Lab Assignment #3

EE 126: Computer Engineering

Tufts Univerisity, Fall 2019

Name: Ruoxi Ren Student ID: 1327867 E-mail: Ruoxi.Ren@tufts.edu

1 Waveforms of Single Cycle CPU

The CPU runs a MIPS program as below:

```
    lw $t2, 0($z)
L1: beq $t0, $t2, L2
    sub $s0, $s0, $s1
    add $t0, $t0, $t1
    j L1
L2: or $s2, $s0, $t3
    and $s2, $s2, $s3
    sw $s2, 4($t3)
    nop
```

Before running the program, some registers are set as below:

Registers	Value
$t0	0
$t1	1
$t2	4
$t3	8
$s0	21
$s1	7
$s2	0
$s3	22

And the memory is set as below:

Address	Value
DMEM(0x0)	0x00 00 00 02
DMEM(0x4)	0x00 00 00 00
DMEM(0x8)	0x00 00 00 00
DMEM(0xC)	0x00 00 00 00

The waveforms of cycle 1 are shown in Fig. 1. We can divide it into two parts. The first part, as is labeled in "1" in the picture, it can be seen that we first assign the signal *rst* to be 1, which means we first reset the CPU to make *PC* equal 1. It fetch the first instruction, which means put the value in 0x0 of memory into register $t2. So at the beginning of the second part, which is the right time to write back the value from the memory to the registers, the value of register $t2 change to 0x00000002.

The waveforms of cycle 2 to 7 are shown in Fig. 2.

In the cycle 2, the second instruction is fetched, which means we compare the value of register $t0 and register $t2. If they are the same, then jump to L2. Now we have $t2=0x00000002, $t0=0x00000000, since they are different, we would not jump to L2.

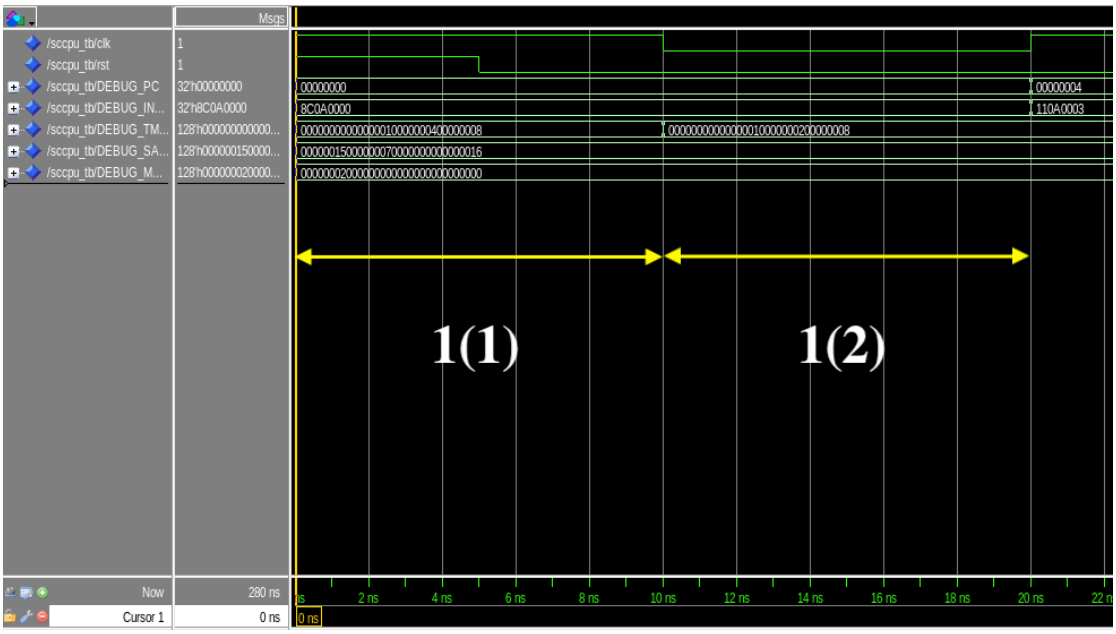

Figure 1: Simulation of single cycle CPU for cycle 1

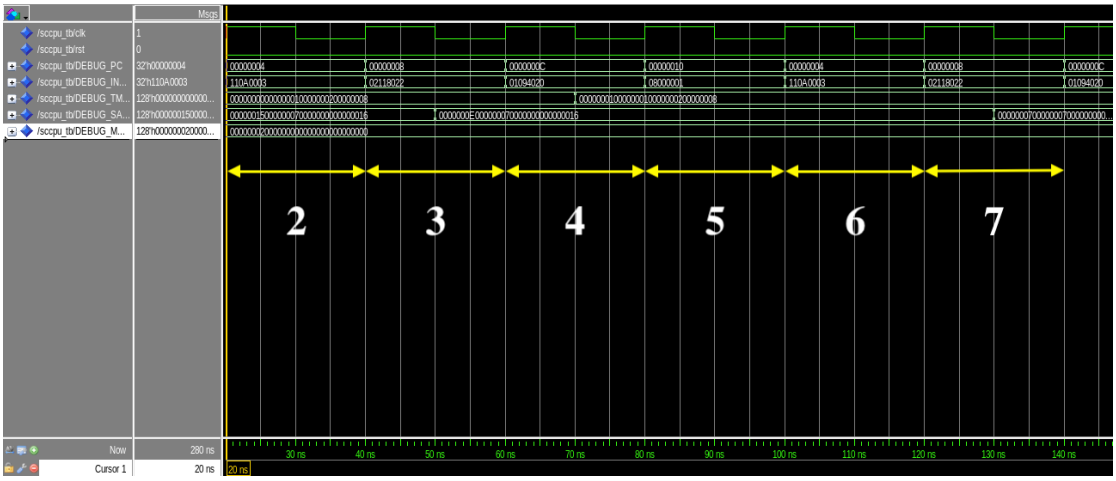

Figure 2: Simulation of single cycle CPU for cycle 2 to 7

In the cycle 3, the next instruction is fetched, which means $s0=$s0-$s1. Because $s0=0x00000015, $s1=0x00000007, the new value of $s0 would be 0x00000015-0x00000007=0x0000000E. Thus in the second half of cycle 3, the value of $s1 changes to 0x0000000E.

In the cycle 4, the next instruction is fetched, which means $t0=$t0+$t1. Then the new value of $t0 would be 0x00000000+0x00000001=0x00000001. So in the second half of cycle 4, the value of $t0 changes to 0x00000001.

In the cycle 5, the next instruction is fetched, which means unconditionally jumping to L1. In this cycle, no registers' value would be changed.

In the cycle 6, we can see that the instruction fetched is the instruction in L1, and PC is 0x00000004, which is the position of the second instruction in the program. Still we compare the value of $t0 and $t2. Apparently, $t0=0x00000001, $t2=0x00000002, which is not equal, so we do not jump to L2.

In the cycle 7, we assign the value of $s0-$s1 to $s0. Then the new value of $s0 is 0x0000000E-0x00000007=0x00000007. As is shown in the picture, the value of $s0 is changed to 0x00000007

in the second half of the cycle.

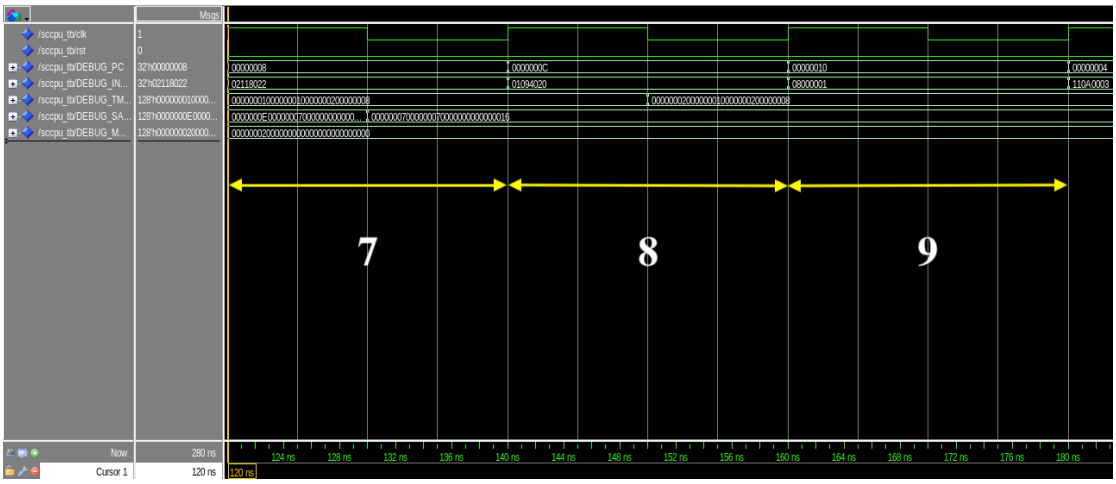

Figure 3: Simulation of single cycle CPU for cycle 7 to 9

The waveforms of cycle 7 to 9 are shown in Fig. 3.

In the cycle 8, we assign the value of $t0+$t1 to $t0. The new value of $t0 is 0x00000001+0x00000001 =0x00000002. Then in the second half of the cycle, the value of $t0 is changed to 0x00000002.

In the cycle 9, we reach the Jump instruction. No registers is modified in this cycle.

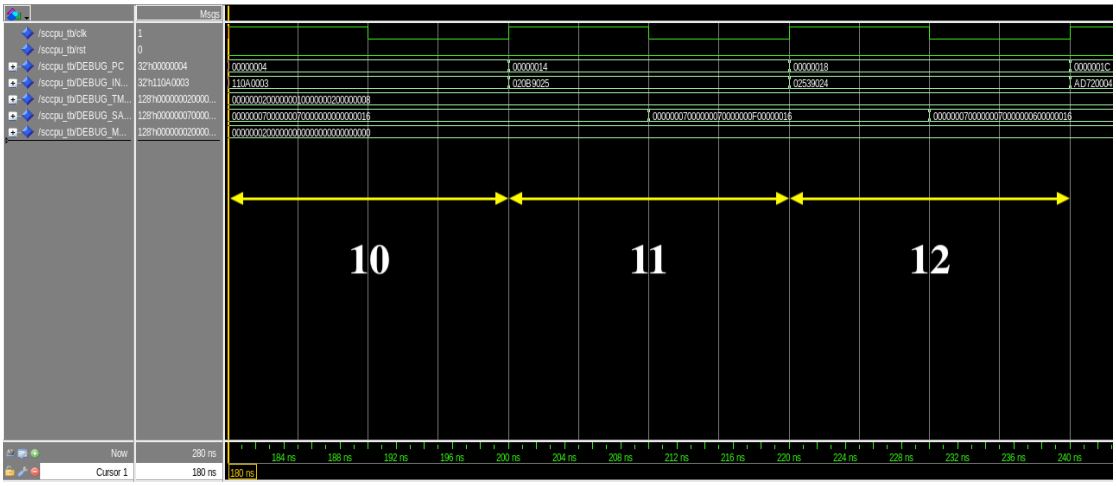

Figure 4: Simulation of single cycle CPU for cycle 10 to 12

The waveforms of cycle 10 to 12 are shown in Fig. 4.

In the cycle 10, we compare the value of $t0 and $t2. It can be seen that they are both equal to 0x00000002. Then in the next cycle, 11, we jump to L2.

In the cycle 11, we assign the value of $s0 *or* $t3 to $s2. In this case, $s0=0x00000007, $t3=0x00000008. So in the second half of the cycle, the value of $s2 is 0x0000000F.

In the cycle 12, we assign the value of $s2 *and* $s3 to $s2. In this case, $s3=0x00000016, $s2=0x000000F. So in the second half of the cycle, the value of $s2 is 0x00000006.

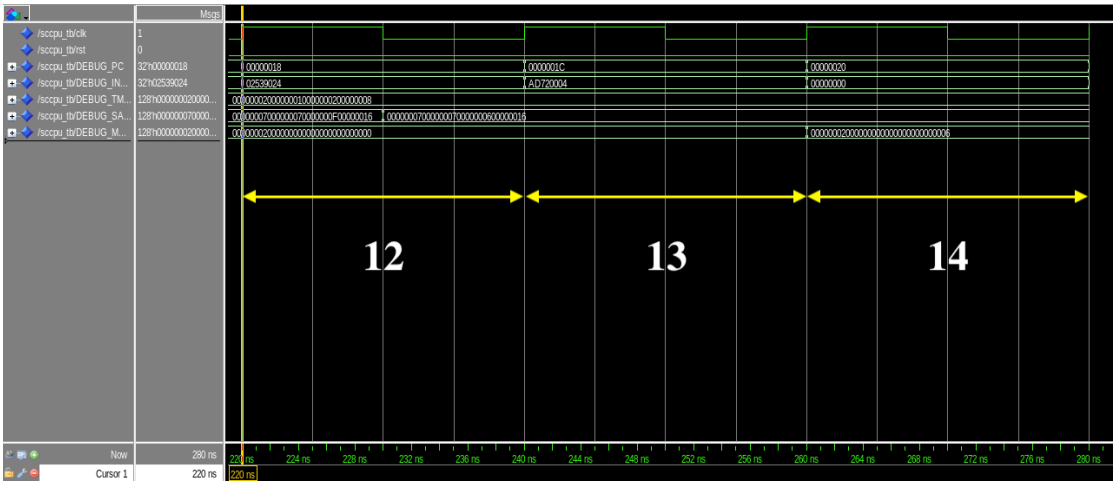

Figure 5: Simulation of single cycle CPU for cycle 12 to 14

The waveforms of cycle 12 to 14 are shown in Fig. 5.

In the cycle 13, we store the value of $s2 to the address of 4+$t3 in the memory. Now $t3=0x00000008, $s2=0x00000006, so at the end of cycle 13, it can be seen that DMEM(0xC)=0x00000006.

2 Answers to the Question

If DMEM(0x0)=15, then according to the waveforms of the output of DEBUG_MEM as shown in Fig. 6, then 0x00000004 will be written to the memory.

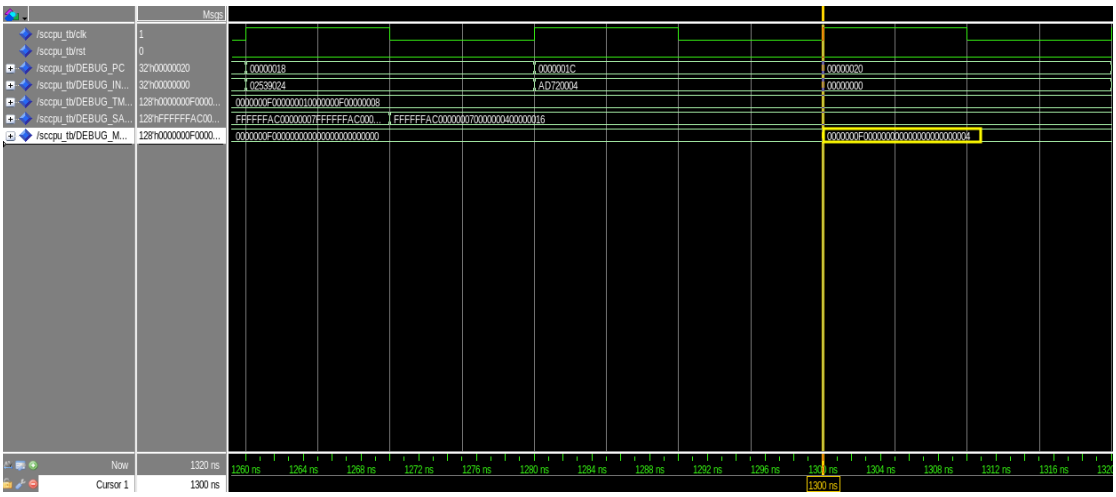

Figure 6: Simulation with the new value of DMEM(0)