01 March 2022 22:06

<u>Planned code launch flow (the packet, cmd, accel-cmd processor must be launched in parallel?)=></u>

Host PC updates AQL packet->send doorbell signal->notify kernel agent->

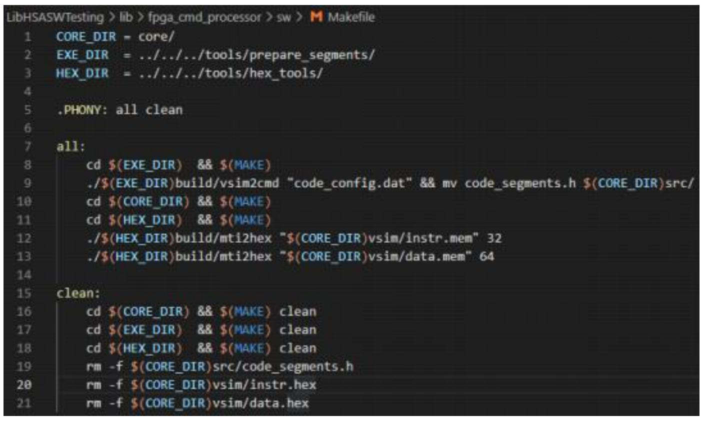
GPU sends an ACK Signal->Packet_processor->FCC->DMA->on board DRAM->FCC->DMA->completion signal

Host PC ACK->Host PC uses these contents->

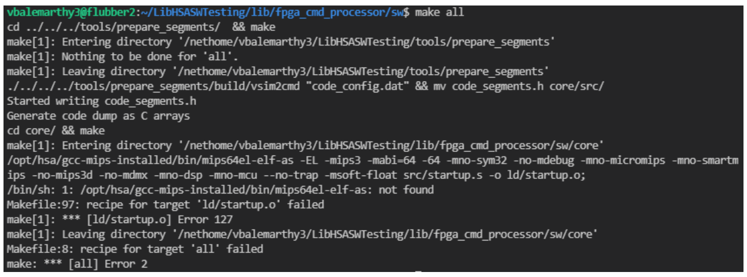
<u>Tools to generate .mti, .hex, code_segments.h file =></u>

1. **<u>vsim2cmd.cpp</u>**
   a. Reads a config file and an mti file and outputs a 'code_segments.h' file.
   b. This requires a config file with specific imem and dmem sizes?
   c. I tried to create one containing the instr and data mem location paths but it did not work.
   d.

```
/nethome/vbalemarthy3/LibHSASWTesting 0 0
```

   e. Found the sample config file to be given as an argument to vsim2cmd.cpp. A mention of this can be found in the Makefile of the fpga_cmd_processor/sw/ location. Contents of this Makefile are=>
      i. 

```
LibHSASWTesting > lib > fpga_cmd_processor > sw > M Makefile
1   CORE_DIR = core/
2   EXE_DIR  = ../../../tools/prepare_segments/
3   HEX_DIR  = ../../../tools/hex_tools/
4
5   .PHONY: all clean
6
7   all:
8       cd $(EXE_DIR)  && $(MAKE)
9       ./$(EXE_DIR)build/vsim2cmd "code_config.dat" && mv code_segments.h $(CORE_DIR)src/
10      cd $(CORE_DIR) && $(MAKE)
11      cd $(HEX_DIR)  && $(MAKE)
12      ./$(HEX_DIR)build/mti2hex "$(CORE_DIR)vsim/instr.mem" 32
13      ./$(HEX_DIR)build/mti2hex "$(CORE_DIR)vsim/data.mem" 64
14
15  clean:
16      cd $(CORE_DIR) && $(MAKE) clean
17      cd $(EXE_DIR)  && $(MAKE) clean
18      cd $(HEX_DIR)  && $(MAKE) clean
19      rm -f $(CORE_DIR)src/code_segments.h
20      rm -f $(CORE_DIR)vsim/instr.hex
21      rm -f $(CORE_DIR)vsim/data.hex
```

   f. Steps to run this Makefile=>
      i. cd ~/LibHSASWTesting/lib/fpga_cmd_processor/sw
      ii. make all
      iii. 

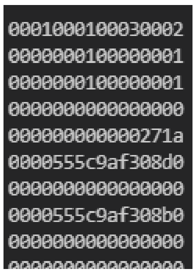
```
vbalemarthy3@flubber2:~/LibHSASWTesting/lib/fpga_cmd_processor/sw$ make all
cd ../../../tools/prepare_segments/  && make
make[1]: Entering directory '/nethome/vbalemarthy3/LibHSASWTesting/tools/prepare_segments'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/nethome/vbalemarthy3/LibHSASWTesting/tools/prepare_segments'
./../../../tools/prepare_segments/build/vsim2cmd "code_config.dat" && mv code_segments.h core/src/
Started writing code_segments.h
Generate code dump as C arrays
cd core/ && make
make[1]: Entering directory '/nethome/vbalemarthy3/LibHSASWTesting/lib/fpga_cmd_processor/sw/core'
/opt/hsa/gcc-mips-installed/bin/mips64el-elf-as -EL -mips3 -mabi=64 -64 -mno-sym32 -no-mdebug -mno-micromips -mno-smartm
ips -no-mips3d -no-mdmx -mno-dsp -mno-mcu --no-trap -msoft-float src/startup.s -o ld/startup.o;
/bin/sh: 1: /opt/hsa/gcc-mips-installed/bin/mips64el-elf-as: not found
Makefile:97: recipe for target 'ld/startup.o' failed
make[1]: *** [ld/startup.o] Error 127
make[1]: Leaving directory '/nethome/vbalemarthy3/LibHSASWTesting/lib/fpga_cmd_processor/sw/core'
Makefile:8: recipe for target 'all' failed
make: *** [all] Error 2
```

      iv. Running this Make errors out at the line 10:

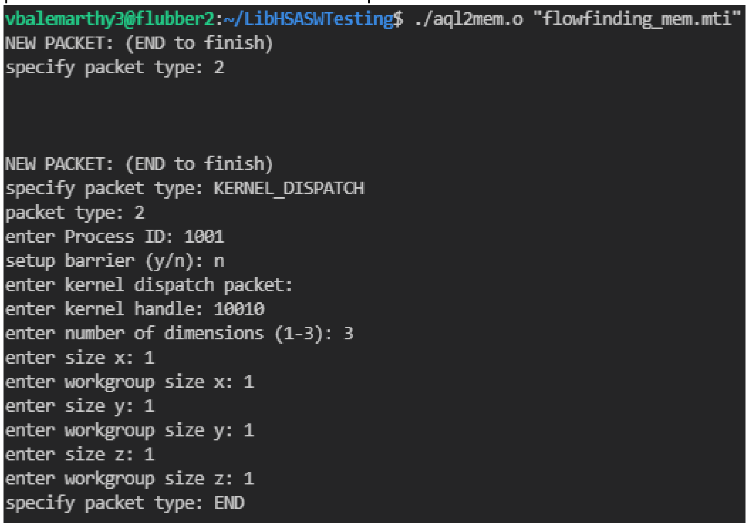
```
cd $(CORE_DIR) && $(MAKE)
```

      v. The Make file within core causes the error. Missing /opt/hsa/gcc-mips-installed/bin/mips64el-elf-as.

2. **<u>mti2hex.cpp</u>**
   a. Takes an mti file as input and converts it to a hex format file.
   b. g++ ./tools/hex_tools/src/mti2hex.cpp -o mti2hex.o
   c. ./mti2hex.o ./flowfinding_mem.mti
   d. Large file containing hex =>

i. 

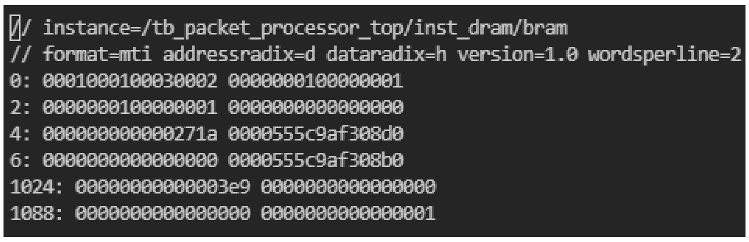
```
0001000100030002
0000000100000001
0000000100000001
0000000000000000
000000000000271a
0000555c9af308d0
0000000000000000
0000555c9af308b0
0000000000000000
```

3. **<u>Aql2mem.cpp</u>**
    a. This code generates an mti file.
    b. g++ ./tools/packet_tools/src/aql2mem.cpp -o aql2mem.o -I ./tools/packet_tools/include/
    c. ./aql2mem.o "test" "default"
    d. First argument is the name of the mti file to create, second argument is "default" for one packet or a number for more than one packet.
    e. 

```
vbalemarthy3@flubber2:~/LibHSASWTesting$ ./aql2mem.o "flowfinding_mem.mti"
NEW PACKET: (END to finish)
specify packet type: 2



NEW PACKET: (END to finish)
specify packet type: KERNEL_DISPATCH
packet type: 2
enter Process ID: 1001
setup barrier (y/n): n
enter kernel dispatch packet:
enter kernel handle: 10010
enter number of dimensions (1-3): 3
enter size x: 1
enter workgroup size x: 1
enter size y: 1
enter workgroup size y: 1
enter size z: 1
enter workgroup size z: 1
specify packet type: END
```

    f. 

```
// instance=/tb_packet_processor_top/inst_dram/bram
// format=mti addressradix=d dataradix=h version=1.0 wordsperline=2
0: 0001000100030002 0000000100000001
2: 0000000100000001 0000000000000000
4: 000000000000271a 0000555c9af308d0
6: 0000000000000000 0000555c9af308b0
1024: 00000000000003e9 0000000000000000
1088: 0000000000000000 0000000000000001
```

<u>LibHSA Processor components=></u>

1. <u>Fpga_cmd_processor=></u>
    a. Note this is the file where I have commented the 'code_segments.h'. Commenting this results in no error.
    b. Steps=>
        i. Invalidate all packet queues.
        ii. Initialize cores. The number of cores is determined by the number of '_' found in the 'code_config.dat'
        iii. Initializes a packet object with kernel arguments, grid sizes, completion signal values etc.
        iv. Assigns it to the packet processor and sends an interrupt to the packet processor via a 'send_aql_interrupt'.
        v. Wait for completion signal from the packet processor.
        vi. Frees up the memory for dst_image. Kernel arguments in the end.
    c. Also contains exception handler code.
    d. The fpga_cmd_processor.h header file contains helper functions to send interrupts to cores and to create headers for HAS packets.
    e. <u>Compile:</u>
        i. g++ ./lib/fpga_cmd_processor/sw/core/src/fpga_cmd_processor.c -o fpga_cmd_processor.o -I ./lib/fpga_cmd_processor/sw/include/
    f. <u>Error:</u>
        i. Getting a segmentation fault upon running fpga_cmd_processor. This might be due to the fpga_cmd_processor/sw/Makefile failing due to missing /opt/hsa/gcc-mips-installed/bin/mips64el-elf-as

2. <u>Packet_processor=></u>
    a. Runs an infinite loop that calls the following functions=>
        i. Process_aql_packets()

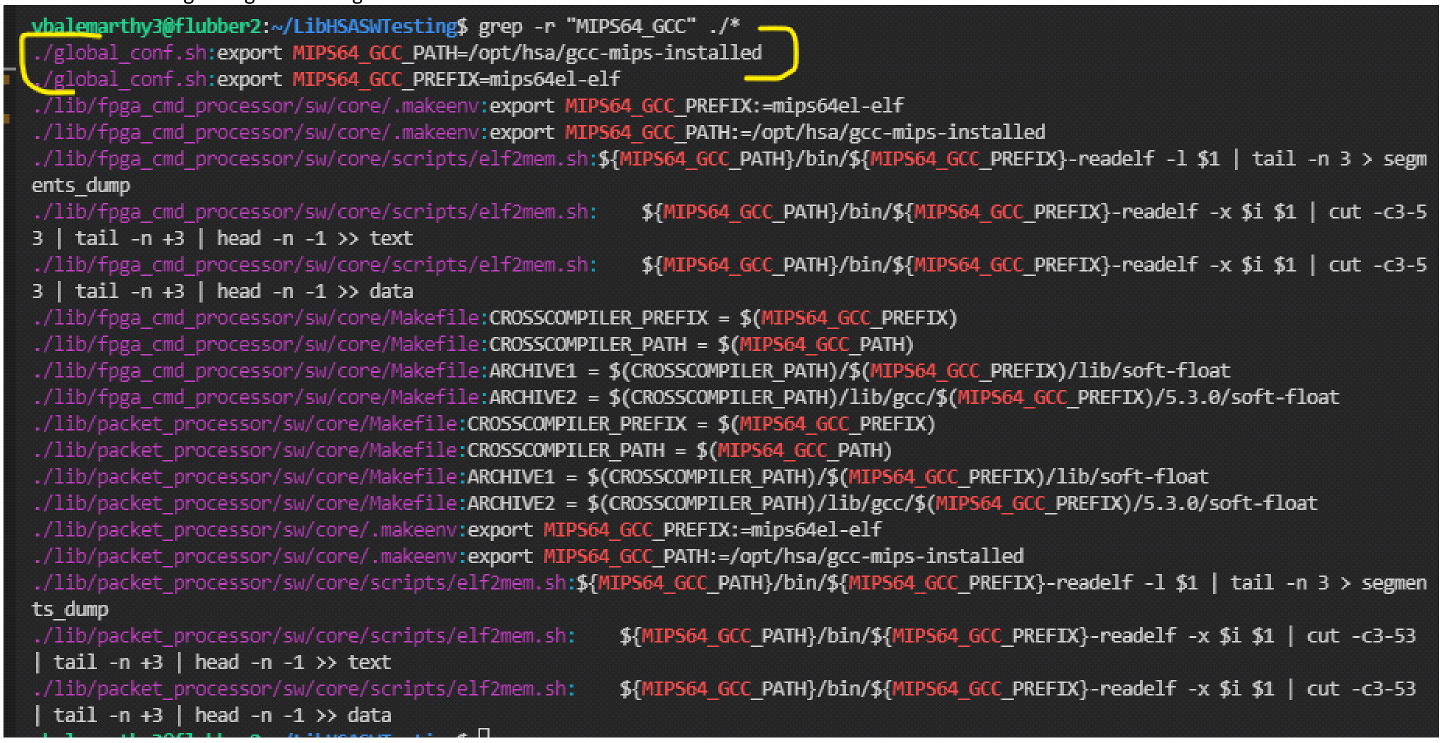
        ii. Process_dma_queue()
        iii. Process_launch_queue()
        iv. Process_dec_queue()
    b. Process_aql_packets()
        i. Handles processing of packets in the queue and handles the barrier sign.
    c. Process_dma_queue()
        i. Prioritize DMA writes. Basically this stops all interrupts, and then copies over the results (obtained from processing the packets) to the CPU memory (DMA_HOST_ADDR pointer being used).
    d. Process_launch_queue()
        i. Triggers work for a new free core if possible.
        ii. Works with the ACCEL command processor? There is a mention of the BASE_ACCEL_ADDR.
    e. Process_dec_queue()
        i. This function sends the completion signal. Call for 'send_completion_interrupt' present here.

3. Rom_accel_cmd_processor=>
    a. Handles interrupts from both the data-mover and the packet processor.
    b. A forever loop containing the following
        i. Reading config (from where? Config.dat?)
        ii. Updating the FPGA PE config w.r.t the task.
        iii. Reset PE.
        iv. Write config to datamover.
        v. Signal interrupt to packet processor that the computation is complete.

Solution: This is where code fails=>

```
# build startup object code
$(LD_DIR)startup.o:
    $(AS) $(ASFLAGS) $(SRC_DIR)startup.s -o $@;
```

Need to make changes to global config



```
vbalemarthy3@flubber2:~/LibHSASWTesting$ grep -r "MIPS64_GCC" ./*
./global_conf.sh:export MIPS64_GCC_PATH=/opt/hsa/gcc-mips-installed
./global_conf.sh:export MIPS64_GCC_PREFIX=mips64el-elf
./lib/fpga_cmd_processor/sw/core/.makeenv:export MIPS64_GCC_PREFIX:=mips64el-elf
./lib/fpga_cmd_processor/sw/core/.makeenv:export MIPS64_GCC_PATH:=/opt/hsa/gcc-mips-installed
./lib/fpga_cmd_processor/sw/core/scripts/elf2mem.sh:${MIPS64_GCC_PATH}/bin/${MIPS64_GCC_PREFIX}-readelf -l $1 | tail -n 3 > segm
ents_dump
./lib/fpga_cmd_processor/sw/core/scripts/elf2mem.sh:    ${MIPS64_GCC_PATH}/bin/${MIPS64_GCC_PREFIX}-readelf -x $i $1 | cut -c3-5
3 | tail -n +3 | head -n -1 >> text
./lib/fpga_cmd_processor/sw/core/scripts/elf2mem.sh:    ${MIPS64_GCC_PATH}/bin/${MIPS64_GCC_PREFIX}-readelf -x $i $1 | cut -c3-5
3 | tail -n +3 | head -n -1 >> data
./lib/fpga_cmd_processor/sw/core/Makefile:CROSSCOMPILER_PREFIX = $(MIPS64_GCC_PREFIX)
./lib/fpga_cmd_processor/sw/core/Makefile:CROSSCOMPILER_PATH = $(MIPS64_GCC_PATH)
./lib/fpga_cmd_processor/sw/core/Makefile:ARCHIVE1 = $(CROSSCOMPILER_PATH)/$(MIPS64_GCC_PREFIX)/lib/soft-float
./lib/fpga_cmd_processor/sw/core/Makefile:ARCHIVE2 = $(CROSSCOMPILER_PATH)/lib/gcc/$(MIPS64_GCC_PREFIX)/5.3.0/soft-float
./lib/packet_processor/sw/core/Makefile:CROSSCOMPILER_PREFIX = $(MIPS64_GCC_PREFIX)
./lib/packet_processor/sw/core/Makefile:CROSSCOMPILER_PATH = $(MIPS64_GCC_PATH)
./lib/packet_processor/sw/core/Makefile:ARCHIVE1 = $(CROSSCOMPILER_PATH)/$(MIPS64_GCC_PREFIX)/lib/soft-float
./lib/packet_processor/sw/core/Makefile:ARCHIVE2 = $(CROSSCOMPILER_PATH)/lib/gcc/$(MIPS64_GCC_PREFIX)/5.3.0/soft-float
./lib/packet_processor/sw/core/.makeenv:export MIPS64_GCC_PREFIX:=mips64el-elf
./lib/packet_processor/sw/core/.makeenv:export MIPS64_GCC_PATH:=/opt/hsa/gcc-mips-installed
./lib/packet_processor/sw/core/scripts/elf2mem.sh:${MIPS64_GCC_PATH}/bin/${MIPS64_GCC_PREFIX}-readelf -l $1 | tail -n 3 > segmen
ts_dump
./lib/packet_processor/sw/core/scripts/elf2mem.sh:    ${MIPS64_GCC_PATH}/bin/${MIPS64_GCC_PREFIX}-readelf -x $i $1 | cut -c3-53
| tail -n +3 | head -n -1 >> text
./lib/packet_processor/sw/core/scripts/elf2mem.sh:    ${MIPS64_GCC_PATH}/bin/${MIPS64_GCC_PREFIX}-readelf -x $i $1 | cut -c3-53
| tail -n +3 | head -n -1 >> data
```