



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Э. БАУМАНА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
(МГТУ им. Н.Э. БАУМАНА)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Программное обеспечение ЭВМ и информационные технологии»

НАПРАВЛЕНИЕ ПОДГОТОВКИ _____ «09.03.04 Программная инженерия»

ОТЧЕТ ПО ЧАСТИ ПРАКТИКУМА №1

Название: _____ Изучение принципов работы микропроцессорного ядра RISC-V

Дисциплина: _____ Архитектура ЭВМ

| | | | |
|---------------|---------|---------------|---------------|
| Студент | ИУ7-56Б | _____ | А. Д. Ковель |
| | Группа | Подпись, дата | И. О. Фамилия |
| Преподаватель | | _____ | А. Ю. Попов |
| | | Подпись, дата | И. О. Фамилия |

Москва, 2022 г.

В данной работе будет выполнен 6 вариант.

Листинг 1 – Код host_main.cpp

```
1 #include <iostream>
2 #include <stdio.h>
3 #include <stdexcept>
4 #include <iomanip>
5 #ifdef _WINDOWS
6 #include <io.h>
7 #else
8 #include <unistd.h>
9 #endif
10
11
12 #include "experimental/xrt_device.h"
13 #include "experimental/xrt_kernel.h"
14 #include "experimental/xrt_bo.h"
15 #include "experimental/xrt_ini.h"
16
17 #include "gpc_defs.h"
18 #include "leonhardx64_xrt.h"
19 #include "gpc_handlers.h"
20
21 #define BURST 128
22
23 union uint64 {
24     uint64_t    u64;
25     uint32_t    u32[2];
26     uint16_t    u16[4];
27     uint8_t     u8[8];
28 };
29
30 typedef struct adr{
31     uint8_t     u8[4];
32 } adr_t;
33
34 uint64_t rand64() {
35     uint64 tmp;
36     tmp.u32[0] = rand();
37     tmp.u32[1] = rand();
38     return tmp.u64;
39 }
40
41 static void usage()
42 {
43     std::cout << "usage: <xclbin> <sw_kernel>\n\n";
```

```

44 }
45
46 int main(int argc, char** argv)
47 {
48
49
50     uint16_t val;
51     uint64 valad;
52     printf("input addt:\n");
53     scanf("%d.%d.%d.%d", &(valad.u16[0]), &(valad.u16[1]), &(valad.u16[2]), &(valad.u16[3]));
54     val = (valad.u16[2] - 32) * 256 + valad.u16[3];
55
56     uint16_t n;
57     printf("input number:\n");
58     scanf("%d", &n);
59
60
61     __foreach_core(group, core) {
62         lnh_inst.gpc[group][core]->mq_send(val);
63         lnh_inst.gpc[group][core]->mq_send(n);
64     }
65
66     printf("out\n");
67     // __foreach_core(group, core) {
68         // lnh_inst.gpc[group][core]->buf_read(count[group][core]*2*sizeof(uint64_t));
69         // }
70
71     // __foreach_core(group, core) {
72         // lnh_inst.gpc[group][core]->buf_read_join();
73         // }
74
75     // uint64 val;
76     // printf("input addr:\n");
77     // scanf("%d %d %d %d", &(val.u16[0]), &(val.u16[1]), &(val.u16[2]), &(val.u16[3]));
78     uint16_t vaaa;
79     __foreach_core(group, core) {
80         vaaa = lnh_inst.gpc[group][core]->mq_receive();
81         printf("sch 1: %d\n", vaaa);
82         vaaa = lnh_inst.gpc[group][core]->mq_receive();
83         printf("sch 2: %d\n", vaaa);
84         vaaa = lnh_inst.gpc[group][core]->mq_receive();
85         printf("sch 3: %d\n", vaaa);
86         vaaa = lnh_inst.gpc[group][core]->mq_receive();
87         printf("sch 4: %d\n", vaaa);
88     }
89
90
91     __foreach_core(group, core) {

```

```
92         free(host2gpc_buffer[group][core]);
93         //free(gpc2host_buffer[group][core]);
94     }
95
96
97     return 0;
98 }
```

Листинг 2 – Код sw_kernel_main.cpp

```
1  /*
2  * gpc_test.c
3  *
4  * sw_kernel library
5  *
6  * Created on: April 23, 2021
7  * Author: A.Popov
8  */
9
10 #include <stdlib.h>
11 #include <unistd.h>
12 #include "lnh64.h"
13 #include "gpc_io_swk.h"
14 #include "gpc_handlers.h"
15
16 #define SW_KERNEL_VERSION 26
17 #define DEFINE_LNH_DRIVER
18 #define DEFINE_MQ_R2L
19 #define DEFINE_MQ_L2R
20 #define __fast_recall__
21
22 #define TEST_STRUCTURE 1
23
24 extern lnh lnh_core;
25 extern global_memory_io gmio;
26 volatile unsigned int event_source;
27
28
29 // union uint64 {
30 //     uint64_t      u64;
31 //     uint32_t      u32[2];
32 //     uint16_t      u16[4];
33 //     uint8_t       u8[8];
34 // };
35
36
37 int main(void) {
38     //////////////////////////////////////
39     //                               Main Event Loop
40     //////////////////////////////////////
41     //Leonhard driver structure should be initialised
42     lnh_init();
43     //Initialise host2gpc and gpc2host queues
44     gmio_init(lnh_core.partition.data_partition);
45     for (;;) {
46         //Wait for event
47         while (!gpc_start());
```

```

48     //Enable RW operations
49     set_gpc_state(BUSY);
50     //Wait for event
51     event_source = gpc_config();
52     switch(event_source) {
53         ///////////////////////////////////////////////////
54         // Measure GPN operation frequency
55         ///////////////////////////////////////////////////
56         case __event__(insert_burst) : insert_burst(); break;
57         case __event__(search_burst) : search_burst(); break;
58     }
59     //Disable RW operations
60     set_gpc_state(IDLE);
61     while (gpc_start());
62
63 }
64 }
65
66
67 void insert_burst() {
68
69     lnh_del_str_sync(TEST_STRUCTURE);
70     unsigned int count = mq_receive();
71     unsigned int size_in_bytes = count*(sizeof(uint16_t));
72     uint16_t *buffer = (uint16_t*)malloc(size_in_bytes);
73     buf_read(size_in_bytes, (char*)buffer);
74     for (int i=0; i<count; i++) {
75         lnh_ins_sync(TEST_STRUCTURE, buffer[i], 0);
76     }
77     lnh_sync();
78     free(buffer);
79 }
80
81
82 void search_burst() {
83
84     lnh_sync();
85     unsigned int count = lnh_get_num(TEST_STRUCTURE);
86     uint16_t ip = mq_receive();
87     uint8_t n = mq_receive();
88     //mq_send(ip);
89     uint64 val;
90
91     if (lnh_search(TEST_STRUCTURE, ip)) {
92         val.u64 = lnh_core.result.value;
93         val.u16[n]++;
94         lnh_ins_async(TEST_STRUCTURE, ip, val.u64);
95     } else {

```

```

96     val.u64 = 0ull;
97     val.u16[n]++;
98     lnh_ins_async(TEST_STRUCTURE, ip, val.u64);
99 }
100 mq_send(val.u16[0]);
101 mq_send(val.u16[1]);
102 mq_send(val.u16[2]);
103 mq_send(val.u16[3]);
104 }

```

Изображение работы программы:

```

(iu7012) 195.19.32.95 — Терминал Fly
Файл  Правка  Настройка  Справка
[Icons]  [Search]  [Dropdown: ls]  [Checkmark]

LD_LIBRARY_PATH : /opt/xilinx/xrt/lib:
PYTHONPATH      : /opt/xilinx/xrt/python:
iu7012@d1580:~$ cd Desktop/worksp/
iu7012@d1580:~/Desktop/worksp$ ./host/host_main leonhard_2cores_267mhz.xclbin ./
sw-kernel/sw_kernel.rawbinary
-bash: ./host/host_main: No such file or directory
iu7012@d1580:~/Desktop/worksp$ ./host_main leonhard_2cores_267mhz.xclbin sw_kern
el_main.rawbinary
-bash: ./host_main: No such file or directory
iu7012@d1580:~/Desktop/worksp$ ./host_main leonhard_2cores_267mhz.xclbin sw_kern
el_main.rawbinary
-bash: ./host_main: No such file or directory
iu7012@d1580:~/Desktop/worksp$ make
make: *** No targets specified and no makefile found. Stop.
iu7012@d1580:~/Desktop/worksp$ ls
disc-example  IDE.log  Leonhardx64_xrt_system  RemoteSystemsTempFiles
iu7012@d1580:~/Desktop/worksp$ cd disc-example/
iu7012@d1580:~/Desktop/worksp/disc-example$ ./host_main leonhard_2cores_267mhz.x
clbin sw_kernel_main.rawbinary
terminate called after throwing an instance of 'xrt_core::error'
  what():  failed to open ip context
Aborted (core dumped)
iu7012@d1580:~/Desktop/worksp/disc-example$ xbutil reset
All existing processes will be killed.
Are you sure you wish to proceed? [y/n]: y
iu7012@d1580:~/Desktop/worksp/disc-example$ ./host_main leonhard_2cores_267mhz.x
clbin sw_kernel_main.rawbinary
Bbegute agpec:
23
Bbegute номер:
3
out
Счетчик №1: 0
Счетчик №2: 0
Счетчик №3: 0
Счетчик №4: 1
iu7012@d1580:~/Desktop/worksp/disc-example$

```