nodeos`
`OS : Nodeos :`
`#include <mppa/osconfig.h>`
`COM : Network On Chip`
`#include <mppaipc.h>`
`Lib : -lmppaipc`

I/O :

Use `k1-gcc` as the compiler, with flag `-mos=rtems`
`OS : rtems :`
`#include <mppa/osconfig.h>`
`COM : Network On Chip, PCIe`
`#include <mppaipc.h>`
`Lib : -lmppaipc`

HOST :

Use `gcc` as the compiler
`COM : PCIe`
`#include <mppaipc.h>`
`Lib hw: -lmppaipc lib ISS : -lmppaipc_sim`

With kalrayMakefile :

`platforms := simulator hardware`
bare-mod : don't include `<mppa/osconfig.h>`

MPPA Cheat Sheet 2/2

Remember : `k1-command --helpor -h`

HW : Hardware SIMU: simulator

HowToK1DTV :

The K1 Disassembly Trace*

`/usr/local/k1tools/doc/Manuals/HowToK1DTV/HowToK1DTV.pdf`

Compile with `-g`

Launch your application :

SIMU: I/O or Cluster :

`$k1-cluster -p -- clusterexec`

`$k1-mppa -p -- io_exec`

SIMU: I/O+Cluster :

`$k1-mppa -p`

`--elf-symbols=5,cluster5exec --`

`./output/bin/myappli.mpk`

K1 Disassembly trace viewer :

`$k1-disasm-all -o <outputfile> profile/`

`$k1-dtv Cluster_1/PE.*`

**available in simulator only*

Callgrind Profiling :

⌘see Callgrinf Profiling*

`/usr/local/k1tools/doc/Book/PosixProgramming.pdf`

Support :

- Executed bundles
- Executed instructions (Equivalent to executed bundles in functional simulation)
- Executed cycles
- Instruction cache hits count
- Instruction cache misses count
- Data cache hits count
- Data cache misses count

Compile with `-g`

Launch your application :

SIMU: Cluster :

`$k1-cluster -p -- clusterexec`

KCachegrind tool :

`$k1-kcachegrind -o callgrind profile/Cluster_1/PE.0`

**available in simulator only*

DSU Timer

Return the DSU cycle

`uint64_t cpt= __k1_read_dsu_timestamp();`

k1-power

Power consumption

`/usr/local/k1tools/doc/Manuals/k1power/k1power.pdf`

Launch your application :

`$k1-power -- k1-jtag-runner --multibinary output/bin/myapply.mpk --exec-multibin=IODDR0:exec_io`

MPPAIPC Pathname

⌘see Process Management and Communication API

`/usr/local/k1tools/doc/Book/PosixProgramming.pdf`

Network-On-Chip Objects

Sync `/mppa/sync/rx_nodes:cnoc_tag`

Portal `/mppa/portal/rx_nodes:dnoc_tag`

RQueue `/mppa/rqueue/rx_node:dnoc_tag/tx_nodes:cnoc_tag/credits.mszie`

Channel `/mppa/channel/rx_node:dnoc_tag/tx_node:cnoc_tag`

`tx/rx_node(s) :`

* single node, e.g. 7

* range, e.g. [0..4]

* mixed, e.g. [0..4],7,128

`cnoc_tag` : Tag ID in range [1,127]

`dnoc_tag` : Tag ID in range [1,255]

PCIe Objects

Buffer `/mppa/buffer/rx_node#number/tx_node#number`

MQueue `/mppa/mqueue/rx_node#number/tx_node#number/mcount.tszie`
(value of rx/tx_node : pcie0 or pcie1)

More documentation :

Kalray Programming :

`/usr/local/k1tools/doc/Book/PosixProgramming.pdf`

Kalray tools and target :

`/usr/local/k1tools/doc/Book/SimulationTraceDebug.pdf`

ETHERNET documentation :

`/usr/local/k1tools/doc/Specifications/EthernetUsage/EthernetUsage.pdf`

ETHERNET Example :

`/usr/share/AccessCore/Apps/EthernetExamples/`