

PART I

First example I use the `ovp/example/using systemc/helloworld/example.sh`

Hello world is the most easiest program

there are three parts to the example

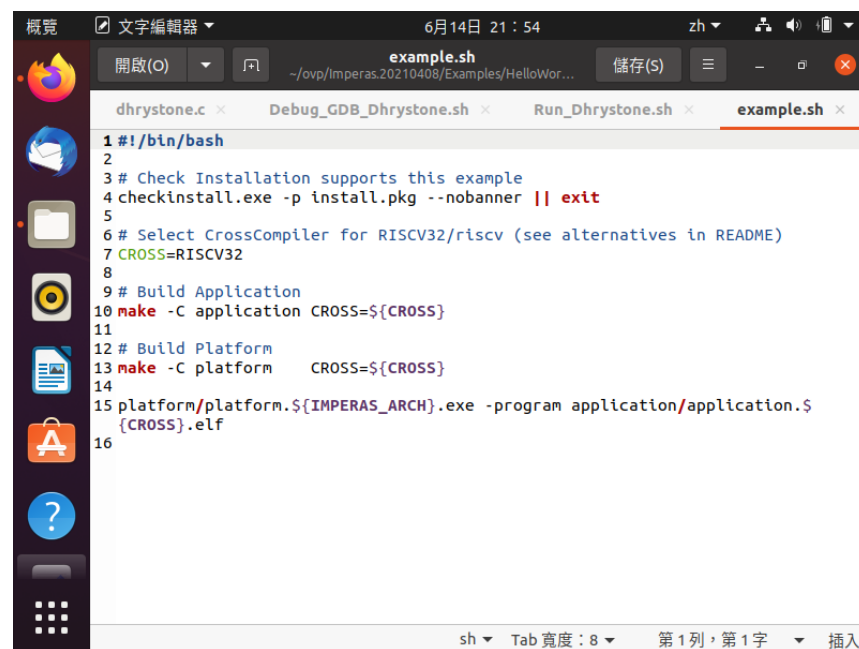
1. An application; this is found as `application/application.c`
2. A `CpuManager/OVPsim SystemC TLM2.0` platform; this is in `platform/platform.cpp`
3. A Makefile to build the application in 1; this is in `application/Makefile`,

In addition a standard Makefile is used to build the platform. This is included into `platform/Makefile` and is found at

`IMPERAS_HOME/ImperasLib/buildutils/Makefile.TLM.platform`

With this `example.sh` I can see how to set platform & how to change `c` to the `elf` file

And next I find the `platform.cpp` I know what I need to build in BareMetal format in figure 3 it also teach me how to set parser & use platform as a top module



```
概覽 文字編輯器 6月14日 21:54 zh 儲存(S)
example.sh
~/ovp/Imperas.20210408/Examples/HelloWor...
dhystone.c x Debug_GDB_Dhystone.sh x Run_Dhystone.sh x example.sh x
1 #!/bin/bash
2
3 # Check Installation supports this example
4 checkinstall.exe -p install.pkg --nobanner || exit
5
6 # Select CrossCompiler for RISCV32/riscv (see alternatives in README)
7 CROSS=RISCV32
8
9 # Build Application
10 make -C application CROSS=${CROSS}
11
12 # Build Platform
13 make -C platform CROSS=${CROSS}
14
15 platform/platform.${IMPERAS_ARCH}.exe -program application/application.$
16 {CROSS}.elf
```

sh Tab 寬度: 8 第 1 列, 第 1 字 插入

```
make: 離開目錄「/home/chini88/ovp/Imperas.20210408/Examples/HelloWorld/usingSystemC/platform」

SystemC 2.3.3-Accellera --- Jun  7 2021 21:06:57
Copyright (c) 1996-2018 by all Contributors,
ALL RIGHTS RESERVED

OVPsim (32-Bit) v20210408.0 Open Virtual Platform simulator from www.OVPworld.org.
Copyright (c) 2005-2021 Imperas Software Ltd.  Contains Imperas Proprietary Information.
Licensed Software, All Rights Reserved.
Visit www.IMPERAS.com for multicore debug, verification and analysis solutions.

OVPsim started: Mon Jun 14 21:54:22 2021

Hello World

OVPsim finished: Mon Jun 14 21:54:22 2021

OVPsim (32-Bit) v20210408.0 Open Virtual Platform simulator from www.OVPworld.org.
Visit www.IMPERAS.com for multicore debug, verification and analysis solutions.

chini88@chini88-VirtualBox:~/ovp/Imperas.20210408/Examples/HelloWorld/usingSystemC$
```

```
64     }
65
66 }; /* BareMetal */
67
68 BareMetal::BareMetal (sc_module_name name)
69     : sc_module (name)
70     , Platform ("")
71     , bus1 (Platform, "bus1", 2, 1)
72     , ram1 (Platform, "ram1", 0xffffffff)
73     , cpu1 ( Platform, "cpu1",  paramsForcpu1())
74 {
75     bus1.connect(cpu1.INSTRUCTION);
76     bus1.connect(cpu1.DATA);
77     bus1.connect(ram1.sp1, 0x0, 0xffffffff);
78 }
79
80 int sc_main (int argc, char *argv[]) {
81
82     // start the CpuManager session
83     session s;
84
85     // create a standard command parser and parse the command line
86     parser p(argc, (const char**) argv);
87
88     // create an instance of the platform
89     BareMetal top ("top");
90 }
```

second example I choose

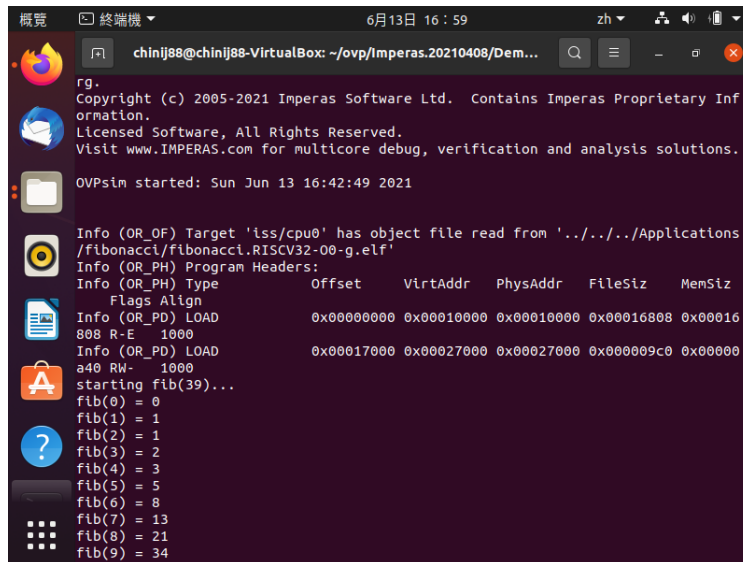
IMPERAS_HOME/Demo/Processors/RISCV/riscv32/RV32G/single_core/ Fibonacci.sh
to execute and get the information I want

1. I choose it because I think it is easy to help me to realize what I need to write in the c code and whether I can use the common library in OVP to model it
2. I find the virtual address / physical address and memory size are include in the result ◦ These information may need to be set by myself when I try to model the DMA & CPU system using OVP and help me to know how to debug with memory

problem i.e. segmentation fault or something else

3. I know I have to set the processor number in the .sh file when I use .sh to execute elf file and to write the processor vendor as riscv.ovpworld.org

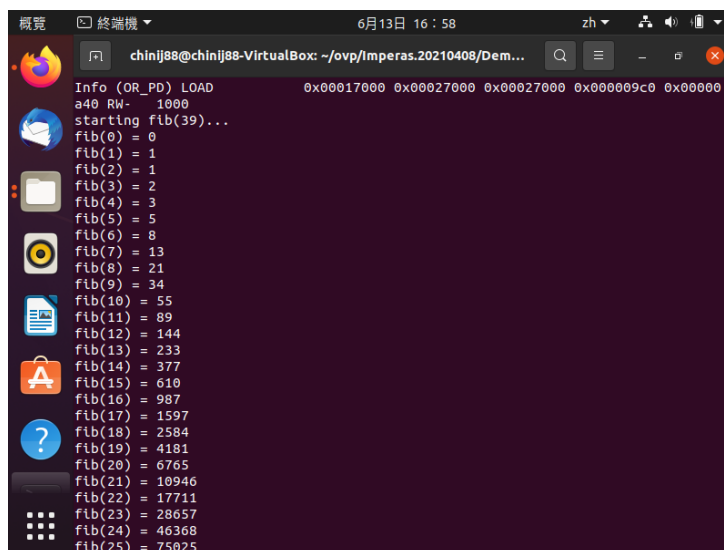
4. I choose it because I think the recurrence and math function is a basic function when we use to write code and I will use this math function when I try to design other thing



```
rg.
Copyright (c) 2005-2021 Imperas Software Ltd.  Contains Imperas Proprietary Inf
ormation.
Licensed Software, All Rights Reserved.
Visit www.IMPERAS.com for multicore debug, verification and analysis solutions.

OVPsim started: Sun Jun 13 16:42:49 2021

Info (OR_OF) Target 'iss/cpu0' has object file read from '../Applications
/fibonacci/fibonacci.RISCV32-00-g.elf'
Info (OR_PH) Program Headers:
Info (OR_PH) Type      Offset      VirtAddr      PhysAddr      FileSiz      MemSiz
Flags Align
Info (OR_PD) LOAD      0x00000000 0x00010000 0x00010000 0x00016808 0x00016
808 R-E 1000
Info (OR_PD) LOAD      0x00017000 0x00027000 0x00027000 0x000009c0 0x00000
a40 RW- 1000
starting fib(39)...
fib(0) = 0
fib(1) = 1
fib(2) = 1
fib(3) = 2
fib(4) = 3
fib(5) = 5
fib(6) = 8
fib(7) = 13
fib(8) = 21
fib(9) = 34
```



```
Info (OR_PD) LOAD      0x00017000 0x00027000 0x00027000 0x000009c0 0x00000
a40 RW- 1000
starting fib(39)...
fib(0) = 0
fib(1) = 1
fib(2) = 1
fib(3) = 2
fib(4) = 3
fib(5) = 5
fib(6) = 8
fib(7) = 13
fib(8) = 21
fib(9) = 34
fib(10) = 55
fib(11) = 89
fib(12) = 144
fib(13) = 233
fib(14) = 377
fib(15) = 610
fib(16) = 987
fib(17) = 1597
fib(18) = 2584
fib(19) = 4181
fib(20) = 6765
fib(21) = 10946
fib(22) = 17711
fib(23) = 28657
fib(24) = 46368
fib(25) = 75025
```

```
6月13日 16:58
chini88@chini88-VirtualBox: ~/ovp/imperas.20210408/Dem...
finishing...
Info -----
Info CPU 'lss/cpu0' STATISTICS
Info Type : riscv (RV32G)
Info NonInal MIPS : 100
Info Final program counter : 0x203ac
Info Simulated instructions: 7,120,126,553
Info Simulated MIPS : 1358.8
Info -----
Info -----
Info SIMULATION TIME STATISTICS
Info Simulated time : 71.20 seconds
Info User time : 5.24 seconds
Info System time : 0.00 seconds
Info Elapsed time : 5.40 seconds
Info Real time ratio : 13.19x faster
Info -----
OVPsim finished: Sun Jun 13 16:42:55 2021

OVPsim (32-Bit) v20210408.0 Open Virtual Platform simulator from www.OVPworld.org.
Visit www.IMPERAS.com for multicore debug, verification and analysis solutions.
```

third example I choose

IMPERAS_HOME/Demo/Processors/RISCV/riscv32/RV32G/single_core/Run_linpack.sh

1. I choose it because I am interested in it since I never see a how to construct a benchmark program in real and it is more complicated than I think
2. It helps me to know how to deal with 2D array this may be helpful for me to construct a DMA when I try to construct a more complicated memory system
3. It also help me how to deal with double precision number ,I think this may be helpful for construct a CPU system related to RISCV ,since I usually don't know how to design a precise and clear floating point processor in hardware

```
6月13日 20:34
chini88@chini88-VirtualBox: ~/ovp/imperas.20210408/Dem...
OVPsim started: Sun Jun 13 20:27:49 2021

Info (OR_OF) Target 'lss/cpu0' has object file read from '../Applications/linpack/linpack.RISCV32-00-g.elf'
Info (OR_PH) Program Headers:
Info (OR_PH) Type Offset VirtAddr PhysAddr FileSiz MemSiz
Flags Align
Info (OR_PD) LOAD 0x00000000 0x00010000 0x00010000 0x0001cad0 0x0001c
a00 R-E 1000
Info (OR_PD) LOAD 0x0001d000 0x0002d000 0x0002d000 0x000009c0 0x0004f
a20 RW- 1000
Begin 100
1: b[0]=0.000000
1: b[1]=0.000000
1: b[2]=0.000000
1: b[3]=0.000000
1: b[4]=0.000000
1: b[5]=0.000000
1: b[6]=0.000000
1: b[7]=0.000000
1: b[8]=0.000000
1: b[9]=0.000000
1: b[10]=0.000000
1: b[11]=0.000000
1: b[12]=0.000000
1: b[13]=0.000000
1: b[14]=0.000000
1: b[15]=0.000000
```

```
1: b[95]=0.000000
1: b[96]=0.000000
1: b[97]=0.000000
1: b[98]=0.000000
1: b[99]=0.000000
2: b[0]=-14.475327
2: b[1]=-10.553467
2: b[2]=20.416260
2: b[3]=-11.188232
2: b[4]=-7.226318
2: b[5]=-10.244873
2: b[6]=-7.228271
2: b[7]=7.651611
2: b[8]=-12.714600
2: b[9]=-1.123779
2: b[10]=-3.810303
2: b[11]=8.803955
2: b[12]=8.359619
2: b[13]=-24.190186
2: b[14]=1.070166
2: b[15]=-20.731201
2: b[16]=-5.003662
2: b[17]=-16.444892
2: b[18]=4.213135
2: b[19]=2.040143
2: b[20]=22.195557
2: b[21]=13.114502
2: b[22]=2.818604
2: b[23]=4.135986

0.000000 0.000038 0.000000 -0.000014 -0.000008
0.000000 0.000038 0.000000 -0.000014 -0.000008
0.000000 0.000038 0.000000 -0.000014 -0.000008
0.000000 0.000038 0.000000 -0.000014 -0.000008
0.000000 0.000038 0.000000 -0.000014 -0.000008
0.000000 0.000038 0.000000 -0.000014 -0.000008
0.000000 0.000038 0.000000 -0.000014 -0.000008
0.000000 0.000038 0.000000 -0.000014 -0.000008
0.000000 0.000038 0.000000 -0.000014 -0.000008
1.596053 0.000038 0.000000 -0.000014 -0.000008
End (ntimes=11) 100
Info
Info -----
Info CPU 'iss/cpu0' STATISTICS
Info Type : riscv (RV32G)
Info Nominal MIPS : 100
Info Final program counter : 0x23928
Info Simulated instructions: 5,541,350,838
Info Simulated MIPS : 2147.8
Info -----
Info
Info SIMULATION TIME STATISTICS
Info Simulated time : 55.41 seconds
Info User time : 2.58 seconds
Info System time : 0.00 seconds
Info Elapsed time : 2.70 seconds
Info Real time ratio : 20.50x faster
Info -----
```

fourth example I choose

IMPERAS_HOME/Demo/Processors/RISCV/riscv32/RV32G/single_core/Run_Peaksped.sh

1. it is a easy code for a person to realize what it is doing but it help me to know how long does it takes to execute about 10 thousand instruction about operation
- 2.It help me to know the real time ratio can be enough large for me to execute a complex program through OVP
- 3.It helps me to evaluate the run time ,when a program simulate more than 1 minute when I testing my code ,it seems like there is something wrong and I don't need to wait more time

```

Info (OR_OF) Target 'iss/cpu0' has object file read from '../Applications/peakSpeed2/peakSpeed2.RISCv32-00-g.elf'
Info (OR_PH) Program Headers:
Info (OR_PH) Type      Offset      VirtAddr    PhysAddr    FileSiz    MemSiz
Flags Align
Info (OR_PD) LOAD      0x00000000  0x00010000  0x00010000  0x00016770 0x00016
770 R-E 1000
Info (OR_PD) LOAD      0x00017000  0x00027000  0x00027000  0x000009c0 0x00000
a40 RW- 1000
result=-999999963
Info
Info -----
Info CPU 'iss/cpu0' STATISTICS
Info Type              : riscv (RV32G)
Info Nominal MIPS      : 100
Info Final program counter : 0x20344
Info Simulated instructions: 27,000,012,477
Info Simulated MIPS     : 10714.3
Info -----
Info
Info -----
Info SIMULATION TIME STATISTICS
Info Simulated time     : 270.00 seconds
Info User time          : 2.52 seconds
Info System time        : 0.00 seconds
Info Elapsed time       : 2.59 seconds
Info Real time ratio    : 104.18x faster
Info -----

```

The example five I choose

IMPERAS_HOME/Example /Platform construction/systemC_TLM/example.sh

This example teach me how to communicate and pass some information through platform ,it also teach me how to construct the platform through these header &cpp file

```

#include "tlm/tlmMemory.hpp"
#include "tlm/tlmDecoder.hpp"
#include "ovpworld.org/processor/or1k/1.0/tlm/processor.igen.hpp"
#include "national.ovpworld.org/peripheral/16550/1.0/tlm/pse.igen.hpp"
#include "tlm.h"
#include "tlm/tlmModule.hpp"
extern "C" {
#include "platform.sc_options.igen.h"
}
#include "platform.sc_constructor.igen.h"

```

And it also teach me the transaction between the module & how to send interrupt signal in this homework and make some point as follow

Note that since a SystemC thread is not normally interrupted, the only source that can interrupt a processor during its quantum is something affected by the processor execution e.g. the processor accessing a model which raises the interrupt. An interrupt raised by another model or processor will be serviced next time the processor is scheduled.

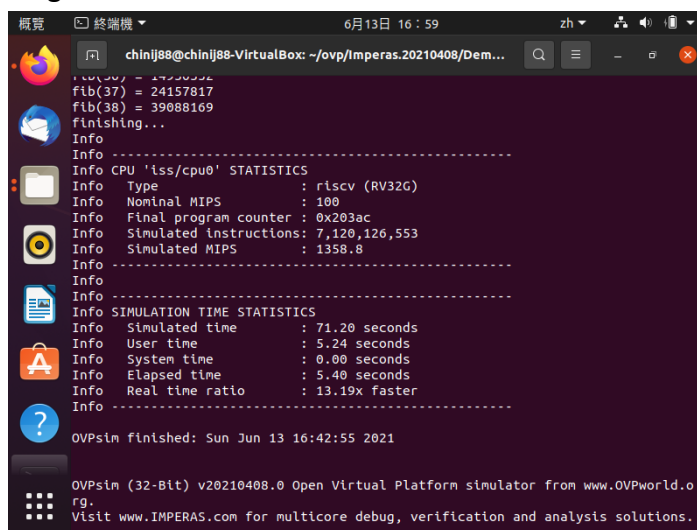
```
概覽 終端機 6月15日 09:48 zh 6月15日 09:48 zh 6月15日 09:48 zh
chini88@chini88-VirtualBox: ~/ovp/Imperas.20210408/Exam...
TEST: Send String: Hello World
Send char H (0x48)
Interrupt Handler 0x02 (2)
Character sent
Send char e (0x65)
Interrupt Handler 0x02 (3)
Character sent
Send char l (0x6c)
Interrupt Handler 0x02 (4)
Character sent
Send char l (0x6c)
Interrupt Handler 0x02 (5)
Character sent
Send char o (0x6f)
Interrupt Handler 0x02 (6)
Character sent
Send char  (0x20)
Interrupt Handler 0x02 (7)
Character sent
Send char W (0x57)
Interrupt Handler 0x02 (8)
Character sent
Send char o (0x6f)
Interrupt Handler 0x02 (9)
Character sent
Send char r (0x72)
Interrupt Handler 0x02 (10)
Character sent
```

```
概覽 終端機 6月15日 09:48 zh 6月15日 09:48 zh 6月15日 09:48 zh
chini88@chini88-VirtualBox: ~/ovp/Imperas.20210408/Exam...
Interrupt Handler 0x02 (5)
Character sent
Send char o (0x6f)
Interrupt Handler 0x02 (6)
Character sent
Send char  (0x20)
Interrupt Handler 0x02 (7)
Character sent
Send char W (0x57)
Interrupt Handler 0x02 (8)
Character sent
Send char o (0x6f)
Interrupt Handler 0x02 (9)
Character sent
Send char r (0x72)
Interrupt Handler 0x02 (10)
Character sent
Send char l (0x6c)
Interrupt Handler 0x02 (11)
Character sent
Send char d (0x64)
Interrupt Handler 0x02 (12)
Character sent
Send char  (0x0a)
Interrupt Handler 0x02 (13)
Character sent
TEST: main done
```

Because I am interested in other example so I choose sixth example
IMPERAS_HOME/Demo/Processors/RISCV/riscv32/RV32G/many_core/Run_Amp4_D
hrystone.sh V.S single_core/ Run_Dhrystone.sh

1.Although this homework is a single core riscv ,I am interested in the difference
between single core and multicore system through ovp platform,since I never design
a multicore hardware ,I try to understand the difference and their speed gap
Comparing them I find their difference in MIPS is little, I think this may represent
each processor is simultaneously executing and they are separate ,it help me to
realize the BareMetal more

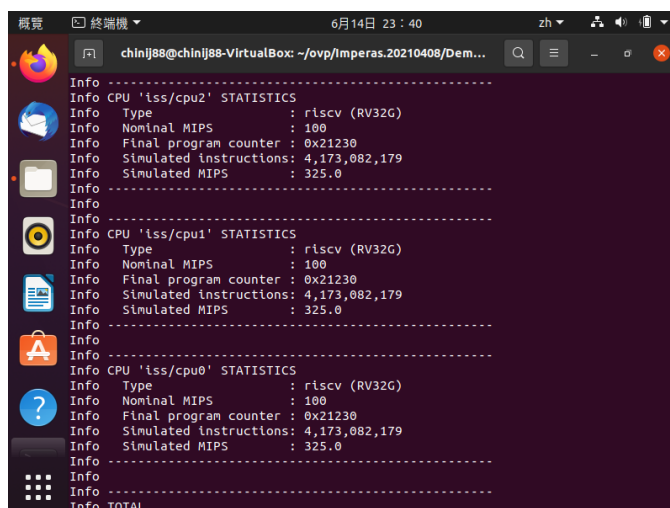
Single core



```
chinj88@chinij88-VirtualBox: ~/ovp/imperas.20210408/Dem...
fib(37) = 24157817
fib(38) = 39088169
finishing...
Info
Info -----
Info CPU 'iss/cpu0' STATISTICS
Info Type : riscv (RV32G)
Info Nominal MIPS : 100
Info Final program counter : 0x203ac
Info Simulated instructions: 7,120,126,553
Info Simulated MIPS : 1358.8
Info -----
Info
Info -----
Info SIMULATION TIME STATISTICS
Info Simulated time : 71.20 seconds
Info User time : 5.24 seconds
Info System time : 0.00 seconds
Info Elapsed time : 5.40 seconds
Info Real time ratio : 13.19x faster
Info -----
Info
OVPsim finished: Sun Jun 13 16:42:55 2021

OVPsim (32-Bit) v20210408.0 Open Virtual Platform simulator from www.OVPworld.org.
Visit www.IMPERAS.com for multicore debug, verification and analysis solutions.
```

Multicore



```
chinj88@chinij88-VirtualBox: ~/ovp/imperas.20210408/Dem...
Info
Info CPU 'iss/cpu2' STATISTICS
Info Type : riscv (RV32G)
Info Nominal MIPS : 100
Info Final program counter : 0x21230
Info Simulated instructions: 4,173,082,179
Info Simulated MIPS : 325.0
Info -----
Info
Info CPU 'iss/cpu1' STATISTICS
Info Type : riscv (RV32G)
Info Nominal MIPS : 100
Info Final program counter : 0x21230
Info Simulated instructions: 4,173,082,179
Info Simulated MIPS : 325.0
Info -----
Info
Info CPU 'iss/cpu0' STATISTICS
Info Type : riscv (RV32G)
Info Nominal MIPS : 100
Info Final program counter : 0x21230
Info Simulated instructions: 4,173,082,179
Info Simulated MIPS : 325.0
Info -----
Info
Info TOTAL
```



```
概覽 終端機 6月14日 23:40 zh 6月14日 23:40  
chlinij88@chlinij88-VirtualBox: ~/ovp/Imperas.20210408/Dem...  
Begin Time = 0  
End Time = 0  
Measured time too small to obtain meaningful results  
Please increase number of runs  
  
Info  
Info -----  
Info CPU 'iss/cpu3' STATISTICS  
Info Type : riscv (RV32G)  
Info Nominal MIPS : 100  
Info Final program counter : 0x21230  
Info Simulated instructions: 4,173,082,179  
Info Simulated MIPS : 325.0  
Info -----  
Info  
Info CPU 'iss/cpu2' STATISTICS  
Info Type : riscv (RV32G)  
Info Nominal MIPS : 100  
Info Final program counter : 0x21230  
Info Simulated instructions: 4,173,082,179  
Info Simulated MIPS : 325.0  
Info -----  
Info  
Info CPU 'iss/cpu1' STATISTICS  
Info Type : riscv (RV32G)  
Info Nominal MIPS : 100  
Info Final program counter : 0x21230
```