CME 433

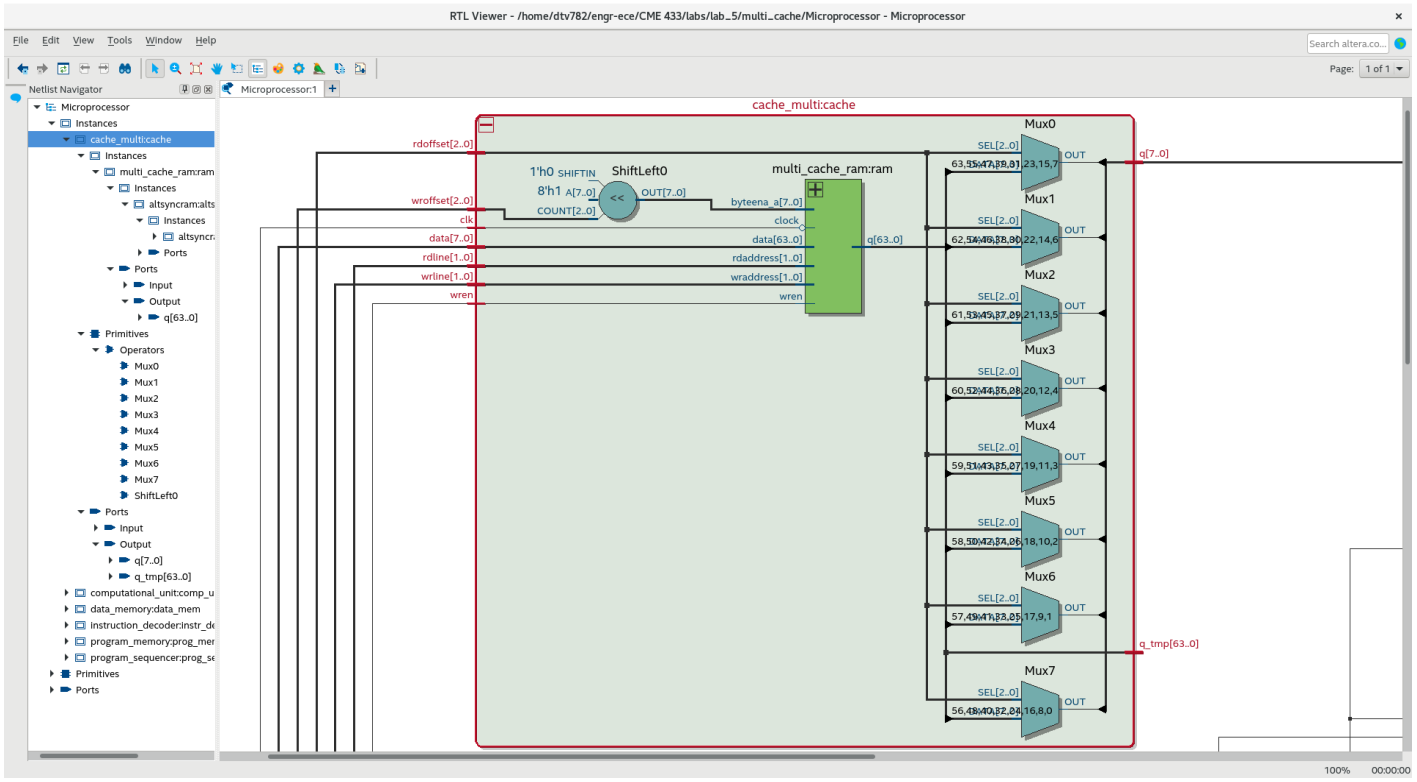
Lab 5: The Direct Mapped & Set-Associative Cache

Student Name: Dillon Vu

Student Number: 11100292

Part I : Multi-Line Direct-Mapped Cache

1. Diagram



2. Explain

   a. A direct-mapped cache operates: each main memory address maps to exactly one cache block. The cache is organized into multiple sets with a single cache line per set.

   b. A cache hit occurs when a requested data is in the cache and that cache is able to fulfill that request. Example of cache hit in figure 7: when instructions 8c, 3f,d9 had already been in cache, instruction register output continued to execute instruction 8c,3f, d9 in the next cycle.

   c. Cache misses occur when the requested data is not in the cache. The requested data then must be accessed via main memory and it will be written and stored into a new line in the cache. Example of cache miss in figure 8: instruction ba was not in the cache, then the processor had to suspended for 8 clock cycle to write ba into cache and then instruction register executed it after

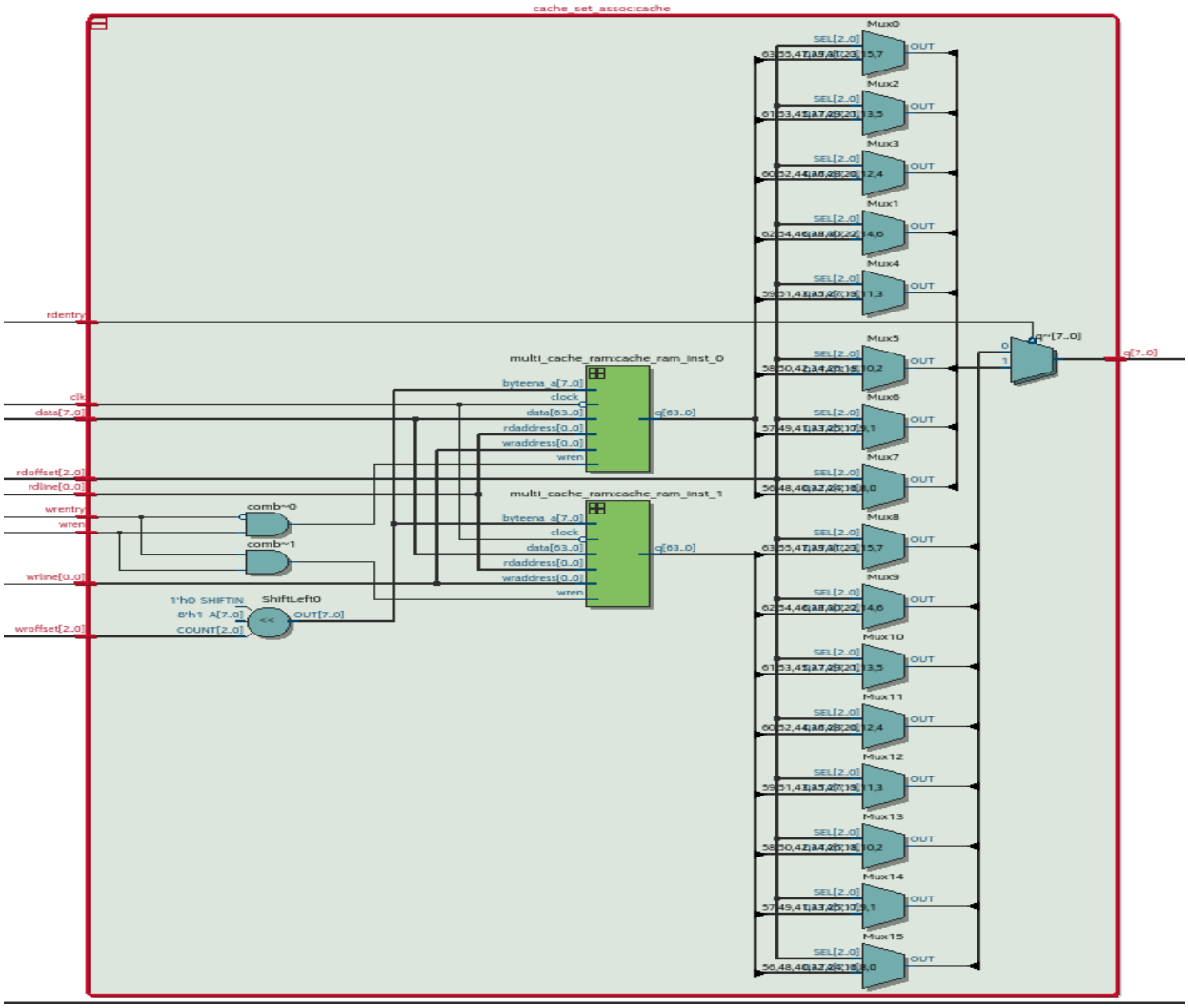
3. Comparison hardware resource utilization
    a. single_cache_init required more memory bits compare to multi_cache
    b. Multi_cache required more total registers but has less logic elements and total fan out compare to single_cache_init

	Multi_cache	Single_cache_init
Total Registers	71	57
Total Logic Elements	320	443
Total Fan out	2089	3349
Total Memory bits	2368	2624

4. Observer the execution time
    a. Base on Figure 4, The execution time of single_cache_init is 803.5 us
    b. Base on Figure 5, The execution time of multi_cache is 694.5 us
    c. Implementing multi_cache reduces the execution time.

Part II: 2-way set associative cache

1. Diagram



2. Describe the operation of the set-associative cache
    a. A set associative cache reduces conflicts by providing extra blocks in each set where data mapping to that set might be found. Each memory address still maps to a specific set but it can map to any of the extra blocks in the set.
    b. A cache hit occurs when a requested data is in the cache and that cache is able to fulfill that request. Example of cache hit in figure 9: when instructions 10, 20,30,40,51,60,80 had already been in cache, instruction register output continued to execute instruction 10, 20,30,40,51,60,80 in the next cycle.
    c. Cache misses occur when the requested data is not in the cache. The requested data then must be accessed via main memory and it will be written and stored

into a new line in the cache. Example of cache miss in figure 10: instruction ba was not in the cache, then the processor had to suspend for 8 clock cycles to write ba into cache and then instruction register executed it after.

3. Comparison hardware resource utilization
    a. Set associative cache and multi_cache have the same memory bits, set associative cache has the most registers. It also has more total logic elements and total fan out than multi_cache but still less than single_cache_unit

	Single_cache_intit	Multi_cache	Set associative cache
Total Registers	57	71	77
Total Logic Elements	443	320	381
Total Fan out	3349	2089	2580
Total Memory bits	2624	2368	2368

4. Observer the execution time

   Execution time:
    A. single cache: 803.5 us
    B. Multi_cache: 698.5 us
    C. Set_assoc_cache: 690.5 us

   Set_asso_cache has a slightly more execution time compared to multi_cache.

pm_address	**Single_cache_intit**	**Multi_cache**	**Set associative cache**
0x70	conflict MISS	Conflict MISS	HIT
0x80	Compulsory MISS	Conflict MISS	Compulsory MISS
0x90	HIT	Conflict MISS	Conflict MISS
0xA0	Compulsory MISS	Compulsory MISS	Compulsory MISS

5. Explain

a. tagID : represents a unique identifier of a group of data in cache, which is determined from main memory.
b. Valid : represent The valid bit marks which determine whether the cache block is used or not.
c. Currdentry: represent current entry data for a given set at the current clock cycle
d. Lastused: represent last written item in the cache in the previous clock cycle. This lastused bit used for LRU replacement policy

	Resource	Usage
1	Estimated Total logic elements	320
2		
3	Total combinational functions	286
4	▼ Logic element usage by number of LUT inputs	
1	-- 4 input functions	194
2	-- 3 input functions	55
3	-- <=2 input functions	37
5		
6	▼ Logic elements by mode	
1	-- normal mode	258
2	-- arithmetic mode	28
7		
8	▼ Total registers	71
1	-- Dedicated logic registers	71
2	-- I/O registers	0
9		
10	I/O pins	149
11	Total memory bits	2368
12		
13	Embedded Multiplier 9-bit elements	0
14		
15	Maximum fan-out node	clk~input
16	Maximum fan-out	147
17	Total fan-out	2089
18	Average fan-out	2.86

Figure 1 :multi_cache Resource Usage Summary

	Resource	Usage
1	Estimated Total logic elements	443
2		
3	Total combinational functions	421
4	▼ Logic element usage by number of LUT inputs	
1	-- 4 input functions	325
2	-- 3 input functions	55
3	-- <=2 input functions	41
5		
6	▼ Logic elements by mode	
1	-- normal mode	393
2	-- arithmetic mode	28
7		
8	▼ Total registers	57
1	-- Dedicated logic registers	57
2	-- I/O registers	0
9		
10	I/O pins	150
11	Total memory bits	2624
12		
13	Embedded Multiplier 9-bit elements	0
14		
15	Maximum fan-out node	clk~input
16	Maximum fan-out	325
17	Total fan-out	3349
18	Average fan-out	3.20

Figure 2 :single_Cache_init Resource Usage Summary

	Resource	Usage
1	Estimated Total logic elements	381
2		
3	Total combinational functions	343
4	▼ Logic element usage by number of LUT inputs	
1	-- 4 input functions	242
2	-- 3 input functions	66
3	-- <=2 input functions	35
5		
6	▼ Logic elements by mode	
1	-- normal mode	315
2	-- arithmetic mode	28
7		
8	▼ Total registers	77
1	-- Dedicated logic registers	77
2	-- I/O registers	0
9		
10	I/O pins	149
11	Total memory bits	2368
12		
13	Embedded Multiplier 9-bit elements	0
14		
15	Maximum fan-out node	clk~input
16	Maximum fan-out	217
17	Total fan-out	2580
18	Average fan-out	3.01

Figure 3 :Set Associative Cache Resource Usage Summary

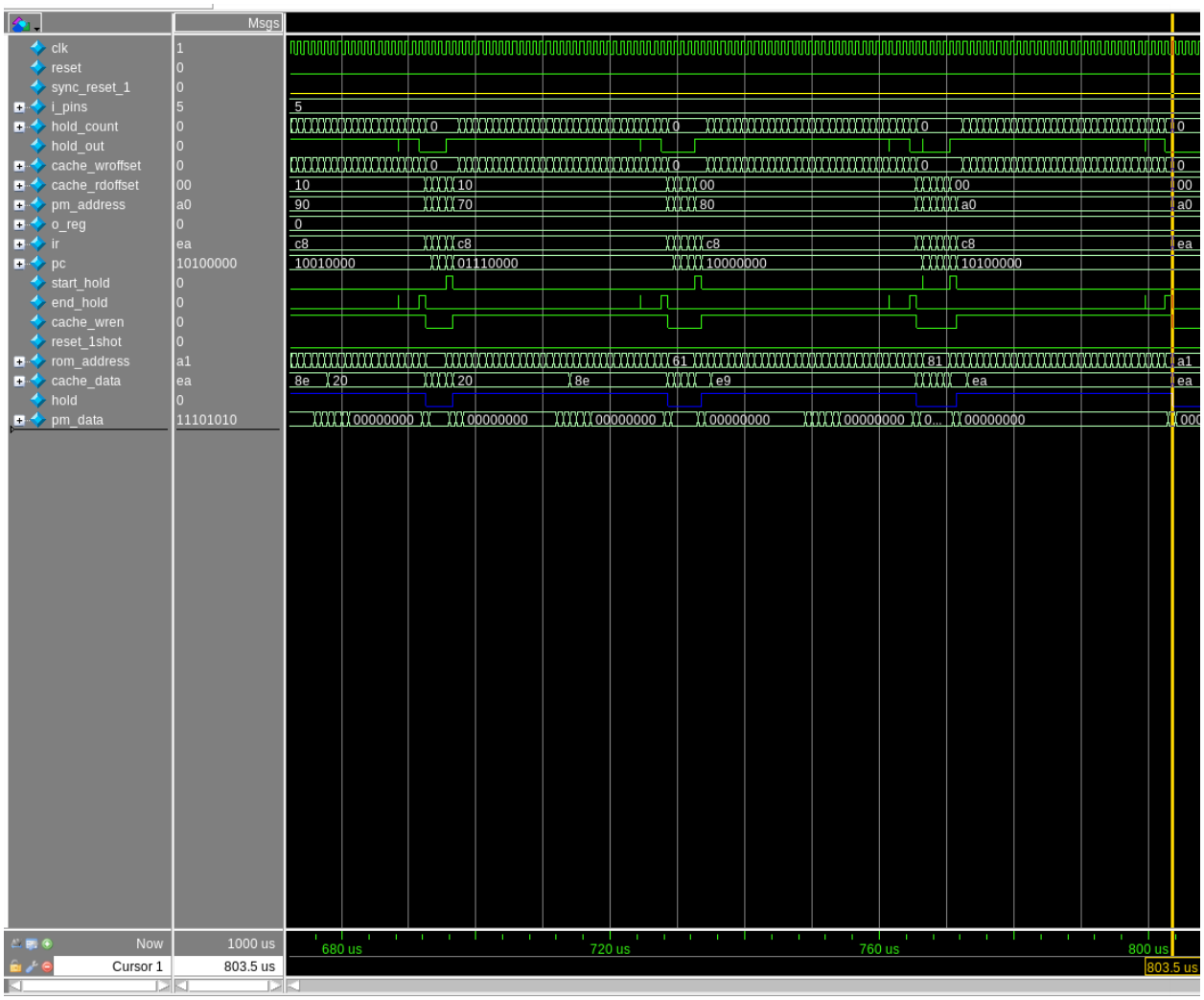


Figure 4 :Single Cache Execution Time

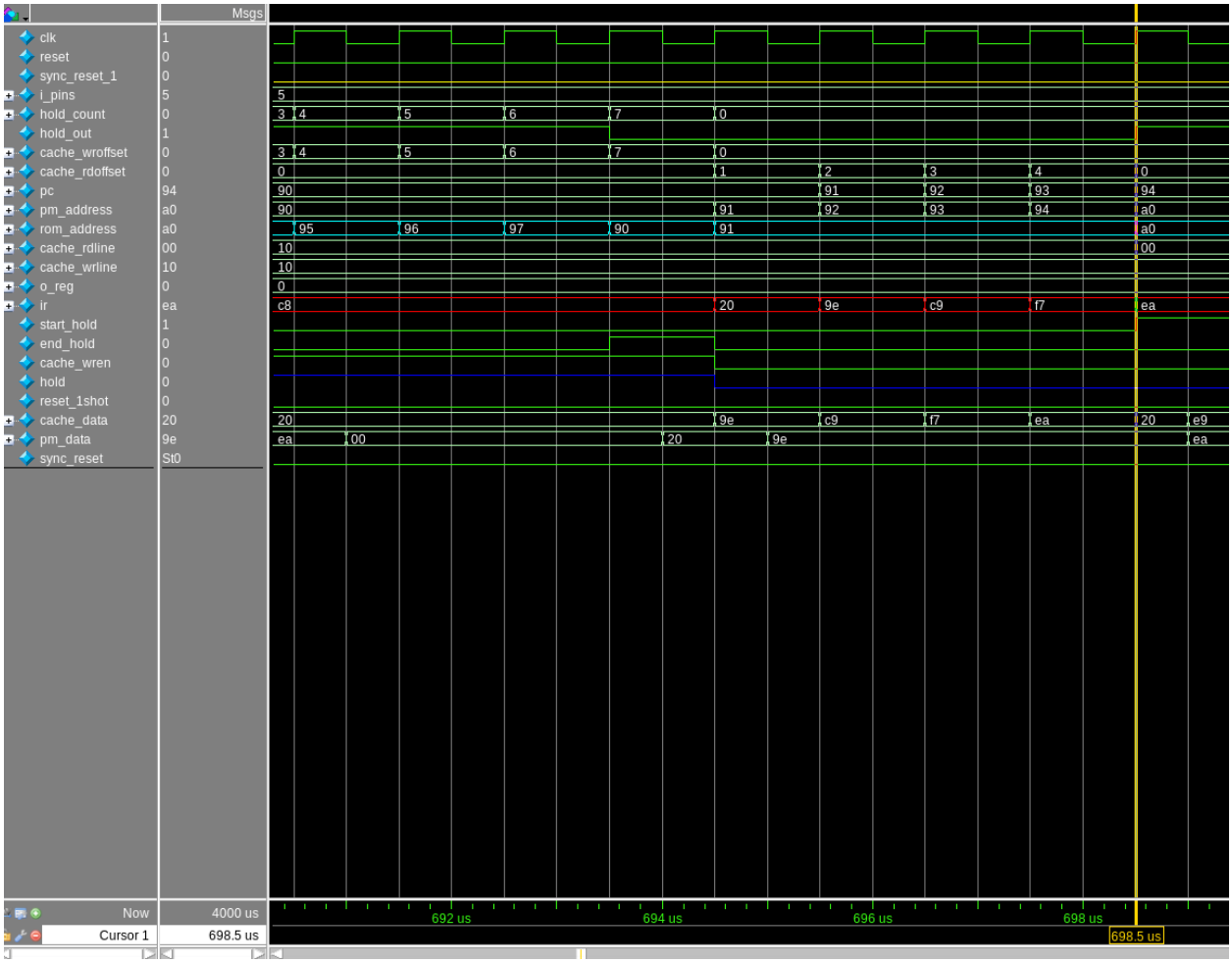


Figure 5 :Multi Cache Execution Time

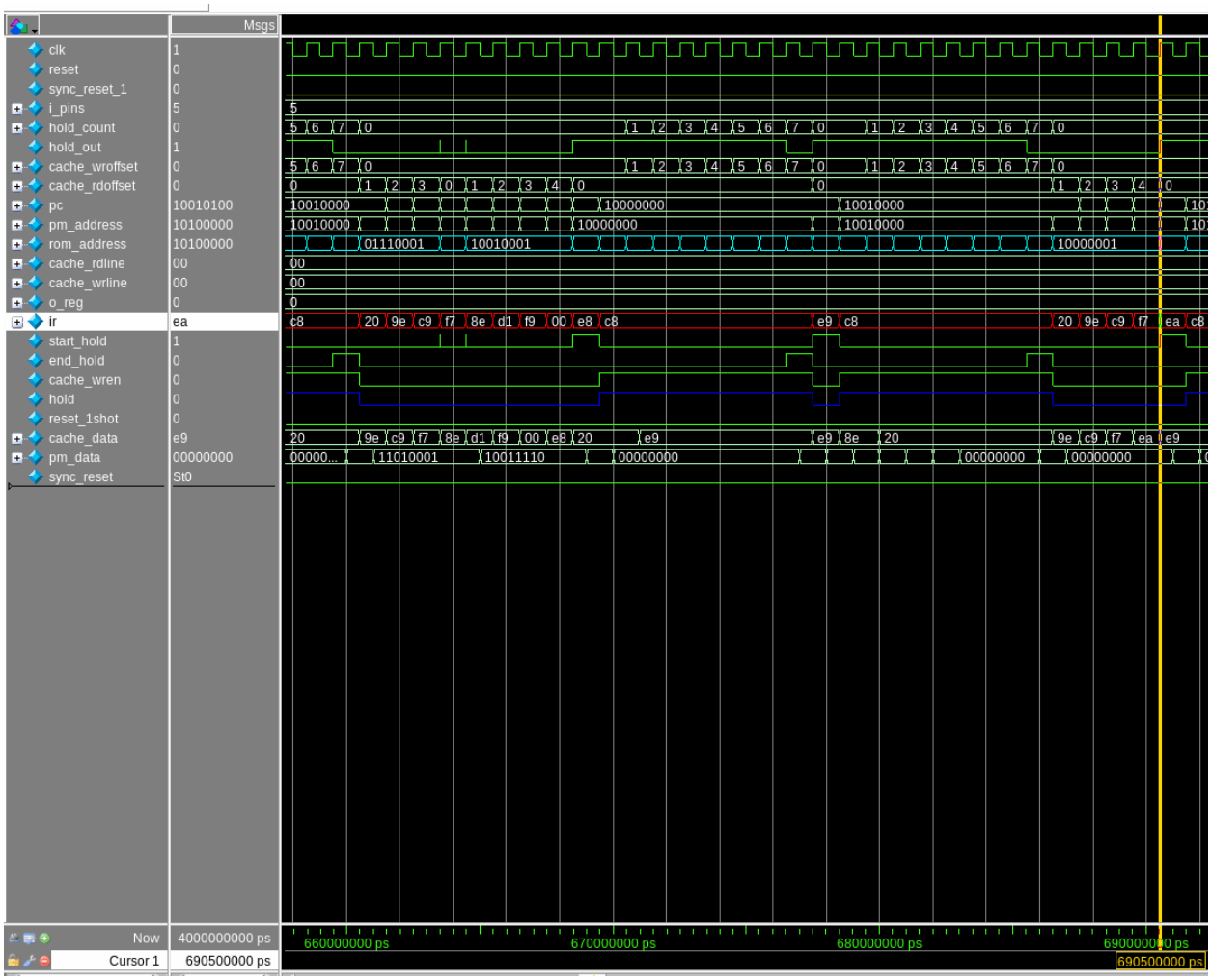


Figure 6 :Set Associative Cache Execution Time

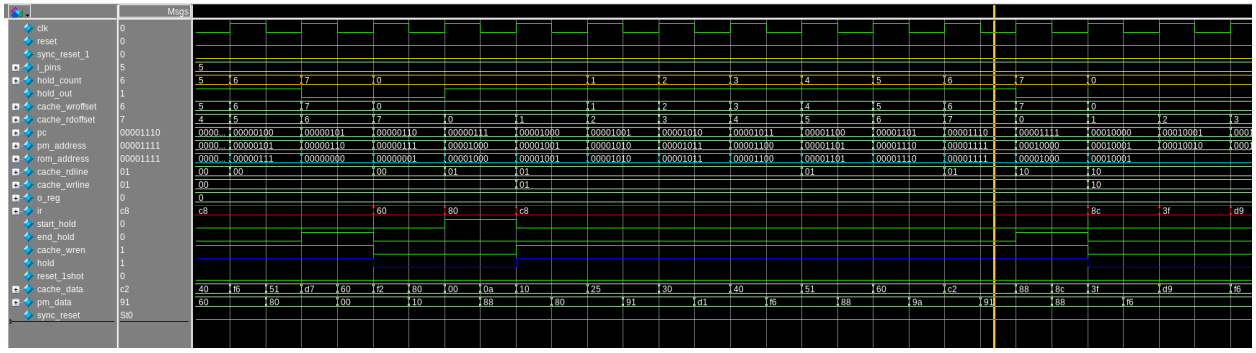


Figure 7: example of cache hit in multi cache

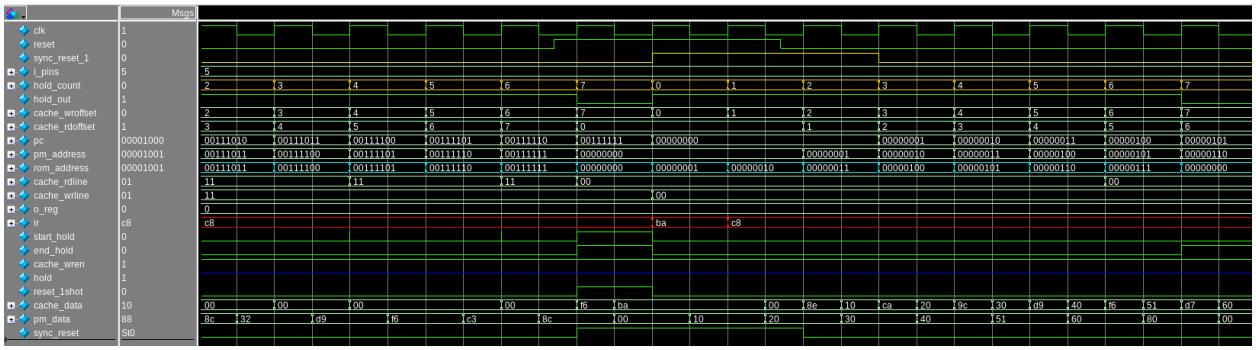


Figure 8: example of cache miss in multi cache.

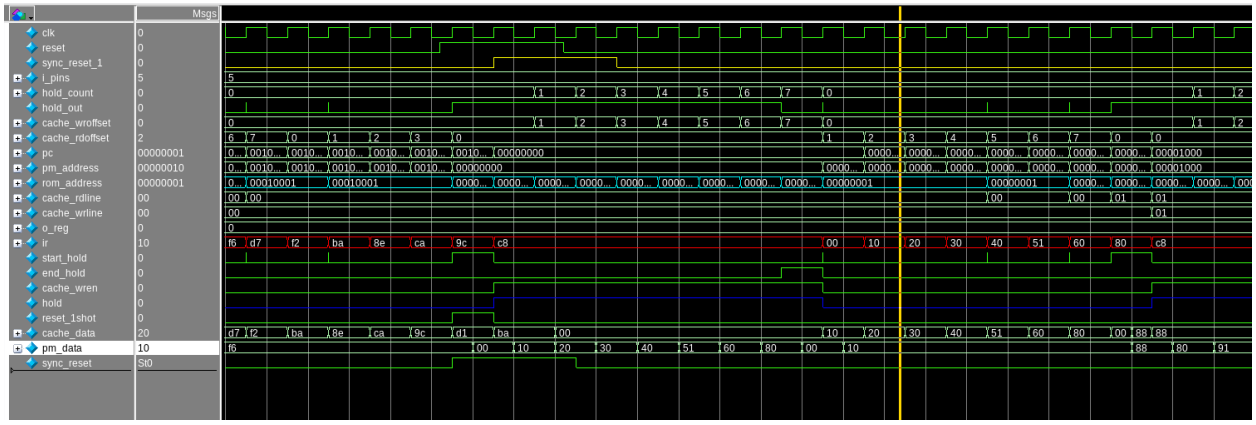


Figure 9: example of cache hit in set associative cache

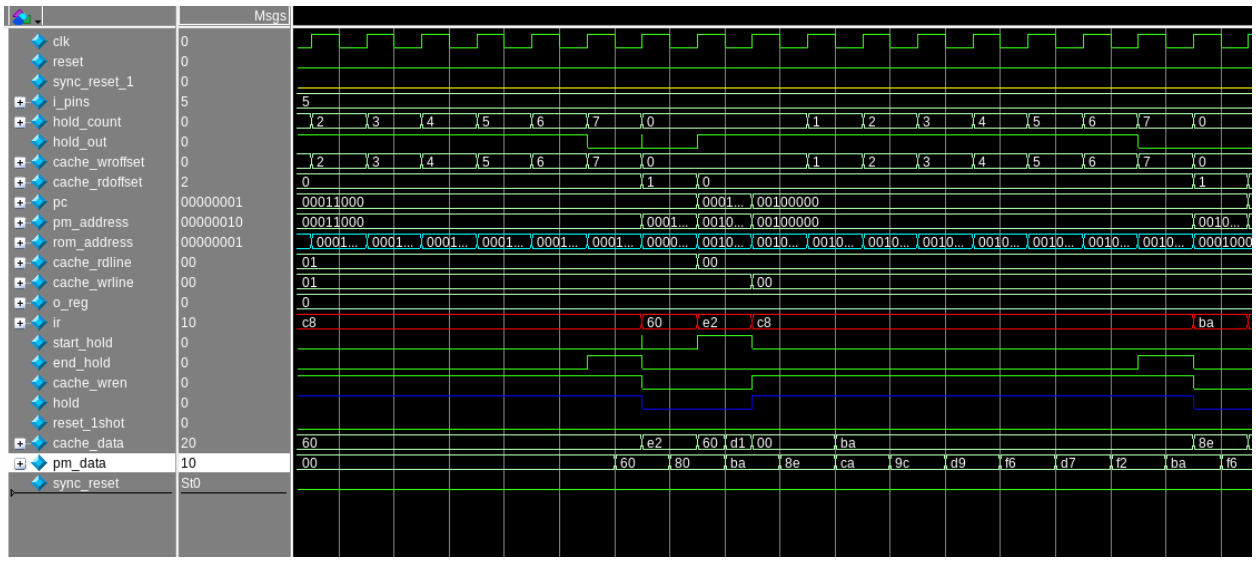


Figure 10: example of cache miss in set associative cache