

# Lab 5 - Part 4

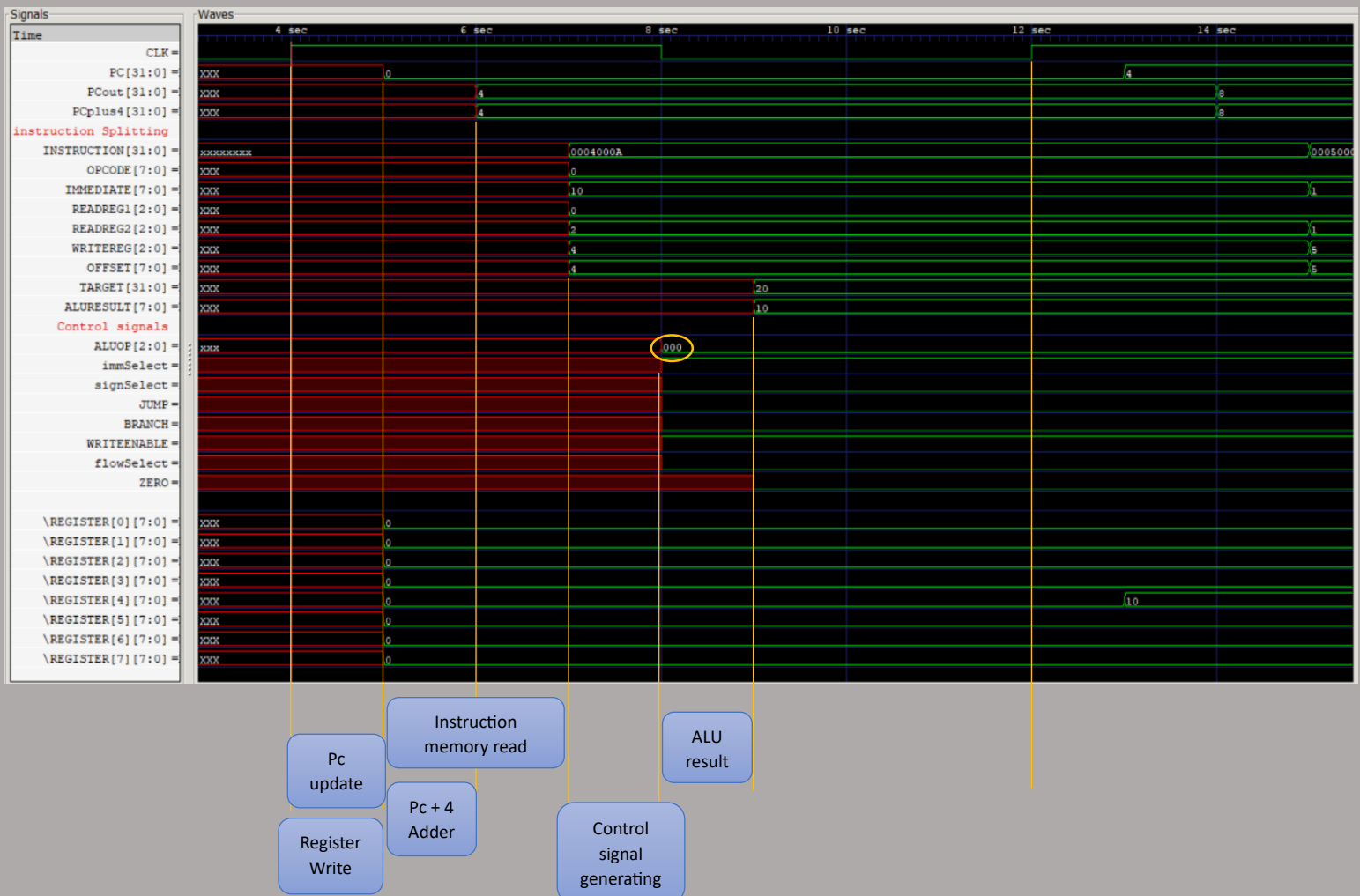
Group 49

## Timing Diagrams

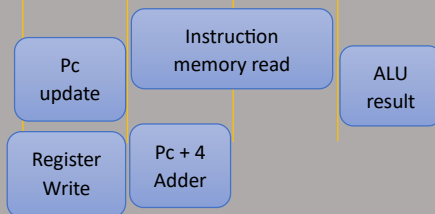
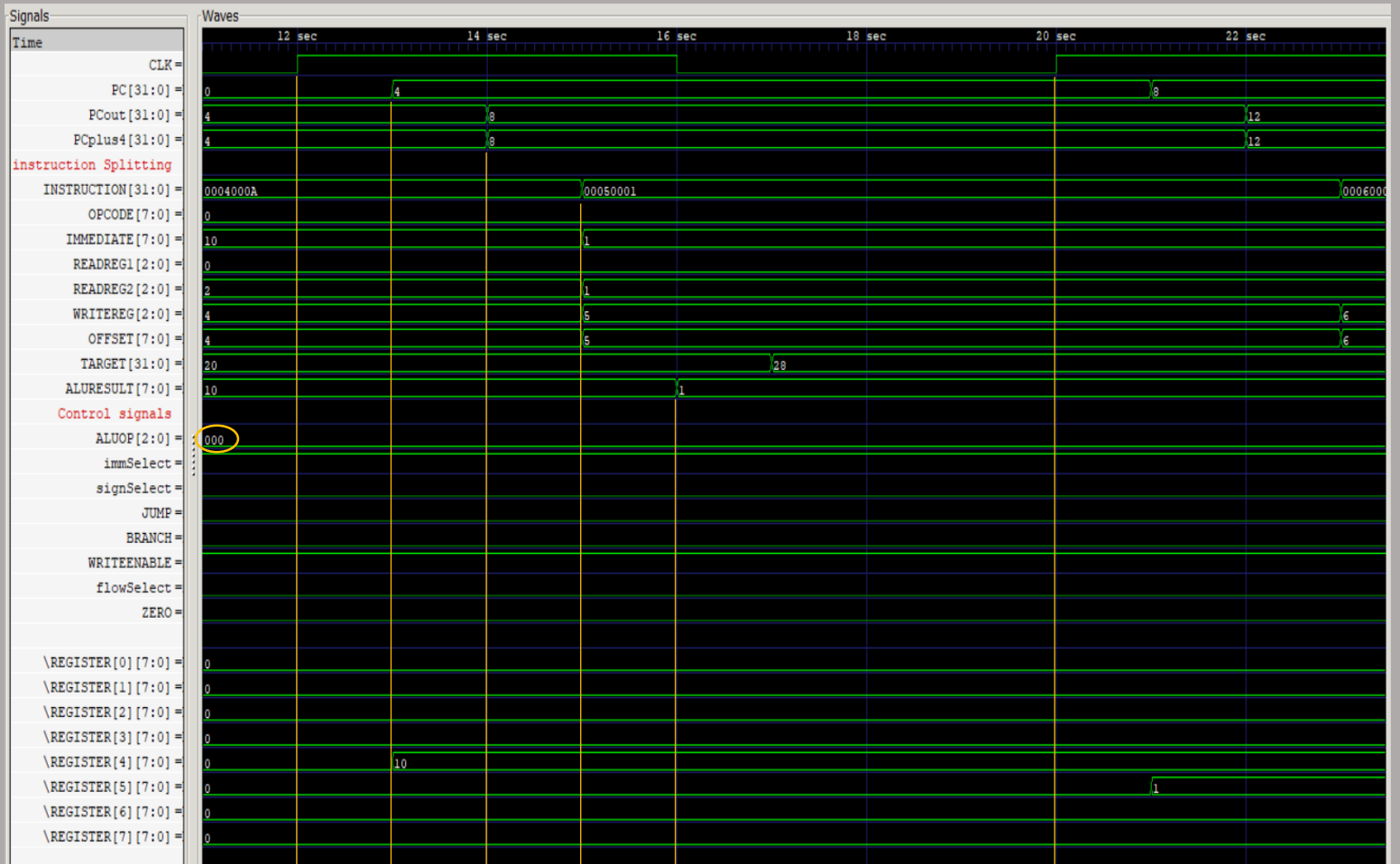
### Instructions

```
ASM my_sample_program.s
1 //This is a sample assembly program for C0224 Lab 5
2 loadi 4 0x0A
3 loadi 5 0x01
4 loadi 6 0x01
5 loadi 7 0x09
6 sub 4 4 5
7 beq 0x01 4 6
8 j 0xFD
9 add 1 4 7
10
```

#### 1. loadi 4 0x0A



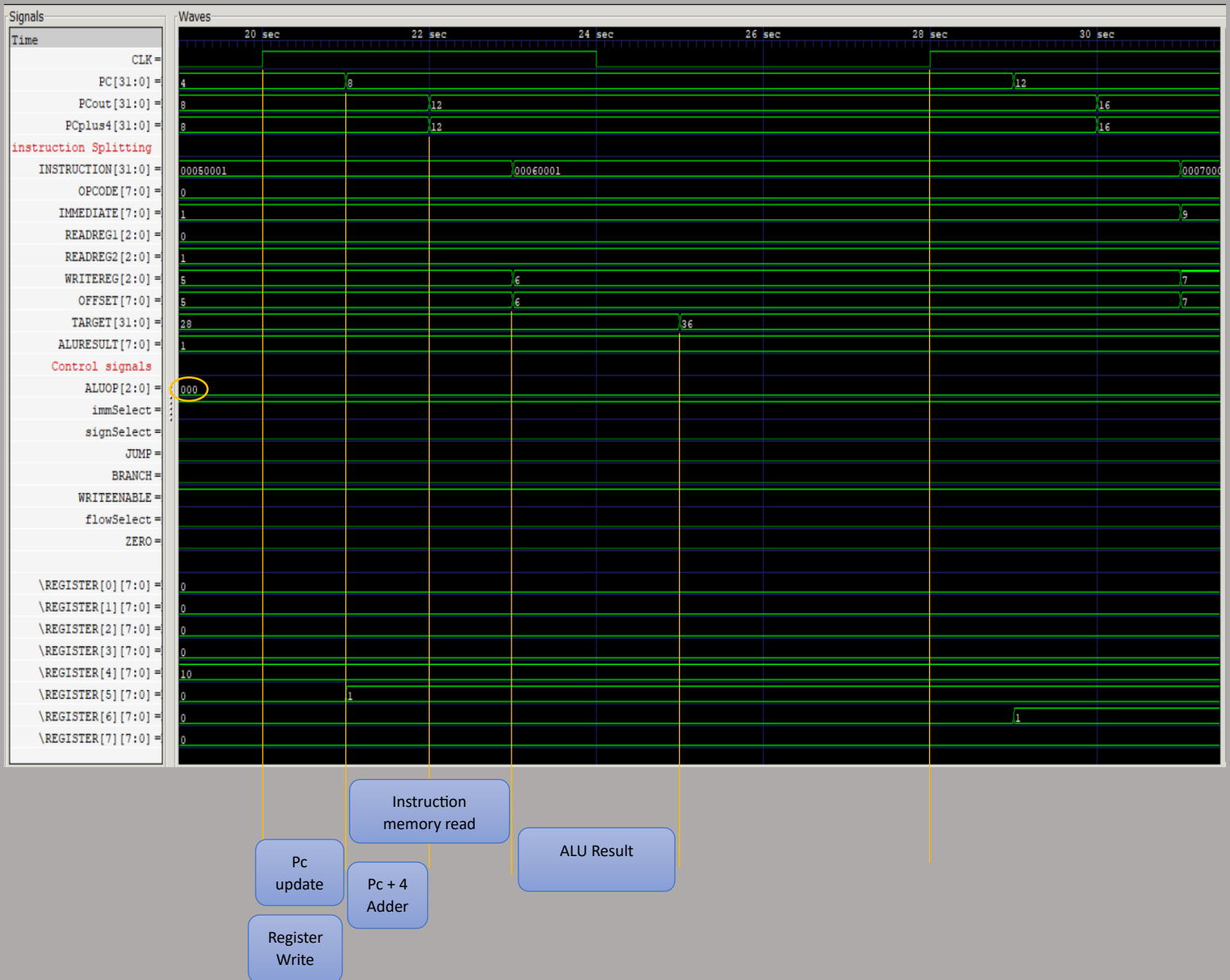
## 2. loadi 5 0x01



loadi:

PC Update	Instruction Memory Read		ALU	
#1	#2		#1	
	PC+4 Adder		Decode	
	#1		#1	
Register Write				
#1				

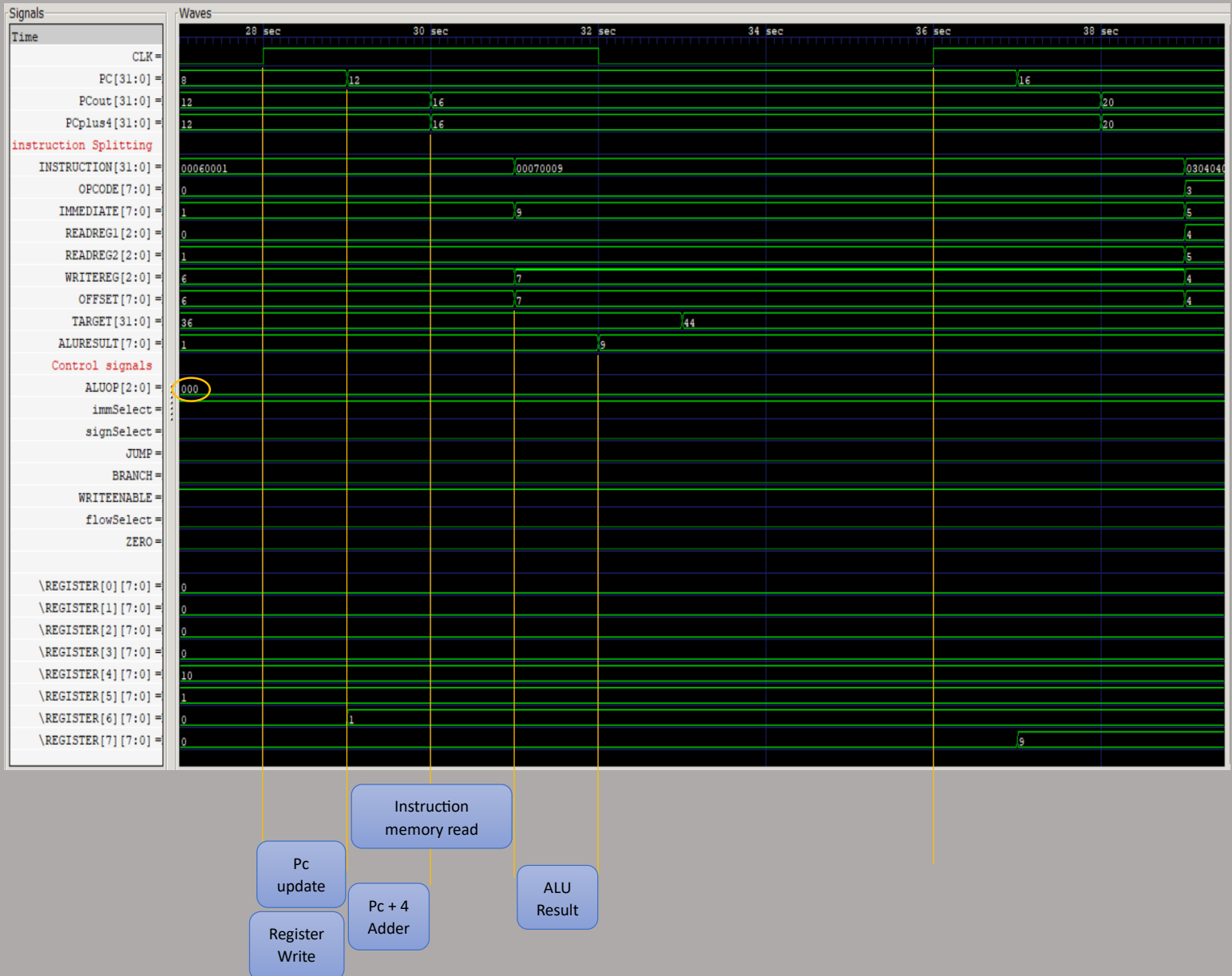
```
3.  loadi 6 0x01
```



loadi:

PC Update	Instruction Memory Read			ALU	
#1	#2			#1	
	PC+4 Adder		Decode		
	#1		#1		
Register Write					
#1					

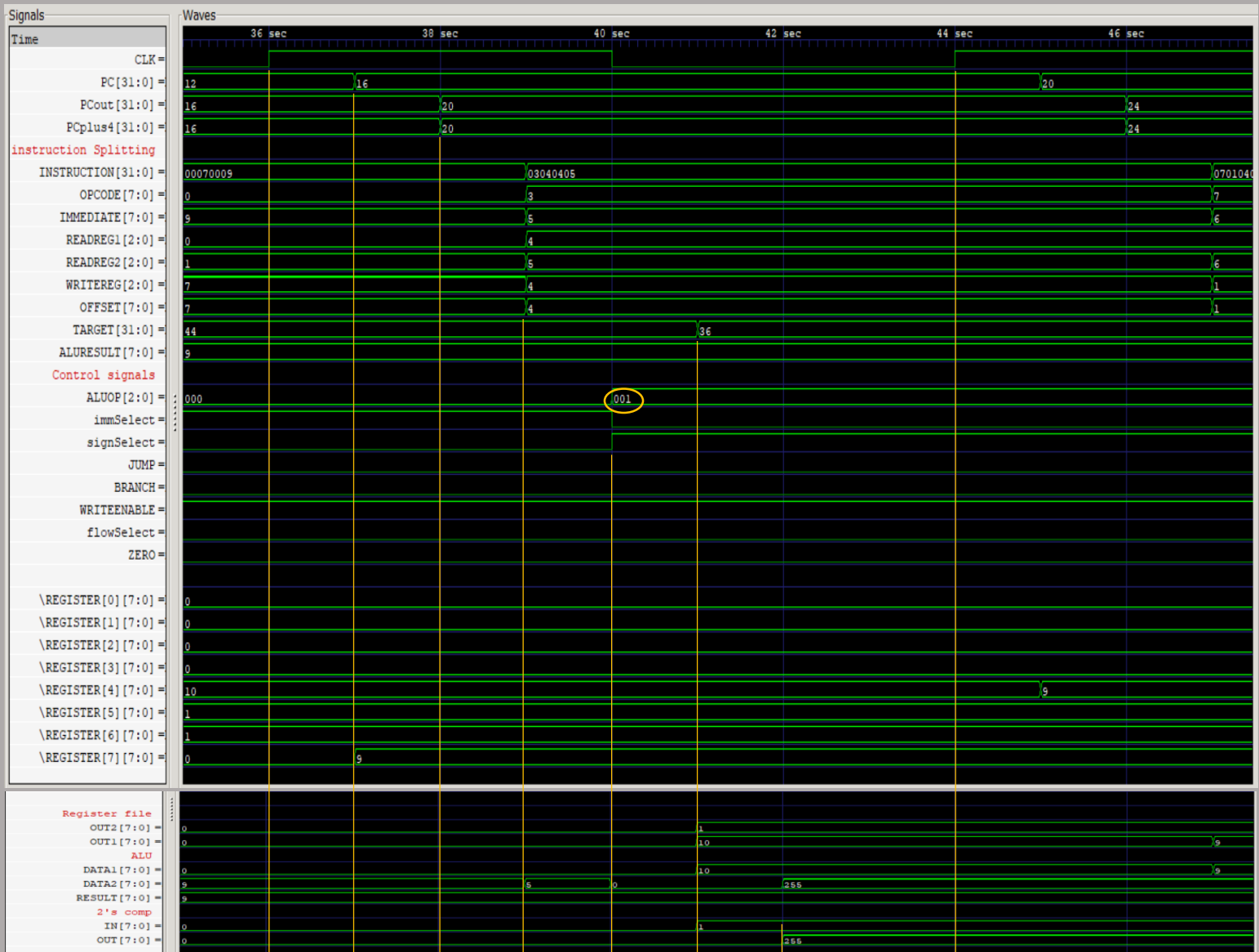
```
4.  loadi 7 0x09
```



loadi:

PC Update	Instruction Memory Read			ALU	
#1	#2			#1	
	PC+4 Adder		Decode		
	#1		#1		
Register Write					
#1					

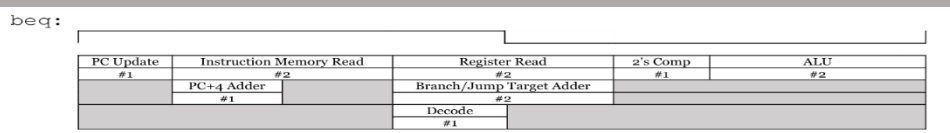
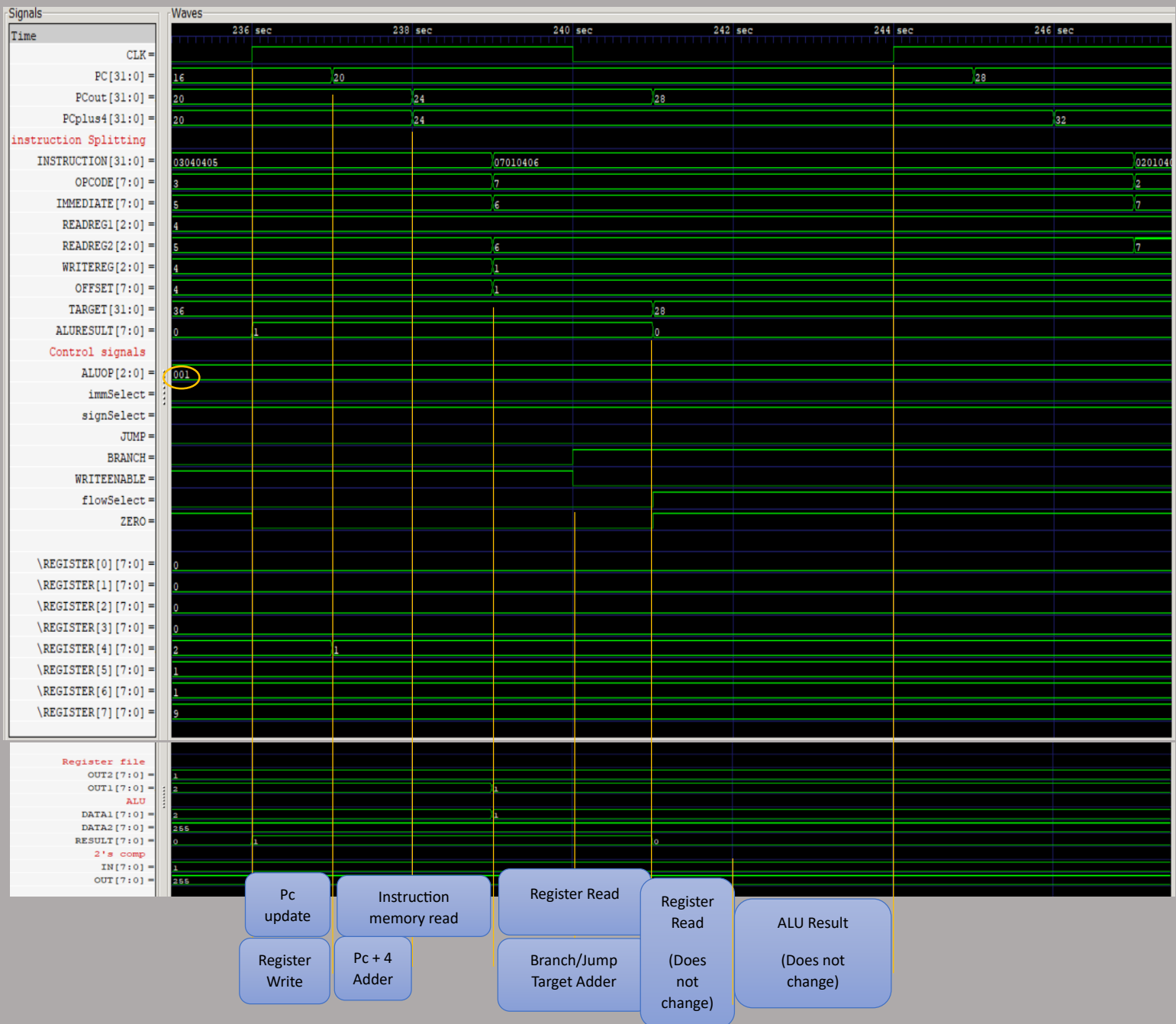
## 5. sub 4 4 5



sub:

sub:				
PC Update	Instruction Memory Read		Register Read	2's Comp
#1	#2		#2	#1
	PC+4 Adder		Decode	
	#1		#1	
Register Write				
#1				

6. beq 0x01 4 6

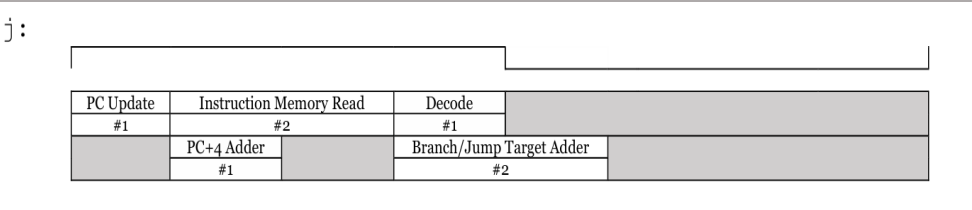
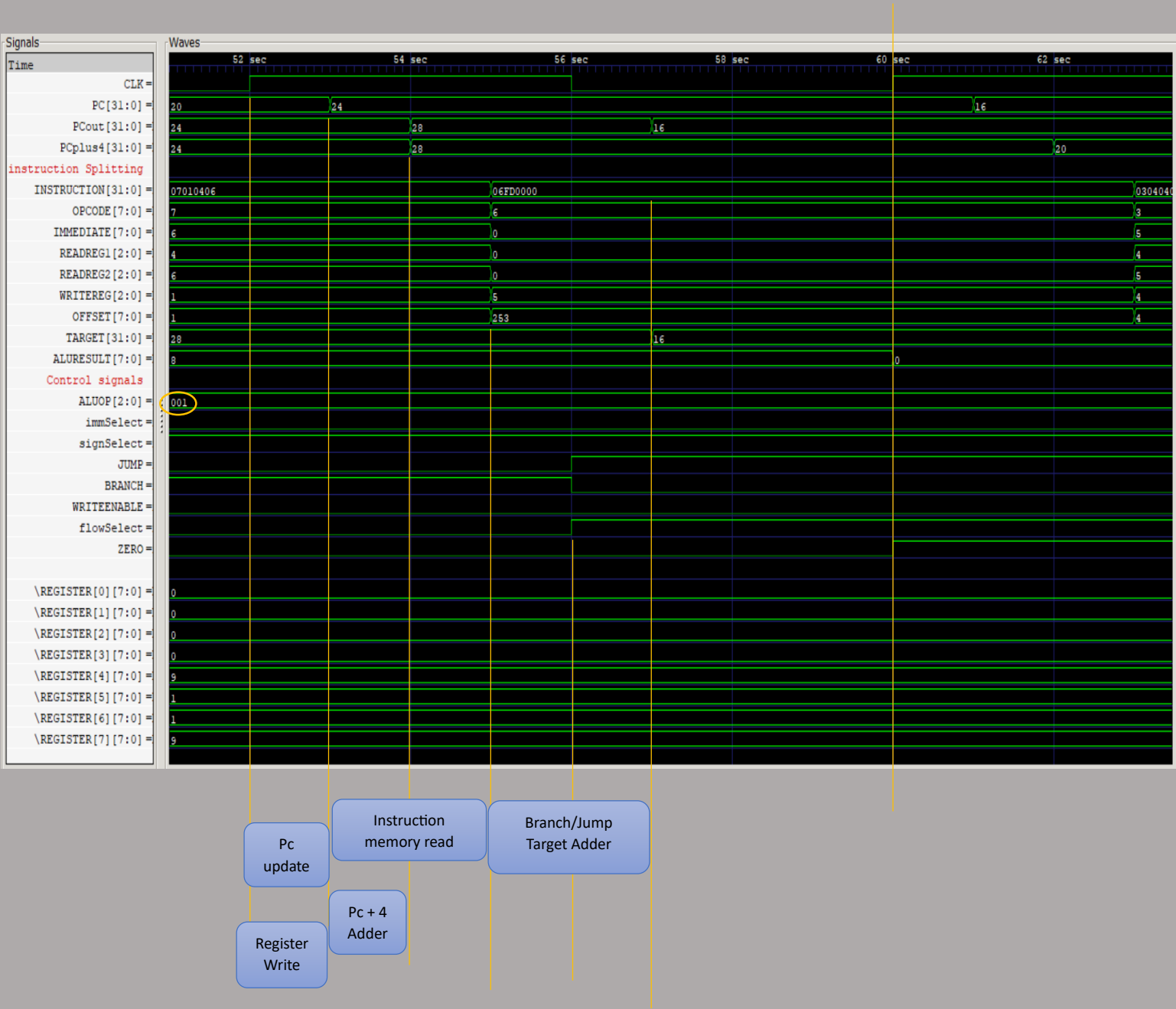


When the jump instruction executes, it jumps back three instructions and re-executes from there. During this process, the sub instruction is also re-executed repeatedly until the beq instruction condition is met. When the values in register 4 and register 6 are equal, the beq instruction is executed, which causes the subsequent jump instruction to be skipped. This breaks the loop, allowing the next instruction, which is the add instruction, to execute.

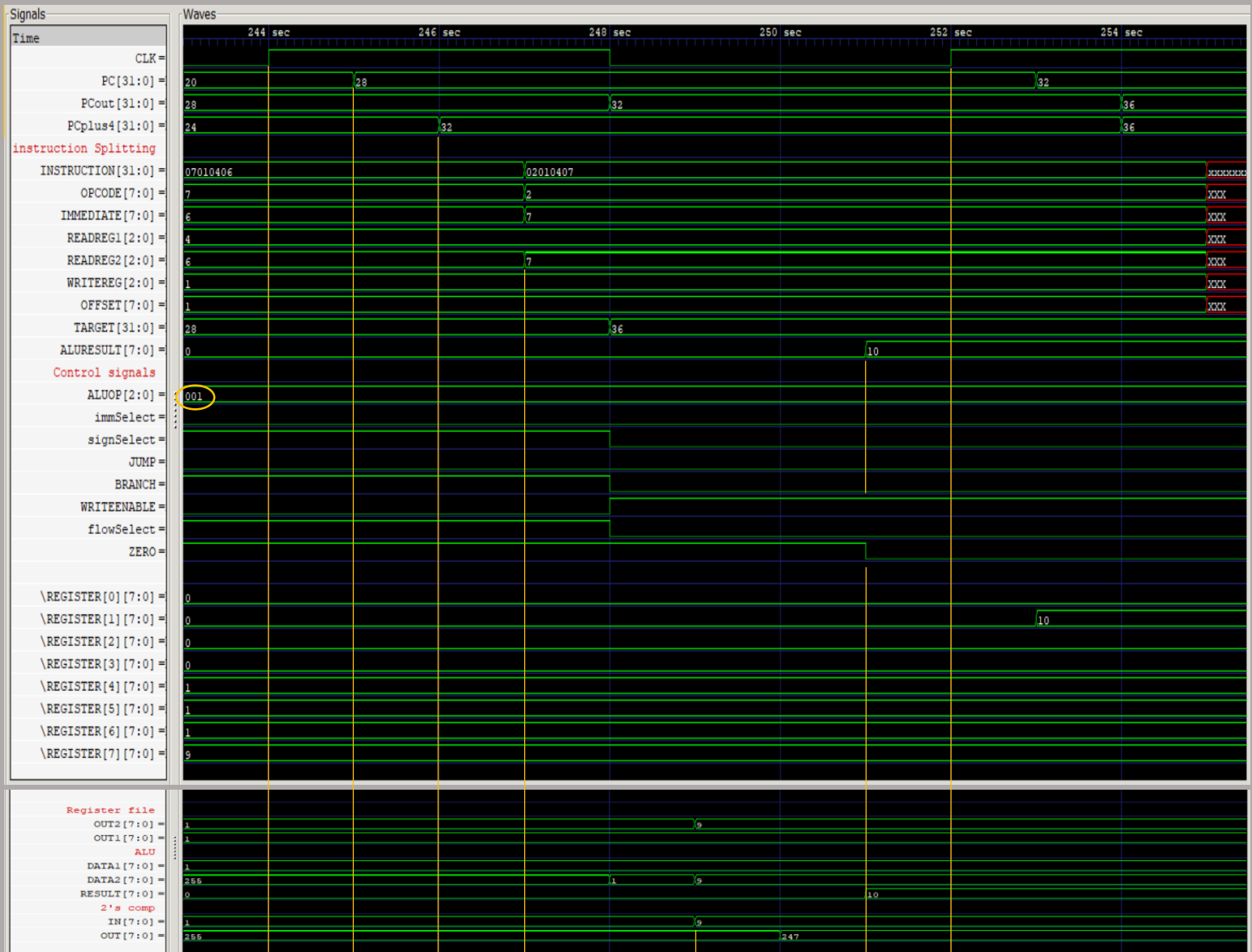
Change of register content Starting from Time #5

time	reg0	reg1	reg2	reg3	reg4	reg5	reg6	reg7
5	0	0	0	0	0	0	0	0
13	0	0	0	0	10	0	0	0
21	0	0	0	0	10	1	0	0
29	0	0	0	0	10	1	1	0
37	0	0	0	0	10	1	1	9
45	0	0	0	0	9	1	1	9
69	0	0	0	0	8	1	1	9
93	0	0	0	0	7	1	1	9
117	0	0	0	0	6	1	1	9
141	0	0	0	0	5	1	1	9
165	0	0	0	0	4	1	1	9
189	0	0	0	0	3	1	1	9
213	0	0	0	0	2	1	1	9
237	0	0	0	0	1	1	1	9
253	0	10	0	0	1	1	1	9

7. j 0xFD



8. add 1 4 7



add:

PC Update #1	Instruction Memory Read #2	Register Read #2	ALU #2	
	PC+4 Adder #1	Decode #1		
Register Write #1				