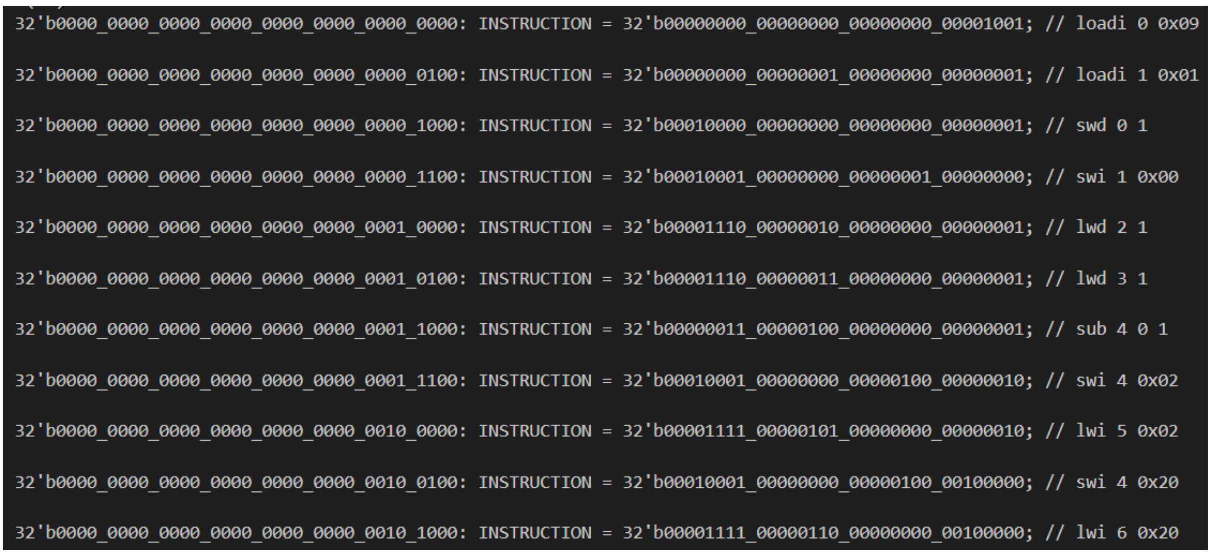
CO224 LAB 06 - PART 2

GROUP 39
E/20/032
E/20/034

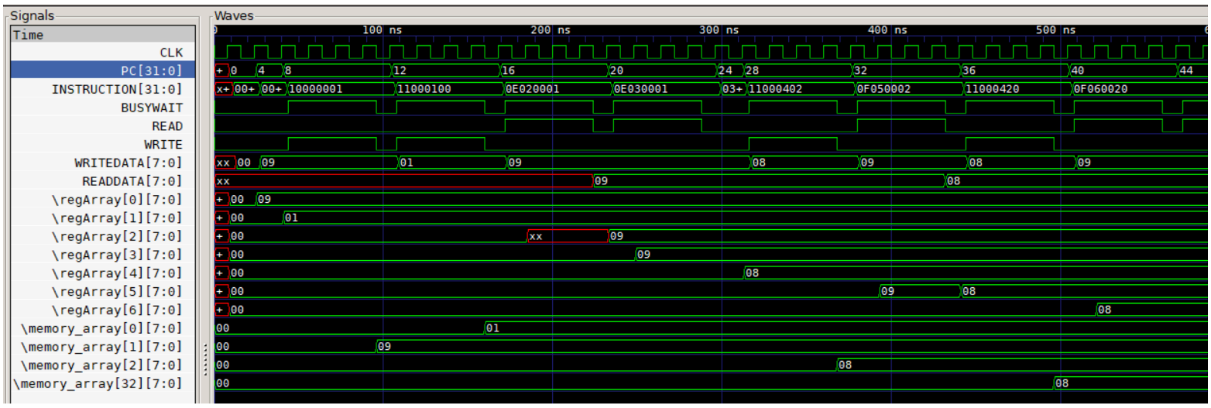
Executed instructions



```
32'b0000_0000_0000_0000_0000_0000_0000_0000: INSTRUCTION = 32'b00000000_00000000_00000000_00001001; // loadi 0 0x09

32'b0000_0000_0000_0000_0000_0000_0000_0100: INSTRUCTION = 32'b00000000_00000001_00000000_00000001; // loadi 1 0x01

32'b0000_0000_0000_0000_0000_0000_0000_1000: INSTRUCTION = 32'b00010000_00000000_00000000_00000001; // swd 0 1

32'b0000_0000_0000_0000_0000_0000_0000_1100: INSTRUCTION = 32'b00010001_00000000_00000001_00000000; // swi 1 0x00

32'b0000_0000_0000_0000_0000_0000_0001_0000: INSTRUCTION = 32'b00001110_00000010_00000000_00000001; // lwd 2 1

32'b0000_0000_0000_0000_0000_0000_0001_0100: INSTRUCTION = 32'b00001110_00000011_00000000_00000001; // lwd 3 1

32'b0000_0000_0000_0000_0000_0000_0001_1000: INSTRUCTION = 32'b00000011_00000100_00000000_00000001; // sub 4 0 1

32'b0000_0000_0000_0000_0000_0000_0001_1100: INSTRUCTION = 32'b00010001_00000000_00000100_00000010; // swi 4 0x02

32'b0000_0000_0000_0000_0000_0000_0010_0000: INSTRUCTION = 32'b00001111_00000101_00000000_00000010; // lwi 5 0x02

32'b0000_0000_0000_0000_0000_0000_0010_0100: INSTRUCTION = 32'b00010001_00000000_00000100_00100000; // swi 4 0x20

32'b0000_0000_0000_0000_0000_0000_0010_1000: INSTRUCTION = 32'b00001111_00000110_00000000_00100000; // lwi 6 0x20
```

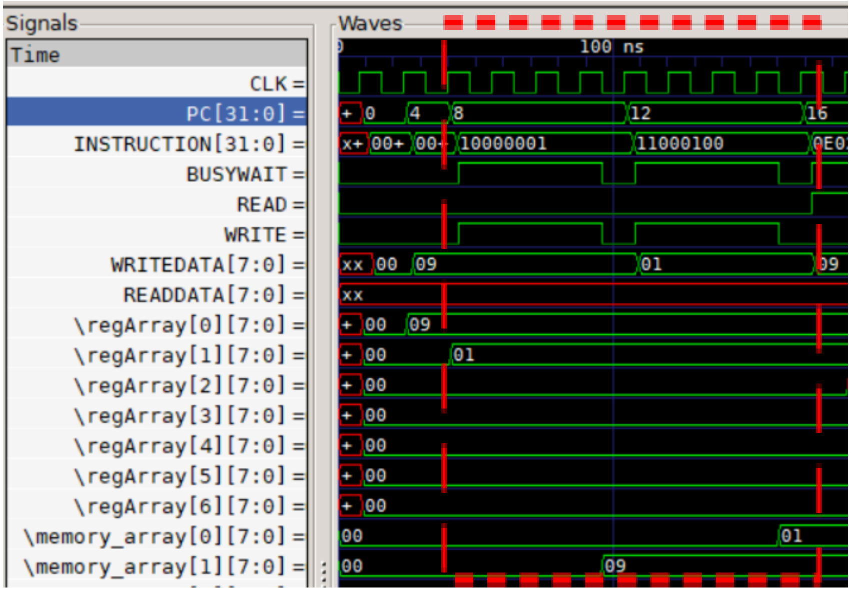
The following set of instructions given above were hardcoded into the test bench for execution on both CPUs with and without cache.

CPU without cache



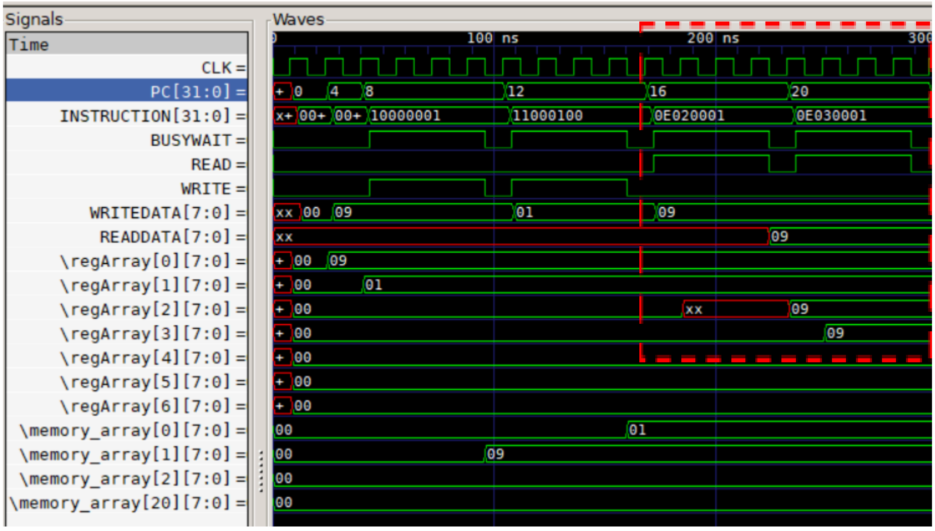
waveform diagram after executing all instructions

Here, in the cache-less approach an external 256x8bit memory module is directly connected to the CPU with additional READ, WRITE, BUSYWAIT, WRITEDATA, READDATA & ADDRESS ports. A Store Word instruction takes multiple clock cycles to complete. An 8 bit data array takes 40 time units to be read or written inside the memory module.



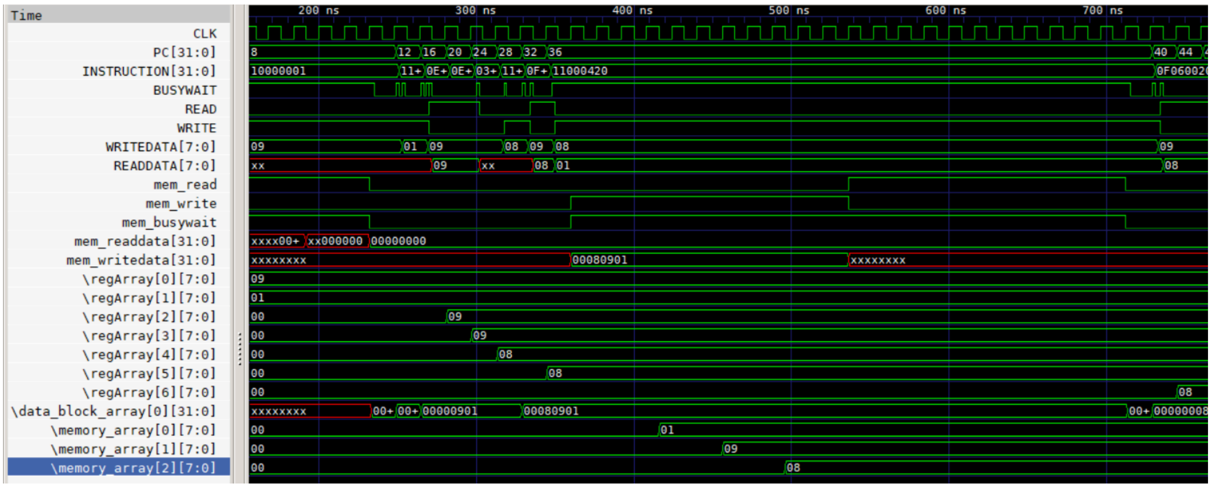
Memory address 0x01 and 0x00 being written by the values stored in register 0 and 1

Moreover, a Load Word instruction will take several clock cycles to complete because of the reading speed of the memory module.



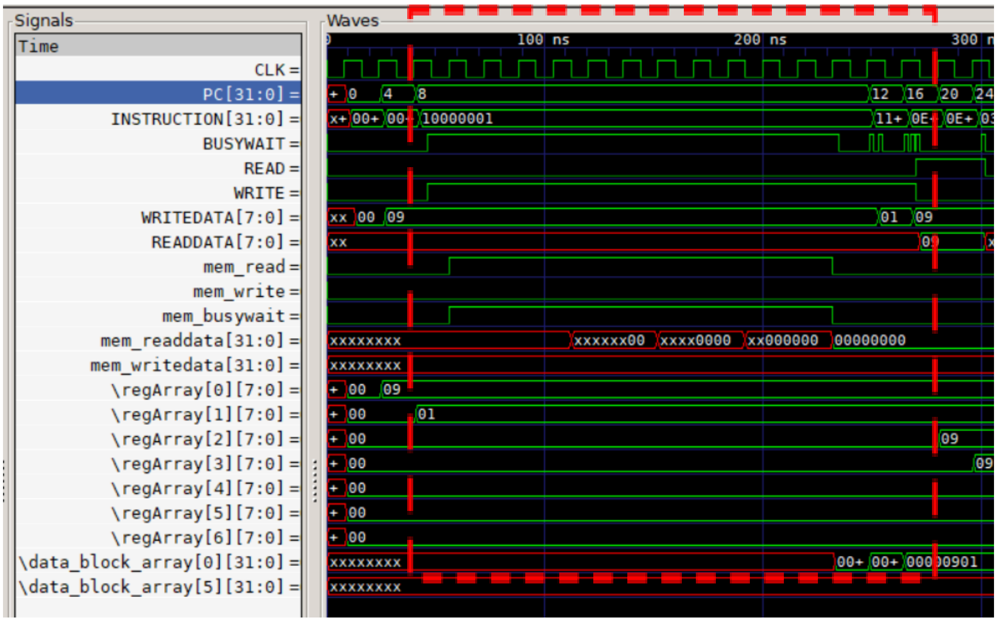
Loading stored values of memory address 0x00 to register 2 and register 3

CPU with cache



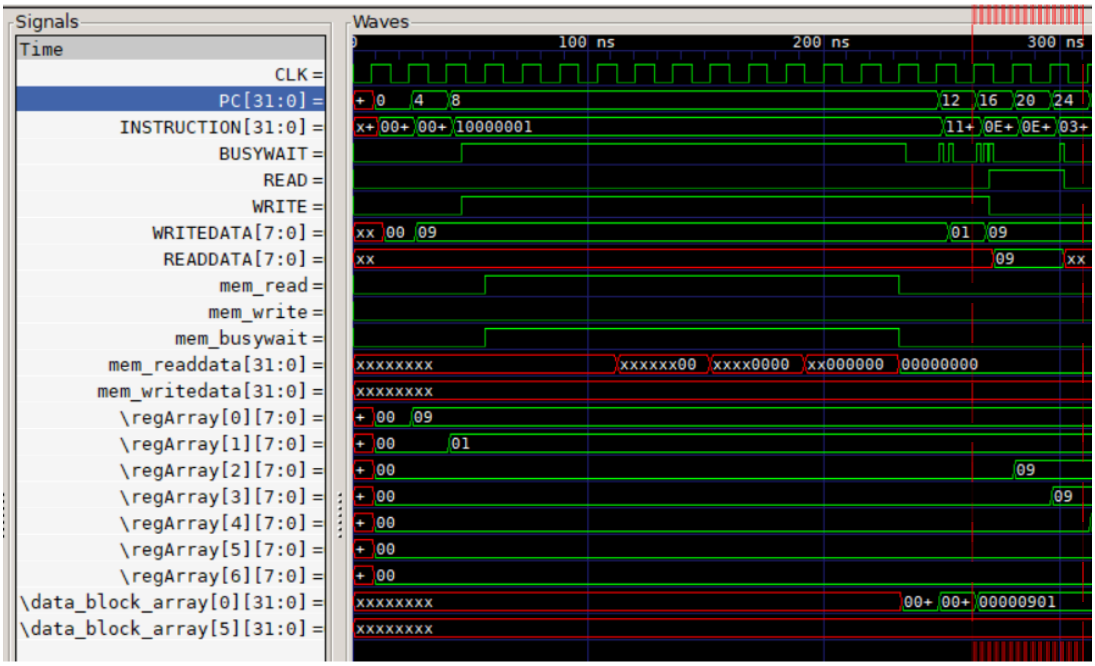
waveform diagram after executing all instructions

Instead of directly connecting the CPU to the memory module, a cache module is placed between the CPU and the Data Memory. It consists of READ, WRITE, BUSYWAIT, WRITEDATA, READDATA, ADDRESS ports for both CPU and Data Memory side connections. Cache consists of 8x32 bit data blocks to store data and the Data Memory consists of 256x8 bit arrays to store data. Here, the cache reads or writes data from the Data Memory as blocks of data (4x8 bit) which takes 40 time units for each 8 bit array to read/write in case of a read or write miss. But in the event of a read or write hit the Load Word or Store Word instruction will be finished executing within a single clock cycle.



<u>storing values of register 0 and 1 into memory address 0x01 and 0x00 (in cache)</u>

It can be observed that the CPU is stalled for multiple clock cycles during the first Store Word instruction. Since it is a write miss at the beginning, the cache is fetching the data block that includes 0x01 address data. Fetching the whole block takes few clock cycles and the fetched block is then written to the cache data block on the subsequent positive clock edge. Even though the first Store Word took several clock cycles, the next Store Word instruction on the 0x00 address is only takes 1 cycle to complete because 0x00 is in the same block as the previously written 0x01. Thus, it is a write hit and the data will be written to the cache data block with the respective index.



<u>loading values in memory address 0x01 onto register 2 and register 3</u>

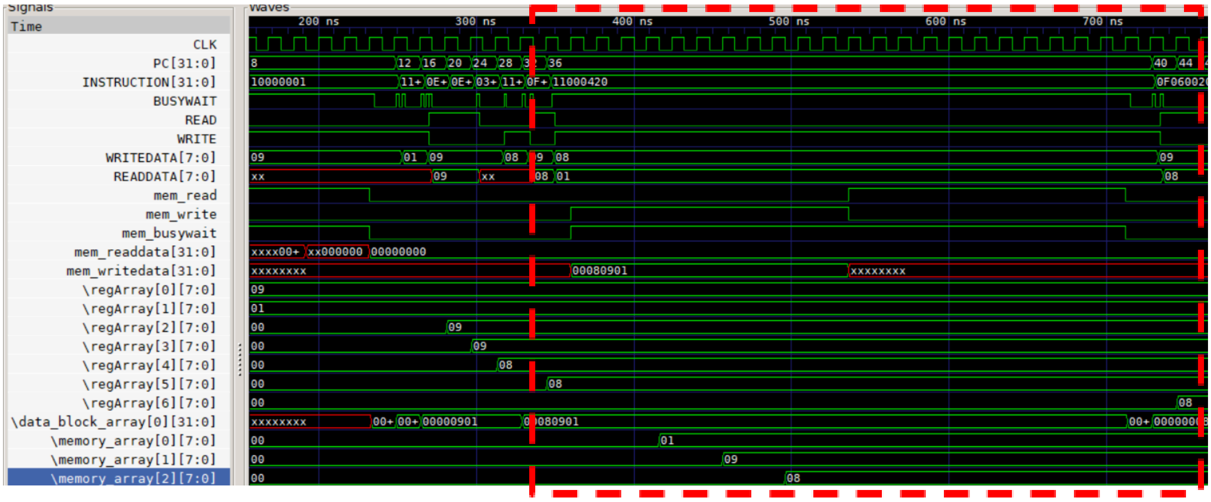
In this load word instruction the value stored in 0x01 is fetched from the cache because it was previously stored in the cache. All data from address 0x00, 0x01, 0x02 and 0x03 are stored in the same data block with the index 0 in cache. Since it is a read hit the instruction will be executed within a single clock cycle. If it was a read miss and the data in the cache entry was dirty, first the data in the cache entry is written to the Data Memory and then the corresponding data from the data memory is written to the cache before sending to the CPU. In the swi and lwi a write miss is occurred during the swi.

0x20 = 0b00100000

0x01 = 0b00000001

0x00 = 0b00000000

These addresses share the same index bits but tag bits are different in 0x20. Therefore, the existing block is written to the Data Memory and then the value for 0x20 is stored in the cache with the new tag bits.



Storing value of register 4 in 0x20 and loading it to register 6