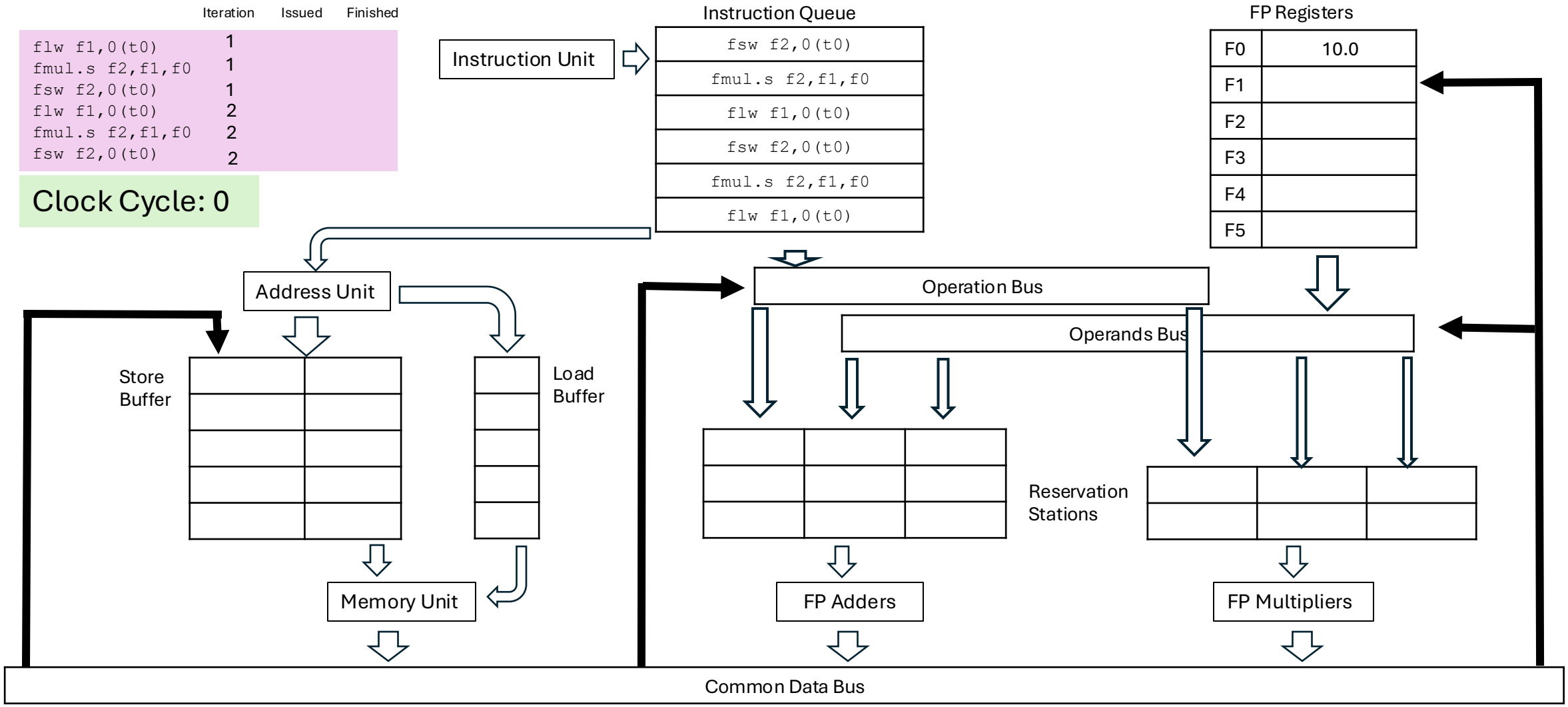
We will run 2 iterations (loop unrolling) of the following program:

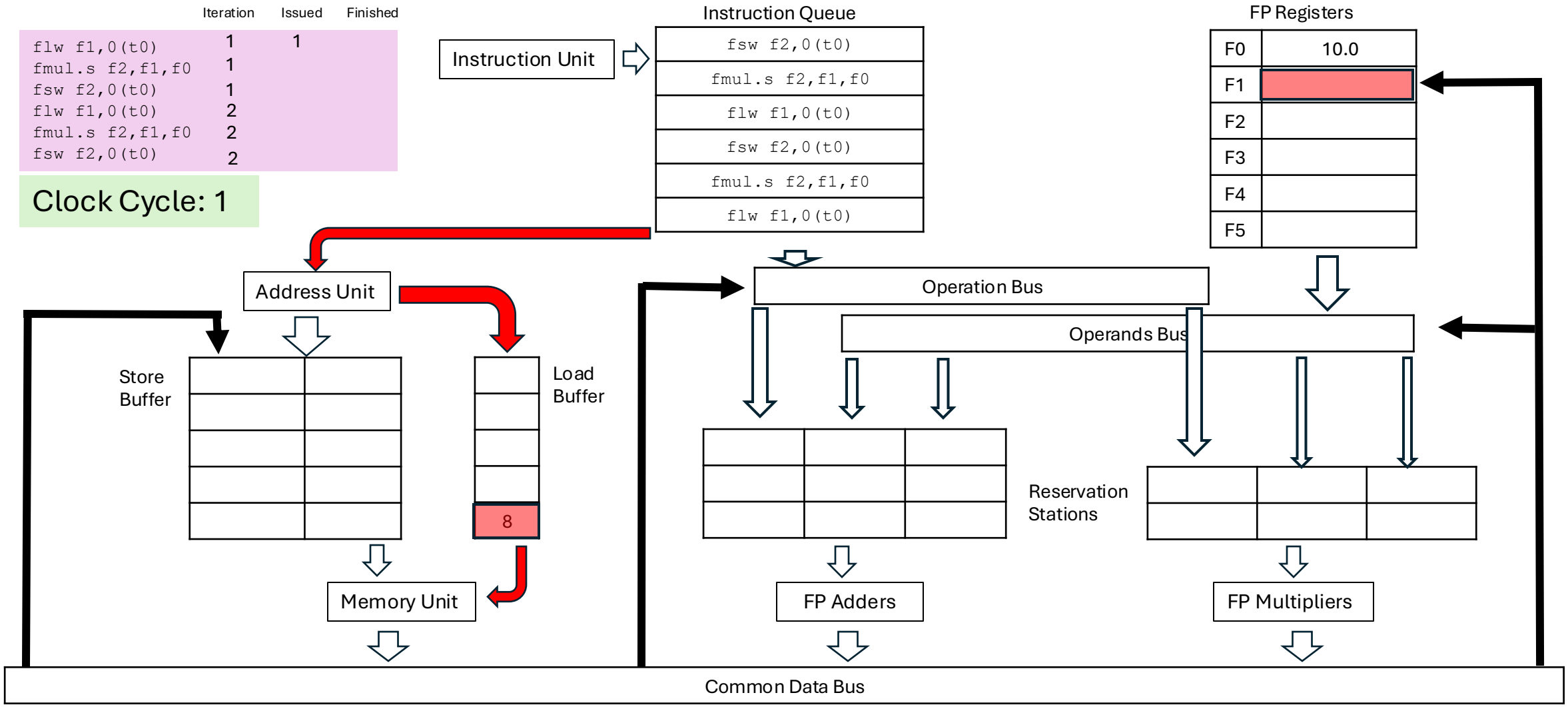
```
loop:   # multiply a vector at Mem[t0] by a scalar in f0
  flw f1, 0(t0)   #load a vector element at mem[t0]
  fmul.s f2, f1, f0 #multiply by f0
  fsw f2, 0(t0)   #store it back at mem[t0]
  addi t0, t0, -4 #point t0 to next element
  bnez t0, loop   # branch if t0 not zero
```

- `flw`, `fsw` will take 2 clock cycles.
- `fmul` will take 4 clock cycles.

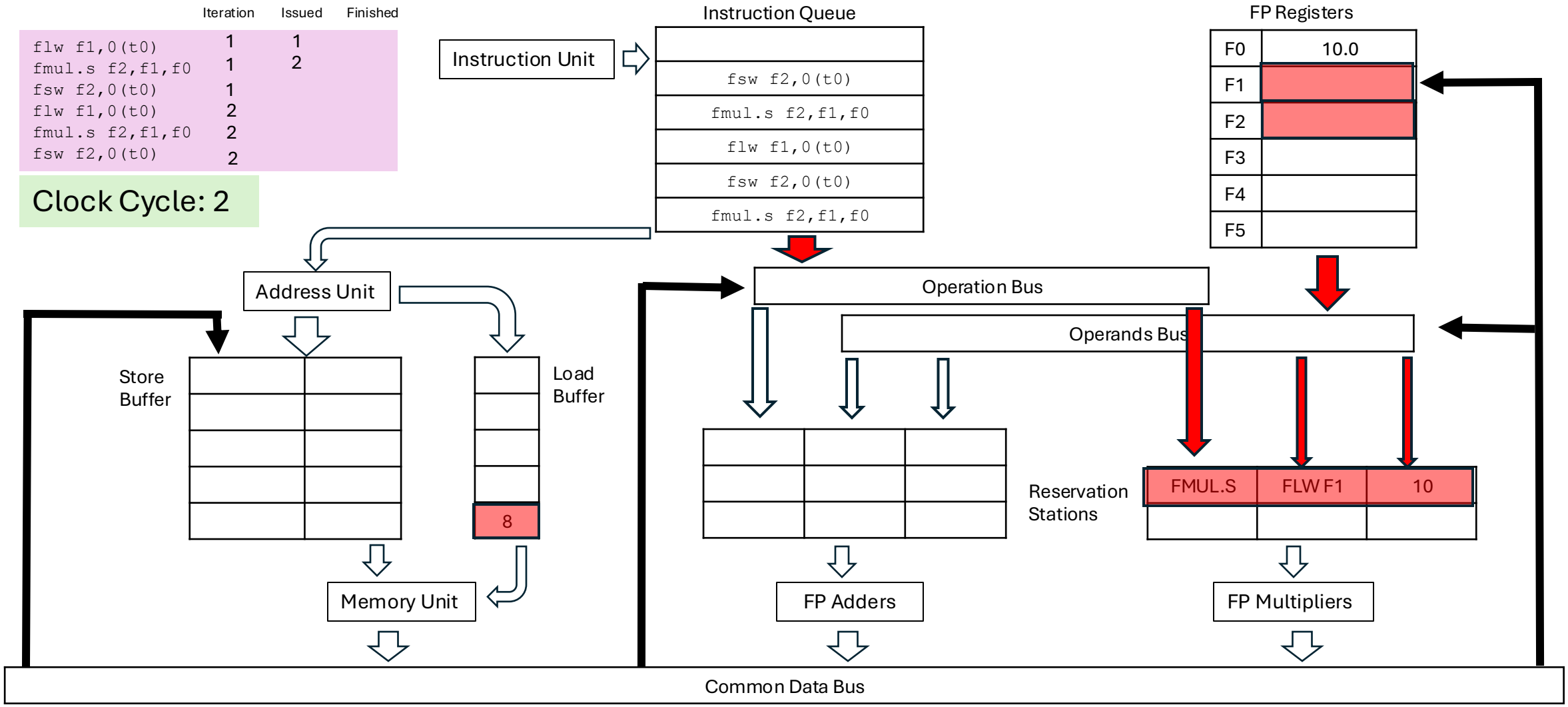
- Integer instructions are pipelined differently, only FP instructions are shown.
- The initial vector loads miss cache and have to wait 5 cycles. Once the vector is in cache, all loads and stores are cache hits.



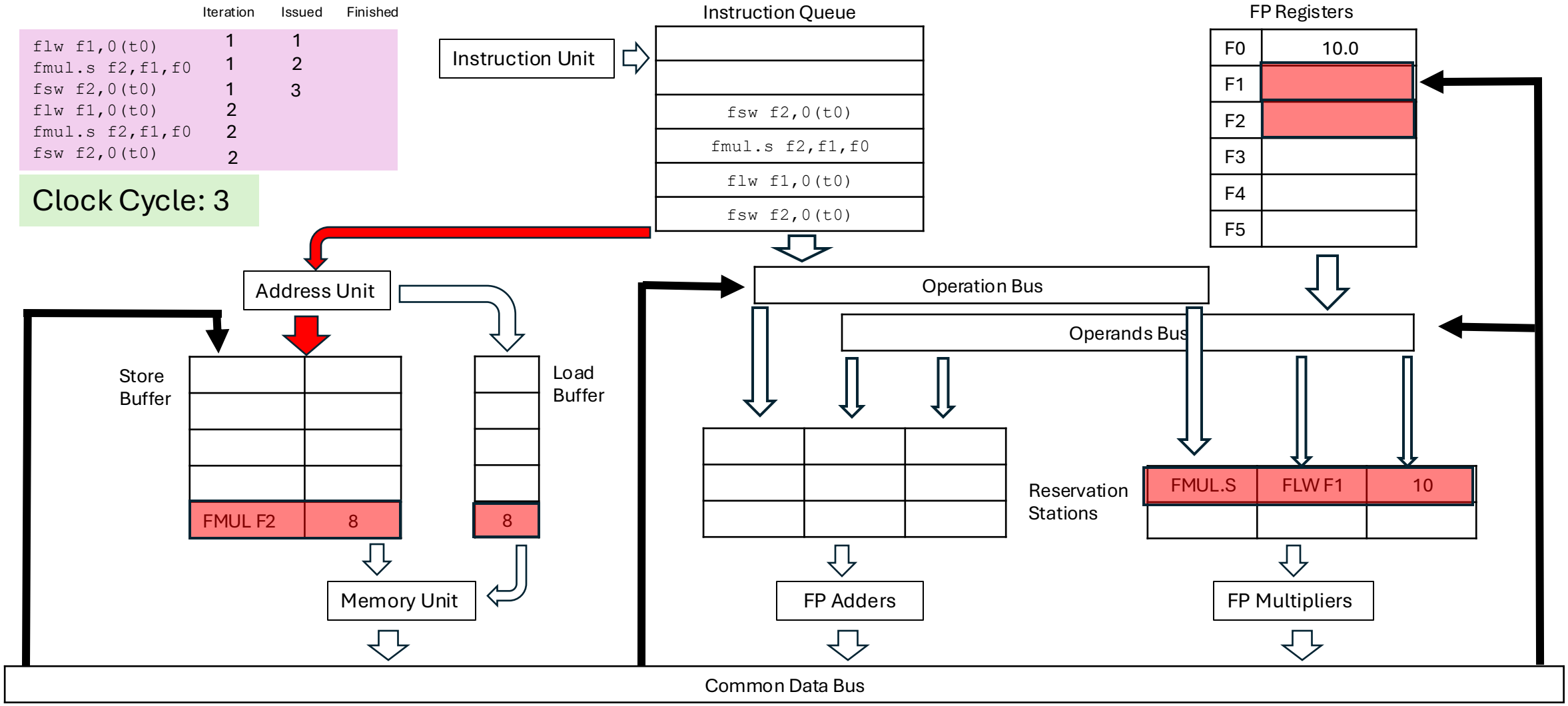
At clock cycle 0, before we begin executing, things are as depicted. Integer registers t0=8 and f0=10. The vector[3] at t0 has values 3.0, 2.0, 1.0.



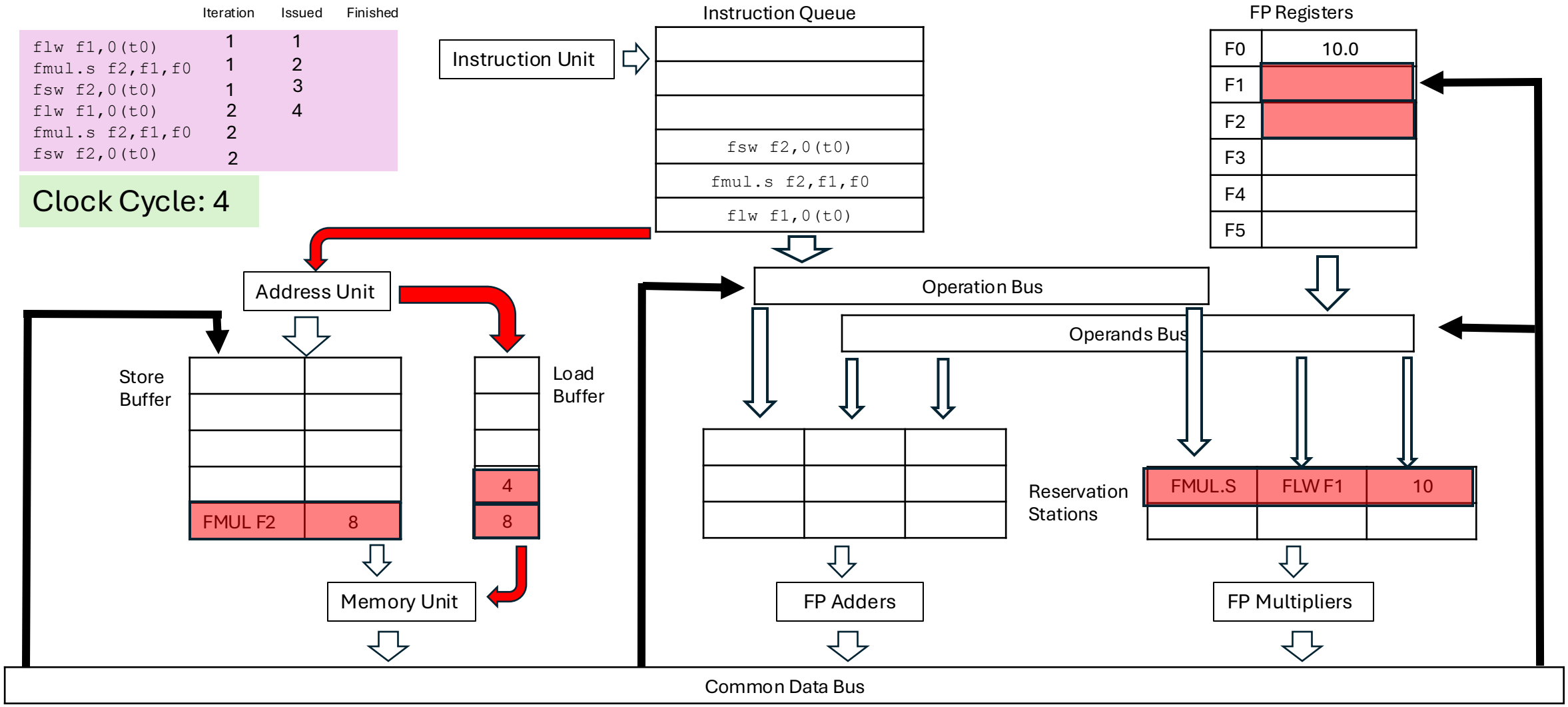
At clock cycle 1 the first load goes to the Memory Unit. We assume a **cache miss** for the whole vector, so all memory operations will have to wait 5 cycles for the vector to load. Destination register F1 is marked as busy waiting for the load.



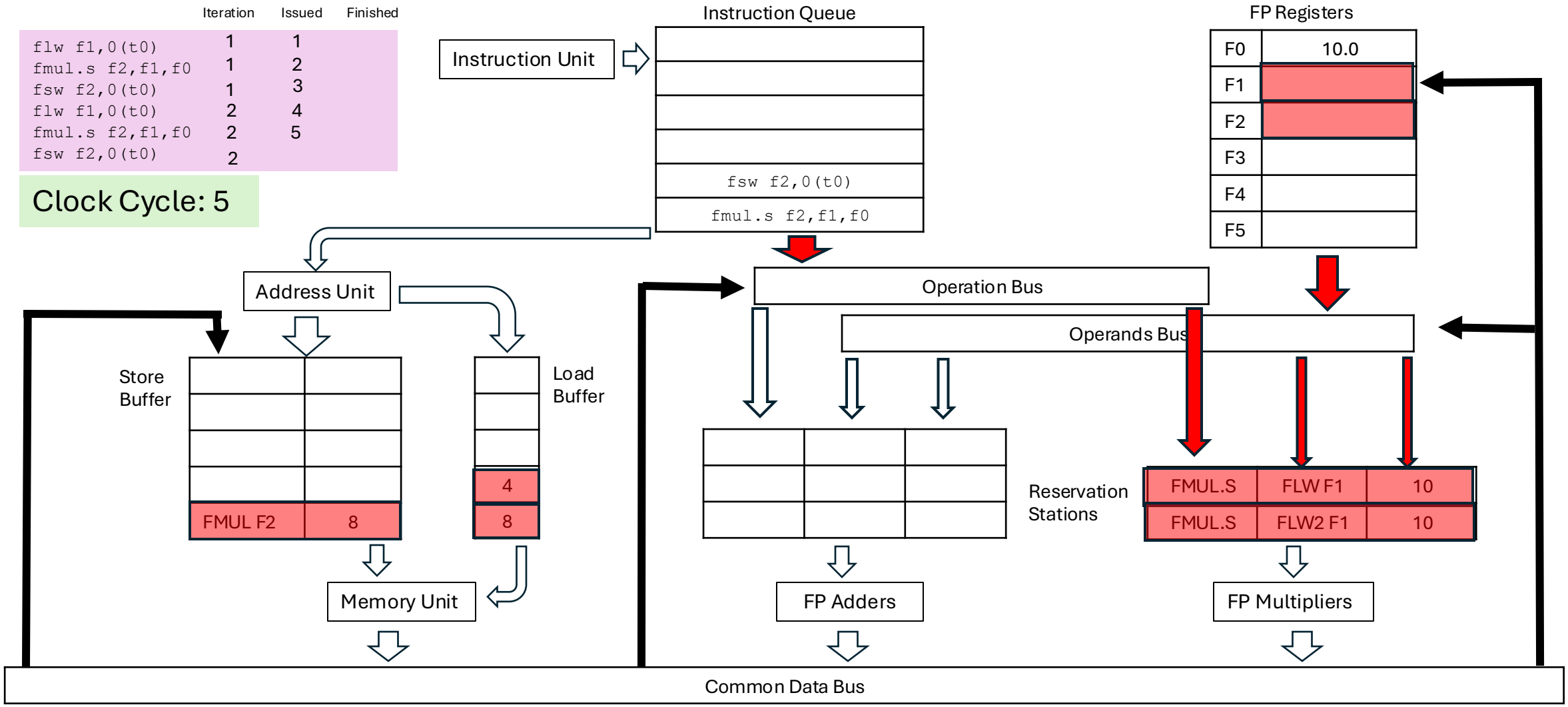
At clock cycle 2 the first FMUL goes to the Reservation Station. Register F1 is waiting for a memory load, so we mark the operand accordingly. Destination register F2 is marked as busy.



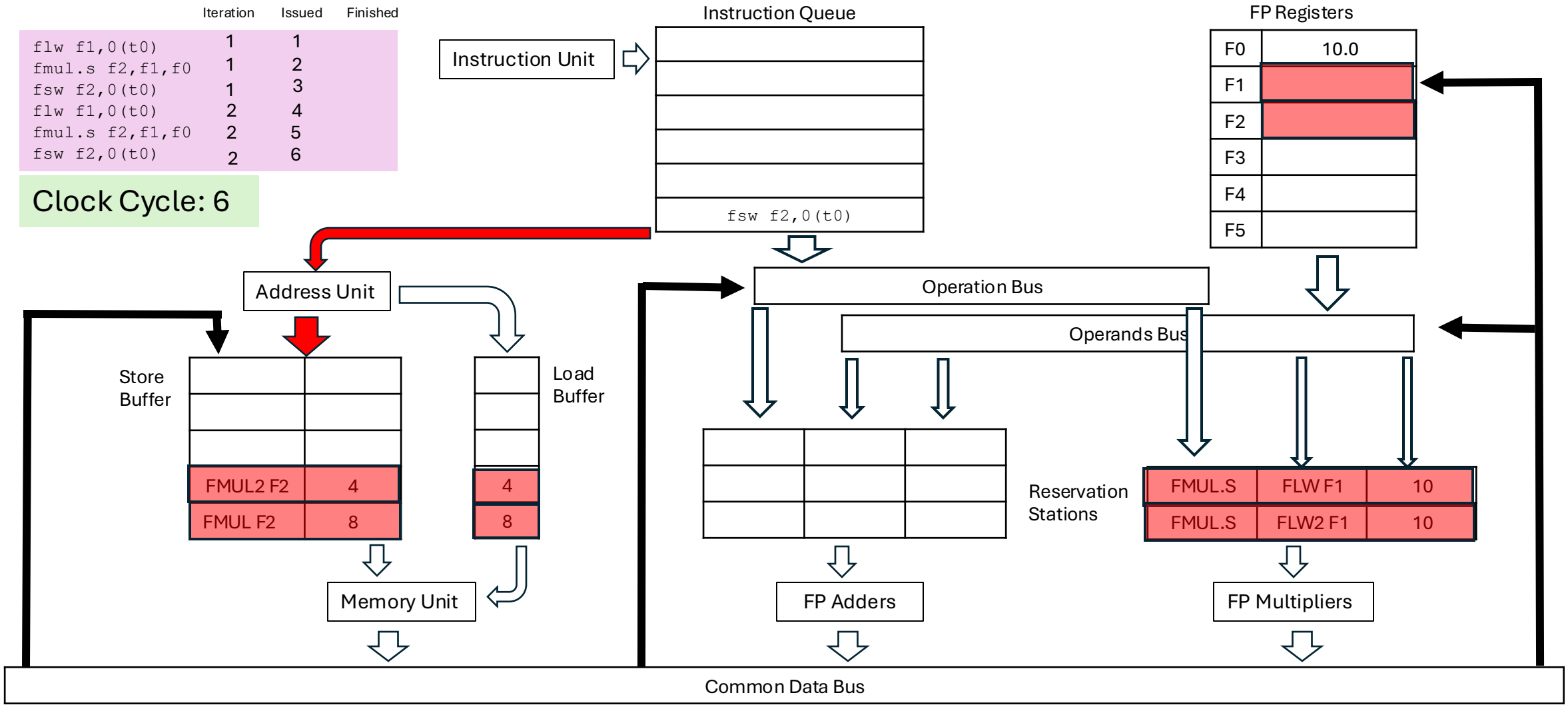
At clock cycle 3 the first memory store operation goes store buffer. At this time 2 things are detected: there is a previous load at the same address waiting, and the value of F2 is waiting for an FMUL operation. This operation will not complete until both conditions are cleared.



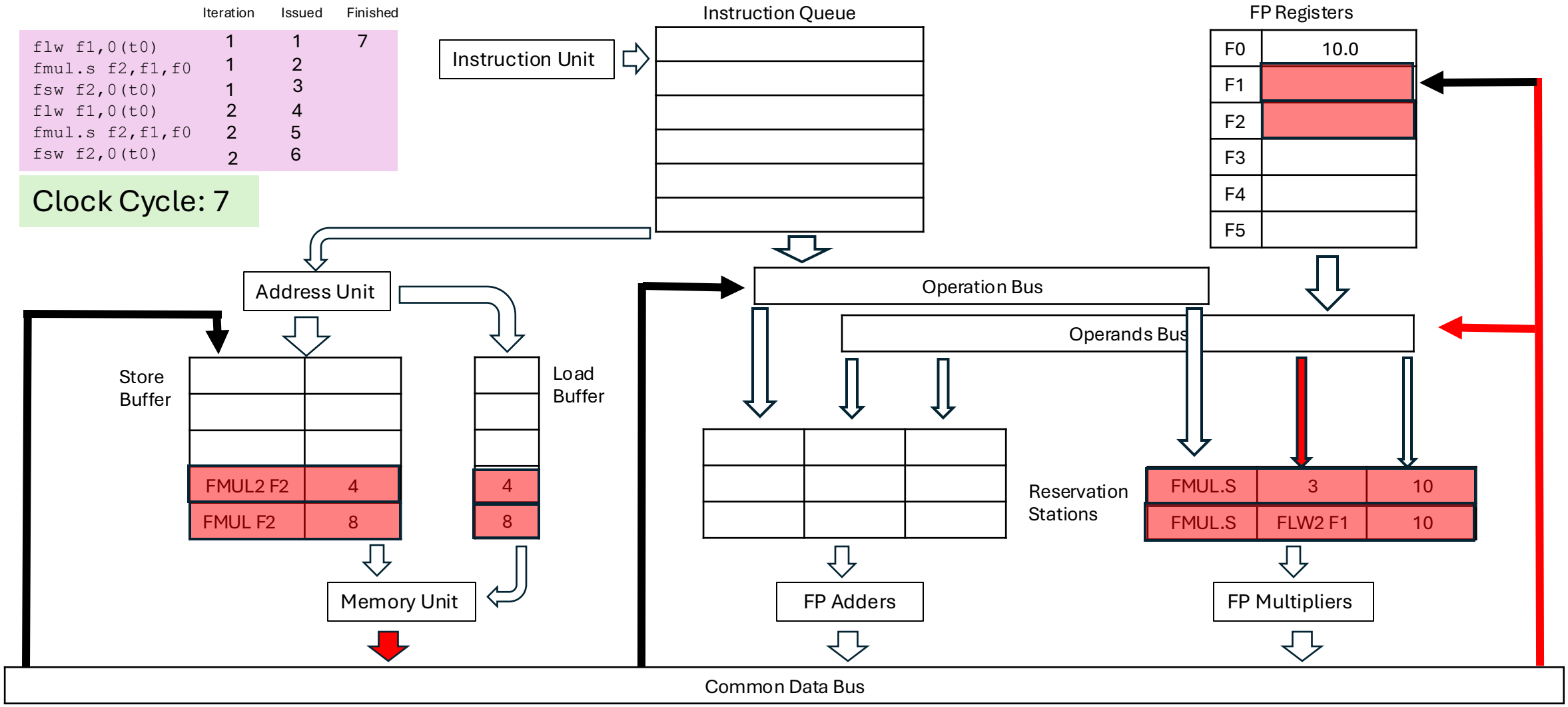
At clock cycle 4 the second memory load operation goes to the Load Buffer. The cache hasn't loaded the vector yet, so it stalls. Register F1 is remarked as waiting for this memory load.



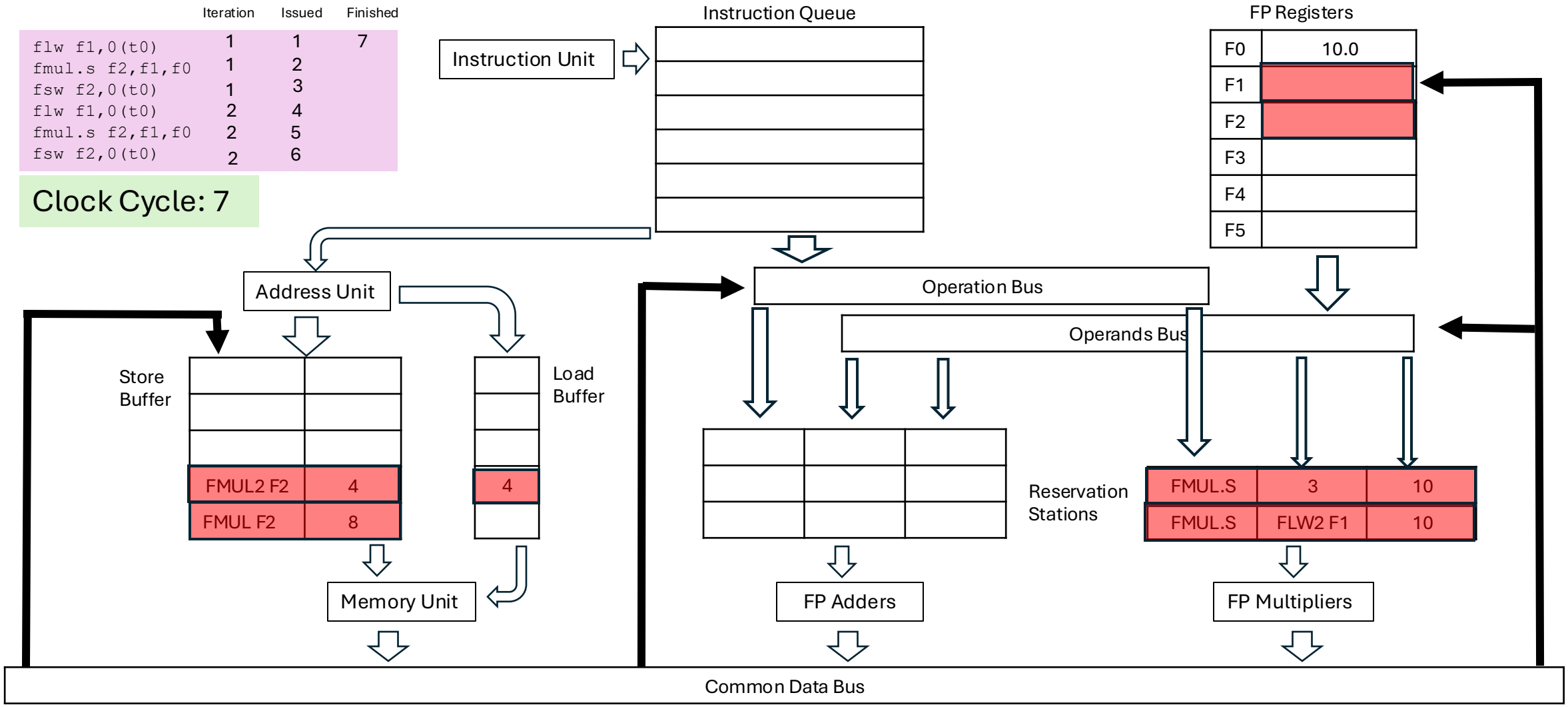
At clock cycle 5 the second multiplication operation goes to the Reservation Station. Register operand F1 is waiting for the second load operation, so it stalls. Register F2 is remarked as waiting for this second FMUL operation.



At clock cycle 6 the second memory store operation goes to the Store Buffer. Two conditions are observed: there is a pending load from the same memory position and source operand F2 is busy waiting for an FMUL (second multiplication) operation. Until both conditions are cleared, the store operation will stall.



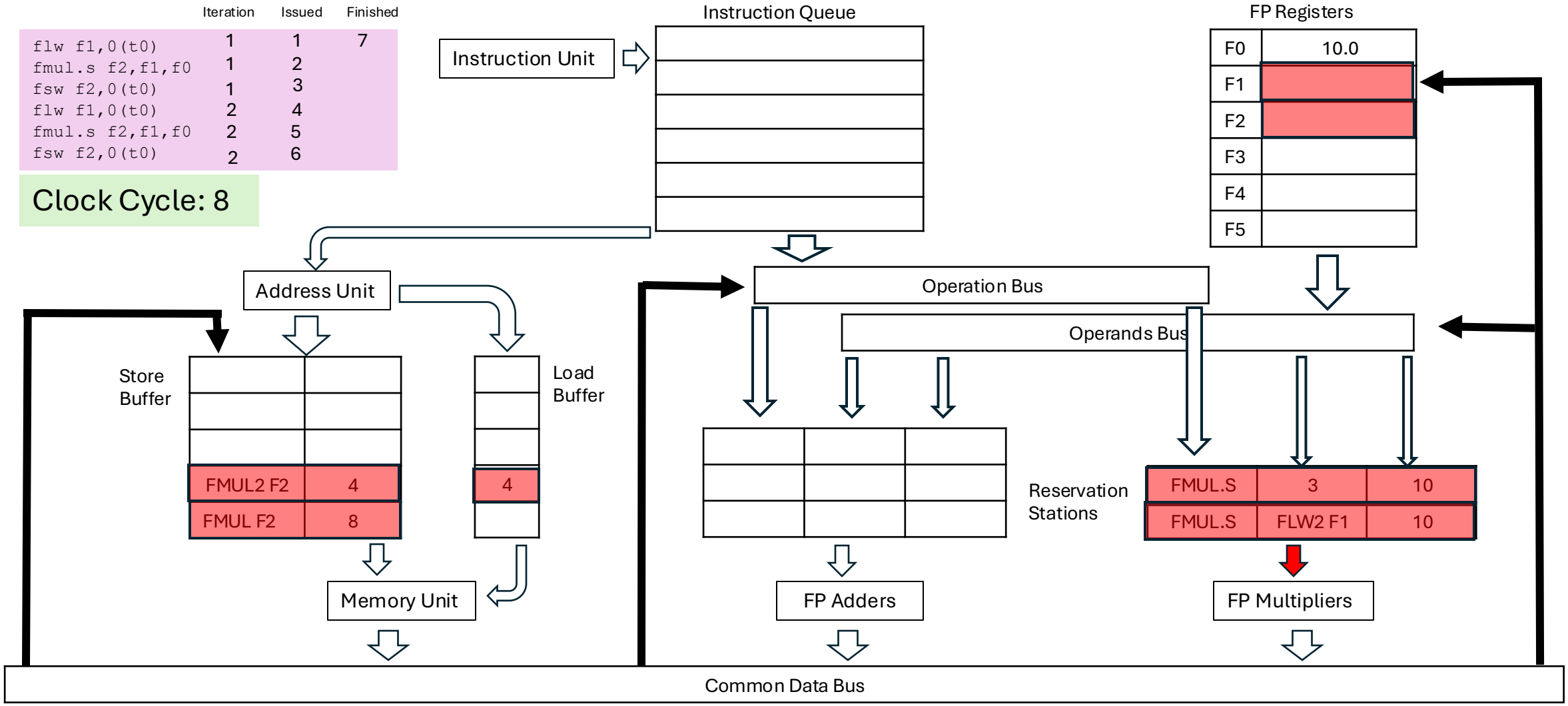
In clock cycle 7 the cache has loaded the vector. The first load from memory is complete and it propagates through the Common Data Bus to all elements waiting for it, like the first FMUL operation.



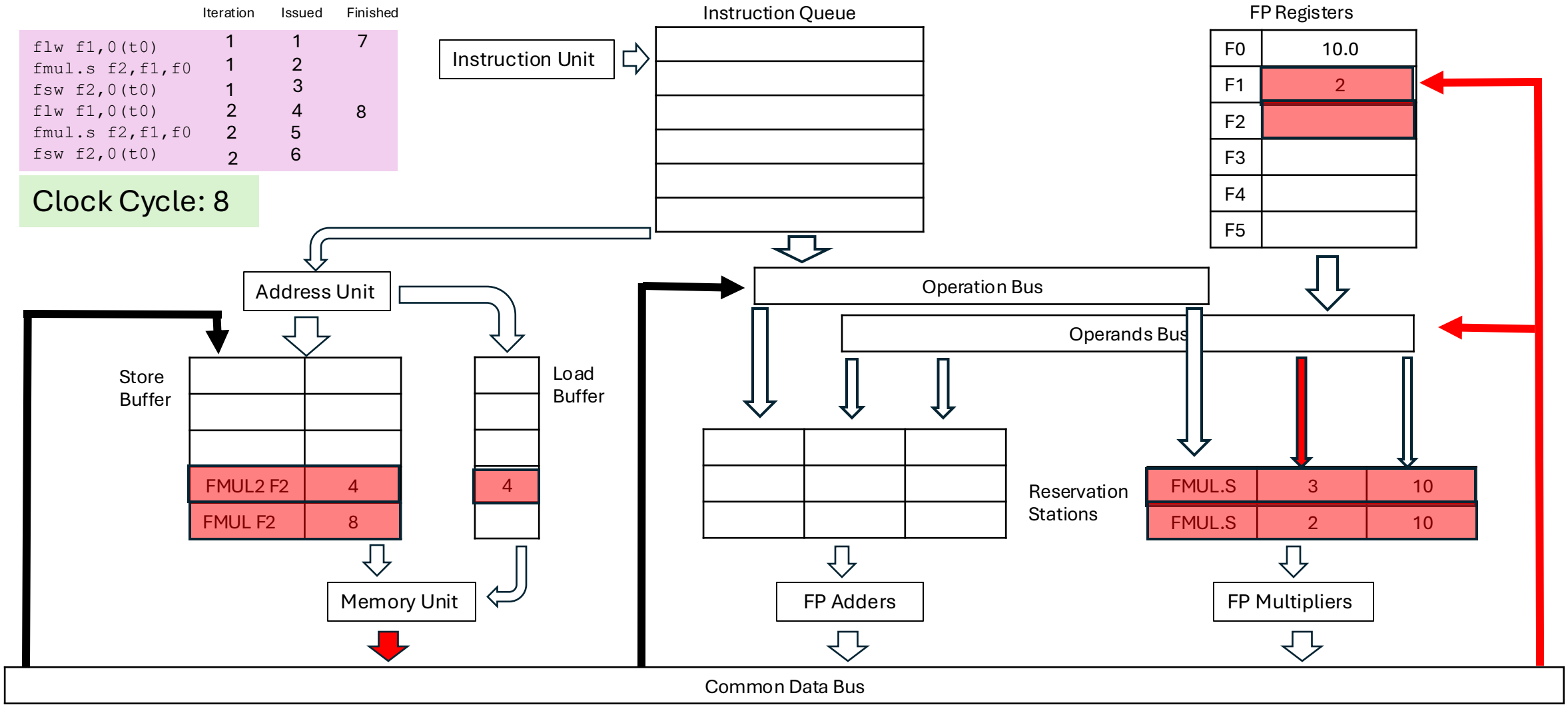
Still in clock cycle 7, since the first load is complete, we remove it from the Load Buffer.

	Iteration	Issued	Finished
`flw f1,0(t0)`	1	1	7
`fmul.s f2,f1,f0`	1	2	
`fsw f2,0(t0)`	1	3	
`flw f1,0(t0)`	2	4	
`fmul.s f2,f1,f0`	2	5	
`fsw f2,0(t0)`	2	6	

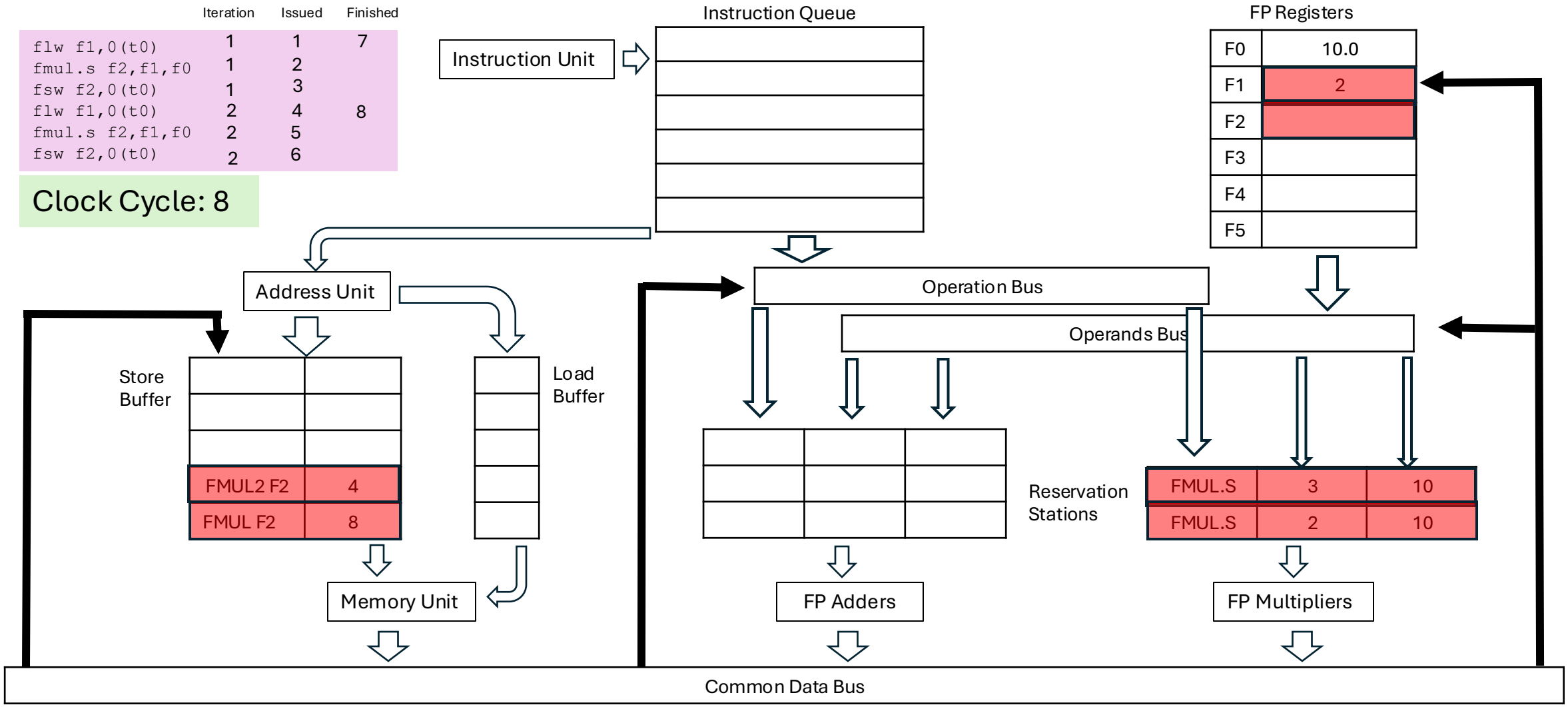
Clock Cycle: 8



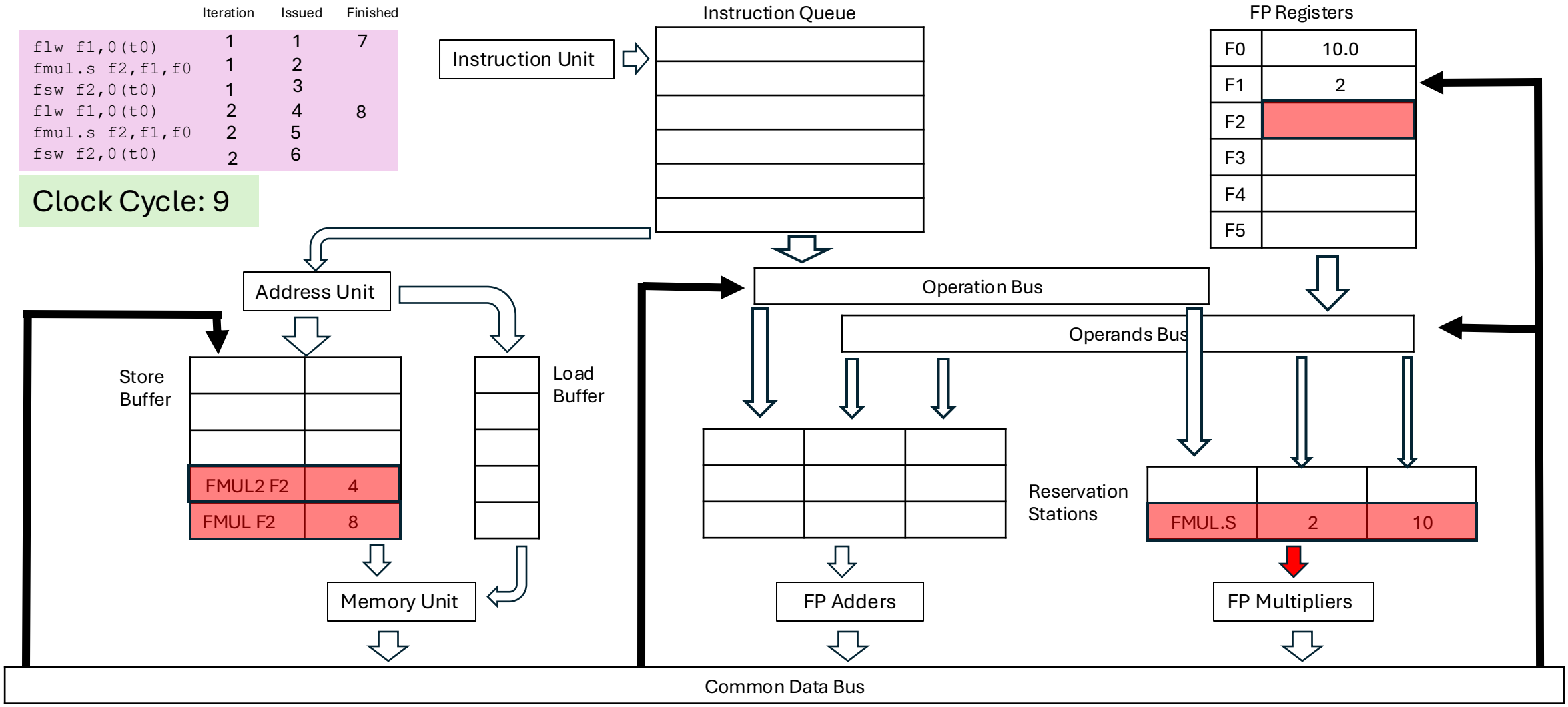
In clock cycle 8, The first FMUL goes to the FP Multipliers since it already has all operands.



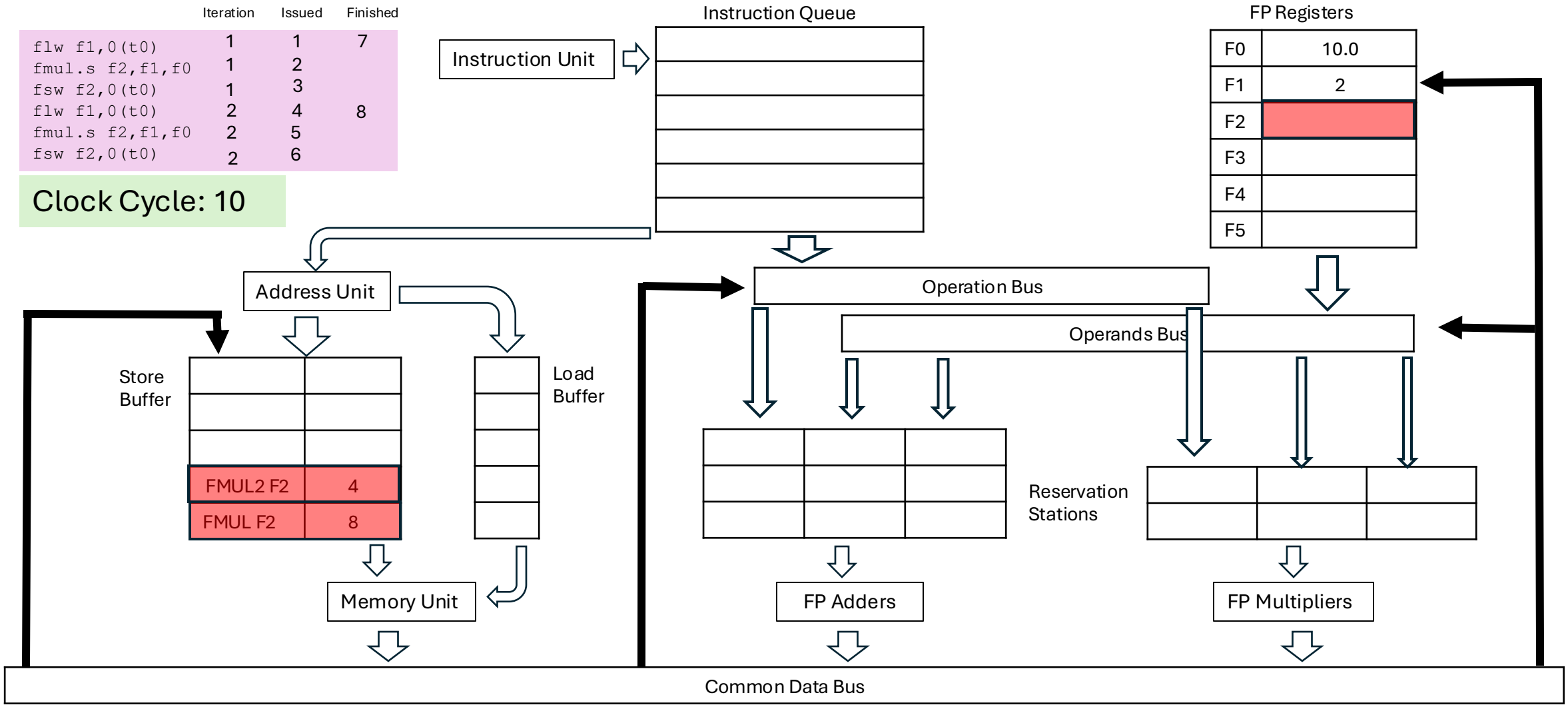
Still in clock cycle 8, the second load gets its result through the Common Data Bus. The value is copied to the all waiting Reservation Stations, like the one of the second FMUL. It also gets copied to the F1 register.



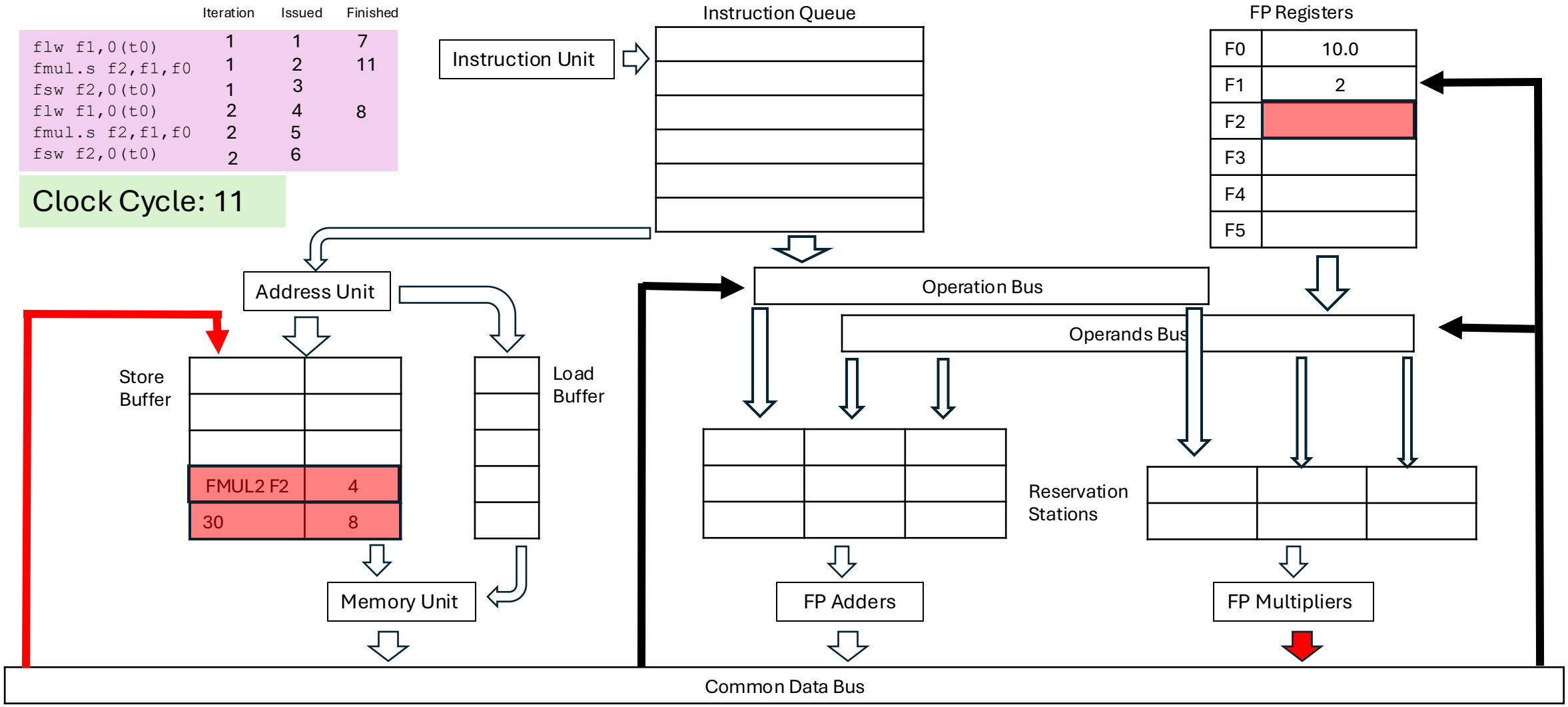
Still in clock cycle 8, the Load Buffer slot where the second load was, is marked as free.



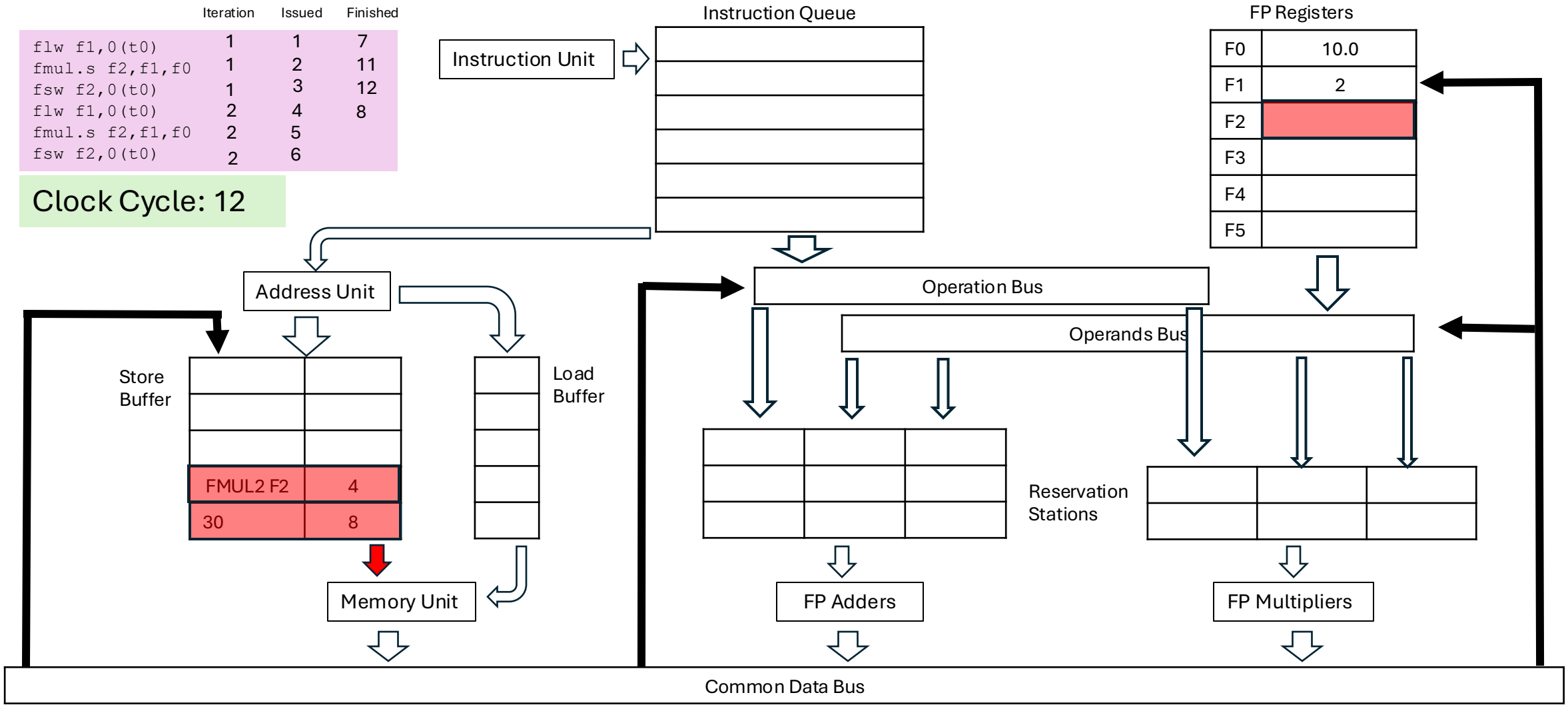
In clock cycle 9, the second FMUL operation goes to execute to the FP Multipliers. The Reservation Station of the first FMUL is marked as free. The F1 register is marked free.



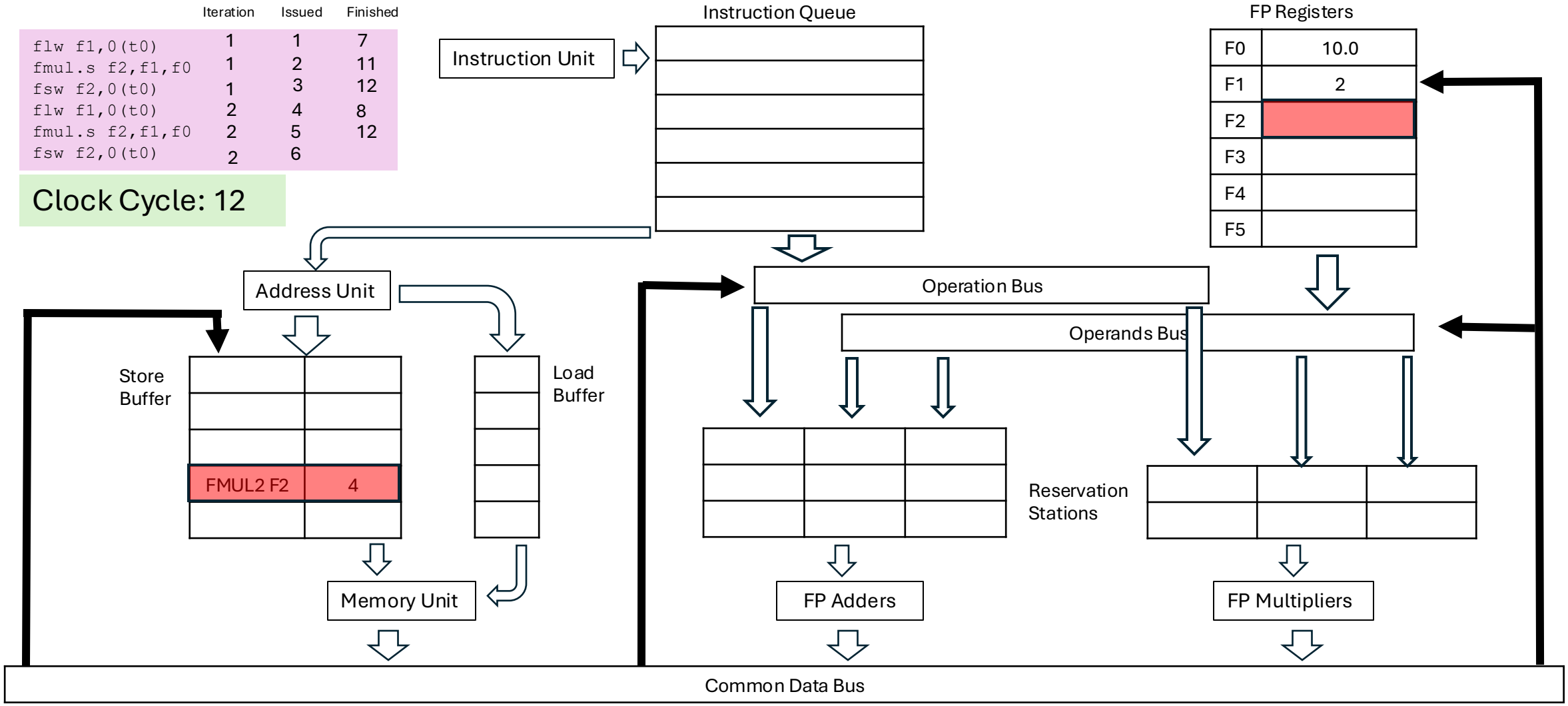
In cycle 10 the Reservation Station of the second FMUL is free.



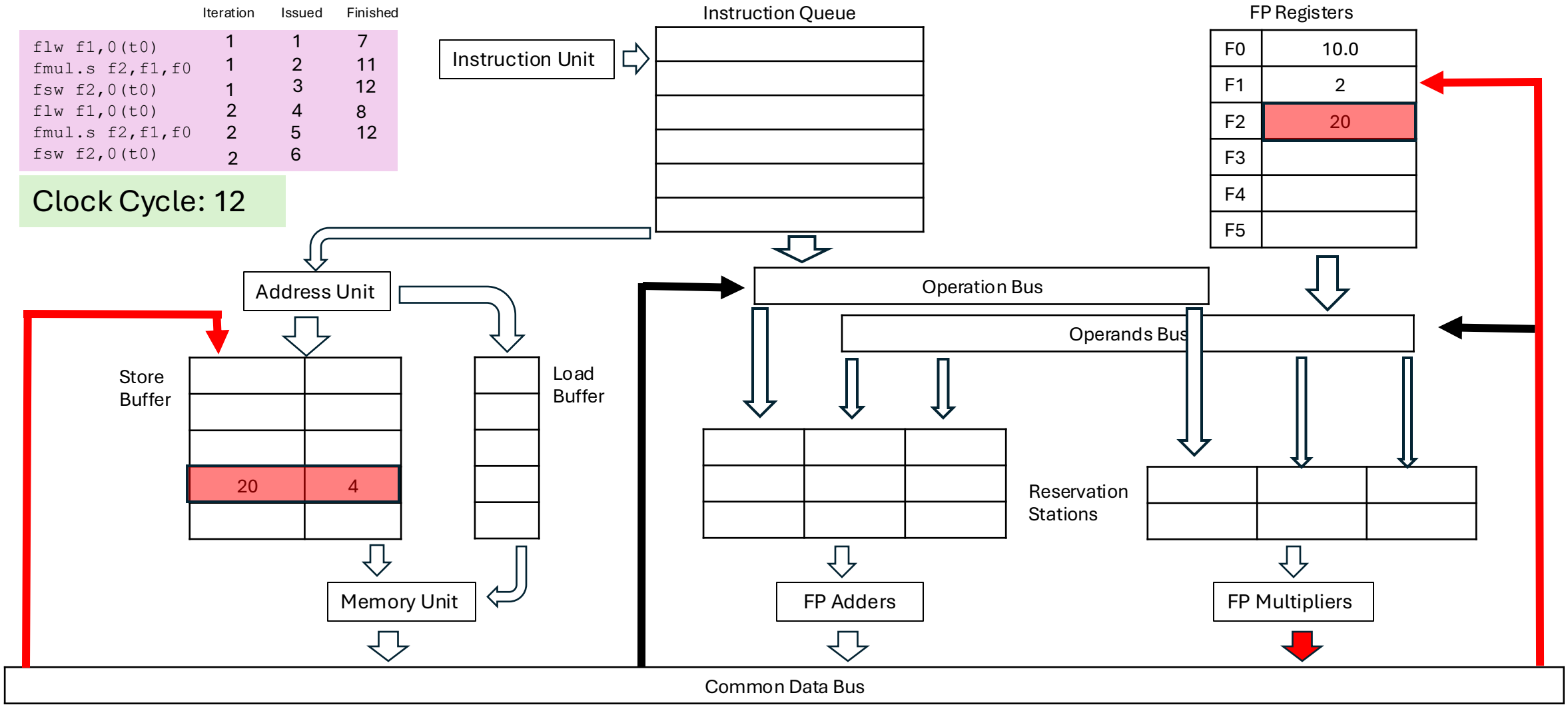
In cycle 11, the first FMUL operation completes and its result is broadcast through the Common Data Bus. The result reaches the Store Buffer where the first store operation was waiting for it.



In cycle 12 the first store operation can finally go to the Memory Unit.



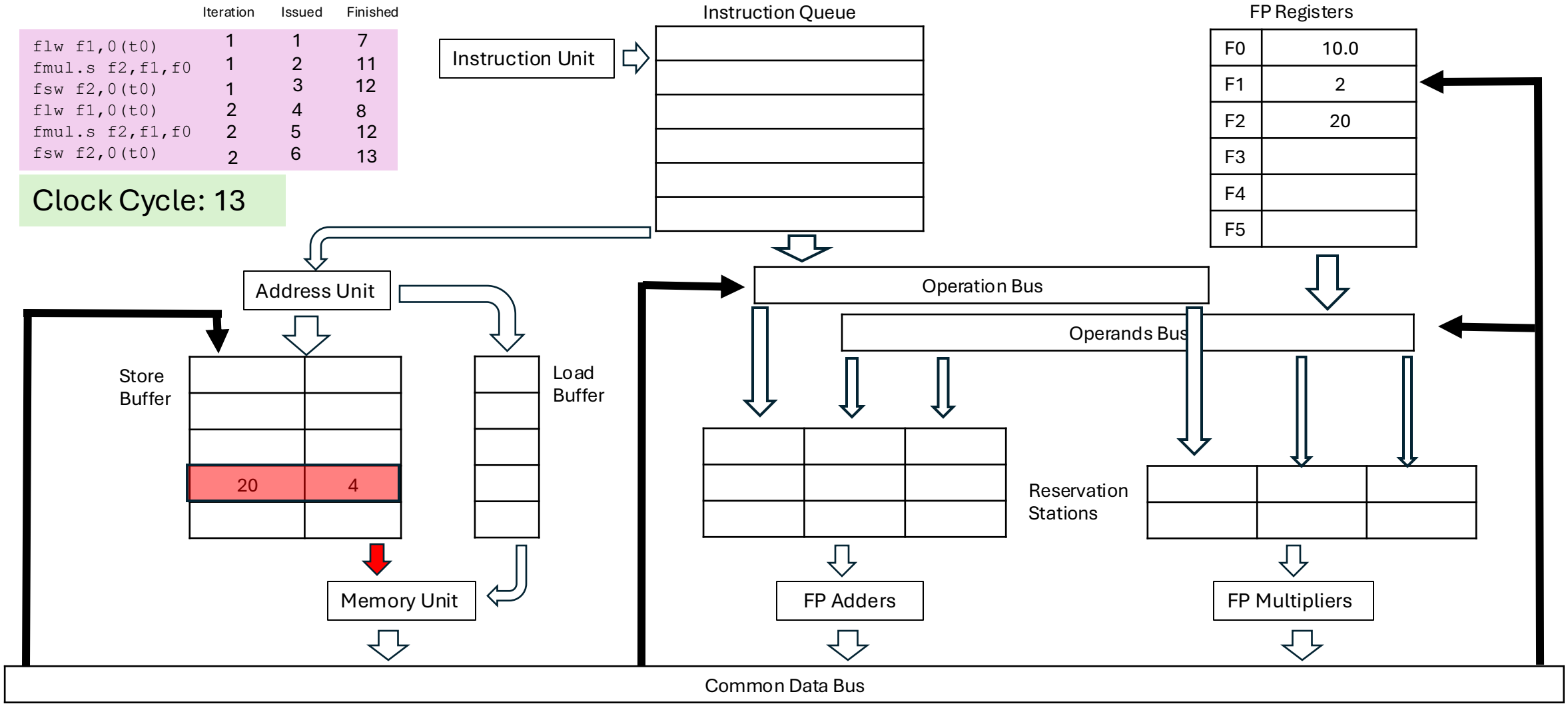
In cycle 12 the Store Buffer slot where the first store operation was, is marked free.



Still in clock cycle 12 the second FMUL operation ends and its results are broadcast through the Common Data Bus to the Store Buffer, where the second store operation was waiting. It also gets copied to the F2 Register.

	Iteration	Issued	Finished
`flw f1,0(t0)`	1	1	7
`fmul.s f2,f1,f0`	1	2	11
`fsw f2,0(t0)`	1	3	12
`flw f1,0(t0)`	2	4	8
`fmul.s f2,f1,f0`	2	5	12
`fsw f2,0(t0)`	2	6	13

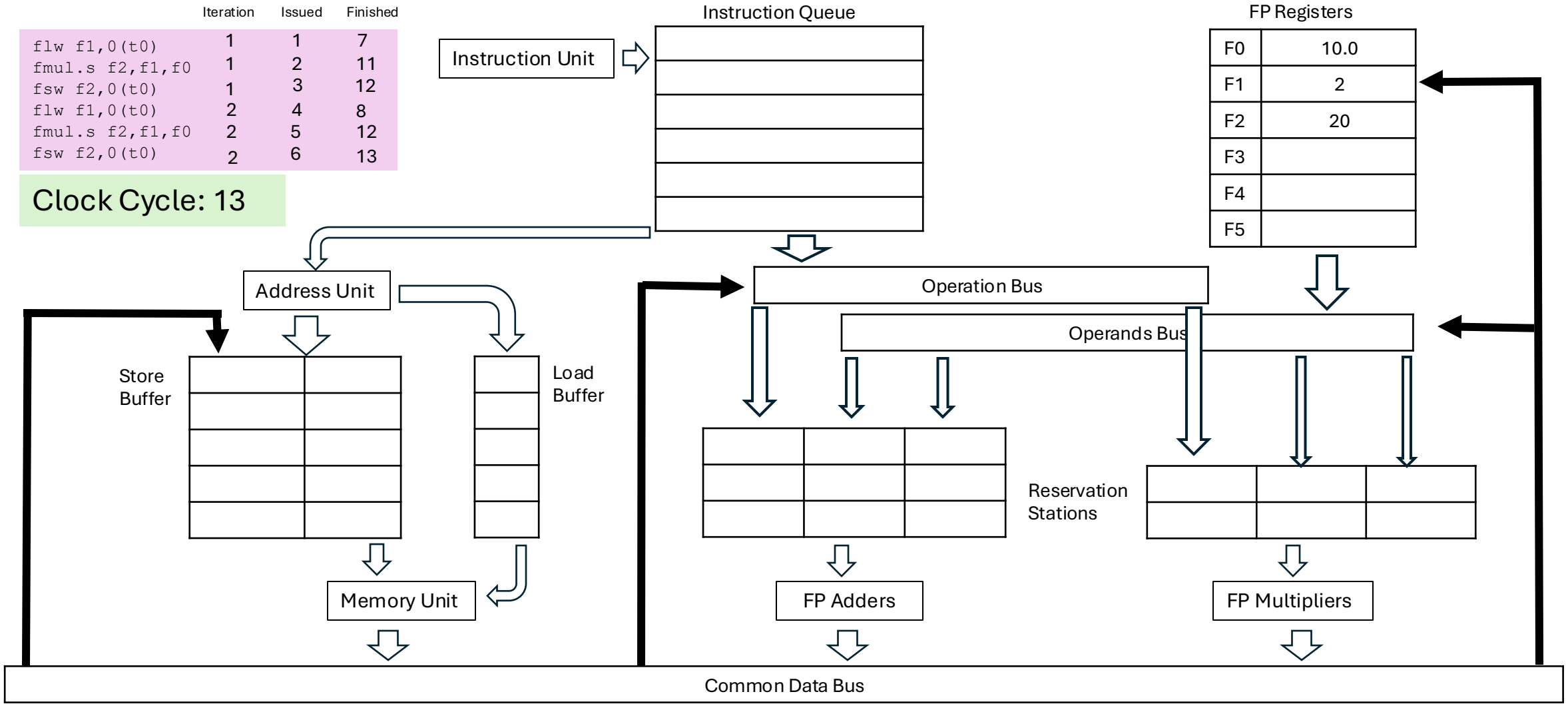
Clock Cycle: 13



In clock cycle 13 the second store operation goes to the Memory Unit. The register F2 is no longer busy.

	Iteration	Issued	Finished
`flw f1,0(t0)`	1	1	7
`fmul.s f2,f1,f0`	1	2	11
`fsw f2,0(t0)`	1	3	12
`flw f1,0(t0)`	2	4	8
`fmul.s f2,f1,f0`	2	5	12
`fsw f2,0(t0)`	2	6	13

Clock Cycle: 13



Still in clock cycle 13, the second store operation slot in the Store Buffer is marked free.