ECE411 MP3 Final CP Report

Group: ZerotoOne

Progress Report:
By the final checkpoint, we successfully rooted out the bugs in our system to correctly run the final test code, as well as implemented advanced features that improves the performance of the system. We implemented the following advanced features:

1. A tournament branch predictor with both local history branch predictor and 2-level global history branch predictor. A four state saturation counter keeps track of the prediction accuracy of the predictors and output the result of the better one.
2. A branch target buffer storing the target to jump to for branching and jumping instructions. It is stored as a fully associative array and has a round robin replacement policy. The branch instructions and jump instructions share the same buffer.
3. A dynamic hardware data prefetch-unit that has a reference prediction table(RPT) that stores memory access info for needed PC. For each PC mapped memory access info, our(RPT) stores previous-access-address, stride, and state. These information are used to possibly generate an outstanding request address whenever a new PC is updated.
4. A 4-line fully associative victim cache with true-LRU replacement policy located between the L2 cache and the eviction write buffer towards the direction of physical memory. During a L2 miss and a victim cache miss, data is fetched from physical memory to L2 cache, while the victim cache catches the data evicted from L2. This mechanism enables the situation where during a L2 miss, the data hits in the victim cache so that the data can be brought directly back into L2 without accessing physical memory and at the same time catching the data evicted from L2, thus improving the performance.

Contribution:
Haiyang Zhang: implementation of tournament predictor, BTB, and debugging
Yuan Ma: implementation of the mem data prefetcher and debugging
Haichuan Xu: implementation of the victim cache

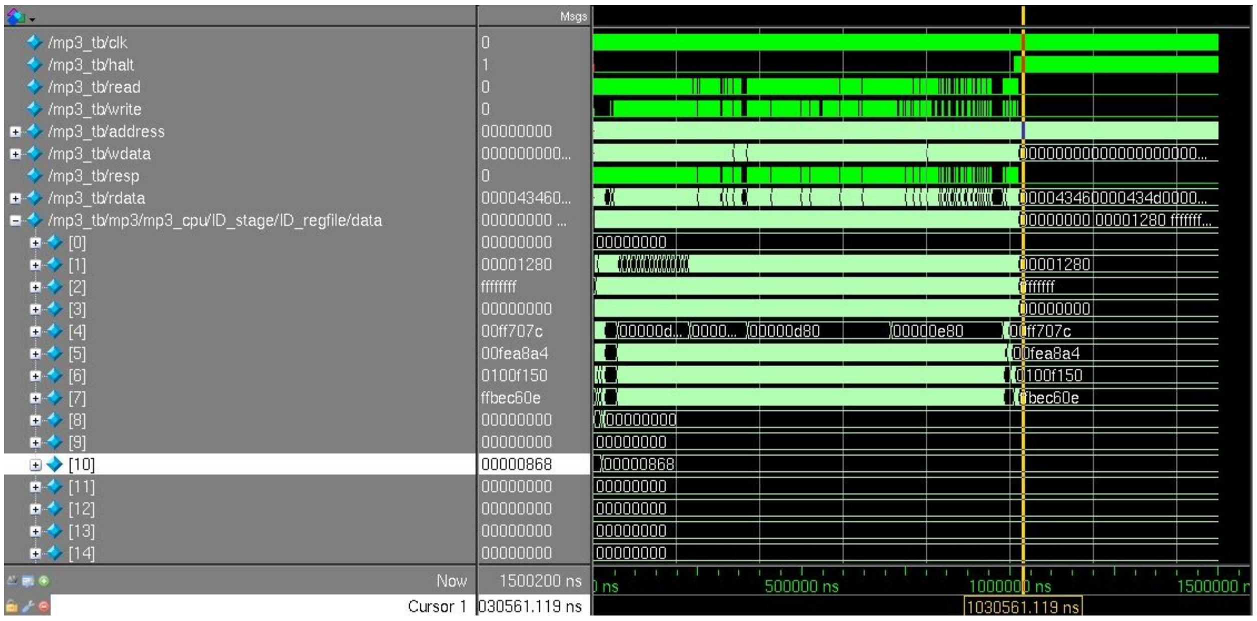
Roadmap:
By the end of the competition demo, we expect to :

1. Continue to improve the performance by pipelining the L2 cache to improve the max frequency.
2. Continue to explore the potential of the advanced features that we implemented, including finding the best size for the BTB or making it set associative, adding support for jalr in the branch predictor

Testing and Performance:

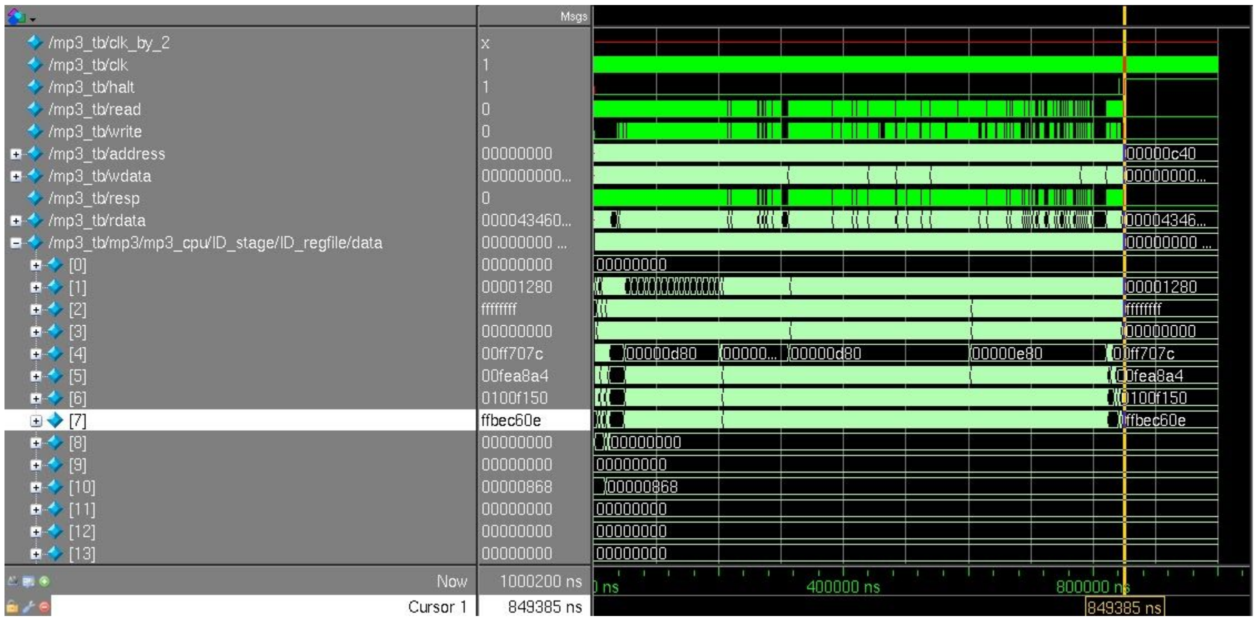
1. BTB and tournament buffer: The best test program is one with a lot of branching instructions , so we mainly use the given cp3-final test code for the testing. We believe it is sufficient because it has sufficient amount of branches to fill the BTB and also covers all kinds of branching and

jumping instructions. By this code we were able to discover a problem associated with jalr instructions where the jump target is not constant.
The way we test its functionality is mainly by comparing how much time it saves to execute a long program full of branches. The picture below shows how much time it takes for the program to finish executing mp3-final with only static prediction.



	Msgs
/mp3_tb/clk	0
/mp3_tb/halt	1
/mp3_tb/read	0
/mp3_tb/write	0
/mp3_tb/address	00000000
/mp3_tb/wdata	000000000...
/mp3_tb/resp	0
/mp3_tb/rdata	000043460...
/mp3_tb/mp3/mp3_cpu/ID_stage/ID_regfile/data	00000000 ...
[0]	00000000
[1]	00001280
[2]	ffffffff
[3]	00000000
[4]	00ff707c
[5]	00fea8a4
[6]	0100f150
[7]	ffbec60e
[8]	00000000
[9]	00000000
[10]	00000868
[11]	00000000
[12]	00000000
[13]	00000000
[14]	00000000
Now	1500200 ns
Cursor 1	030561.119 ns

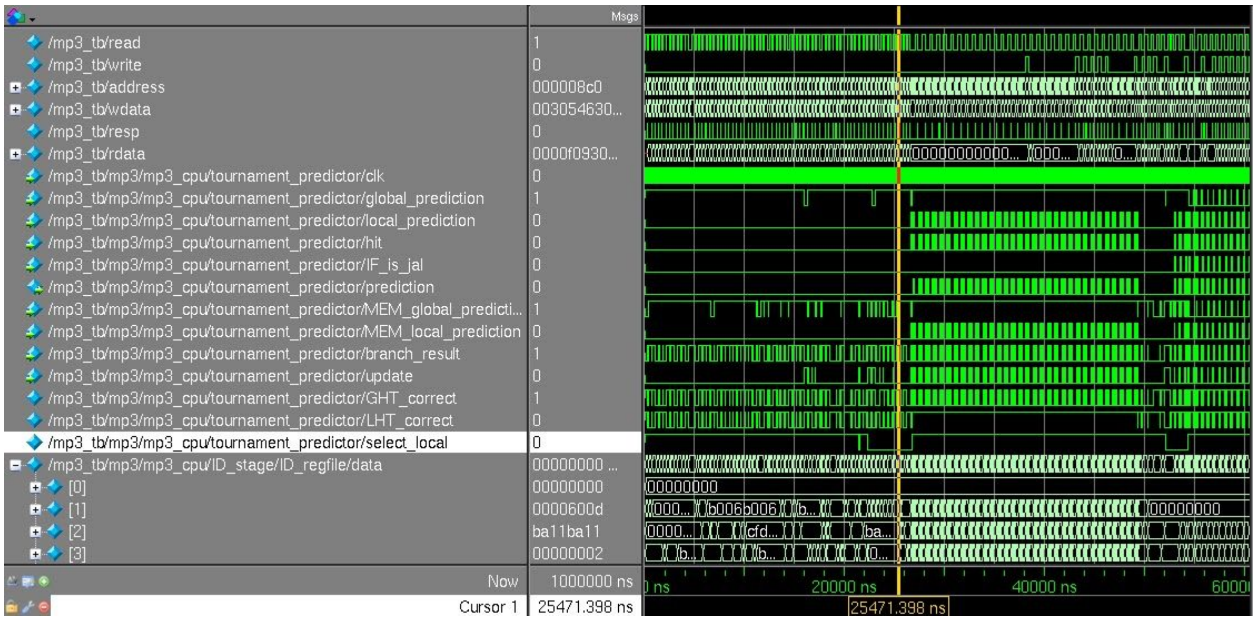
By the end of the program, there are around 0x31e8 branches/jumps and 0x249b misses, which accounts for 73% of the branches. And this picture shows how the performance improves with the tournament predictor



```
/mp3_tb/clk_by_2                              x
/mp3_tb/clk                                   1
/mp3_tb/halt                                  1
/mp3_tb/read                                  0
/mp3_tb/write                                 0
/mp3_tb/address                               00000000
/mp3_tb/wdata                                 000000000...
/mp3_tb/resp                                  0
/mp3_tb/rdata                                 000043460...
/mp3_tb/mp3/mp3_cpu/ID_stage/ID_regfile/data  00000000 ...
  [0]                                         00000000
  [1]                                         00001280
  [2]                                         ffffffff
  [3]                                         00000000
  [4]                                         00ff707c
  [5]                                         00fea8a4
  [6]                                         0100f150
  [7]                                         ffbec60e
  [8]                                         00000000
  [9]                                         00000000
  [10]                                        00000868
  [11]                                        00000000
  [12]                                        00000000
  [13]                                        00000000
Now                                           1000200 ns
Cursor 1                                      849385 ns
```

The execution time shortened from roughly 1030,000 ns to 850,000 ns, which is almost 20 % performance increase. According to the reading of performance registers, by the end of the program, there are 0x31e8 branches and 0xcdf misses, which accounts for 26% of the branches.

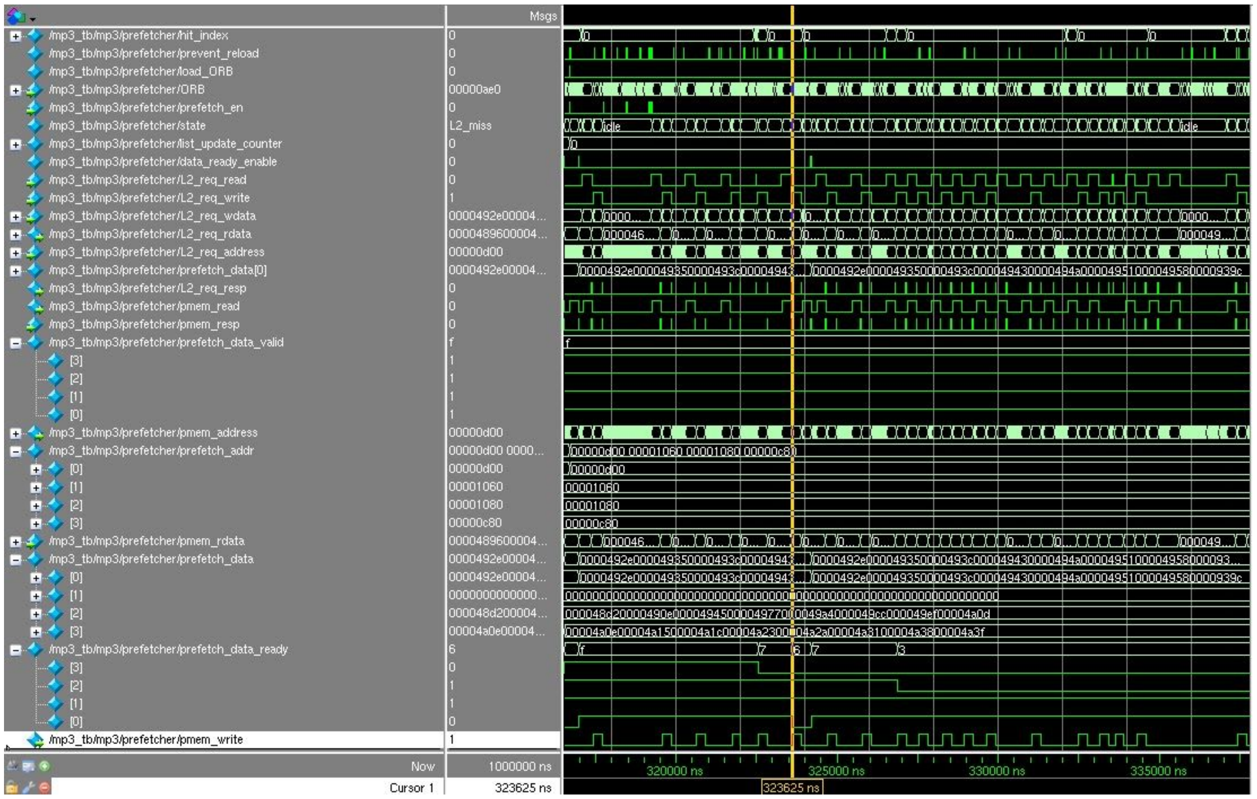
The following picture is a closer look at the tournament predictor. It is clear that the predictor is making a choice from the local and global predictor. The local one is chosen most of the time but the global one is also used sometimes.



```
                                                               Msgs
/mp3_tb/read                                                   1
/mp3_tb/write                                                  0
/mp3_tb/address                                                000008c0
/mp3_tb/wdata                                                  003054630...
/mp3_tb/resp                                                   0
/mp3_tb/rdata                                                  0000f0930...
/mp3_tb/mp3/mp3_cpu/tournament_predictor/clk                   0
/mp3_tb/mp3/mp3_cpu/tournament_predictor/global_prediction     1
/mp3_tb/mp3/mp3_cpu/tournament_predictor/local_prediction      0
/mp3_tb/mp3/mp3_cpu/tournament_predictor/hit                   0
/mp3_tb/mp3/mp3_cpu/tournament_predictor/IF_is_jal             0
/mp3_tb/mp3/mp3_cpu/tournament_predictor/prediction            0
/mp3_tb/mp3/mp3_cpu/tournament_predictor/MEM_global_predicti...1
/mp3_tb/mp3/mp3_cpu/tournament_predictor/MEM_local_prediction  0
/mp3_tb/mp3/mp3_cpu/tournament_predictor/branch_result         1
/mp3_tb/mp3/mp3_cpu/tournament_predictor/update                0
/mp3_tb/mp3/mp3_cpu/tournament_predictor/GHT_correct           1
/mp3_tb/mp3/mp3_cpu/tournament_predictor/LHT_correct           0
/mp3_tb/mp3/mp3_cpu/tournament_predictor/select_local          0
/mp3_tb/mp3/mp3_cpu/ID_stage/ID_regfile/data                   00000000 ...
  [0]                                                          00000000
  [1]                                                          0000600d
  [2]                                                          ba11ba11
  [3]                                                          00000002
                                                    Now        1000000 ns
                                               Cursor 1        25471.398 ns
```

2. Dynamic Stride Hardware Prefetcher: we use the mp3-final.s as our test-code prototype for its significant number of L2 miss and replacement. The corner test includes the scenario that multiple PC access the same address frequently. This tests whether the prefetcher is able to send only one predicted address to the PMEM. Another corner test includes read the memory that is wroten back immediately. This tests whether the prefetcher is able to detect the dirty data that has been accessed.

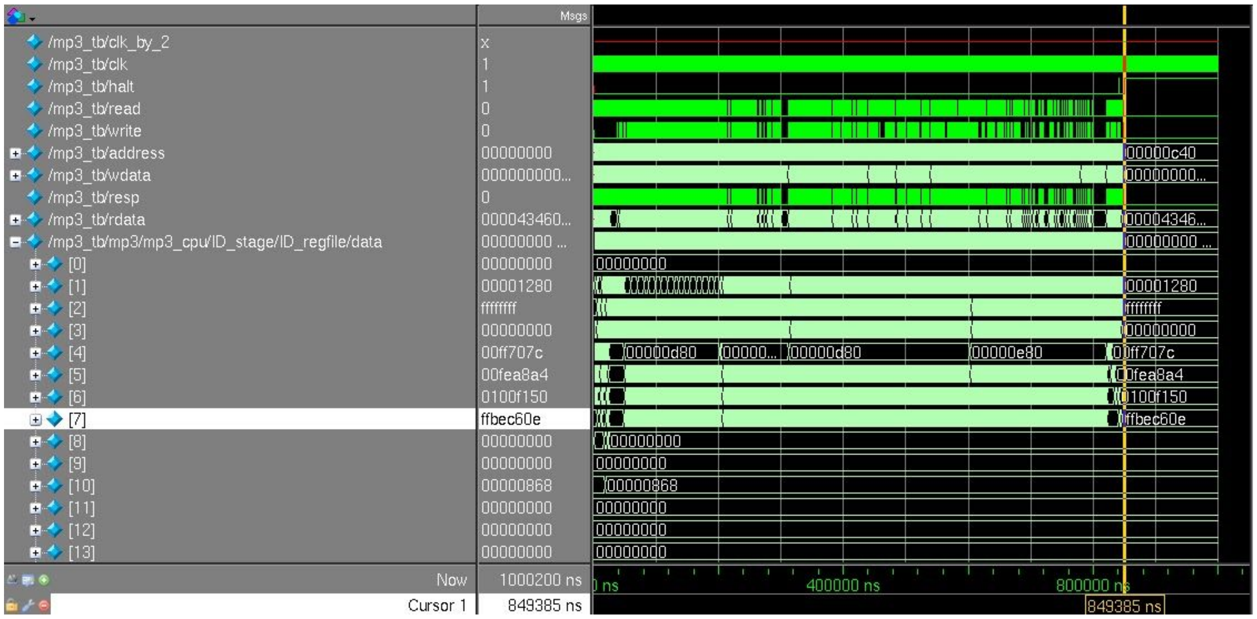
Below is a corner situation that show an immediate pmem write instruction that makes the prefetcher preloaded data invalid.



```
                                                      Msgs
/mp3_tb/mp3/prefetcher/hit_index                      0
/mp3_tb/mp3/prefetcher/prevent_reload                 0
/mp3_tb/mp3/prefetcher/load_ORB                       0
/mp3_tb/mp3/prefetcher/ORB                            00000ae0
/mp3_tb/mp3/prefetcher/prefetch_en                    0
/mp3_tb/mp3/prefetcher/state                          L2_miss
/mp3_tb/mp3/prefetcher/list_update_counter            0
/mp3_tb/mp3/prefetcher/data_ready_enable              0
/mp3_tb/mp3/prefetcher/L2_req_read                    0
/mp3_tb/mp3/prefetcher/L2_req_write                   1
/mp3_tb/mp3/prefetcher/L2_req_wdata                   0000492e00004...
/mp3_tb/mp3/prefetcher/L2_req_rdata                   0000489600004...
/mp3_tb/mp3/prefetcher/L2_req_address                 00000d00
/mp3_tb/mp3/prefetcher/prefetch_data[0]               0000492e00004...
/mp3_tb/mp3/prefetcher/L2_req_resp                    0
/mp3_tb/mp3/prefetcher/pmem_read                      0
/mp3_tb/mp3/prefetcher/pmem_resp                      0
/mp3_tb/mp3/prefetcher/prefetch_data_valid            f
    [3]                                               1
    [2]                                               1
    [1]                                               1
    [0]                                               1
/mp3_tb/mp3/prefetcher/pmem_address                   00000d00
/mp3_tb/mp3/prefetcher/prefetch_addr                  00000d00 0000...
    [0]                                               00000d00
    [1]                                               00001060
    [2]                                               00001080
    [3]                                               00000c80
/mp3_tb/mp3/prefetcher/pmem_rdata                     0000489600004...
/mp3_tb/mp3/prefetcher/prefetch_data                  0000492e00004...
    [0]                                               0000492e00004...
    [1]                                               0000000000000...
    [2]                                               000048d200004...
    [3]                                               00004a0e00004...
/mp3_tb/mp3/prefetcher/prefetch_data_ready            6
    [3]                                               0
    [2]                                               1
    [1]                                               1
    [0]                                               0
/mp3_tb/mp3/prefetcher/pmem_write                     1

Now        1000000 ns
Cursor 1   323625 ns
```

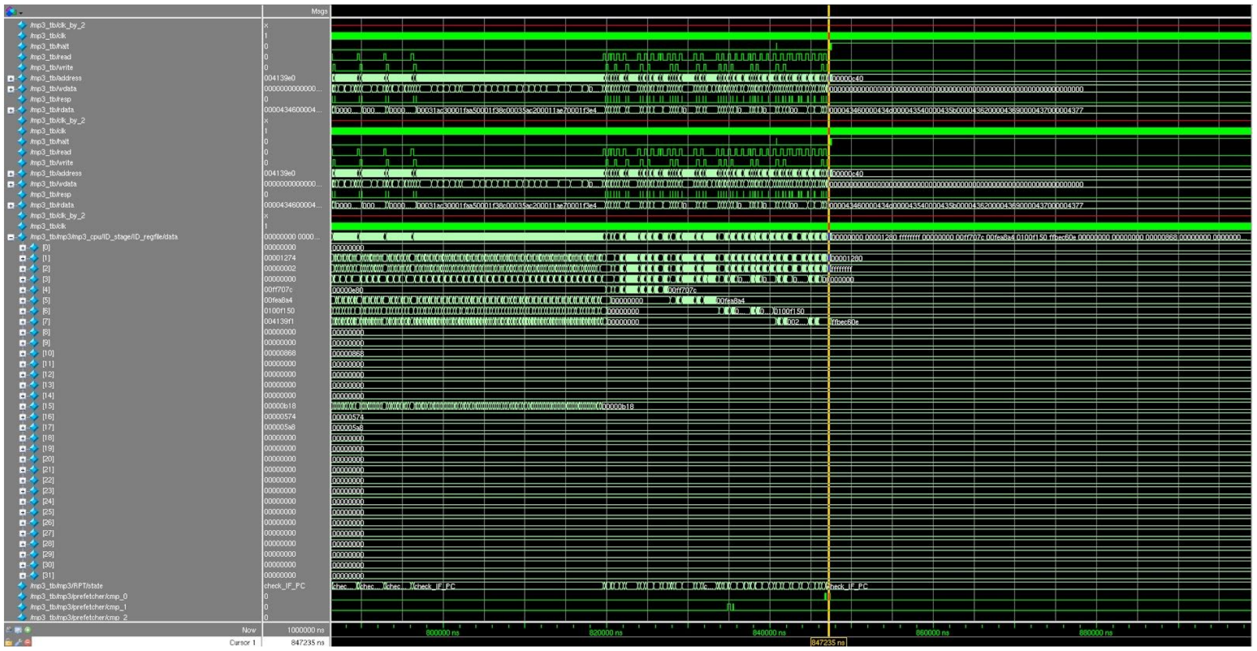
We compare the performance before and after adding the prefetch-unit. The performance before adding the prefetch-unit is around 849,300.



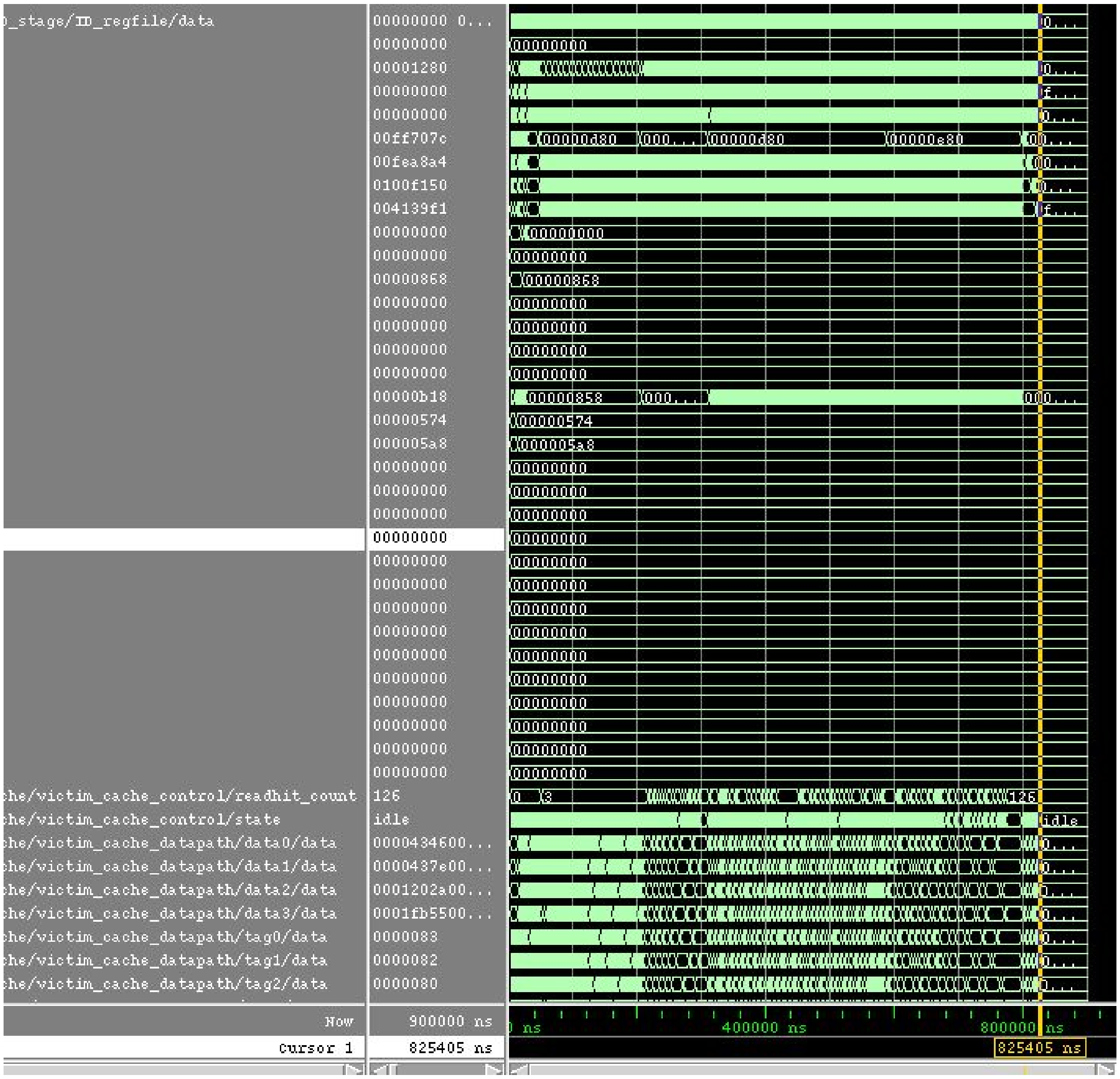
```
                                                  Msgs
/mp3_tb/clk_by_2                                  x
/mp3_tb/clk                                       1
/mp3_tb/halt                                      1
/mp3_tb/read                                      0
/mp3_tb/write                                     0
/mp3_tb/address                                   00000000
/mp3_tb/wdata                                     000000000...
/mp3_tb/resp                                      0
/mp3_tb/rdata                                     000043460...
/mp3_tb/mp3/mp3_cpu/ID_stage/ID_regfile/data      00000000 ...
    [0]                                           00000000
    [1]                                           00001280
    [2]                                           ffffffff
    [3]                                           00000000
    [4]                                           00ff707c
    [5]                                           00fea8a4
    [6]                                           0100f150
    [7]                                           ffbec60e
    [8]                                           00000000
    [9]                                           00000000
    [10]                                          00000868
    [11]                                          00000000
    [12]                                          00000000
    [13]                                          00000000

Now        1000200 ns
Cursor 1   849385 ns
```

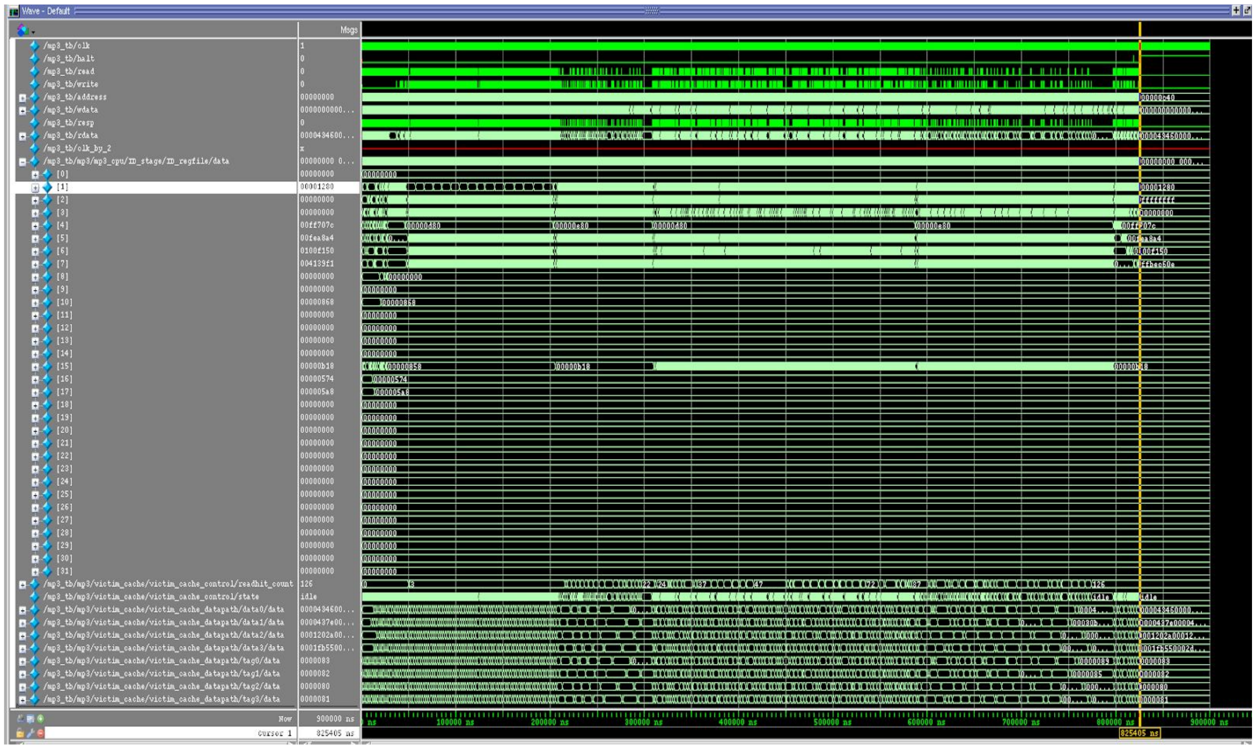
After adding the prefetching unit, we found a minor improvement in runtime from 849,300 to 847,200, nearly 0.5 percent. We think the reason is that we generate the potential prefetchable data in IF stage, only 3 clock cycle ahead and a cache miss immediately block prefetching process. All these reasons significantly reduces the hit rate in the prefetched data.



3. Victim cache: Before adding the victim cache, the mp3-final.s code runs for 847,200 ns. After implementing the victim cache, the performance improved to 825,405ns. The performance boost is mainly due to the reduction of L2 miss but victim hit penalty down to 1 cycle. To visualize the performance, a victim cache hit counter is established that for every L2 miss and victim cache hit, the counter goes up by 1. As indicated in the test waveform, for the mp3-final.s test code, a total of 126 victim cache hit is observed.



```
_stage/ID_regfile/data                       00000000 0...
                                             00000000
                                             00001280
                                             00000000
                                             00000000
                                             00ff707c
                                             00fea8a4
                                             0100f150
                                             004139f1
                                             00000000
                                             00000000
                                             00000868
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000b18
                                             00000574
                                             000005a8
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
                                             00000000
che/victim_cache_control/readhit_count       126
che/victim_cache_control/state               idle
che/victim_cache_datapath/data0/data         0000434600...
che/victim_cache_datapath/data1/data         0000437e00...
che/victim_cache_datapath/data2/data         0001202a00...
che/victim_cache_datapath/data3/data         0001fb5500...
che/victim_cache_datapath/tag0/data          0000083
che/victim_cache_datapath/tag1/data          0000082
che/victim_cache_datapath/tag2/data          0000080
                                     Now     900000 ns
                                Cursor 1     825405 ns
```



```
/mp3_tb/clk                                                     1
/mp3_tb/halt                                                    0
/mp3_tb/read                                                    0
/mp3_tb/write                                                   0
/mp3_tb/address                                                 00000000
/mp3_tb/wdata                                                   0000000000...
/mp3_tb/resp                                                    0
/mp3_tb/rdata                                                   0000434600...
/mp3_tb/clk_by_2                                                x
/mp3_tb/mp3/mp3_cpu/ID_stage/ID_regfile/data                    00000000 0...
  [0]                                                           00000000
  [1]                                                           00001280
  [2]                                                           00000000
  [3]                                                           00000000
  [4]                                                           00ff707c
  [5]                                                           00fea8a4
  [6]                                                           0100f150
  [7]                                                           004139f1
  [8]                                                           00000000
  [9]                                                           00000000
  [10]                                                          00000868
  [11]                                                          00000000
  [12]                                                          00000000
  [13]                                                          00000000
  [14]                                                          00000000
  [15]                                                          00000b18
  [16]                                                          00000574
  [17]                                                          000005a8
  [18]                                                          00000000
  [19]                                                          00000000
  [20]                                                          00000000
  [21]                                                          00000000
  [22]                                                          00000000
  [23]                                                          00000000
  [24]                                                          00000000
  [25]                                                          00000000
  [26]                                                          00000000
  [27]                                                          00000000
  [28]                                                          00000000
  [29]                                                          00000000
  [30]                                                          00000000
  [31]                                                          00000000
/mp3_tb/mp3/victim_cache/victim_cache_control/readhit_count     126
/mp3_tb/mp3/victim_cache/victim_cache_control/state             idle
/mp3_tb/mp3/victim_cache/victim_cache_datapath/data0/data       0000434600...
/mp3_tb/mp3/victim_cache/victim_cache_datapath/data1/data       0000437e00...
/mp3_tb/mp3/victim_cache/victim_cache_datapath/data2/data       0001202a00...
/mp3_tb/mp3/victim_cache/victim_cache_datapath/data3/data       0001fb5500...
/mp3_tb/mp3/victim_cache/victim_cache_datapath/tag0/data        0000083
/mp3_tb/mp3/victim_cache/victim_cache_datapath/tag1/data        0000082
/mp3_tb/mp3/victim_cache/victim_cache_datapath/tag2/data        0000080
/mp3_tb/mp3/victim_cache/victim_cache_datapath/tag3/data        0000081
                                                        Now     900000 ns
                                                   Cursor 1     825405 ns
```