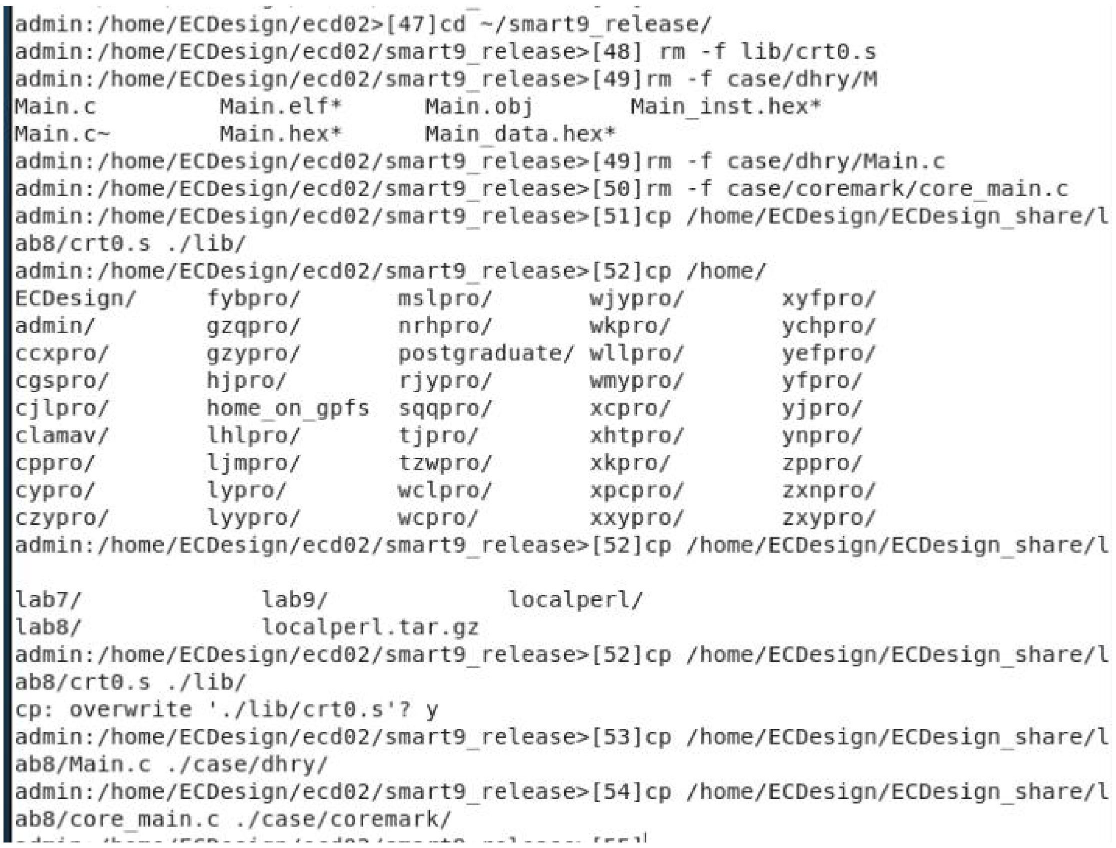
Pipeline 实验

1、实验目的

在 SMART 平台上通过调整分支预测配置情况观察测试程序程序的 CPI、分支预测准确率等，了解各种配置对 CPU 的性能的影响。

2、实验步骤（包括实验结果，数据记录，截图等）

（1）更改以及替换 SMART 平台内对应的文件，包括 crt0.s，dhrystone 程序和 coremark 程序。

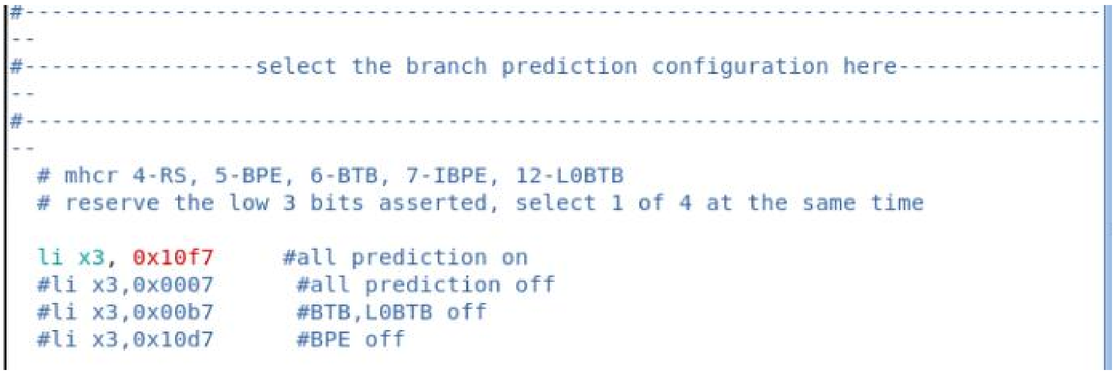


```
admin:/home/ECDesign/ecd02>[47]cd ~/smart9_release/
admin:/home/ECDesign/ecd02/smart9_release>[48] rm -f lib/crt0.s
admin:/home/ECDesign/ecd02/smart9_release>[49]rm -f case/dhry/M
Main.c          Main.elf*       Main.obj        Main_inst.hex*
Main.c~         Main.hex*       Main_data.hex*
admin:/home/ECDesign/ecd02/smart9_release>[49]rm -f case/dhry/Main.c
admin:/home/ECDesign/ecd02/smart9_release>[50]rm -f case/coremark/core_main.c
admin:/home/ECDesign/ecd02/smart9_release>[51]cp /home/ECDesign/ECDesign_share/l
ab8/crt0.s ./lib/
admin:/home/ECDesign/ecd02/smart9_release>[52]cp /home/
ECDesign/     fybpro/       mslpro/       wjypro/       xyfpro/
admin/        gzqpro/       nrhpro/       wkpro/        ychpro/
ccxpro/       gzypro/       postgraduate/ wllpro/       yefpro/
cgspro/       hjpro/        rjypro/       wmypro/       yfpro/
cjlpro/       home_on_gpfs  sqqpro/       xcpro/        yjpro/
clamav/       lhlpro/       tjpro/        xhtpro/       ynpro/
cppro/        ljmpro/       tzwpro/       xkpro/        zppro/
cypro/        lypro/        wclpro/       xpcpro/       zxnpro/
czypro/       lyypro/       wcpro/        xxypro/       zxypro/
admin:/home/ECDesign/ecd02/smart9_release>[52]cp /home/ECDesign/ECDesign_share/l

lab7/             lab9/             localperl/
lab8/             localperl.tar.gz
admin:/home/ECDesign/ecd02/smart9_release>[52]cp /home/ECDesign/ECDesign_share/l
ab8/crt0.s ./lib/
cp: overwrite './lib/crt0.s'? y
admin:/home/ECDesign/ecd02/smart9_release>[53]cp /home/ECDesign/ECDesign_share/l
ab8/Main.c ./case/dhry/
admin:/home/ECDesign/ecd02/smart9_release>[54]cp /home/ECDesign/ECDesign_share/l
ab8/core_main.c ./case/coremark/
```

crt0.s 是 SMART 平台中的 CPU 启动文件，dhrystone、coremark 则是两个 benchmark 文件。更改替换文件就是先把~/smart9_release 路径下的相应文件删除，再从已提供在的目录下将本实验所需的相关文件复制过来。

（2）在启动文件 crt0.s 中选择分支预测的配置，并进行 dhrystone 程序和 coremark 程序的仿真。

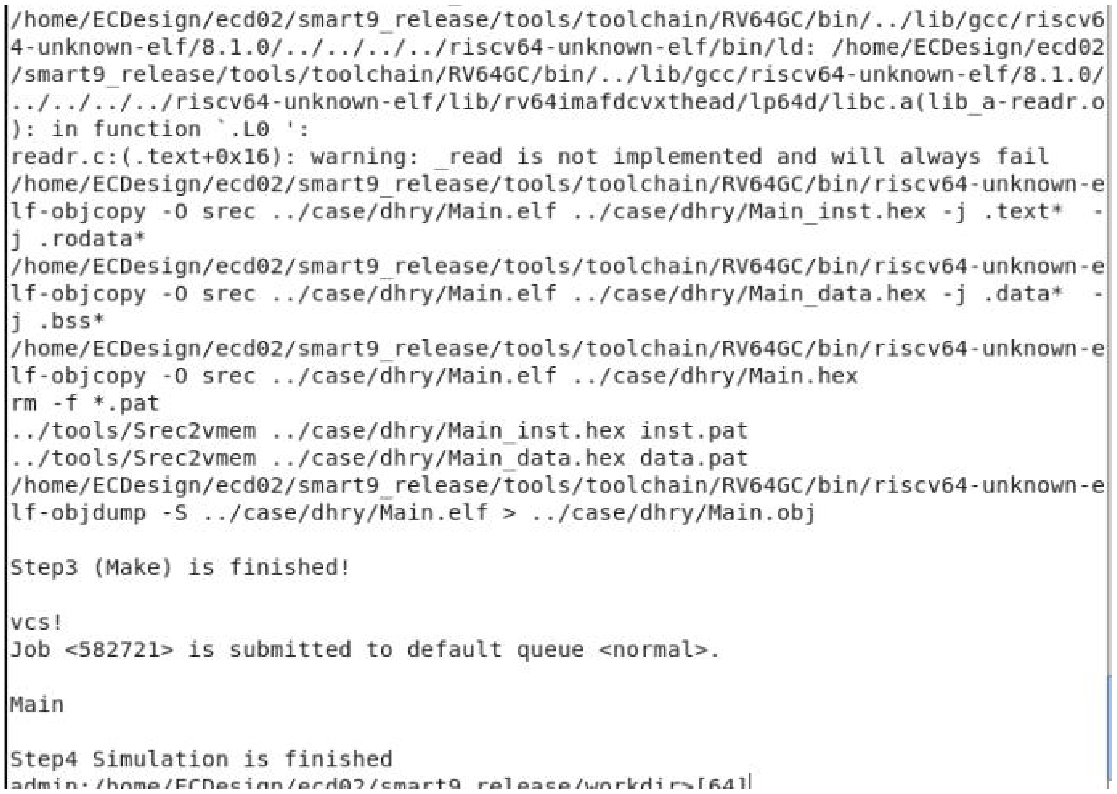
1）选择配置，这里先测试第一种情况，开启所有预测期，将另外三行全部注释掉。



```
#-------------------------------------------------------------------------
--
#-----------------select the branch prediction configuration here---------------
--
#-------------------------------------------------------------------------
--
  # mhcr 4-RS, 5-BPE, 6-BTB, 7-IBPE, 12-L0BTB
  # reserve the low 3 bits asserted, select 1 of 4 at the same time

  li x3, 0x10f7     #all prediction on
  #li x3,0x0007      #all prediction off
  #li x3,0x00b7      #BTB,L0BTB off
  #li x3,0x10d7      #BPE off
```

2）在 SMART 平台上对 dhrystone 和 coremark 进行仿真。

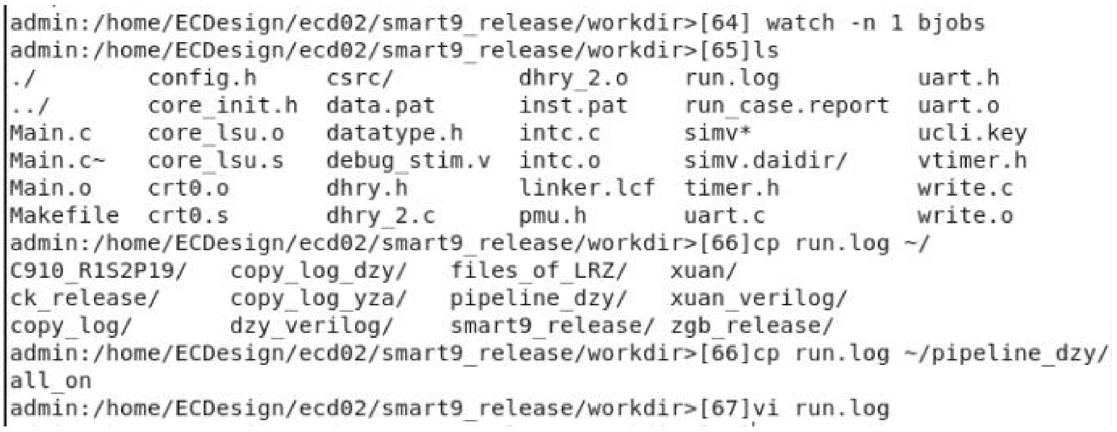
在 smart9_release 下执行 source setup.csh，再进入 workdir 目录执行 run ../case/dhry/Main.c，得到下图界面



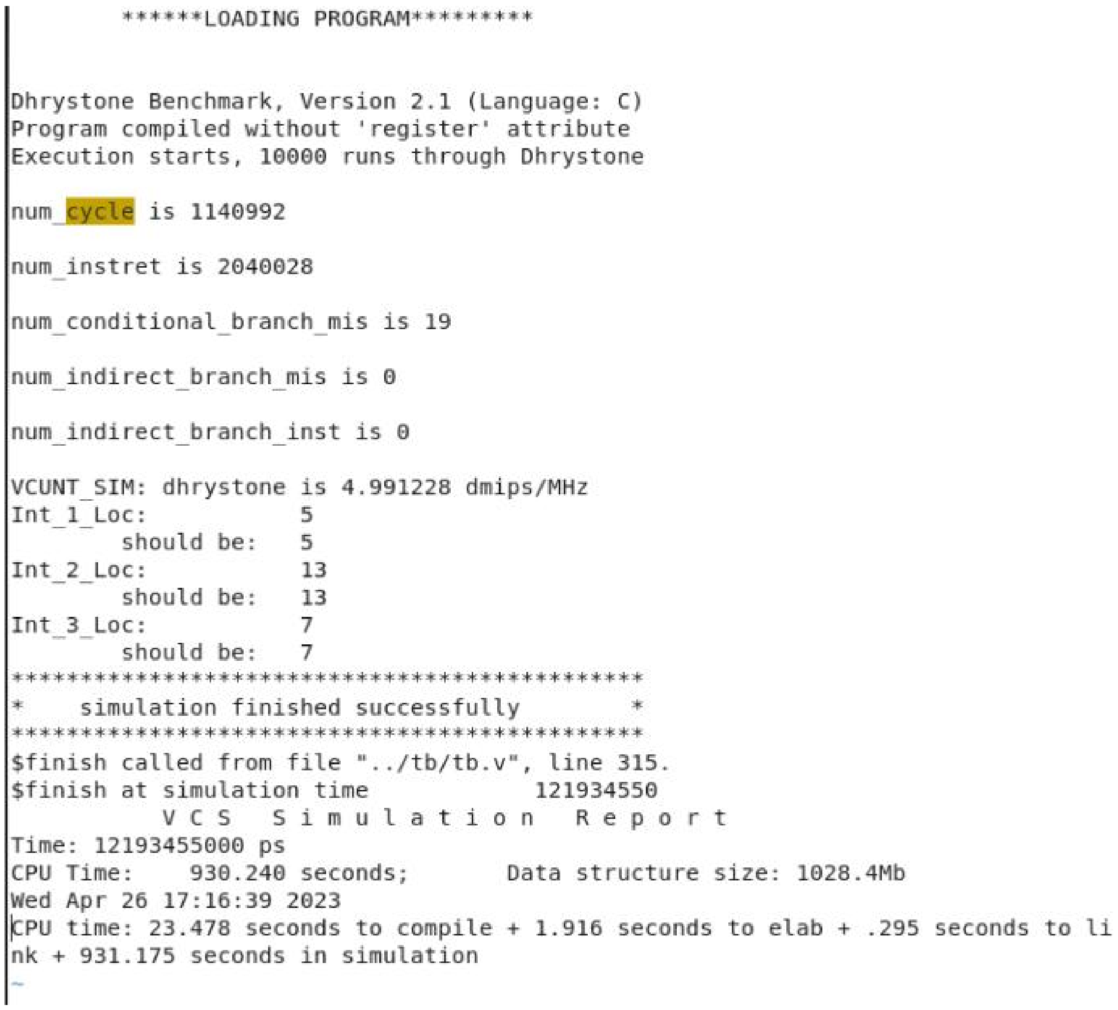
```
/home/ECDesign/ecd02/smart9_release/tools/toolchain/RV64GC/bin/../lib/gcc/riscv6
4-unknown-elf/8.1.0/../../../../riscv64-unknown-elf/bin/ld: /home/ECDesign/ecd02
/smart9_release/tools/toolchain/RV64GC/bin/../lib/gcc/riscv64-unknown-elf/8.1.0/
../../../../riscv64-unknown-elf/lib/rv64imafdcvxthead/lp64d/libc.a(lib_a-readr.o
): in function `.L0 ':
readr.c:(.text+0x16): warning: _read is not implemented and will always fail
/home/ECDesign/ecd02/smart9_release/tools/toolchain/RV64GC/bin/riscv64-unknown-e
lf-objcopy -O srec ../case/dhry/Main.elf ../case/dhry/Main_inst.hex -j .text*  -
j .rodata*
/home/ECDesign/ecd02/smart9_release/tools/toolchain/RV64GC/bin/riscv64-unknown-e
lf-objcopy -O srec ../case/dhry/Main.elf ../case/dhry/Main_data.hex -j .data*  -
j .bss*
/home/ECDesign/ecd02/smart9_release/tools/toolchain/RV64GC/bin/riscv64-unknown-e
lf-objcopy -O srec ../case/dhry/Main.elf ../case/dhry/Main.hex
rm -f *.pat
../tools/Srec2vmem ../case/dhry/Main_inst.hex inst.pat
../tools/Srec2vmem ../case/dhry/Main_data.hex data.pat
/home/ECDesign/ecd02/smart9_release/tools/toolchain/RV64GC/bin/riscv64-unknown-e
lf-objdump -S ../case/dhry/Main.elf > ../case/dhry/Main.obj

Step3 (Make) is finished!

vcs!
Job <582721> is submitted to default queue <normal>.

Main

Step4 Simulation is finished
admin:/home/ECDesign/ecd02/smart9_release/workdir>[64]
```

3）使用 watch -n 1 bjobs 命令动态查看进程是否结束。

4）将 run.log 文件复制到~/piprline_dzy 目录下并使用 vi 查看仿真结果。



```
admin:/home/ECDesign/ecd02/smart9_release/workdir>[64] watch -n 1 bjobs
admin:/home/ECDesign/ecd02/smart9_release/workdir>[65]ls
./        config.h     csrc/          dhry_2.o     run.log          uart.h
../       core_init.h  data.pat       inst.pat     run_case.report  uart.o
Main.c    core_lsu.o   datatype.h     intc.c       simv*            ucli.key
Main.c~   core_lsu.s   debug_stim.v   intc.o       simv.daidir/     vtimer.h
Main.o    crt0.o       dhry.h         linker.lcf   timer.h          write.c
Makefile  crt0.s       dhry_2.c       pmu.h        uart.c           write.o
admin:/home/ECDesign/ecd02/smart9_release/workdir>[66]cp run.log ~/
C910_R1S2P19/   copy_log_dzy/   files_of_LRZ/   xuan/
ck_release/     copy_log_yza/   pipeline_dzy/   xuan_verilog/
copy_log/       dzy_verilog/    smart9_release/ zgb_release/
admin:/home/ECDesign/ecd02/smart9_release/workdir>[66]cp run.log ~/pipeline_dzy/
all_on
admin:/home/ECDesign/ecd02/smart9_release/workdir>[67]vi run.log
```



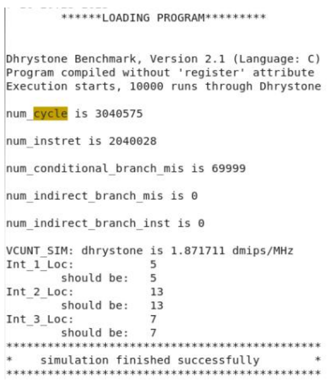
```
        ******LOADING PROGRAM*********


Dhrystone Benchmark, Version 2.1 (Language: C)
Program compiled without 'register' attribute
Execution starts, 10000 runs through Dhrystone

num_cycle is 1140992

num_instret is 2040028

num_conditional_branch_mis is 19

num_indirect_branch_mis is 0

num_indirect_branch_inst is 0

VCUNT_SIM: dhrystone is 4.991228 dmips/MHz
Int_1_Loc:            5
        should be:   5
Int_2_Loc:            13
        should be:   13
Int_3_Loc:            7
        should be:   7
**********************************************
*    simulation finished successfully        *
**********************************************
$finish called from file "../tb/tb.v", line 315.
$finish at simulation time            121934550
           V C S   S i m u l a t i o n   R e p o r t
Time: 12193455000 ps
CPU Time:    930.240 seconds;       Data structure size: 1028.4Mb
Wed Apr 26 17:16:39 2023
CPU time: 23.478 seconds to compile + 1.916 seconds to elab + .295 seconds to li
nk + 931.175 seconds in simulation
~
```

（3)完成分支预测配置下的 dhrystone 程序和 coremark 程序的仿真后，观察仿真结果，记录数据，汇总成上述的两张表格。

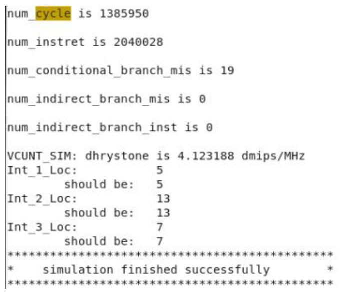
表格 1 dhrystone **测试**

	all prediction on	all prediction off	BTB, LOBTB off	BPE off
cycle	1140992	3040575	1385950	2278724
insts	2040028	2040028	2040028	2040028
CPI	0.559302127	1.490457484	0.67937793	1.117006237
conditional branch miss	19	69999	19	69999
indirect branch miss	0	0	0	0
indirect branch inst	0	0	0	0
DMIPS(dmips/MHz)	4.991228	1.871711	4.123188	2.506608

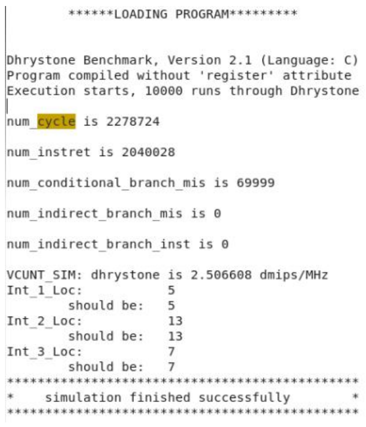
run.log 截图如下



```
      ******LOADING PROGRAM*********


Dhrystone Benchmark, Version 2.1 (Language: C)
Program compiled without 'register' attribute
Execution starts, 10000 runs through Dhrystone

num_cycle is 3040575

num_instret is 2040028

num_conditional_branch_mis is 69999

num_indirect_branch_mis is 0

num_indirect_branch_inst is 0

VCUNT_SIM: dhrystone is 1.871711 dmips/MHz
Int_1_Loc:             5
        should be:     5
Int_2_Loc:             13
        should be:     13
Int_3_Loc:             7
        should be:     7
**********************************************
*    simulation finished successfully        *
**********************************************
```



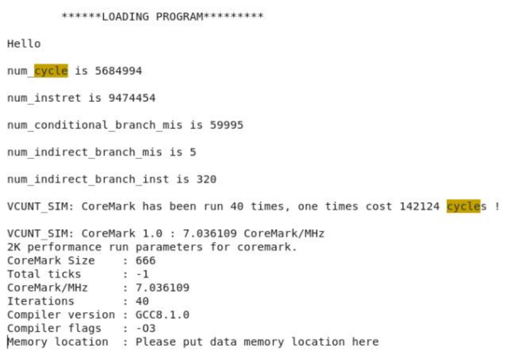
```
num_cycle is 1385950

num_instret is 2040028

num_conditional_branch_mis is 19

num_indirect_branch_mis is 0

num_indirect_branch_inst is 0

VCUNT_SIM: dhrystone is 4.123188 dmips/MHz
Int_1_Loc:             5
        should be:     5
Int_2_Loc:             13
        should be:     13
Int_3_Loc:             7
        should be:     7
**********************************************
*    simulation finished successfully        *
**********************************************
```



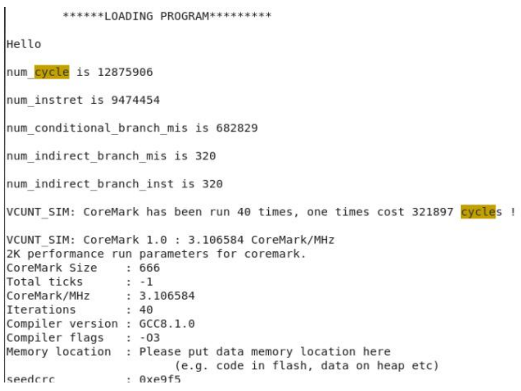
```
        ******LOADING PROGRAM*********


Dhrystone Benchmark, Version 2.1 (Language: C)
Program compiled without 'register' attribute
Execution starts, 10000 runs through Dhrystone

num_cycle is 2278724

num_instret is 2040028

num_conditional_branch_mis is 69999

num_indirect_branch_mis is 0

num_indirect_branch_inst is 0

VCUNT_SIM: dhrystone is 2.506608 dmips/MHz
Int_1_Loc:             5
        should be:     5
Int_2_Loc:             13
        should be:     13
Int_3_Loc:             7
        should be:     7
**********************************************
*    simulation finished successfully        *
**********************************************
```

表格 2 coremark 测试

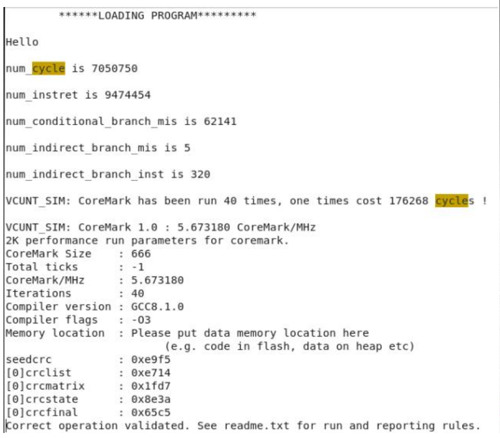
	all prediction on	all prediction off	BTB, LOBTB off	BPE off
cycle	5684994	12875906	7050750	12568284
insts	9474454	9474454	9474454	9474454
CPI	0.600033944	1.359012984	0.744185364	1.326544411
conditional branch miss	59995	682829	62141	682829
indirect branch miss	5	320	5	5
indirect branch inst	320	320	320	320
CoreMark point (CoreMark/MHz)	7.036109	3.106584	5.67318	3.182615



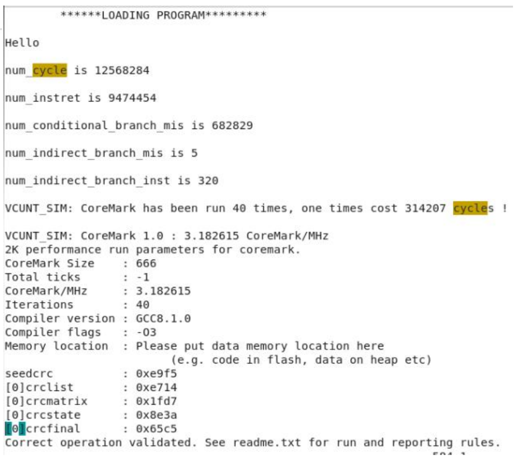
```
        ******LOADING PROGRAM*********

Hello

num_cycle is 5684994

num_instret is 9474454

num_conditional_branch_mis is 59995

num_indirect_branch_mis is 5

num_indirect_branch_inst is 320

VCUNT_SIM: CoreMark has been run 40 times, one times cost 142124 cycles !

VCUNT_SIM: CoreMark 1.0 : 7.036109 CoreMark/MHz
2K performance run parameters for coremark.
CoreMark Size    : 666
Total ticks      : -1
CoreMark/MHz     : 7.036109
Iterations       : 40
Compiler version : GCC8.1.0
Compiler flags   : -O3
Memory location  : Please put data memory location here
```



```
        ******LOADING PROGRAM*********

Hello

num_cycle is 12875906

num_instret is 9474454

num_conditional_branch_mis is 682829

num_indirect_branch_mis is 320

num_indirect_branch_inst is 320

VCUNT_SIM: CoreMark has been run 40 times, one times cost 321897 cycles !

VCUNT_SIM: CoreMark 1.0 : 3.106584 CoreMark/MHz
2K performance run parameters for coremark.
CoreMark Size    : 666
Total ticks      : -1
CoreMark/MHz     : 3.106584
Iterations       : 40
Compiler version : GCC8.1.0
Compiler flags   : -O3
Memory location  : Please put data memory location here
                        (e.g. code in flash, data on heap etc)
seedcrc          : 0xe9f5
```



```
      ******LOADING PROGRAM*********

Hello

num_cycle is 7050750

num_instret is 9474454

num_conditional_branch_mis is 62141

num_indirect_branch_mis is 5

num_indirect_branch_inst is 320

VCUNT_SIM: CoreMark has been run 40 times, one times cost 176268 cycles !

VCUNT_SIM: CoreMark 1.0 : 5.673180 CoreMark/MHz
2K performance run parameters for coremark.
CoreMark Size    : 666
Total ticks      : -1
CoreMark/MHz     : 5.673180
Iterations       : 40
Compiler version : GCC8.1.0
Compiler flags   : -O3
Memory location  : Please put data memory location here
                        (e.g. code in flash, data on heap etc)
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x65c5
Correct operation validated. See readme.txt for run and reporting rules.
```



```
      ******LOADING PROGRAM*********

Hello

num_cycle is 12568284

num_instret is 9474454

num_conditional_branch_mis is 682829

num_indirect_branch_mis is 5

num_indirect_branch_inst is 320

VCUNT_SIM: CoreMark has been run 40 times, one times cost 314207 cycles !

VCUNT_SIM: CoreMark 1.0 : 3.182615 CoreMark/MHz
2K performance run parameters for coremark.
CoreMark Size    : 666
Total ticks      : -1
CoreMark/MHz     : 3.182615
Iterations       : 40
Compiler version : GCC8.1.0
Compiler flags   : -O3
Memory location  : Please put data memory location here
                        (e.g. code in flash, data on heap etc)
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x65c5
Correct operation validated. See readme.txt for run and reporting rules.
```

3、实验分析和总结

1)在四种情况下，跑完 dhrystone 和 coremark 两个程序周期数 cycle 参数有如下的关系：all prediction off > BPE off > BTB,L0BTB off > all prediction on ，CPI 呈相同的关系，DMIPS 呈相反的关系。这是由于，完整的分支预测跳转结构能够最大程度的提升程序运行的效率。而 BTB 被关闭之后，由于间接跳转分支和分支历史表的功能仍然正常运作，因此对程序运行的影响较小。当 BPE 被关闭之后，不管什么方式的分支预测都无法提前跳转，故这种操作对程序运行影响最大，数据接近于 all prediction on 的情况。

2)DMIPS：Dhrystone Million Instructions executed Per Second，每秒百万条指令，主要用于测试整数运算能力。分支跳转预测越完善，CPI 值越低，则运算能力越高，DMIPS 值越高。Coremark point：coremark 跑分测试，这也是一种处理器性能测试基准程序，分支跳转预测越完善，CPI 值越低，则运算能力越高，Coremark point 值越高。

3)按照理论来讲，indirect branch miss 应该一直为零。至于 coremark 中为什么不为零，有可能是用到了使用序号为 1 或 5 的源寄存器的 call 或 return 指令，导致间接分支跳转仍然失败，但 coremark all prediction off 达到 320 确实有一些可疑。

4)总的来说，BTB 和 L0BTB 不是分支跳转预测实现的唯一方式，使能关闭后仍然有 BHT，IBTB 作为补救，因此对 cpu 性能影响不大。而 BPE 是跳转的必经之路，使能一旦关闭代表着无法通过预测的方式跳转，必须严格遵循执行顺序，对 cpu 性能影响较大。

4、实验收获、存在问题、改进措施或建议等

实验收获：在本次试验中，通过在 SMART 平台上通过调整分支预测配置情况观察测试程序程序的 CPI、分支预测准确率等，我了解各种配置对 CPU 的性能的影响，并且仔细阅读了知乎专栏上关于分支预测的相关文献，以及 C910 用户手册，深刻学习了分支历史表(BHT)、快速跳转目标缓冲器(L0 BTB)、分支跳转目标缓冲器(BTB)、间接跳转目标缓冲器(IND BTB)和短循环加速器(Loop Buffer)等分支预测的实现方式。