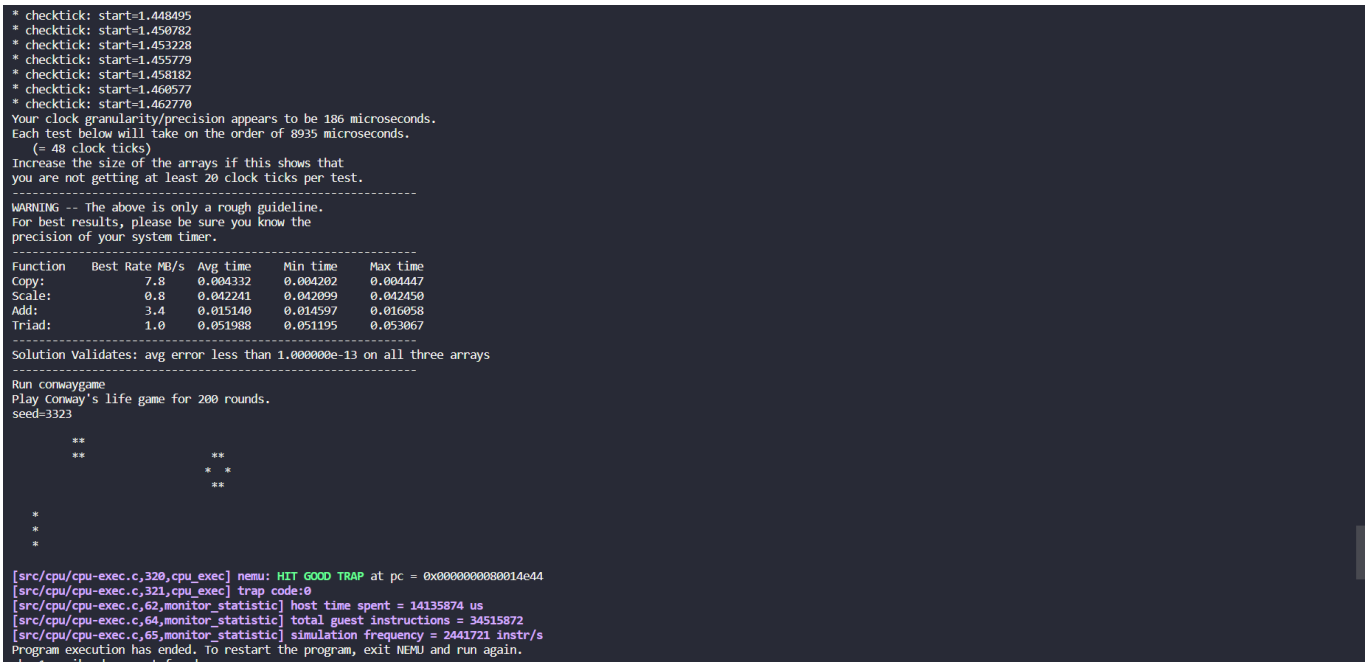
lab3实验报告

20302010059-孙姝然

测试通过截图



```
* checktick: start=1.448495
* checktick: start=1.450782
* checktick: start=1.453228
* checktick: start=1.455779
* checktick: start=1.458182
* checktick: start=1.460577
* checktick: start=1.462770
Your clock granularity/precision appears to be 186 microseconds.
Each test below will take on the order of 8935 microseconds.
   (= 48 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:             7.8     0.004332     0.004202     0.004447
Scale:            0.8     0.042241     0.042099     0.042450
Add:              3.4     0.015140     0.014597     0.016058
Triad:            1.0     0.051988     0.051195     0.053067
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
Run conwaygame
Play Conway's life game for 200 rounds.
seed=3323

        **
        **                  **
                          *  *
                           **


   *
   *
   *

[src/cpu/cpu-exec.c,320,cpu_exec] nemu: HIT GOOD TRAP at pc = 0x0000000080014e44
[src/cpu/cpu-exec.c,321,cpu_exec] trap code:0
[src/cpu/cpu-exec.c,62,monitor_statistic] host time spent = 14135874 us
[src/cpu/cpu-exec.c,64,monitor_statistic] total guest instructions = 34515872
[src/cpu/cpu-exec.c,65,monitor_statistic] simulation frequency = 2441721 instr/s
Program execution has ended. To restart the program, exit NEMU and run again.
```

```
==================================================
CoreMark Iterations/Sec 20
Run dhrystone
Dhrystone Benchmark, Version C, Version 2.2
Trying 10000 runs through Dhrystone.
Finished in 669 ms
==================================================
Dhrystone PASS          26 Marks
                    vs. 100000 Marks (i7-7700K @ 4.20GHz)
Run stream
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 2048 (elements), Offset = 0 (elements)
Memory per array = 0.0 MiB (= 0.0 GiB).
Total memory required = 0.0 MiB (= 0.0 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
* checktick: start=1.591409
* checktick: start=1.622137
* checktick: start=1.652835
* checktick: start=1.683508
* checktick: start=1.714181
* checktick: start=1.744854
* checktick: start=1.775552
* checktick: start=1.806225
* checktick: start=1.836898
* checktick: start=1.867596
* checktick: start=1.898269
* checktick: start=1.928942
* checktick: start=1.959640
* checktick: start=1.990313
* checktick: start=2.020986
* checktick: start=2.051664
* checktick: start=2.082347
* checktick: start=2.113039
* checktick: start=2.143722
* checktick: start=2.174420
Your clock granularity/precision appears to be 90 microseconds.
Each test below will take on the order of 6479 microseconds.
   (= 71 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:            19.2     0.001705     0.001705     0.001705
Scale:            1.0     0.033212     0.033183     0.033269
Add:              5.2     0.009863     0.009538     0.010512
Triad:            1.2     0.040404     0.040051     0.041110
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
```

```
----------------------------------------------------------------
Run conwaygame
Play Conway's life game for 200 rounds.
seed=4536
```

```
[OK] void (7ms)
[--] reset (skipped)
[OK] fake load (82ms)
[OK] fake store (83ms)
[OK] naive (7ms)
[--] akarin~ (skipped)
[OK] strobe (7ms)
[OK] ad hoc (7ms)
[OK] pipelined (7ms)
[OK] memory cell (7ms)
[OK] memory cell array (7ms)
[OK] cmp: word (9ms)
[OK] cmp: halfword (13ms)
[OK] cmp: byte (20ms)
[OK] cmp: random (196ms)
[OK] memset (102ms)
[OK] memcpy (117ms)
[OK] load/store repeat (76ms)
[OK] backward memset (261ms)
[OK] backward load/store (344ms)
[OK] random step (3030ms)
[OK] random load/store (3592ms)
[OK] random block load/store (1566ms)
"std::sort": bingo!
[OK] std::sort (1622ms)
"std::stable_sort": bingo!
[OK] std::stable_sort (5221ms)
"heap sort": bingo!
[OK] heap sort (10750ms)
"binary search tree": bingo!
[OK] binary search tree (6183ms)
(info) 27 tests passed.
make[1]: Leaving directory '/mnt/c/Users/31324/Desktop/Arch-2022Spring-FDU'
ssr@LAPTOP-S2OBLK4Q:/mnt/c/Users/31324/Desktop/Arch-2022Spring-FDU$
```

支持随机延迟，流水线改动

i_wait，e_wait，d_wait分别表示取指、多周期运算、取数据内存时需要等待的信号

- **e_wait与d_wait与流水线阻塞**：多周期乘除法器运算期间，stall F，D，E寄存器（阻塞E阶段之前的流水线），flush M寄存器（防止E阶段指令重复执行）e_wait期间若前一条指令正在d_wait，将flush M 改为stall M。（握手期间保持dreq不变）。d_wait先结束，变回flush M，指令依然不会重复执行。e_wait先结束，变为 stall M，E，D，F，即变回单独d_wait时的情况。
- **e_wait与数据冲突**：多周期运算若使用到前一条或两条指令写入的寄存器时，由于运算期间需阻塞的原因，无法复用转发（尝试使用寄存器存储运算数时失败了）。使用阻塞。对于一条多周期运算指令，在d阶段检测到需使用前两条指令写入寄存器时 stall F，stall D，flush E。（d_wait与e_wait期间自然阻塞前半部分流水线，所以不用考虑）
- **代码实现**：一个always_comb以i_wait，e_wait，d_awit，multi_stall，branch_stall为条件控制所有寄存器和stall。另一个always_comb中以de阶段、dm阶段读取写入寄存器比较为条件，并排除e_wait，控制forward信号