GROUP 15

CO224 Computer Architecture - 2020
Department of Computer Engineering

Lab 6 - Building a Memory Hierarchy

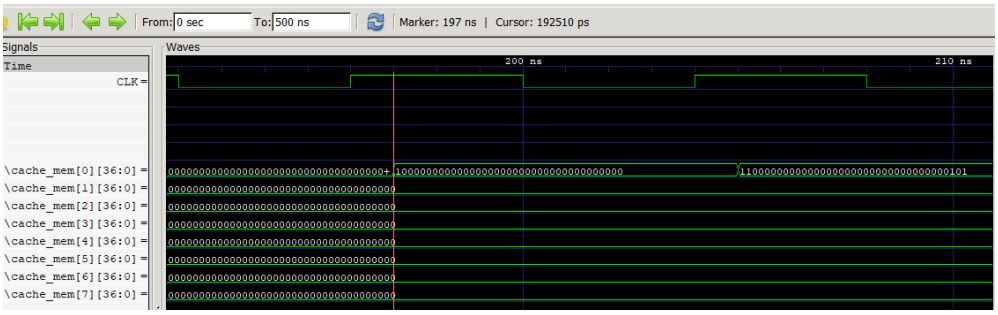
E/17/100
E/17/246

Let’s compare the cache-less memory and the memory module that have the cache. To do that initially we can give simple instruction set to the both module.

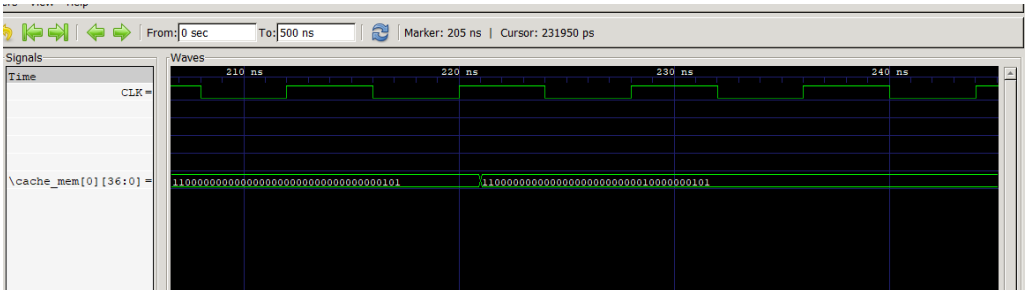
```
loadi 4 0x05        //Load  value 5 into register 4
loadi 5 0x04        //Load  value 4 into register 5
swi 4 0x00          //Store the value in reg 4 to memory
swi 5 0x01          //Store the value in reg 5 to memory
```

Now we can observe the relevant timing diagrams.

1. System with the cache



Above timing diagram shows the behaviour of the cache memory. When the instruction is swi 4 0x00 it miss the cache. (WRITE MISS) . Because initially reset the cache and when reset is given the all valid bits becomes invalid. So that instruction miss the cache and spend 20 clock cycles to read the relevant block from the memory. After that ,the block is fetched to the cache within another clock cycle. After that the value 5 (last 8 bit-> 00000101) is write to the cache.

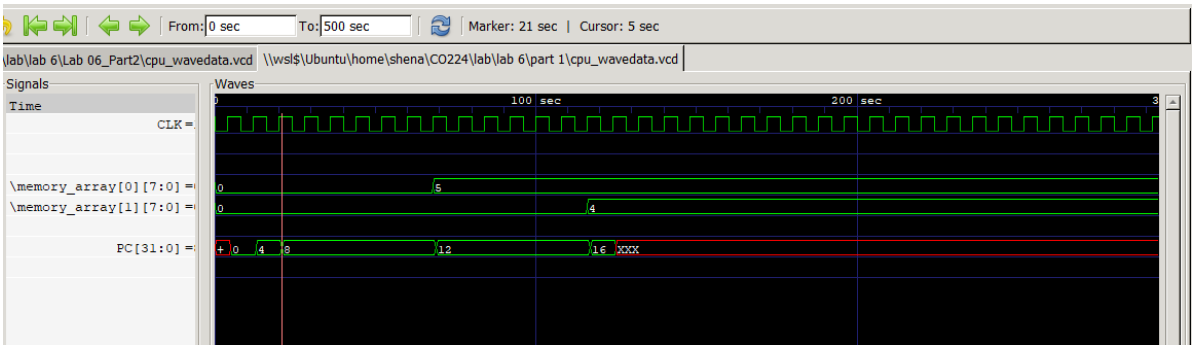


Next instruction is to write the value stored in reg 5 to memory 1. Since memory address 0x01 belongs to the same block of the above and that block is in the cache

now. Therefor that value is write to the cache memory's relevant place (second 8 bits 00000100). It did not consume more clock cycles

2. Cache- less system

.



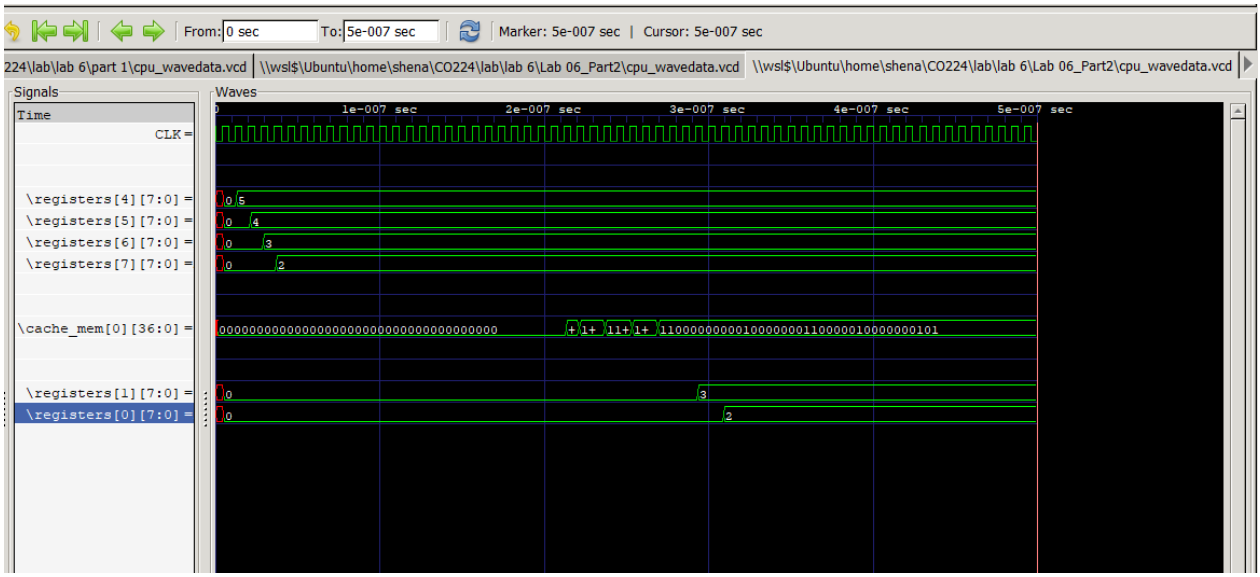
Above timing diagram show the responses to the same instruction above. In this after the swi is load (when pc =8) it consume 5 clock cycles to write into the memory. After that the next instruction is also a swi. Then it also consume another 5 clock cycles to write the memory. Even these two blocks are in the same block(4 size) it consume another 5 clock cycles

In the system with the cache it consume more clock cycles initially but after when we deal with the memory address that belongs to the same block in the cache it reduce the time that cache-less one.

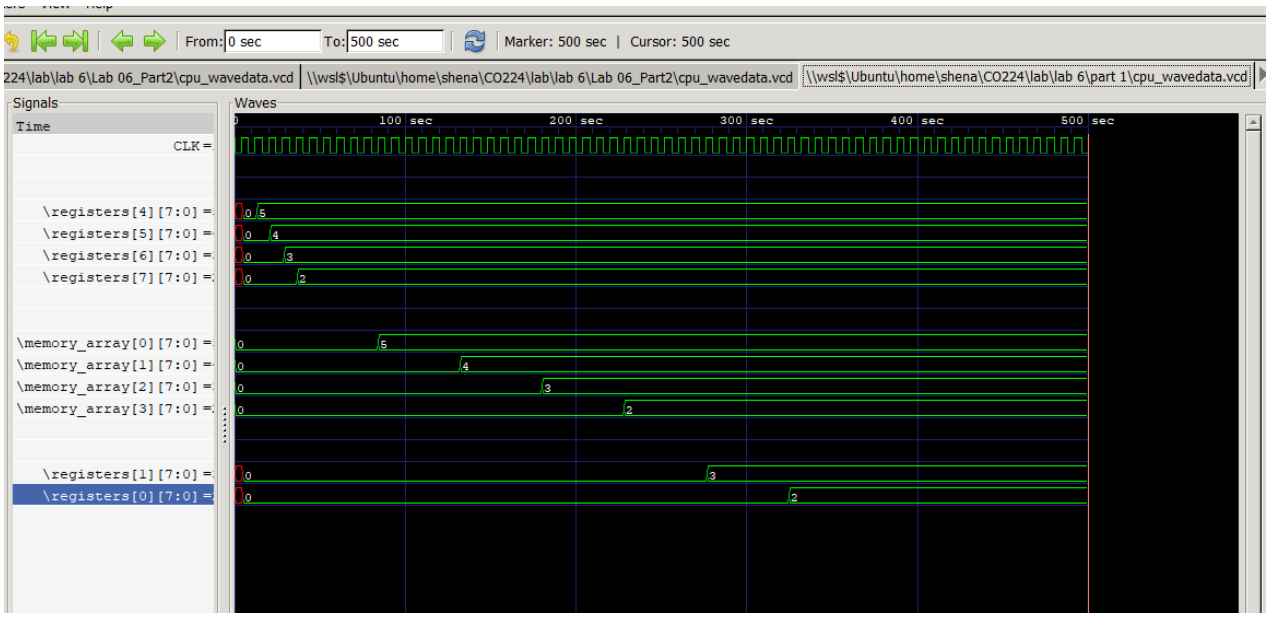
Let's consider another instruction set

loadi 4 0x05
loadi 5 0x04
loadi 6 0x03
loadi 7 0x02
swi 4 0x00
swi 5 0x01
swi 6 0x02
swi 7 0x03
lwi 1 0x02
lwi 0 0x03

1. System with the cache



2. Cache –less system



From above two diagrams we can compare the cache less system with other one.

- If the system does not have cache mem then it consume 5 clock cycles for each time when it access (read or write)
- If the system have cache, it does not consume fixed clock cycles each time. When we accessing same block it's consume big number of clock cycle when it is amiss or the dirty bit is high.

When comparing these two we can say that,

- If our programme works with lots of cache hits then the system with the cache is faster than a cache – less one. Because with the help of the cache memory we can access to a recently use address in a memory faster.
- But if our programme has lot of cache misses it consume lots of clock cycles.