Advance Computer Architecture

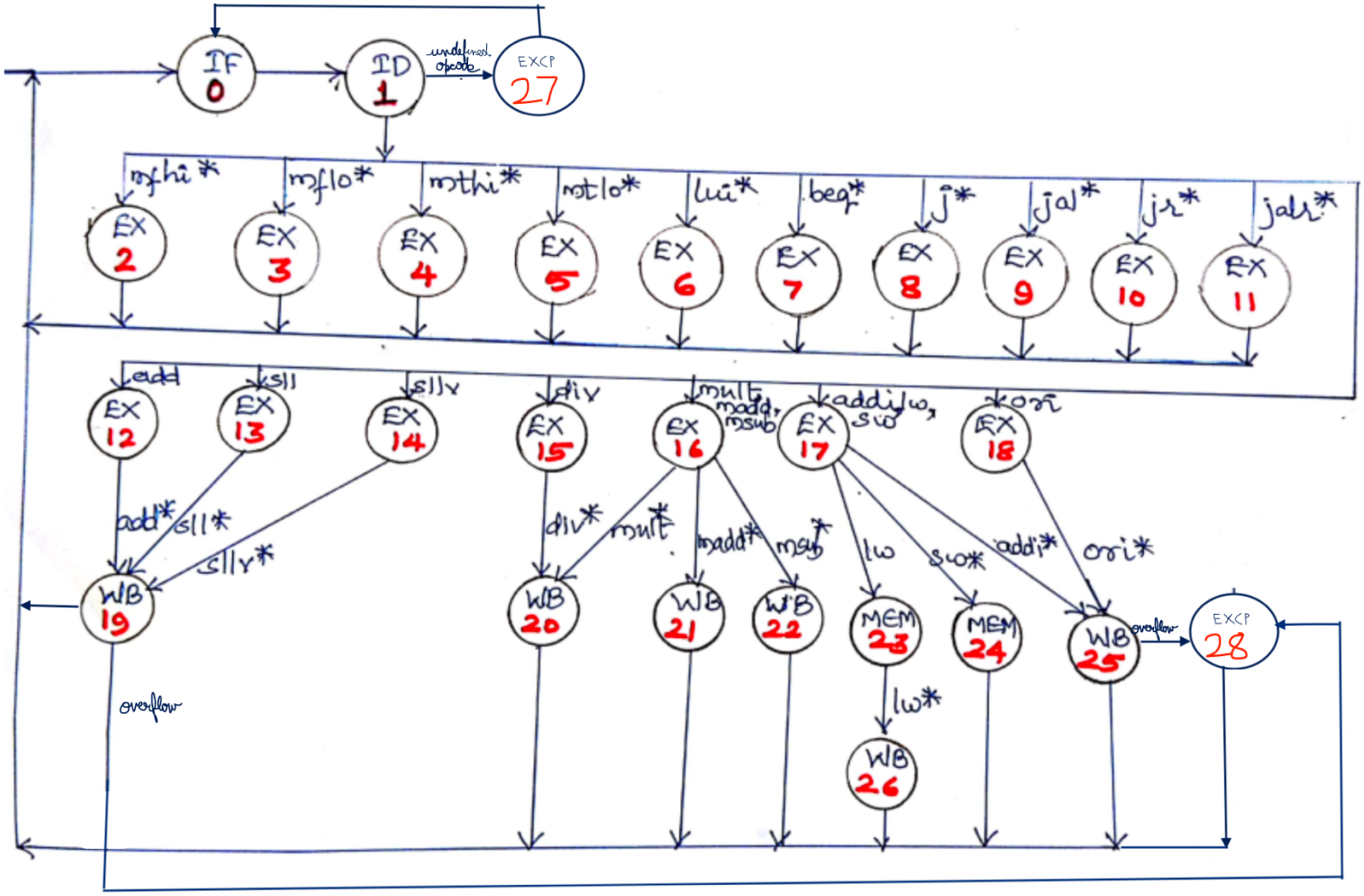
Assignment -2

Microprogrammed Control Unit

Kanika Gera
2018H1030041G

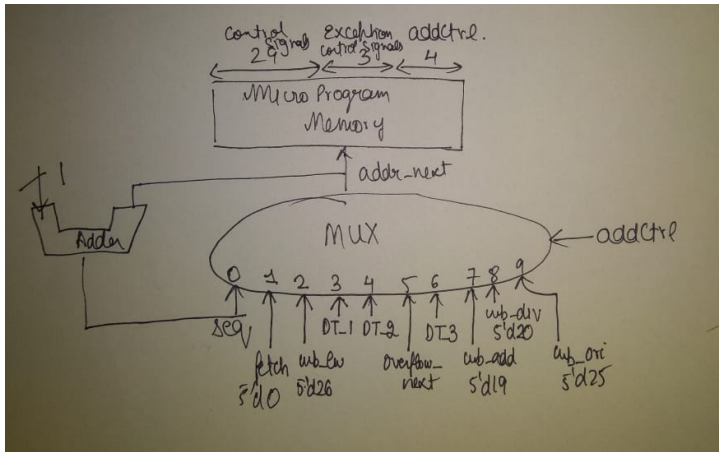
Note : Instructions listed below are in order of execution

Design Constraints
The FSM Followed is :



State 27 : Undefined Instruction State
State 28: Arithmetic Overflow State

Design Used



DT_1 : Output from Dispatch Table 1

Dispatch Table 1

Dispatch Table 1 is used to store the transition from ID state to EX state of each instruction.
It defines the transition from state 1 to state 2-18 or state 27.

DT_2 : Output from Dispatch Table 2

Dispatch Table 2

Dispatch Table 2 is used to store the transition from EX state of instruction to its WB state.
It defines the EX->WB transition defined for ,mult,madd,msub, .

DT_3 : Output from Dispatch Table 3

Dispatch Table 3

Dispatch Table 3 is used to store the transition from EX state of instruction to its MEM/WB state.
It defines the EX->WB transition defined for ,addi, lw,sw

wb_add: Defines the transition from state 12,13,14-> state 19 That is transition from EX -> WB for add,sll,sllv instruction.
wb_lw: Defines the transition from state 23 -> state 24. That is transition from MEM -> WB for lw instruction.
wb_ori: Defines the transition from state 18 -> state 25. That is transition from EX-> WB for ori instruction.
wb_div: Defines the transition from state 15 -> state 20. That is transition from EX-> WB for div instruction.
overflow_next: Output from overflow_checker

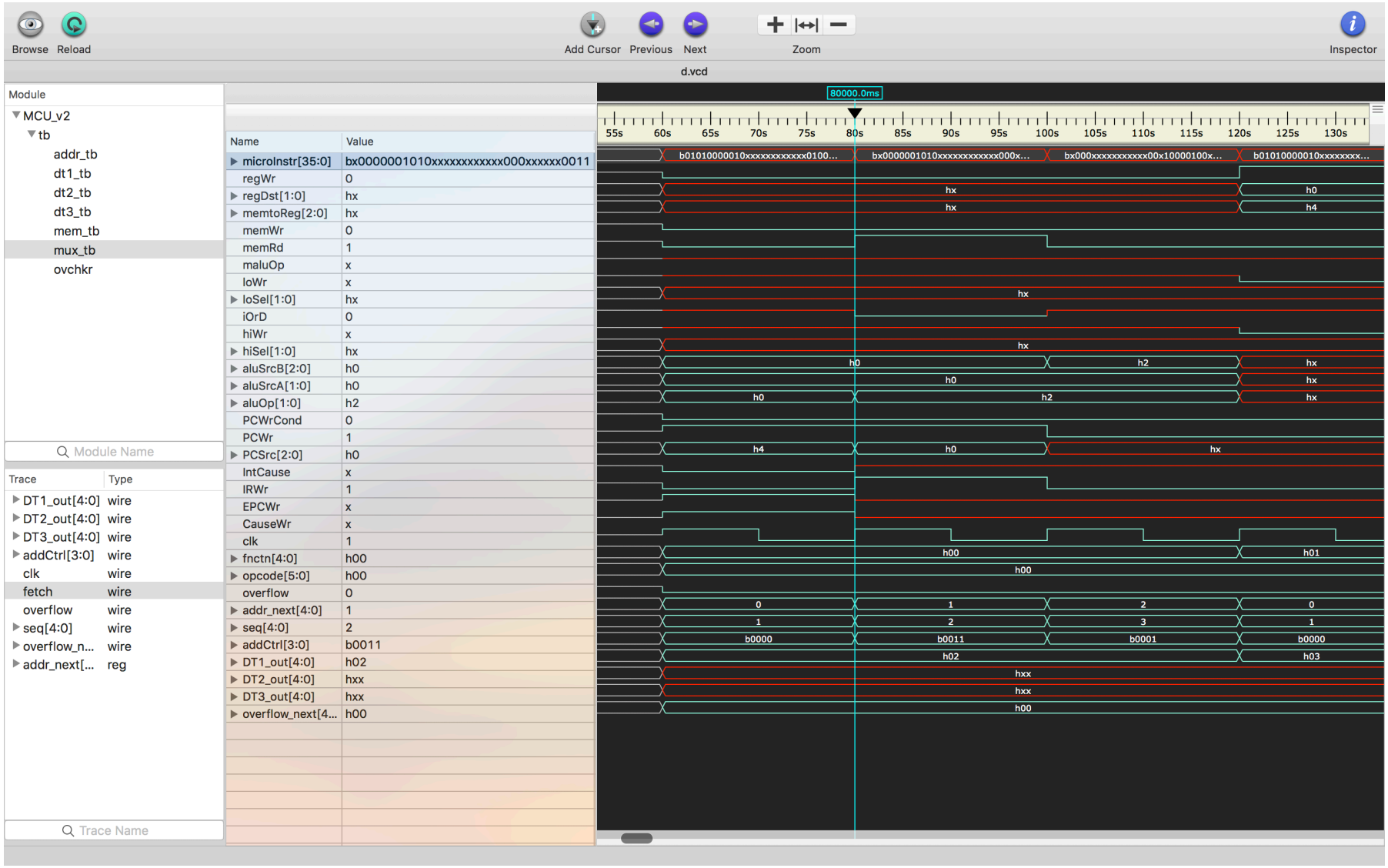
overflow_checker

Defines whether to go to state 28 or not from WB state of add and addi.

seq: Next state of current executing state

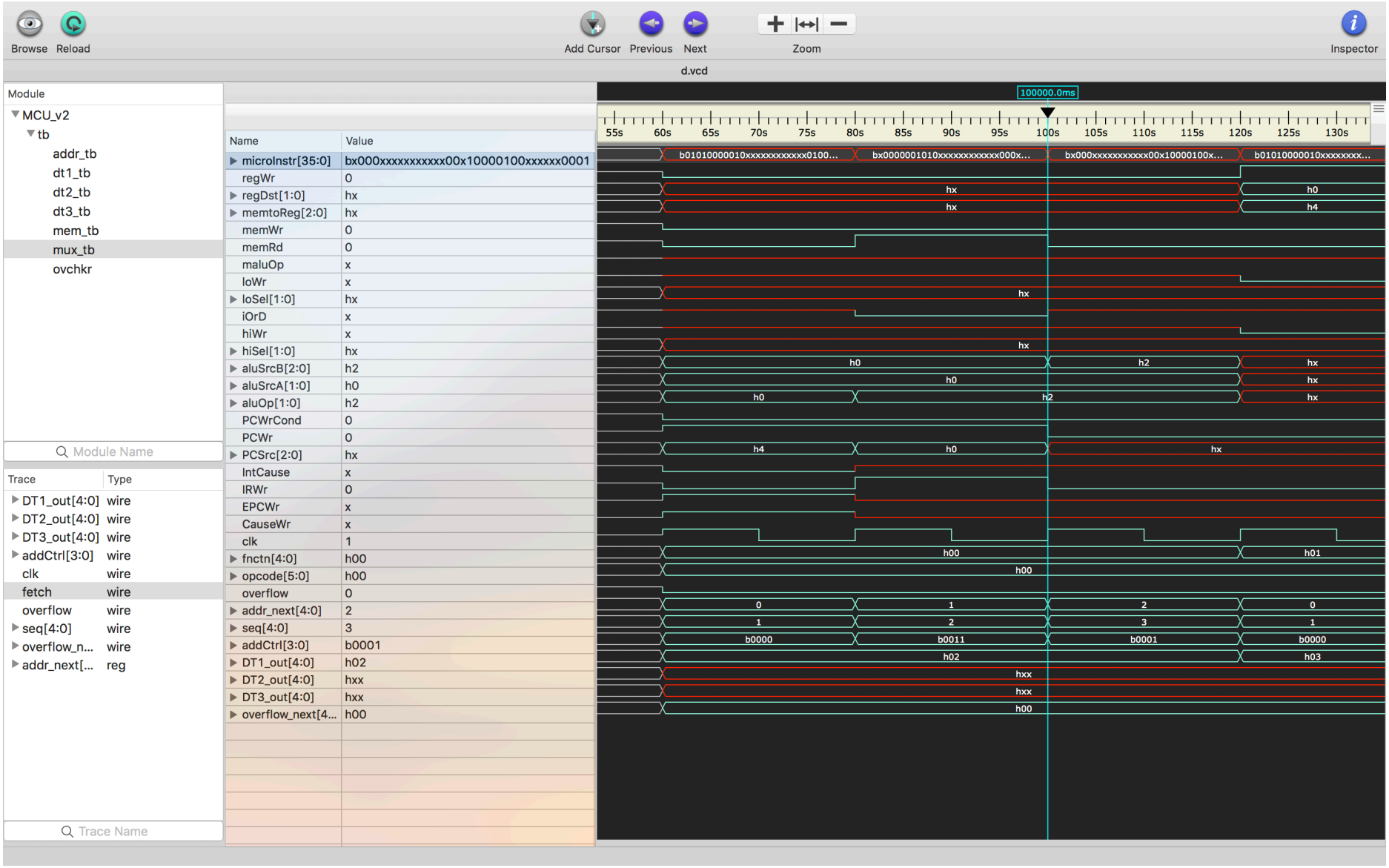
STATE 0

Control Signals generated in IF stage of all instructions are shown in figure below:



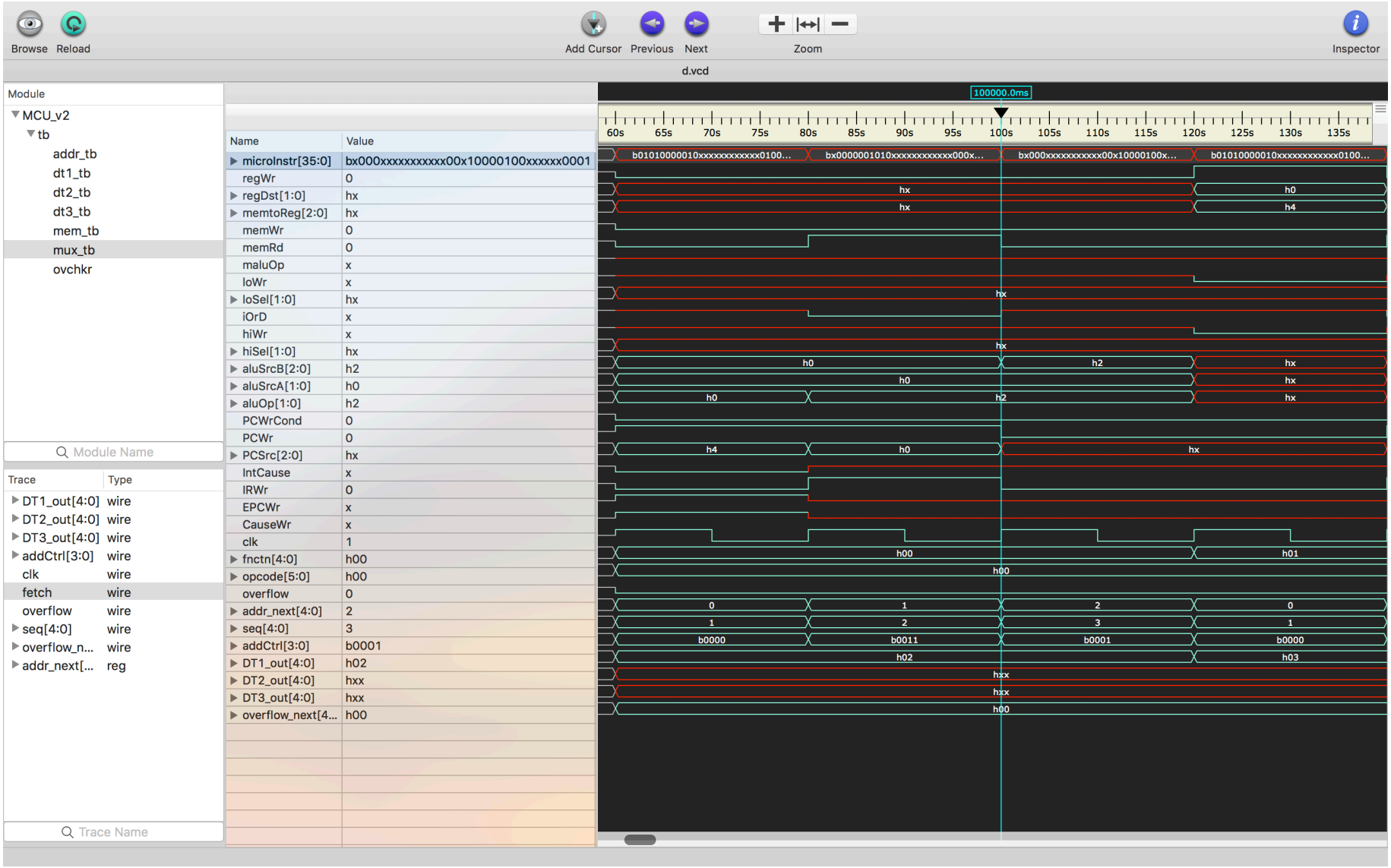
STATE 1

Control Signals generated in ID stage of all instructions are shown in figure below:



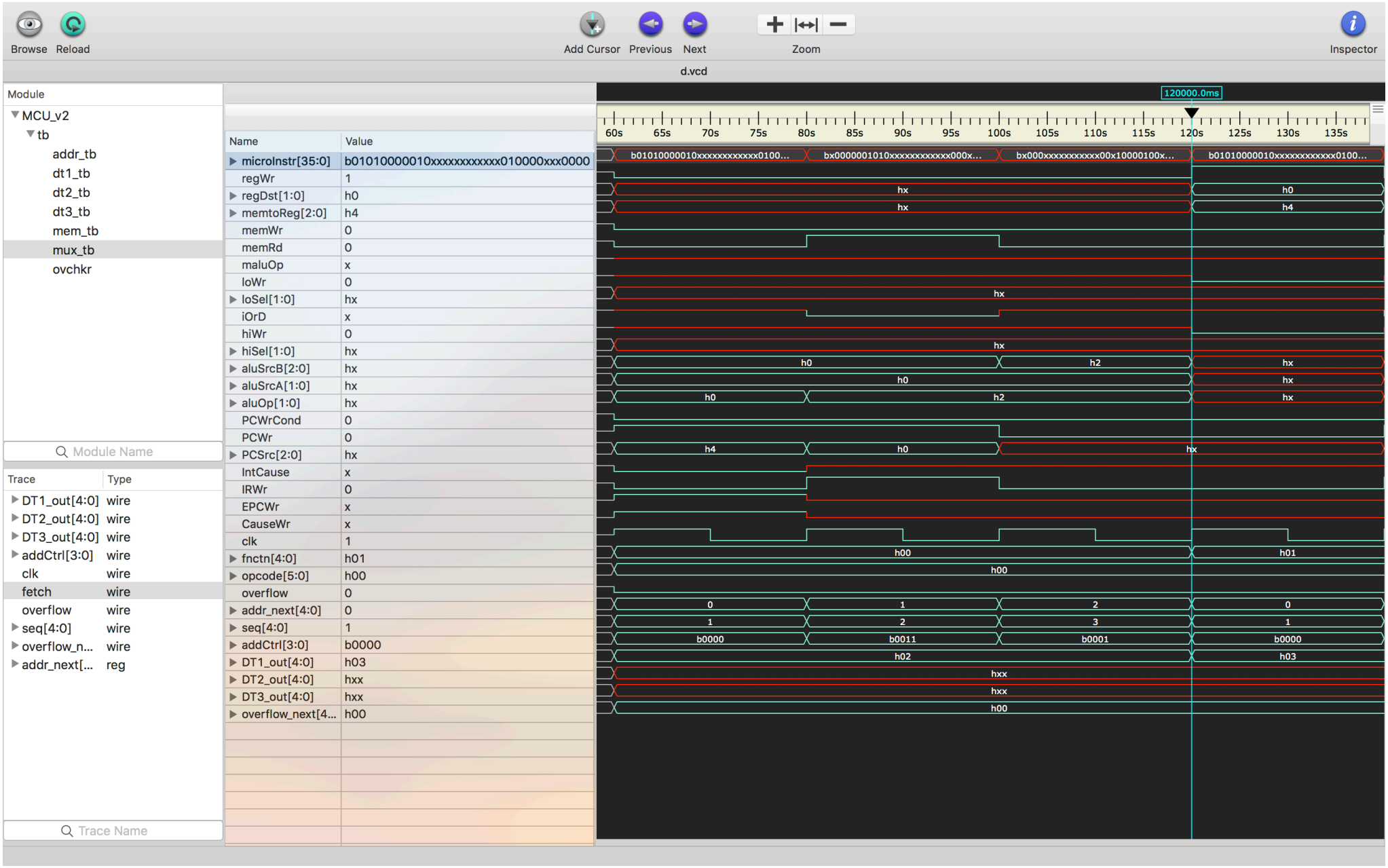
MFHI

Control Signals generated in EX stage of given instruction is shown in following two figures. Instruction in IF and ID stages are shown above.
In the starting of clock cycle microinstruction is read as control enters new state of FSM , by the end of that state microinstruction fetched and each control signal is given respective values.



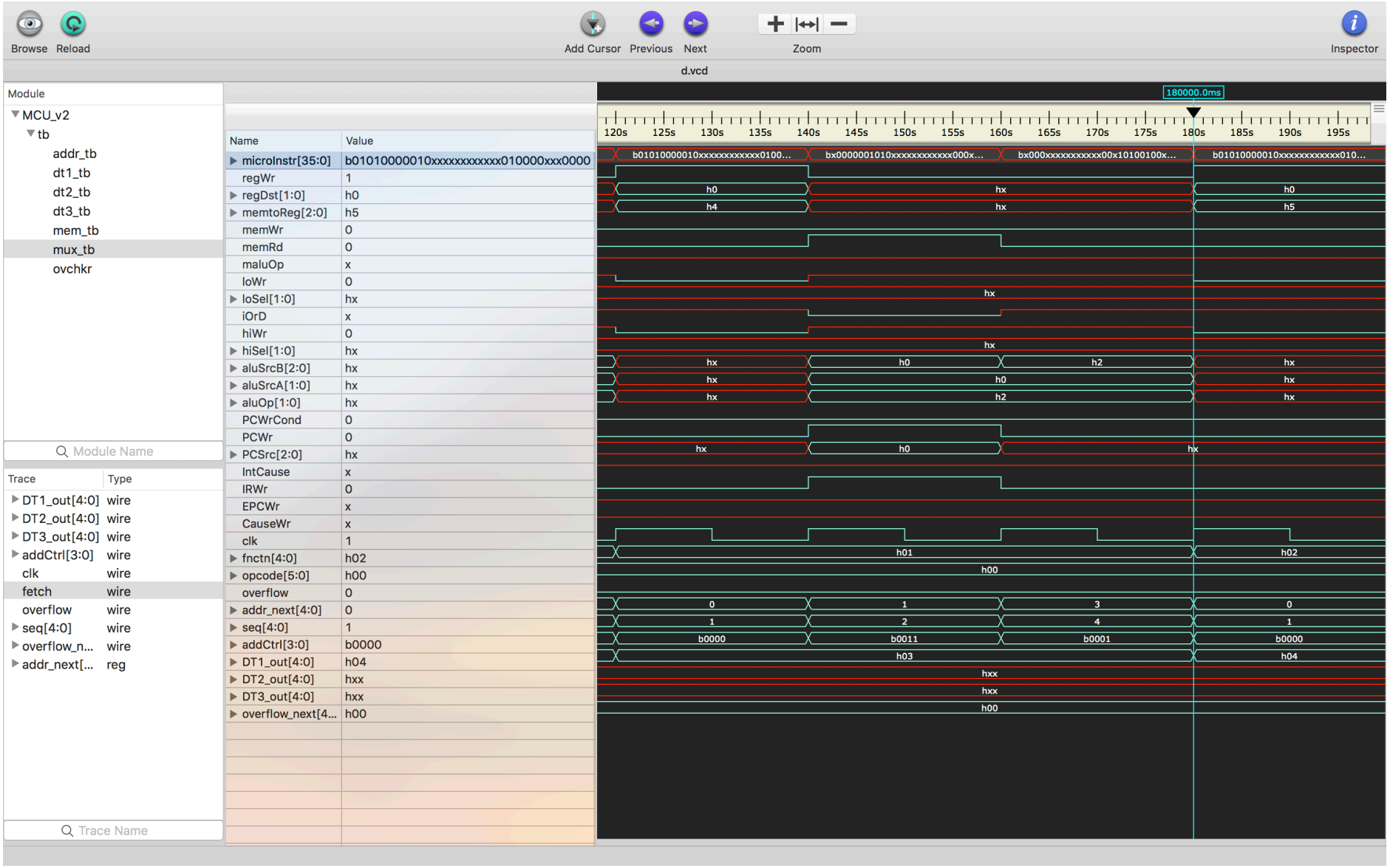
MFHI

State : 2 - MFHI EX



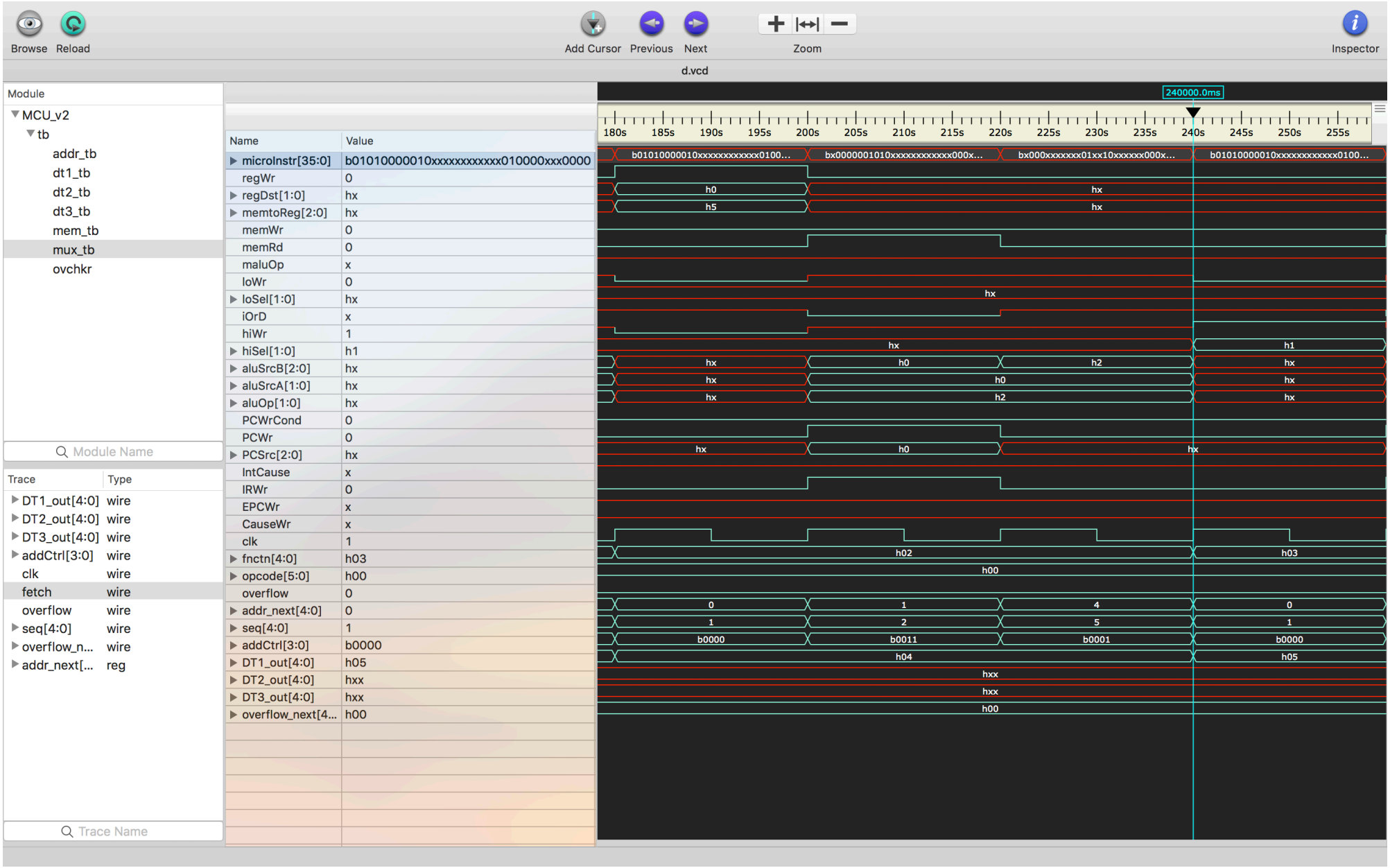
MFLO

State : 3 - MFLO EX



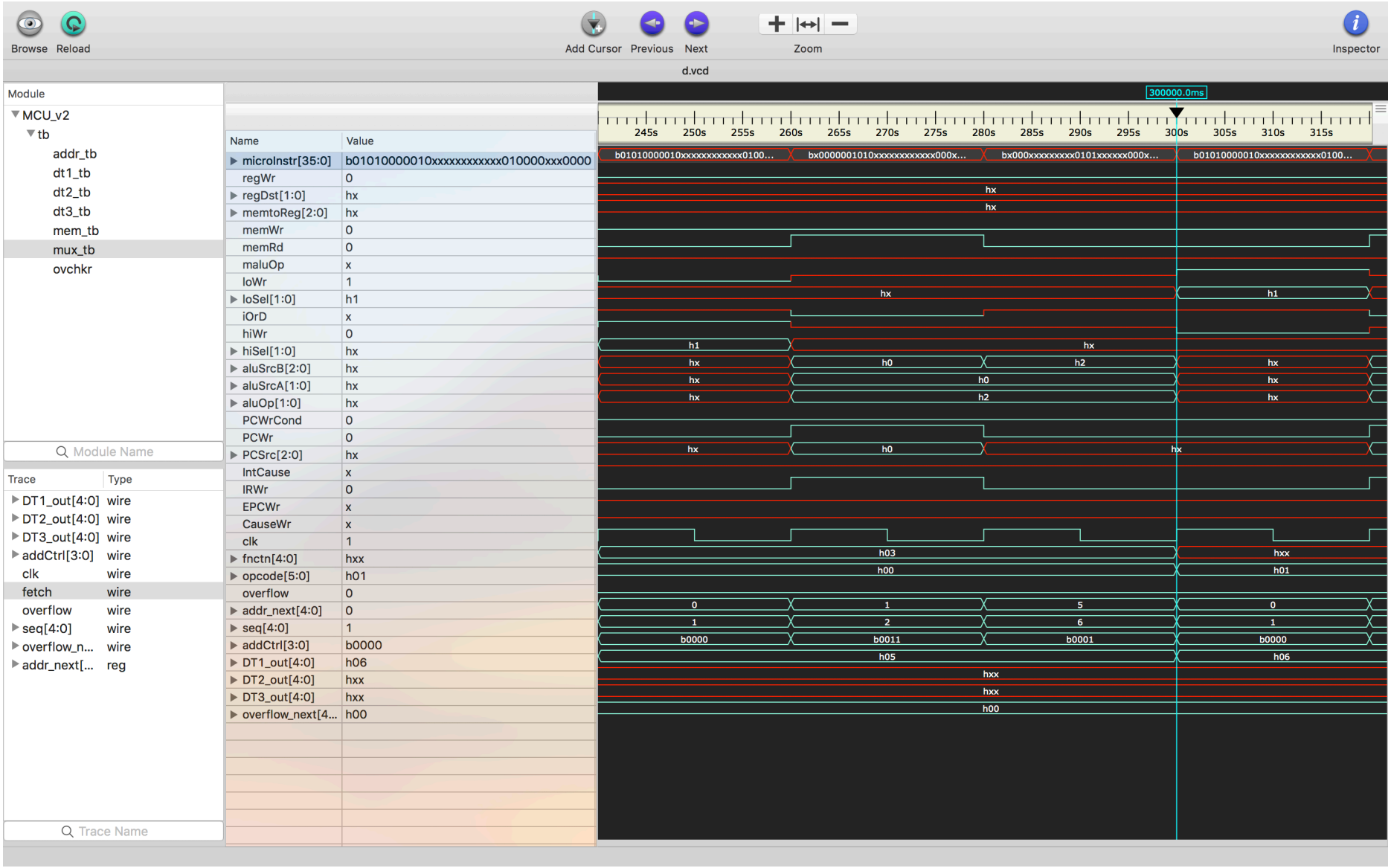
MTHI

State : 4 - MTHI EX



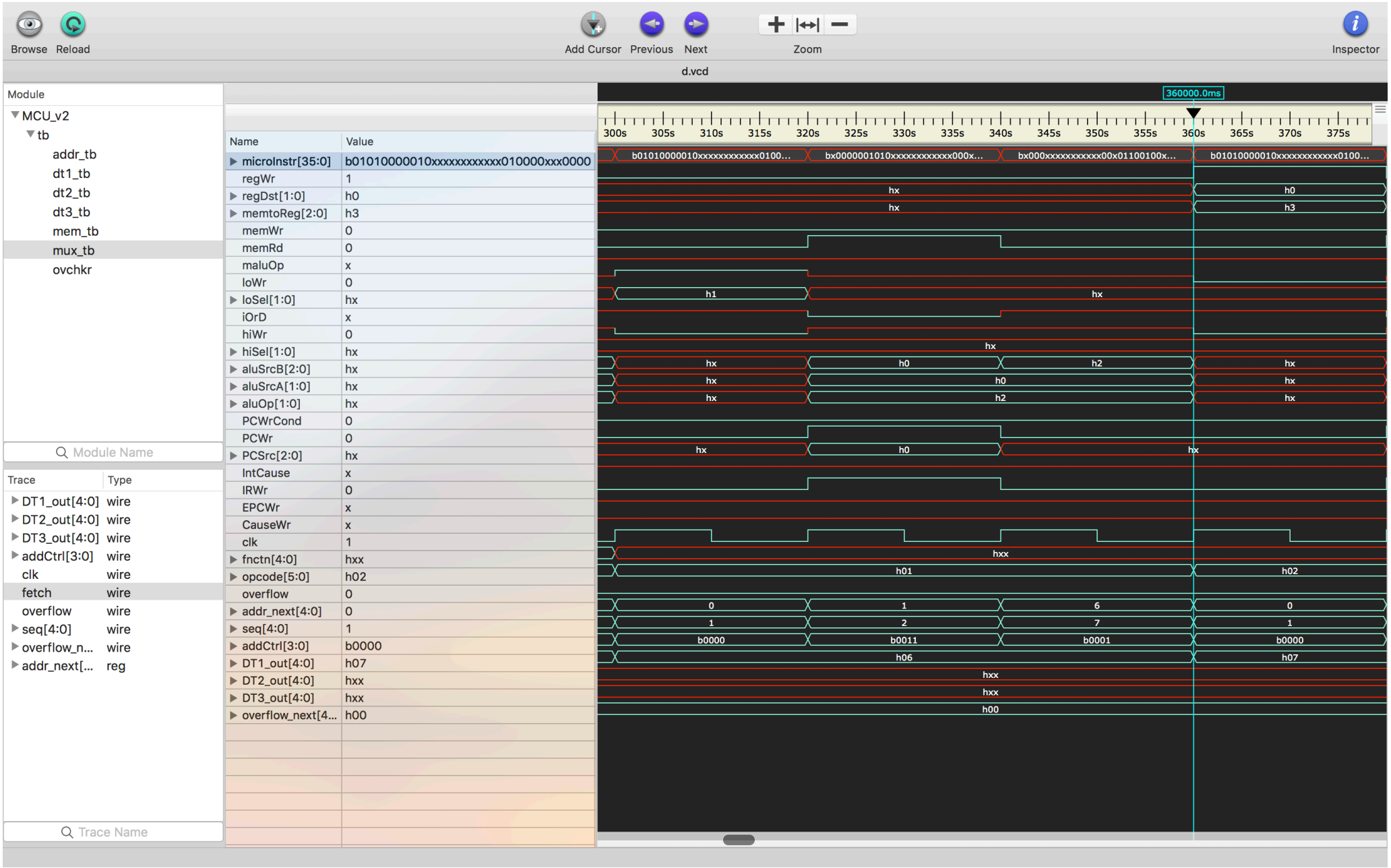
MTLO

State : 5- MTLO EX



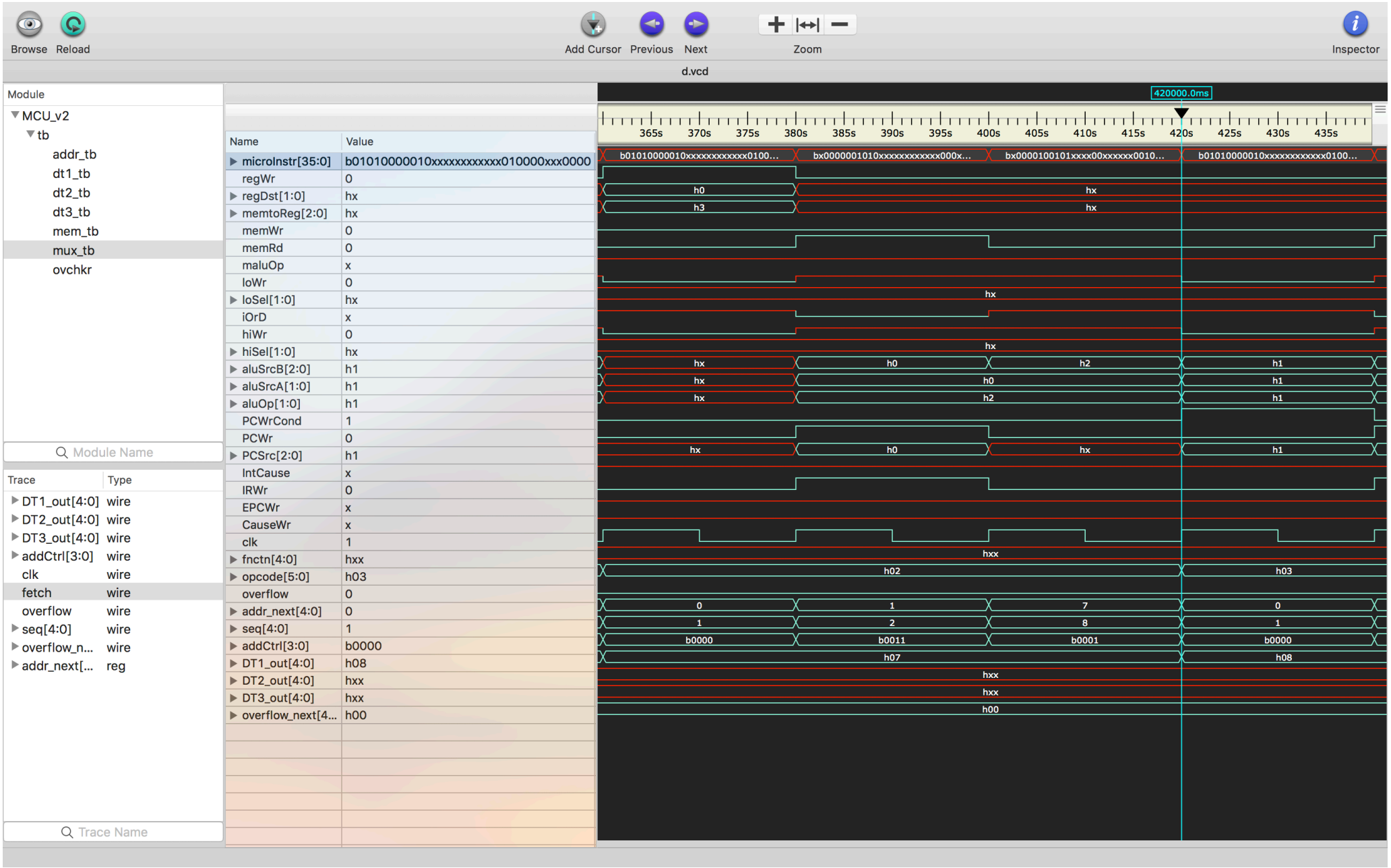
LUI

State : 6- LUI EX



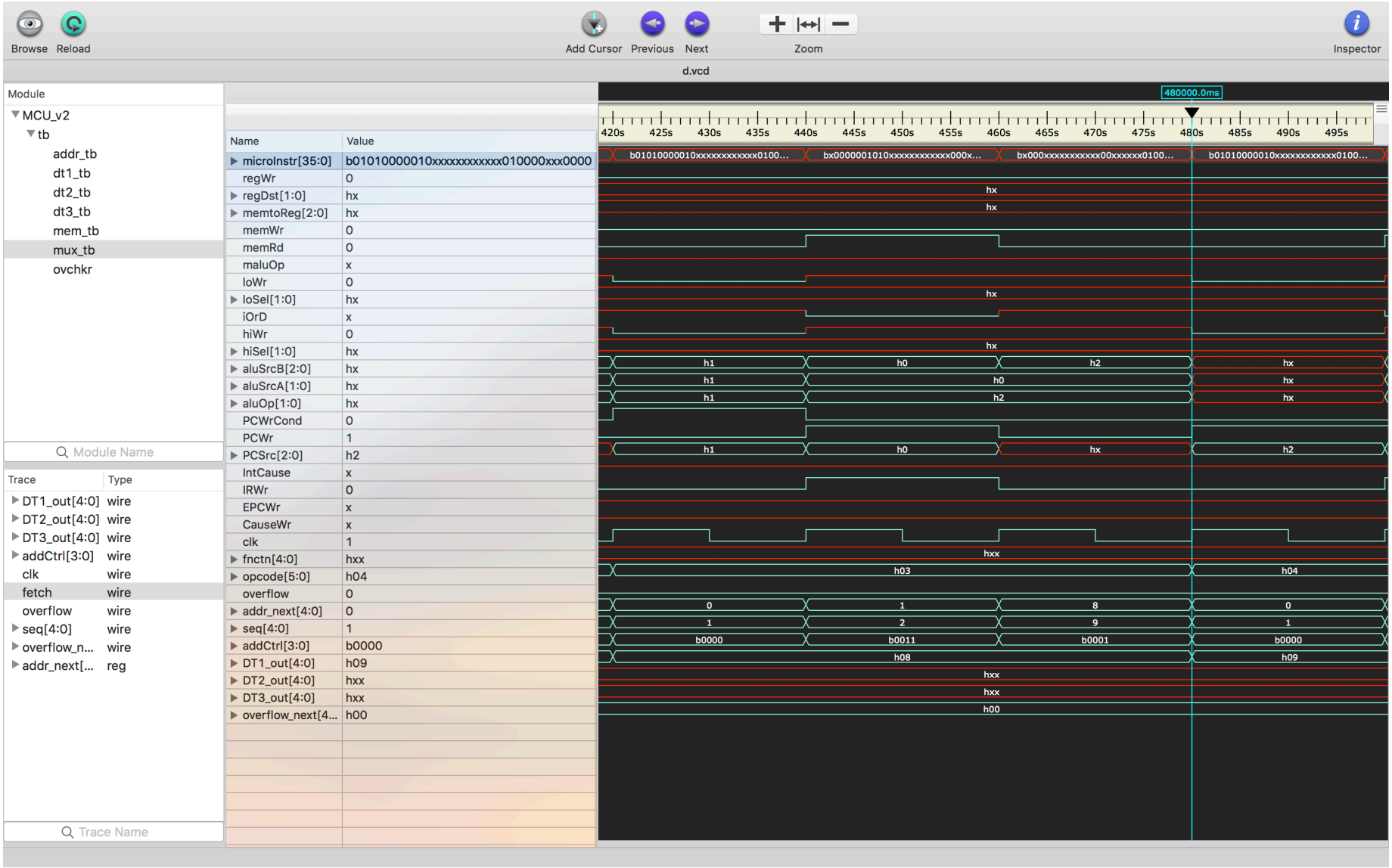
BEQ

State : 7- BEQ EX



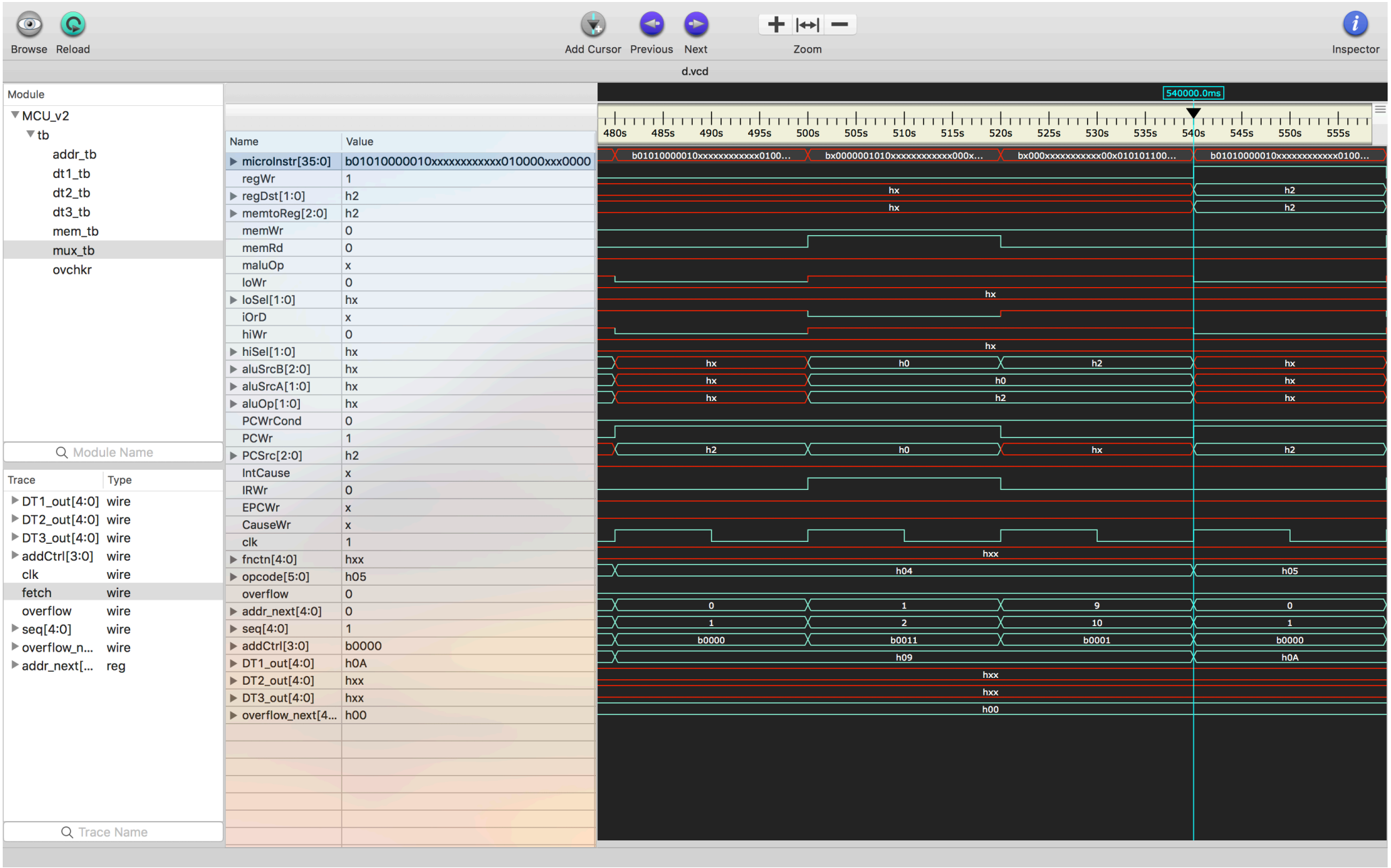
J

State : 8- Jump



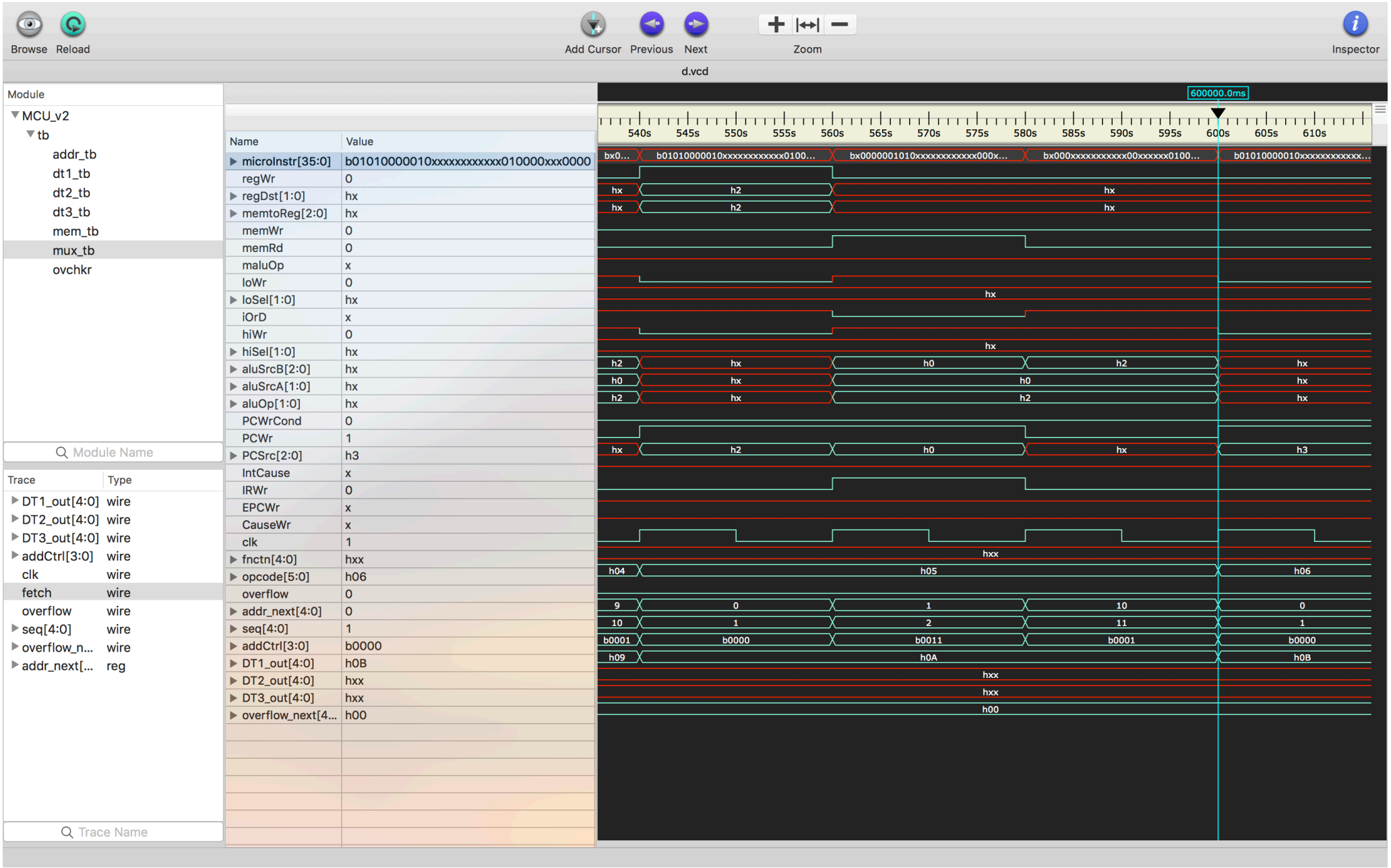
JAL

State : 9- JAL EX



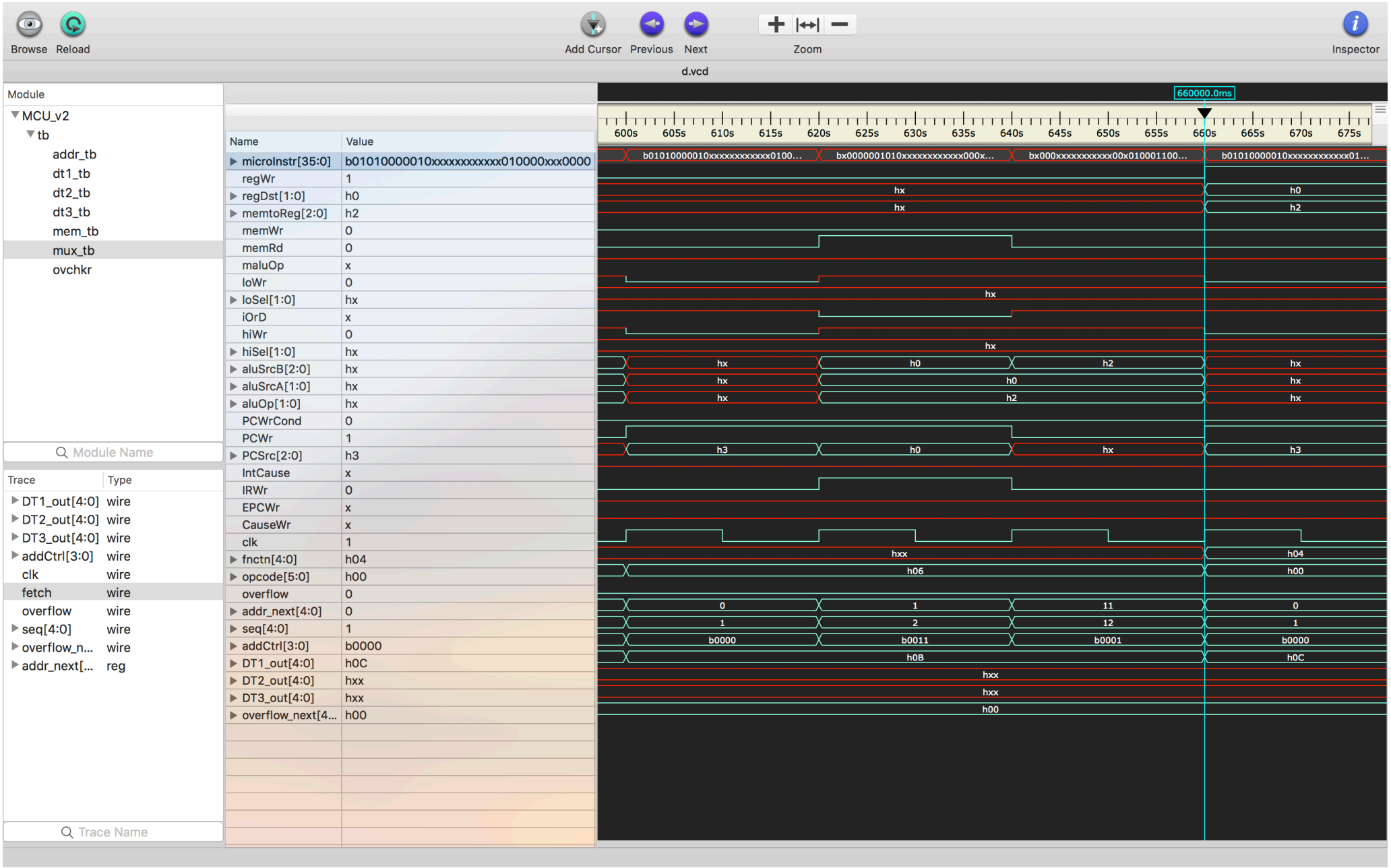
JR

State : 10- JR EX



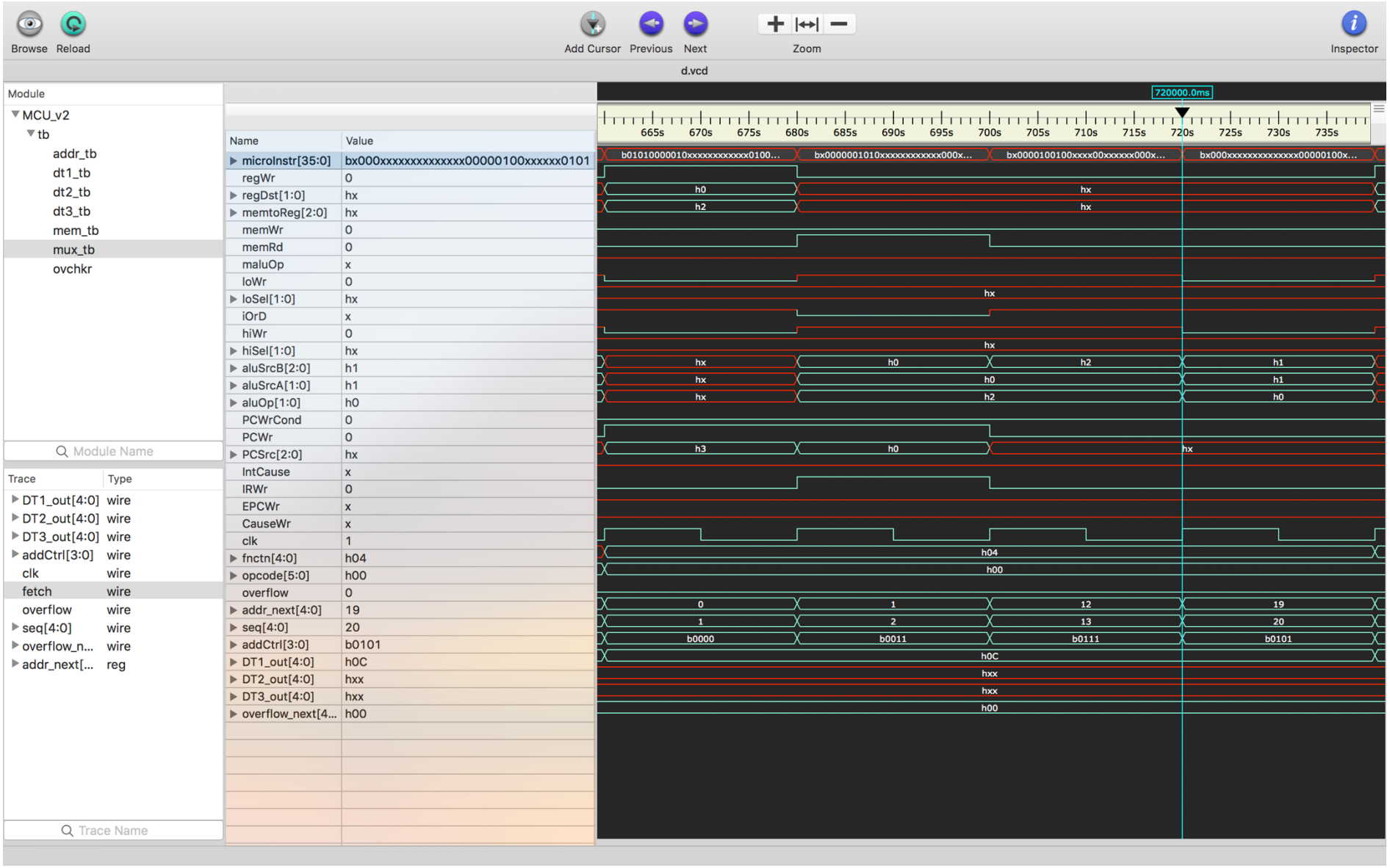
JALR

State : 11 JALR EX

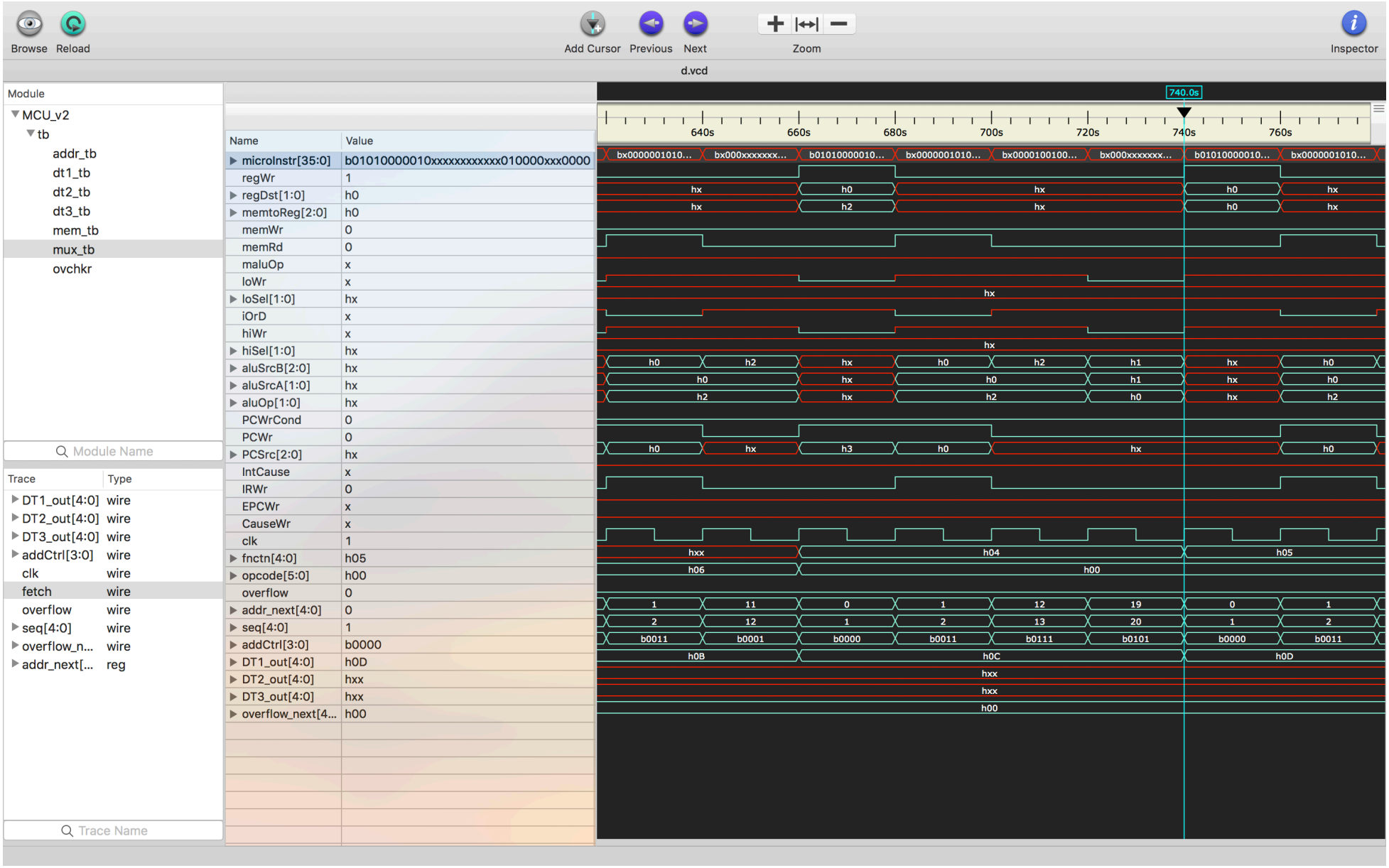


ADD

State : 12 ADD EX

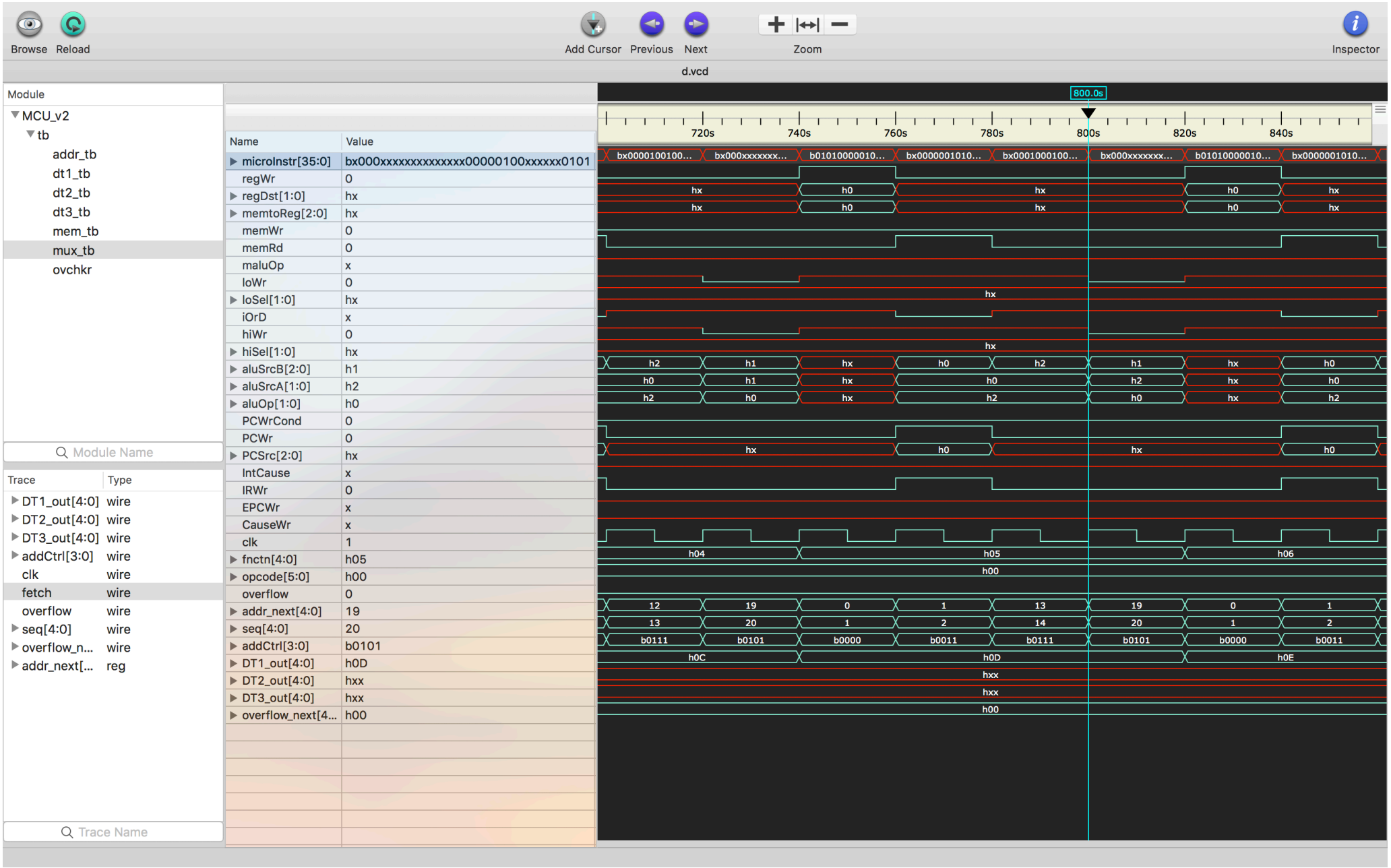


State : 19 ADD WB

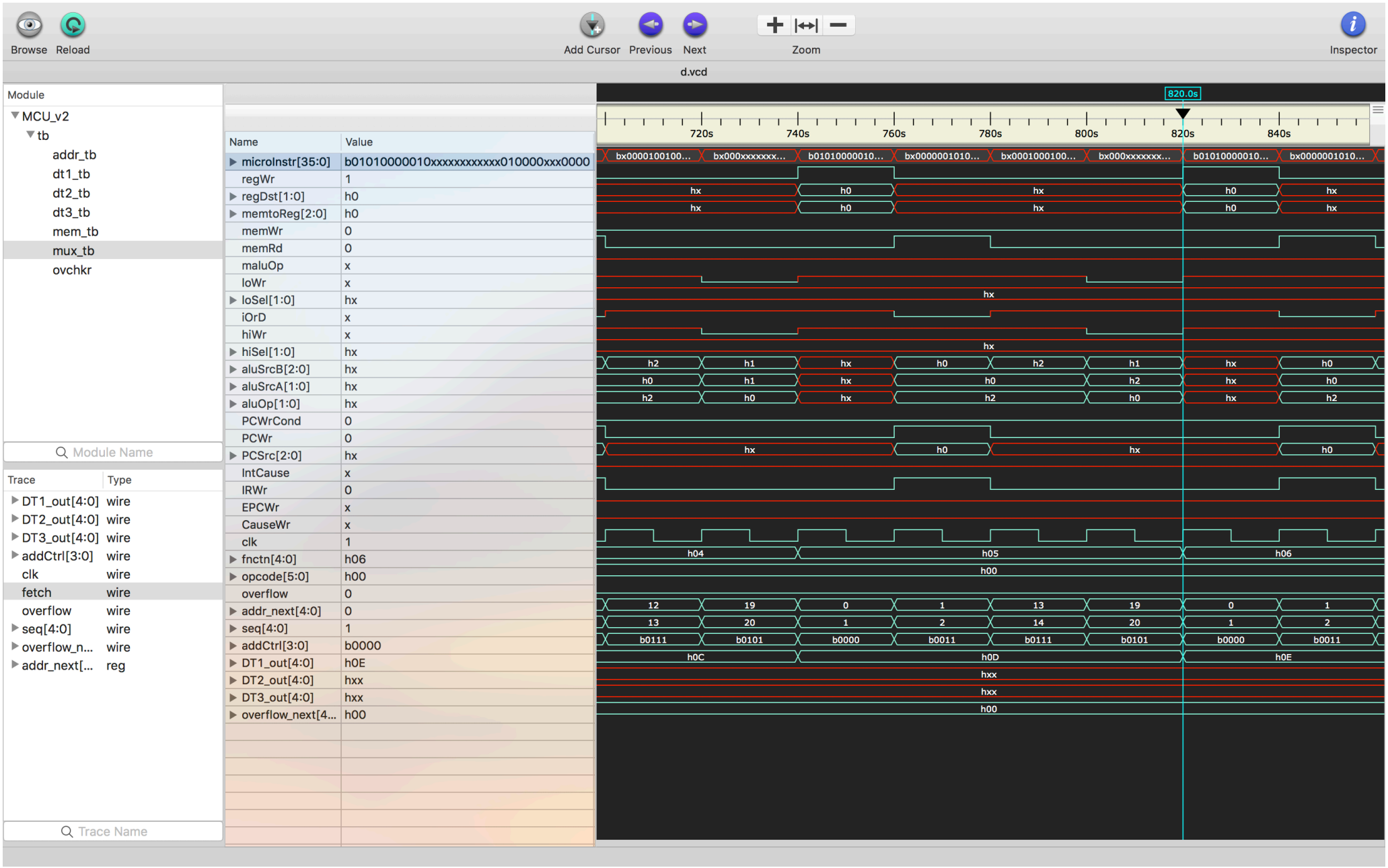


SLL

State : 13 SLL EX

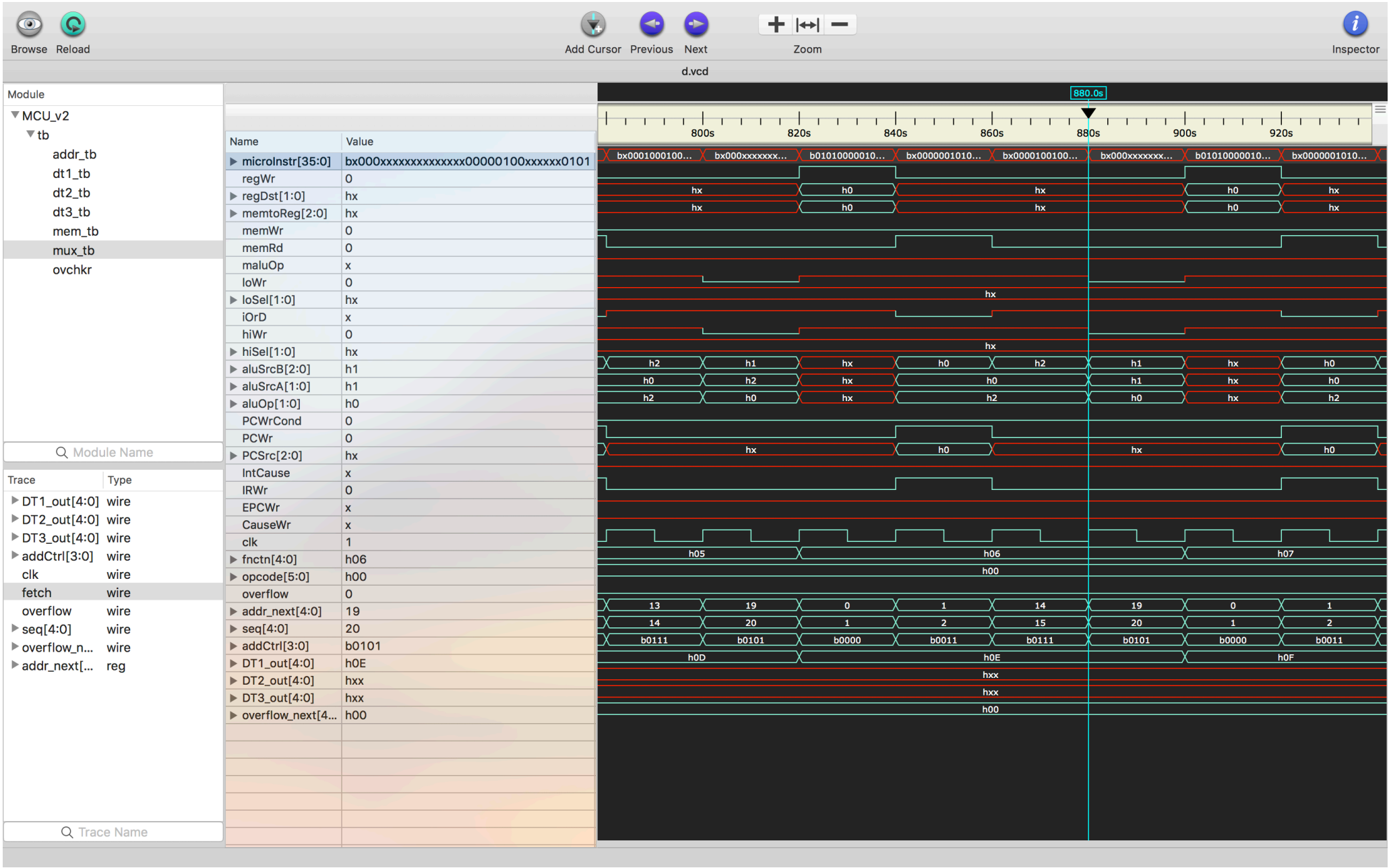


State : 19 SLL WB

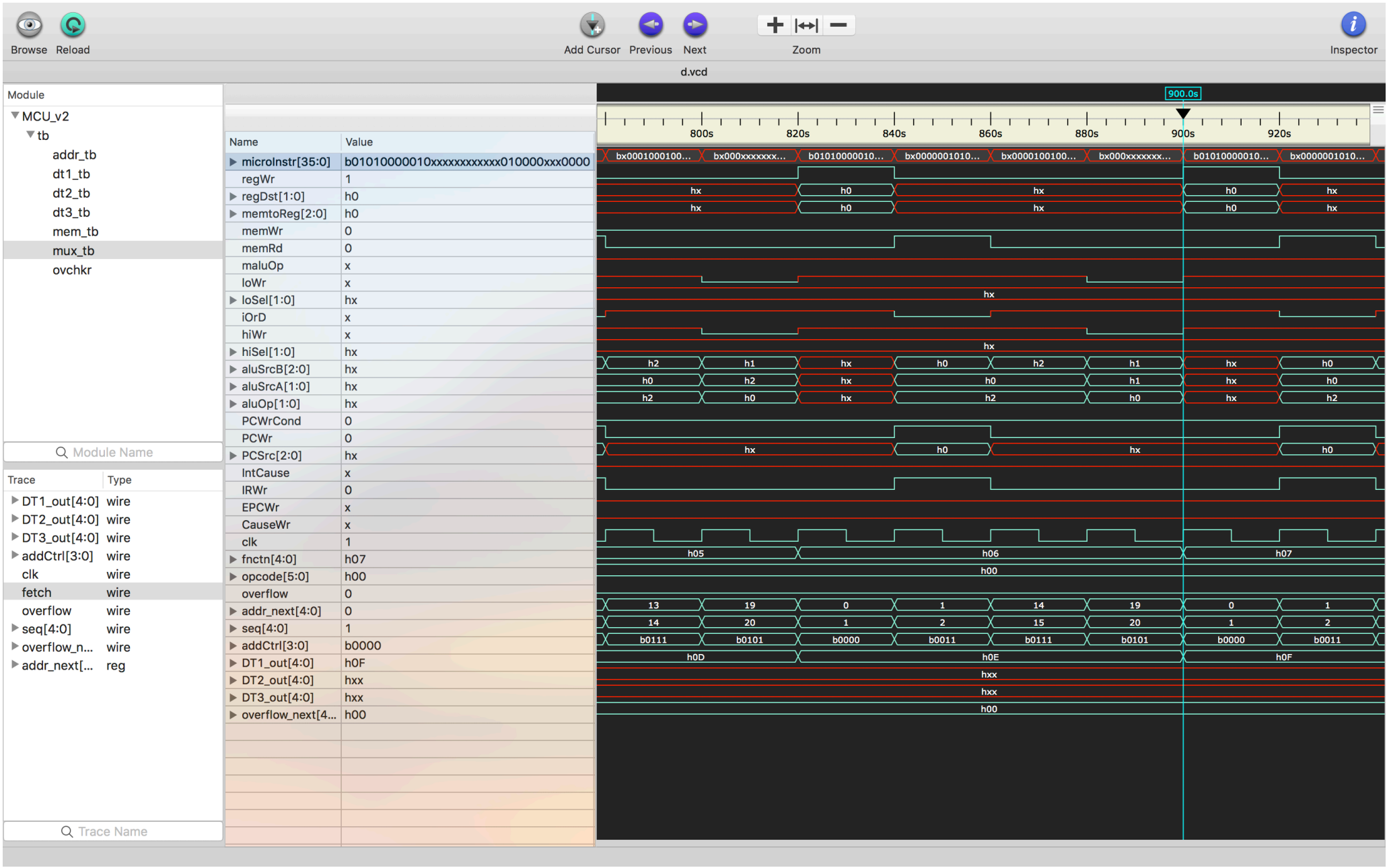


SLLV

State : 14 SLLV EX

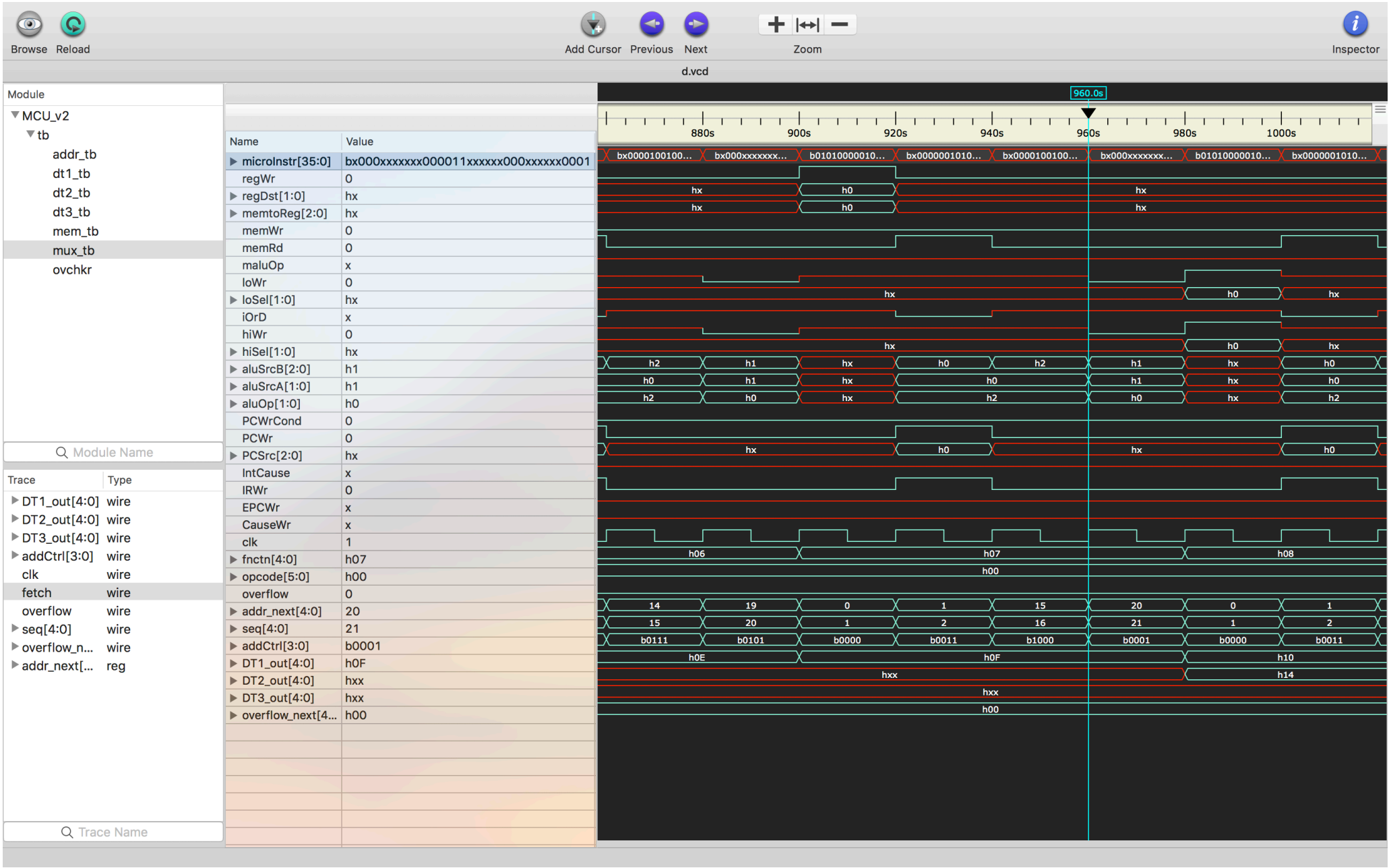


State : 19 SLLV WB

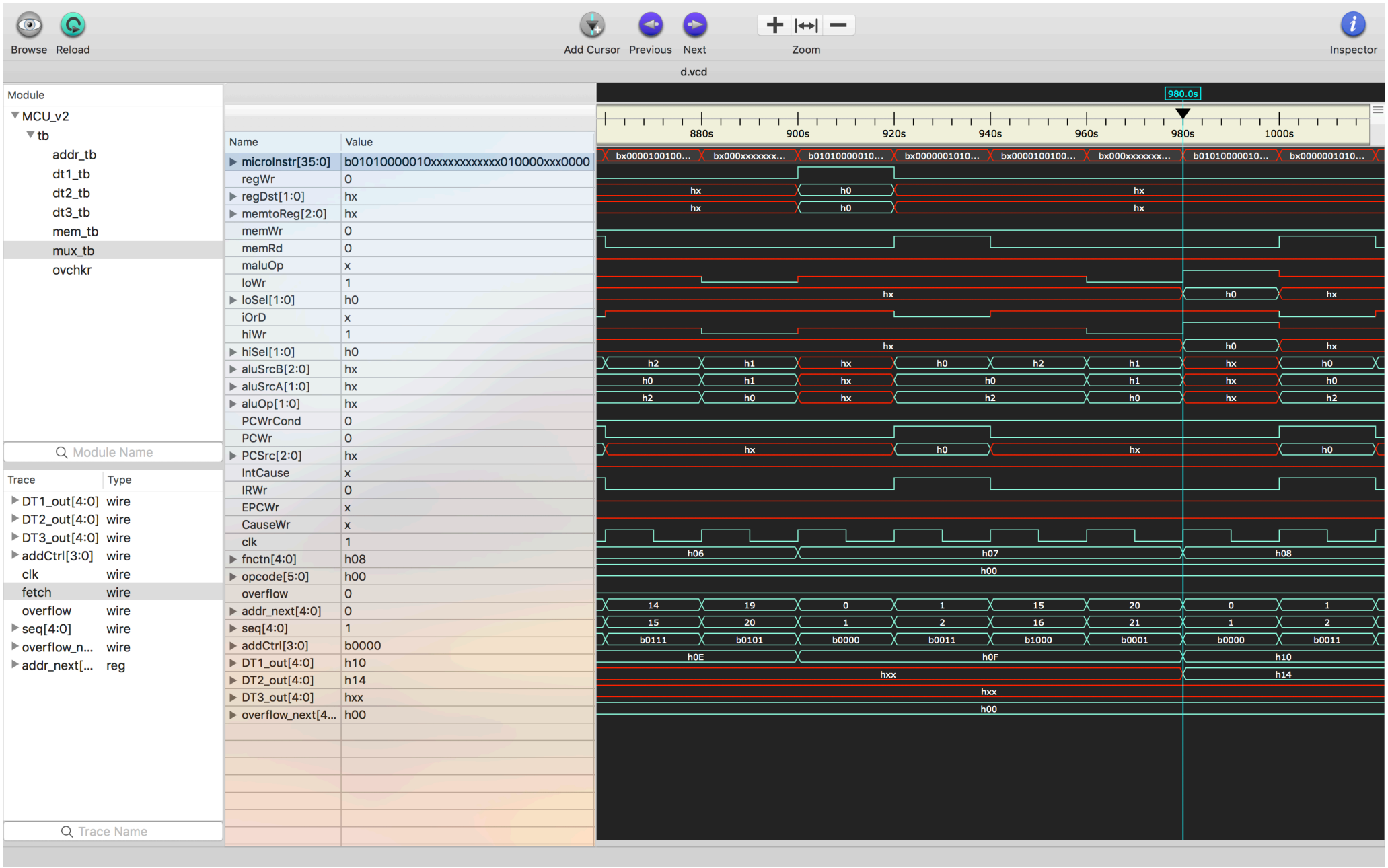


DIV

State : 15 DIV EX

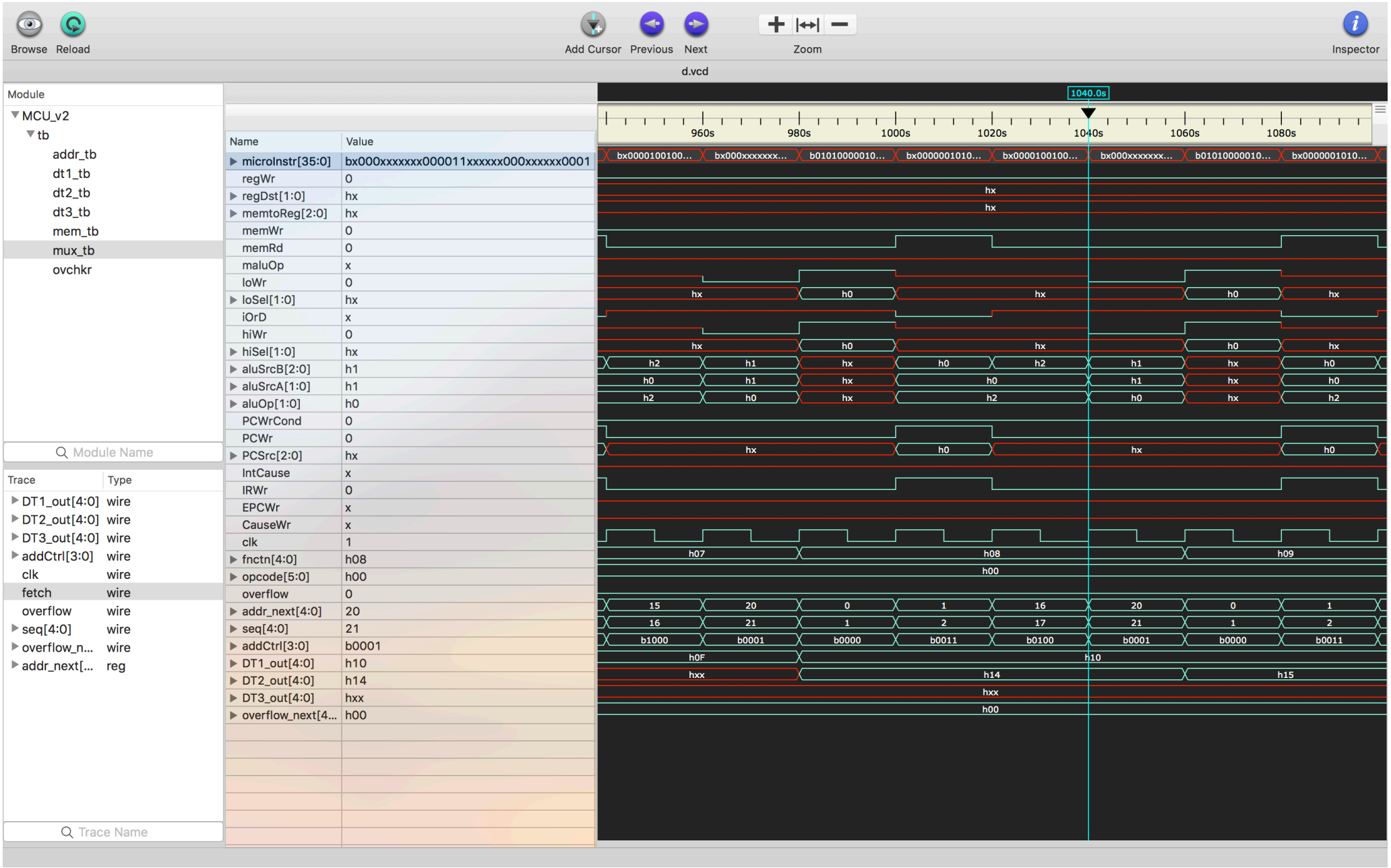


State : 20 DIV WB

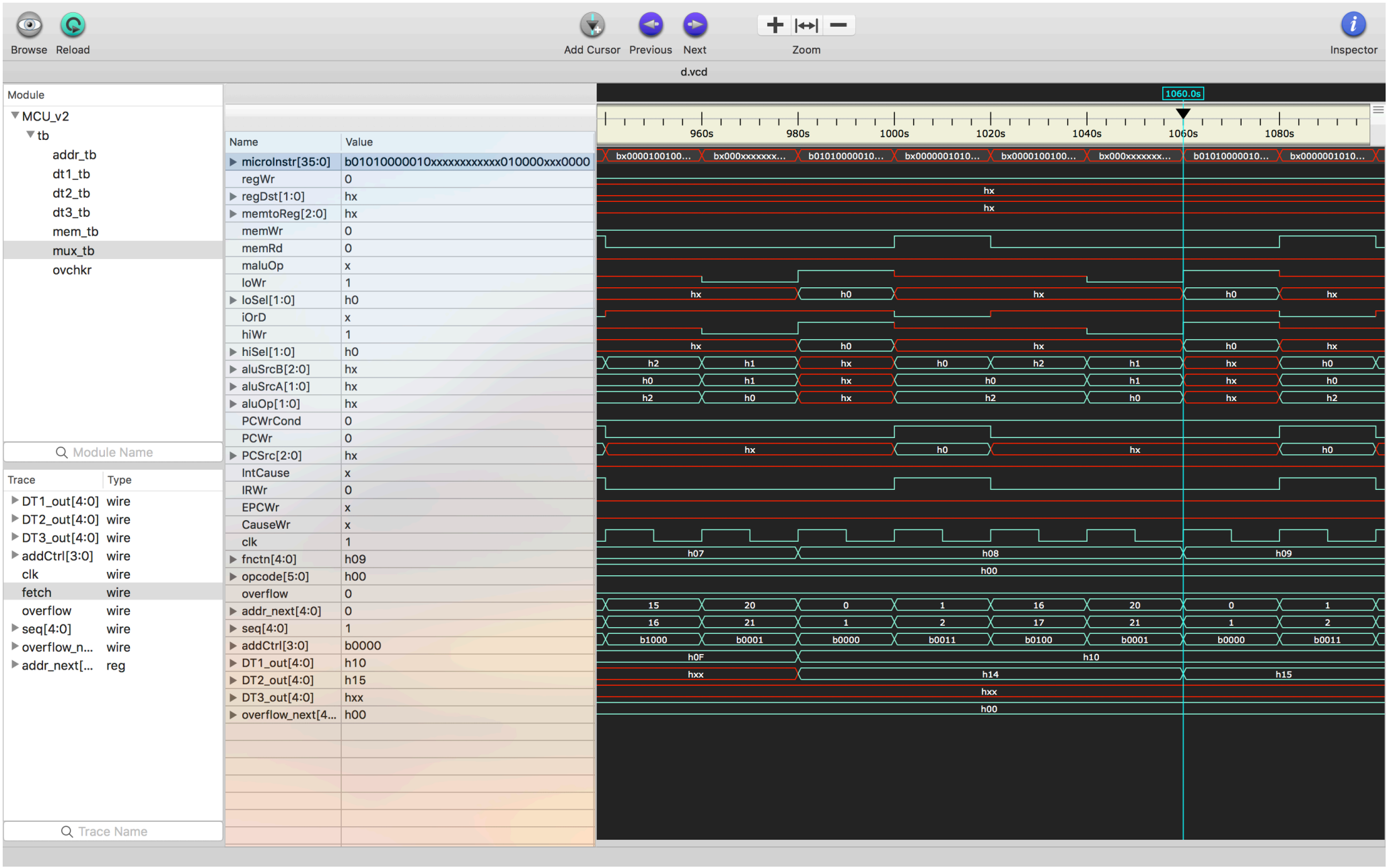


MULT

State : 16 MULT EX

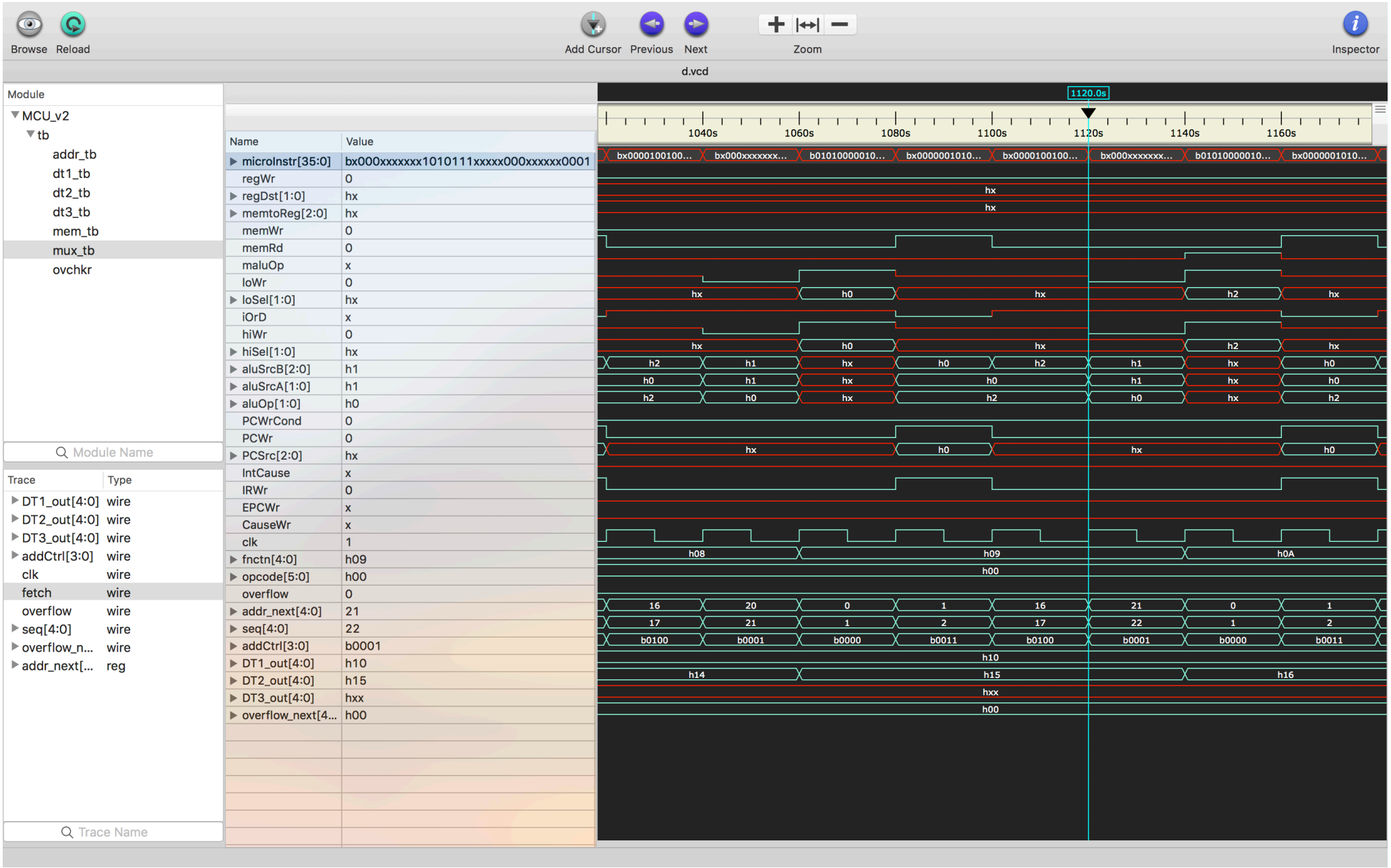


State : 20 MULT WB

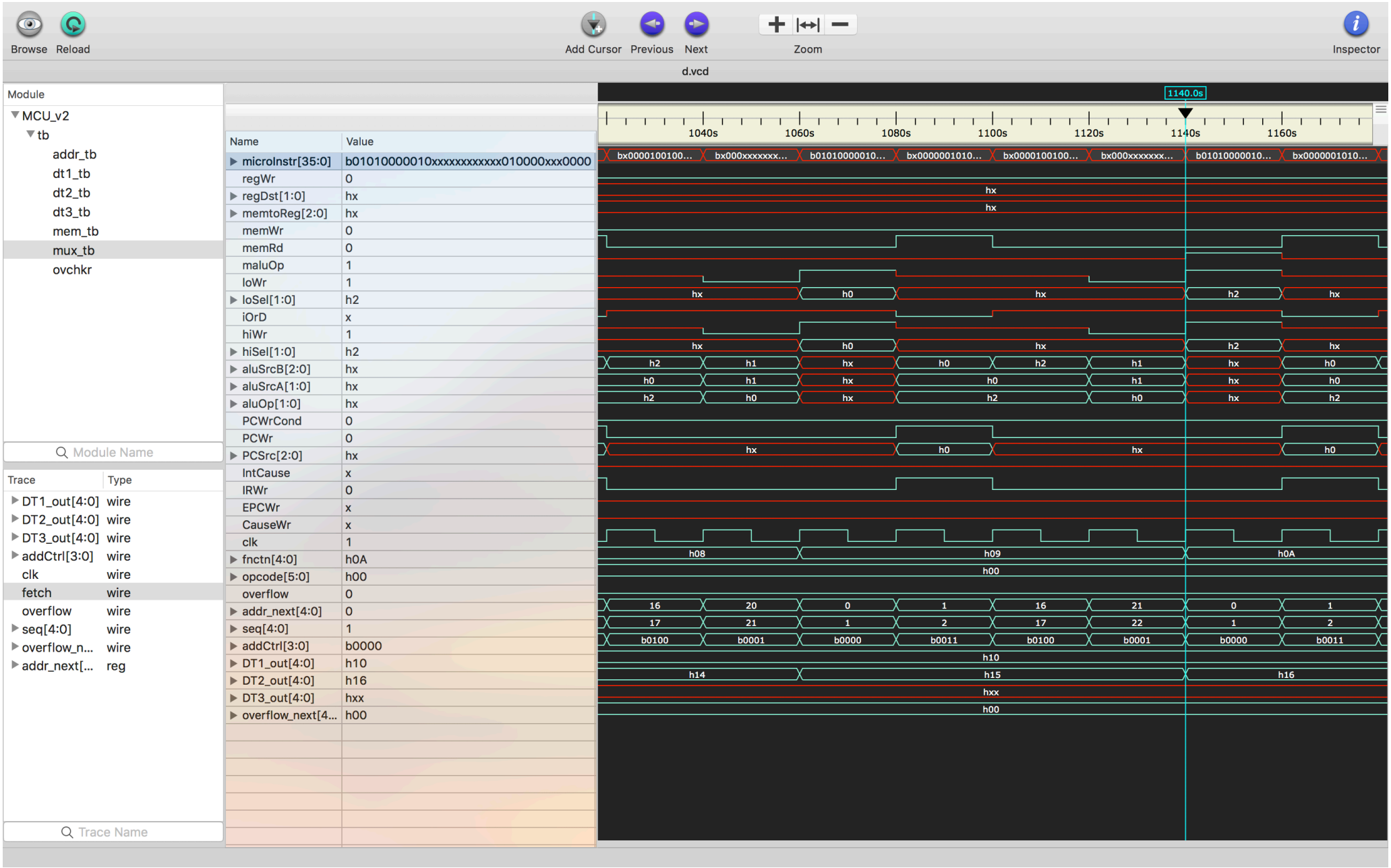


MADD

State : 16 MADD EX

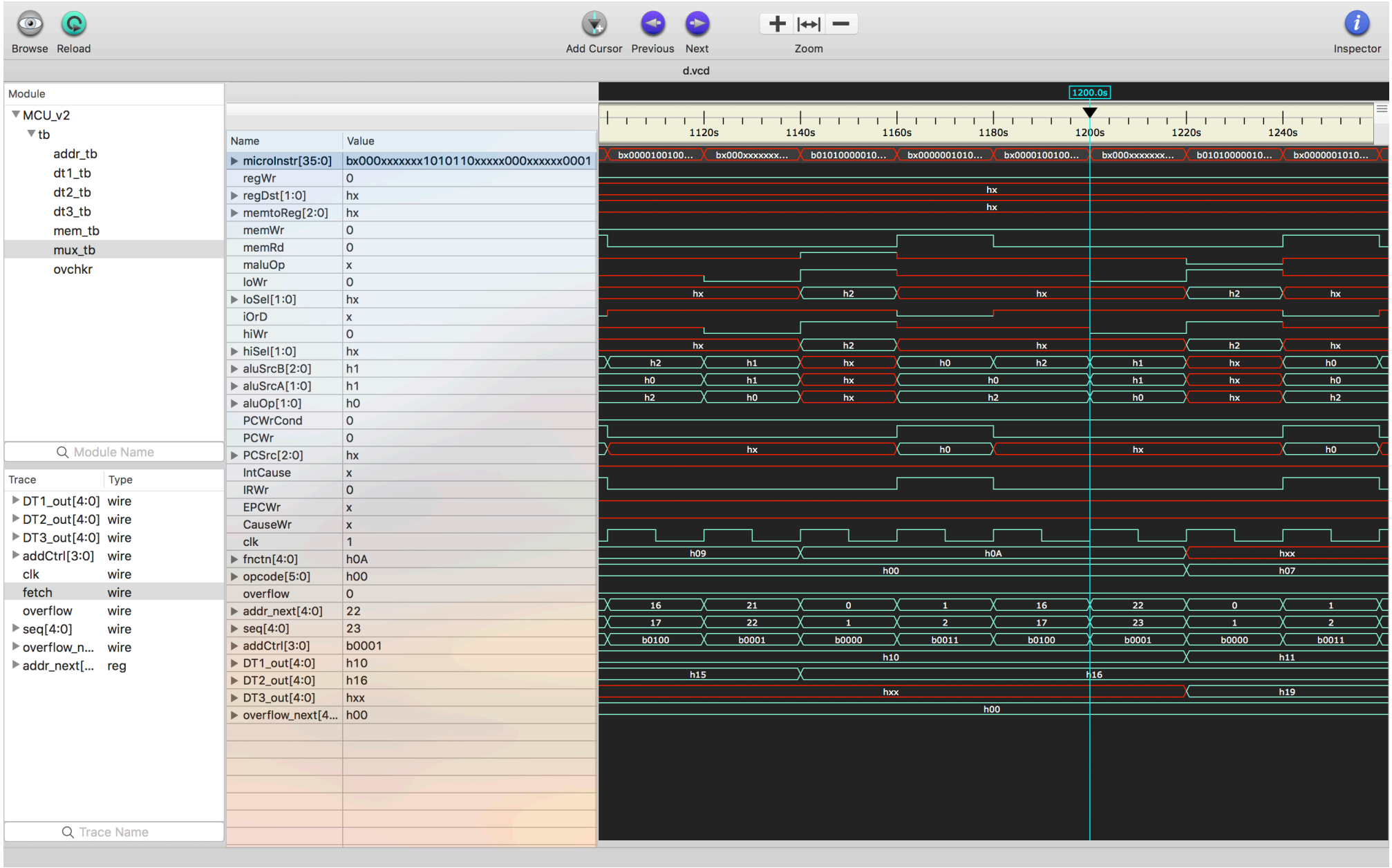


State : 21 MADD WB

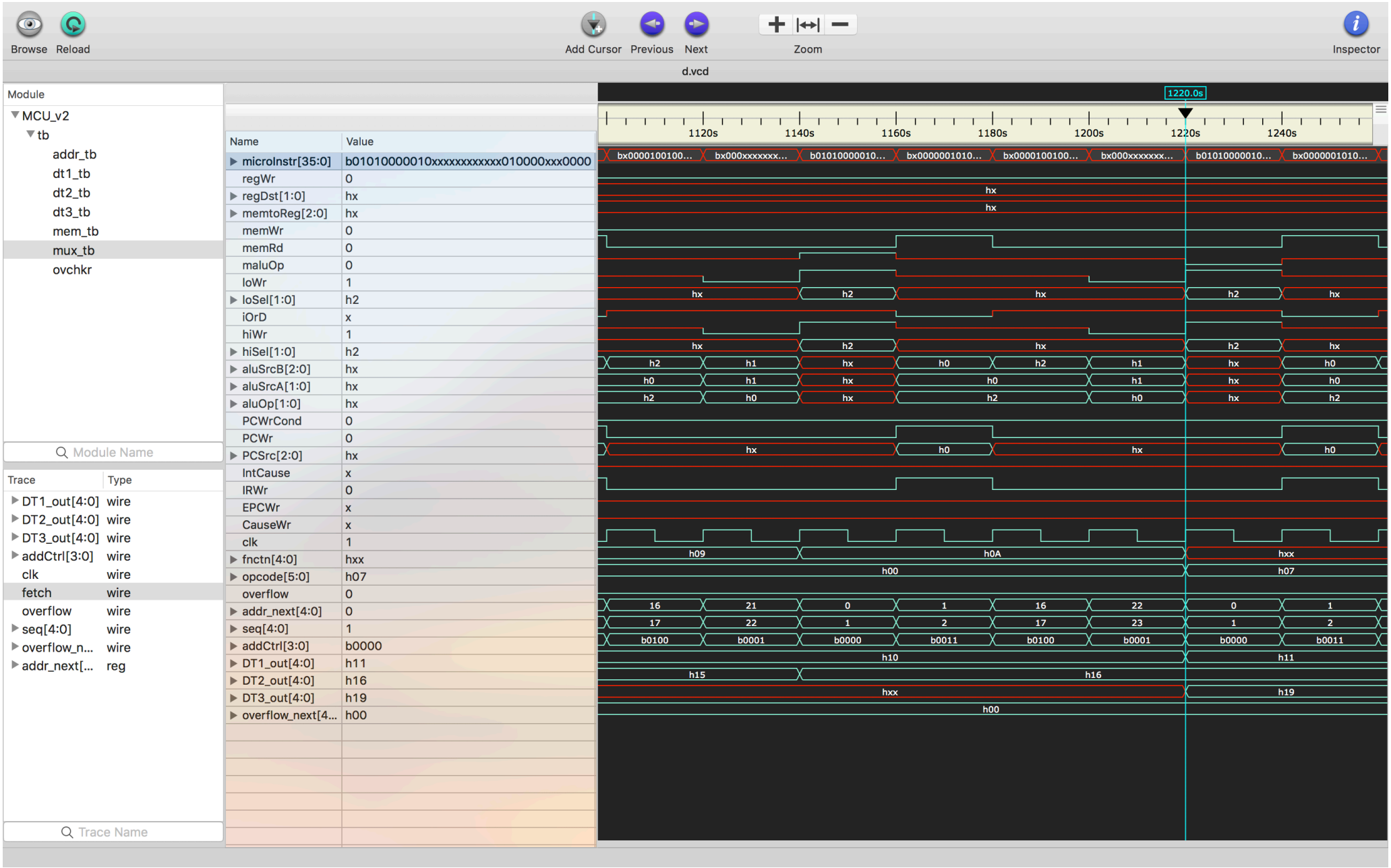


MSUB

State : 16 MSUB EX

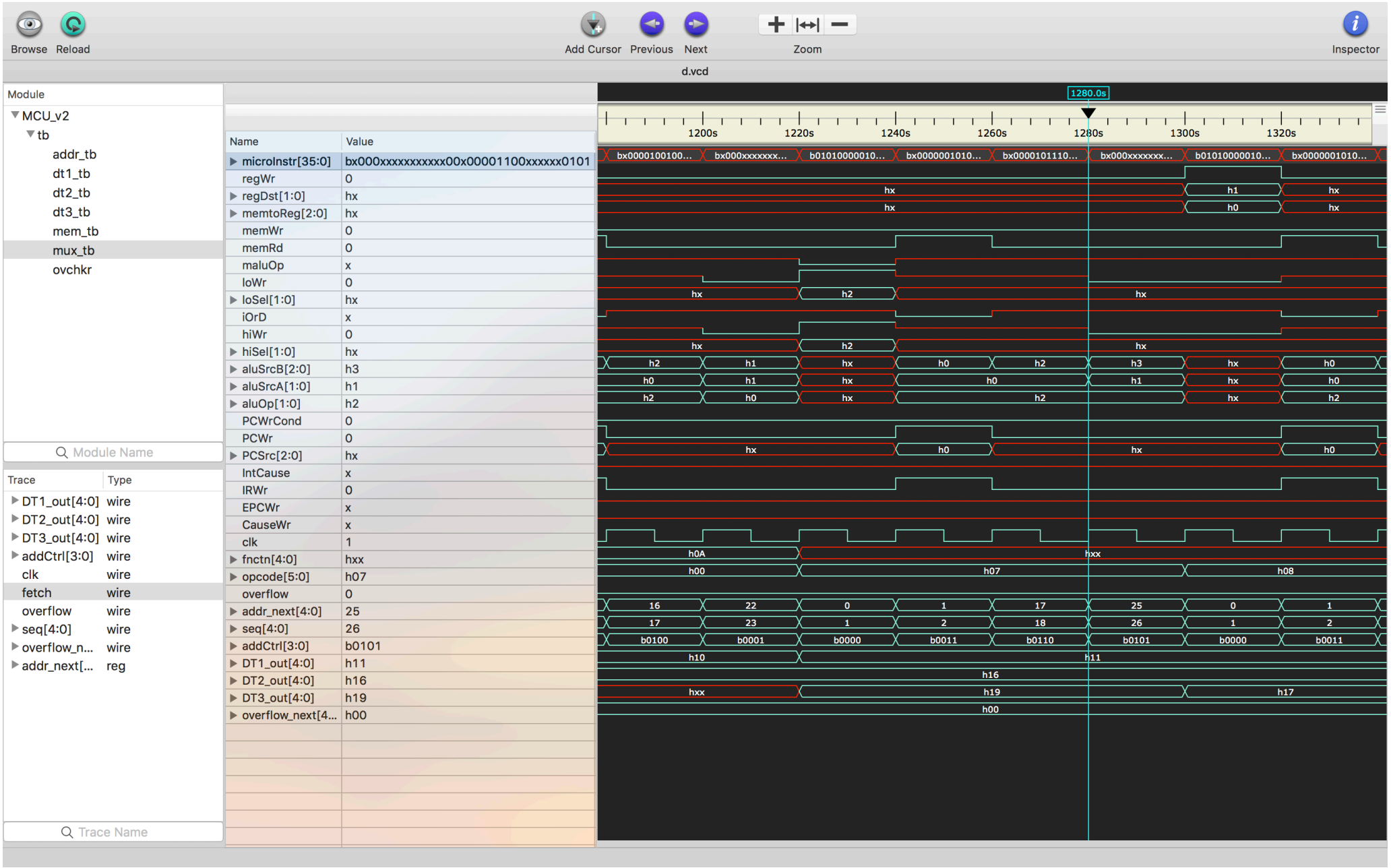


State : 22 MSUB WB

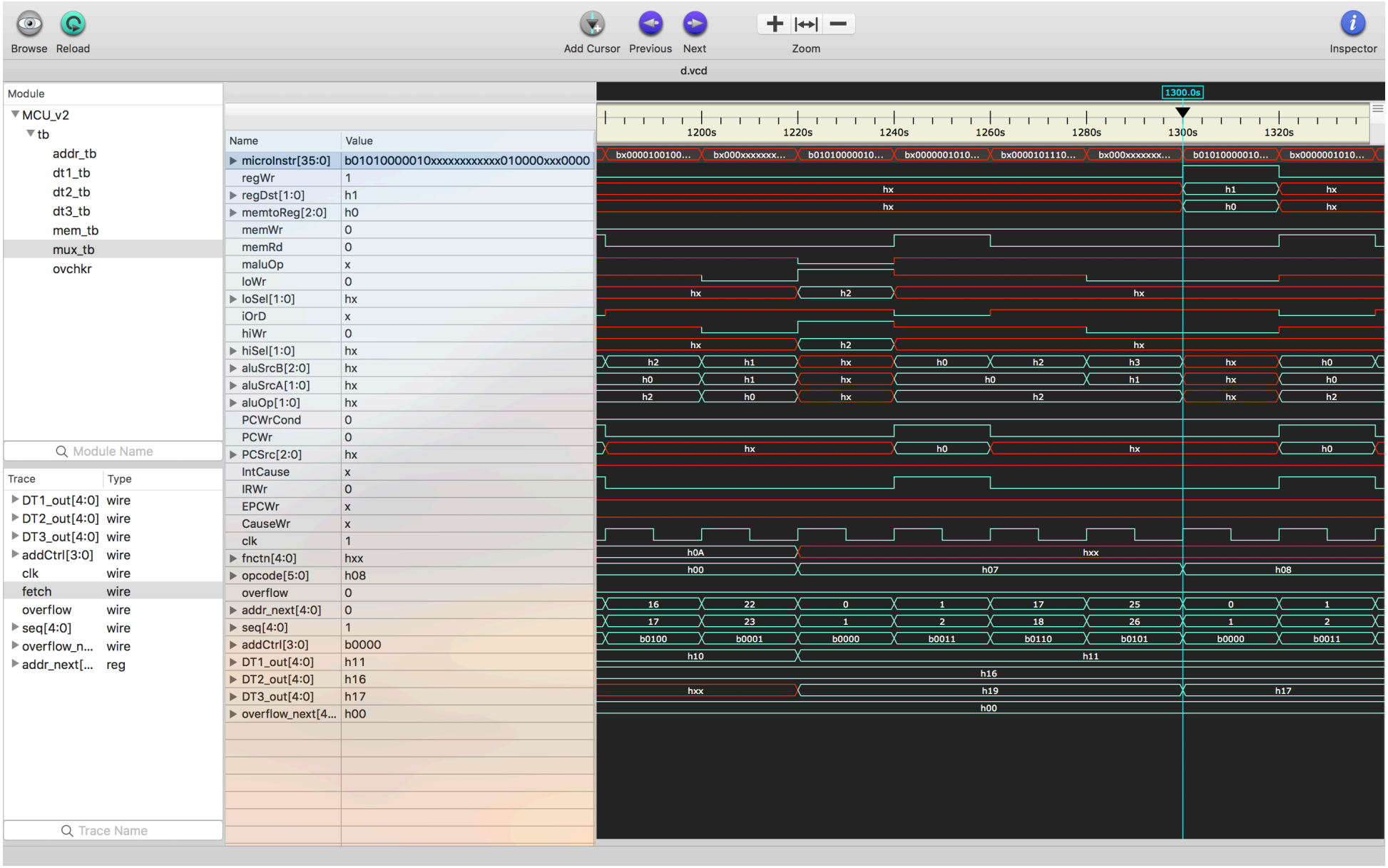


ADDI

State : 17 ADDI EX

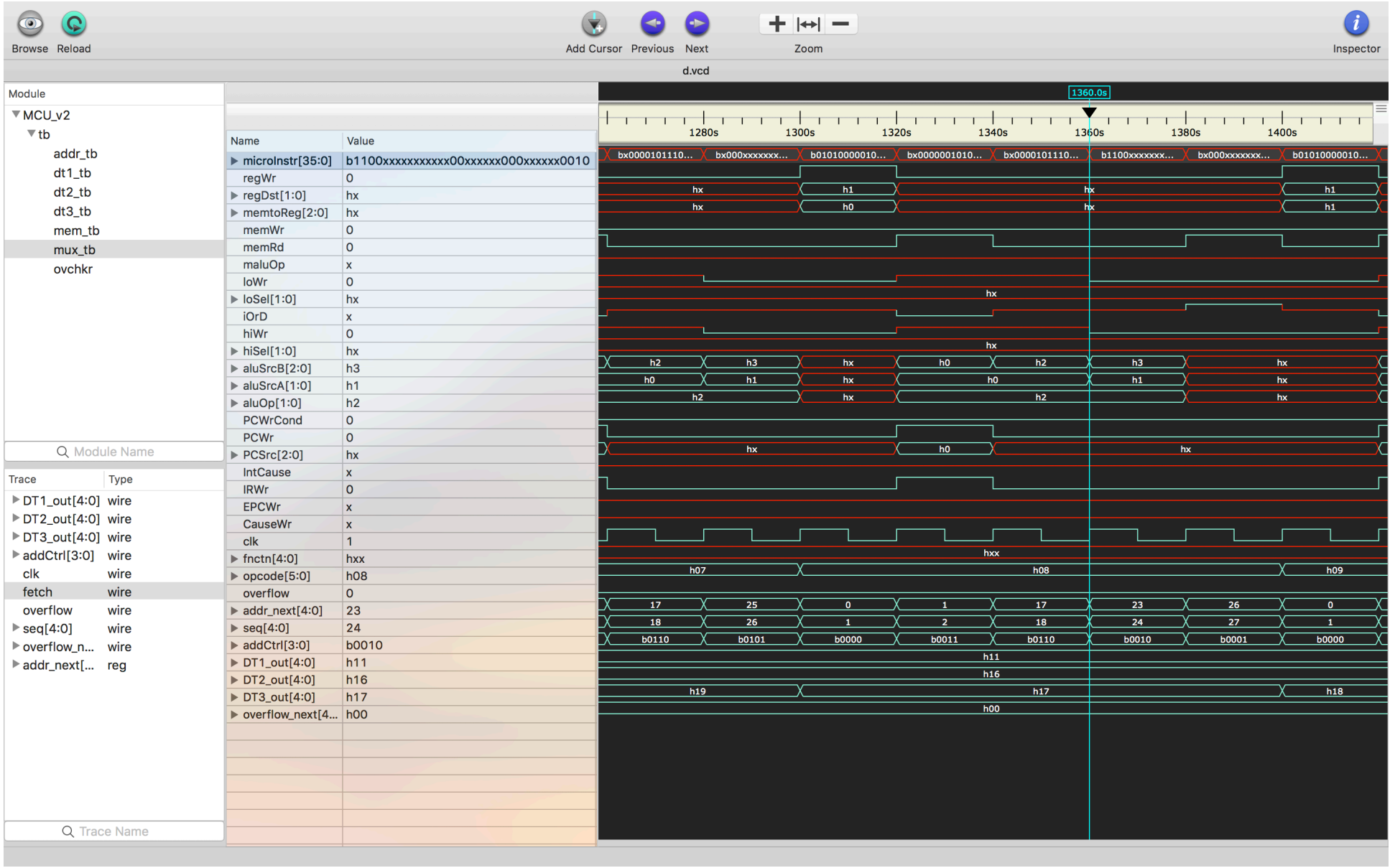


State : 25 ADDI WB

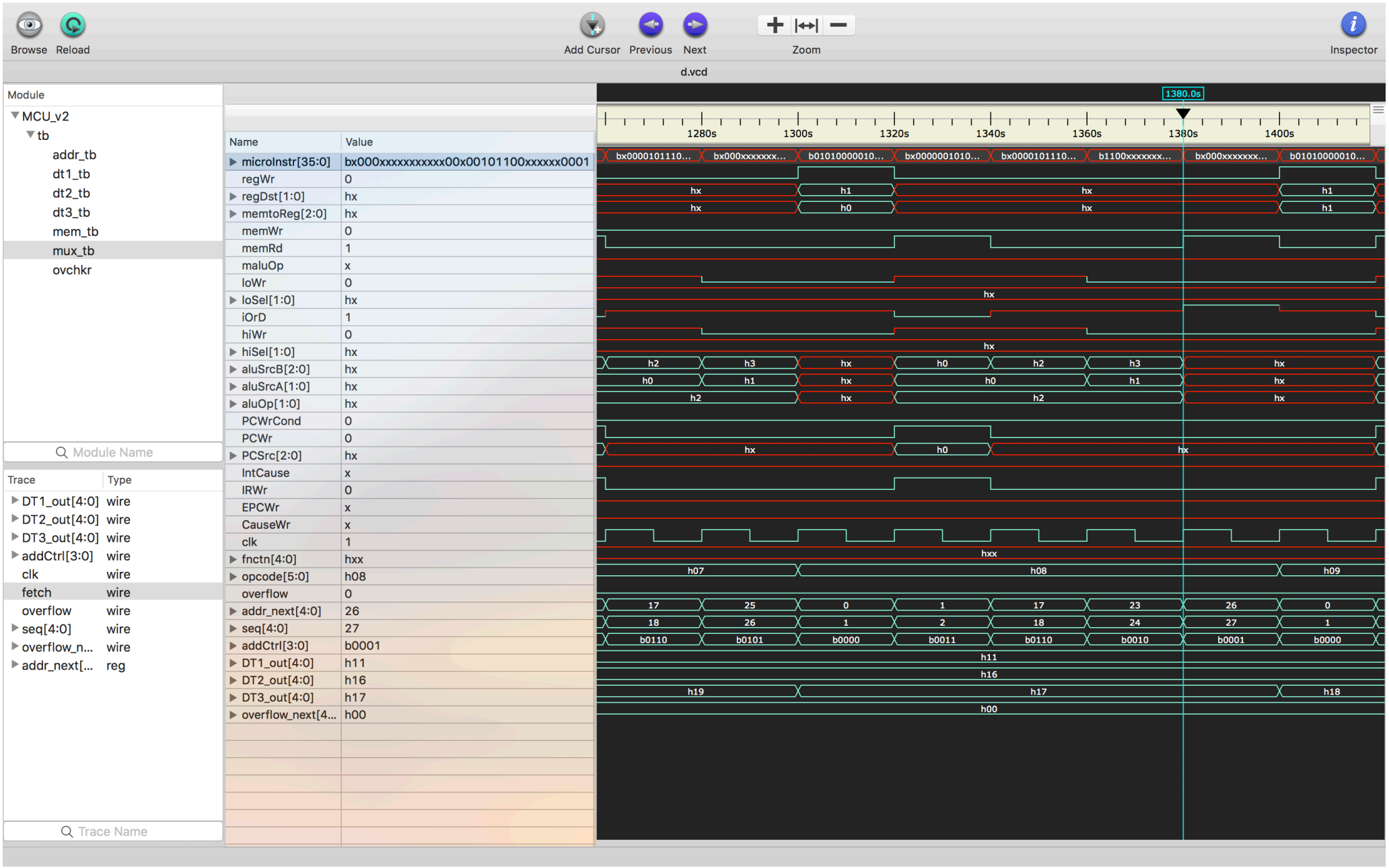


LW

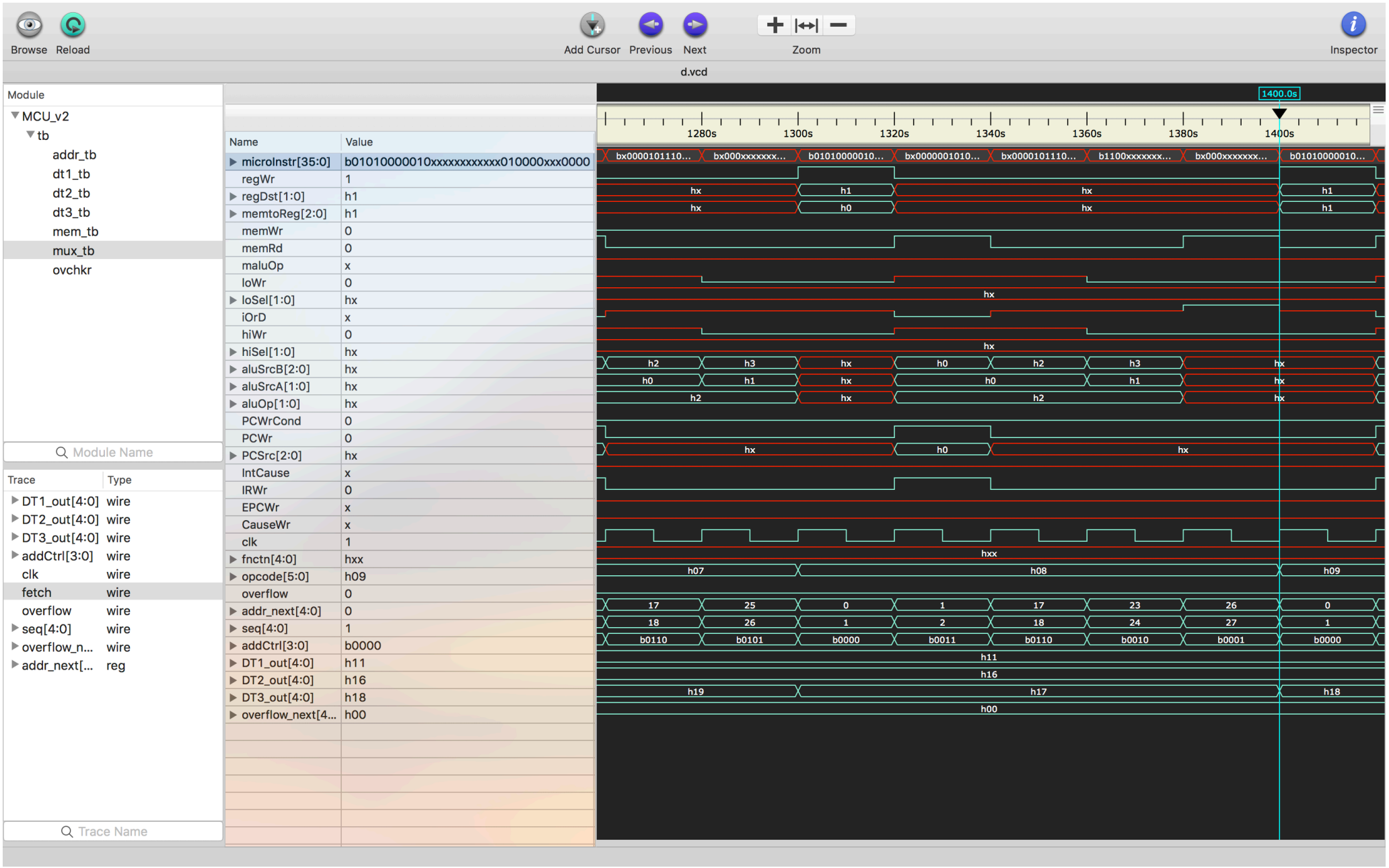
State : 17 LW EX



State : 23 LW MEM

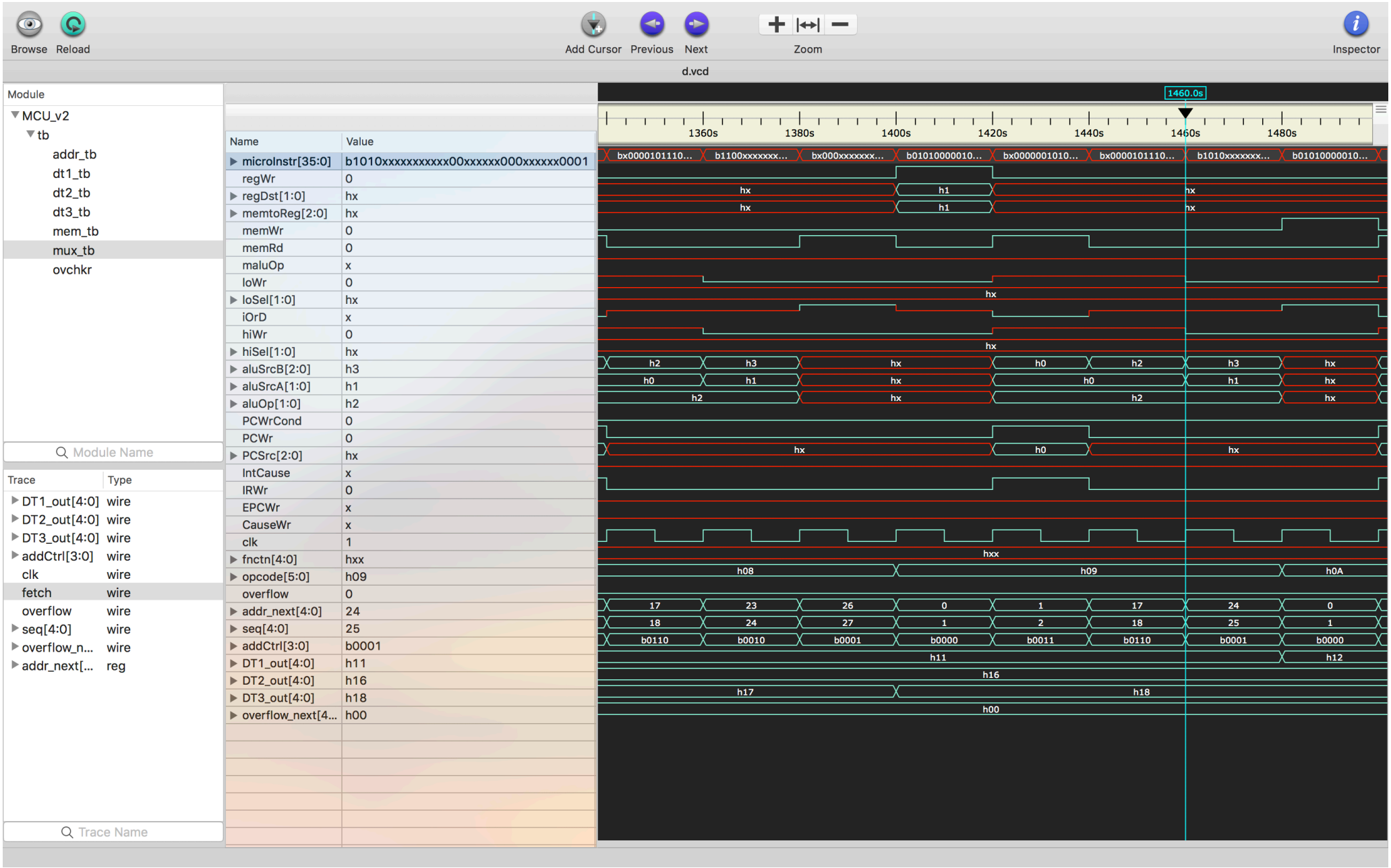


State : 26 LW WB

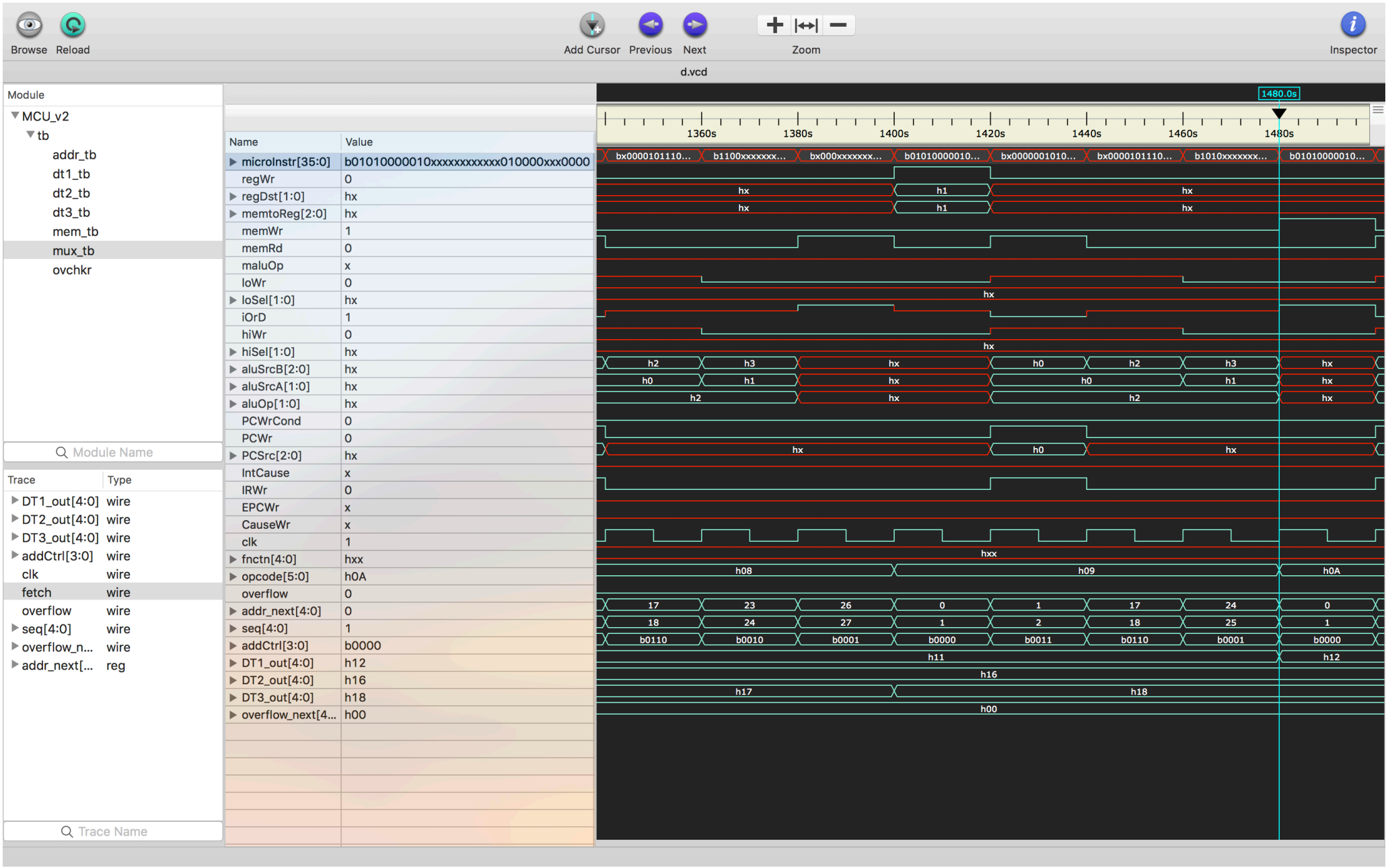


SW

State : 17 SW EX

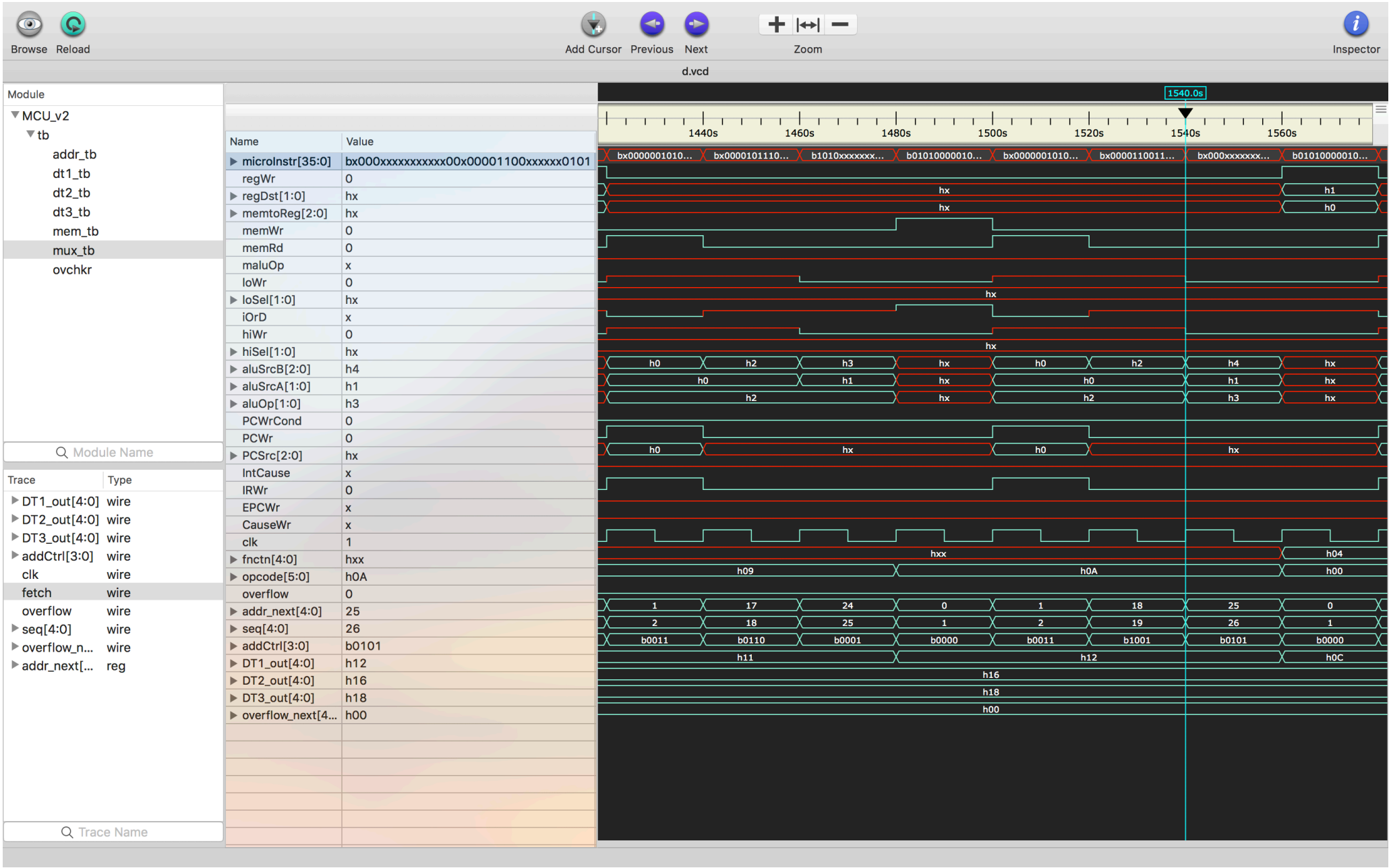


State : 24 SW WB

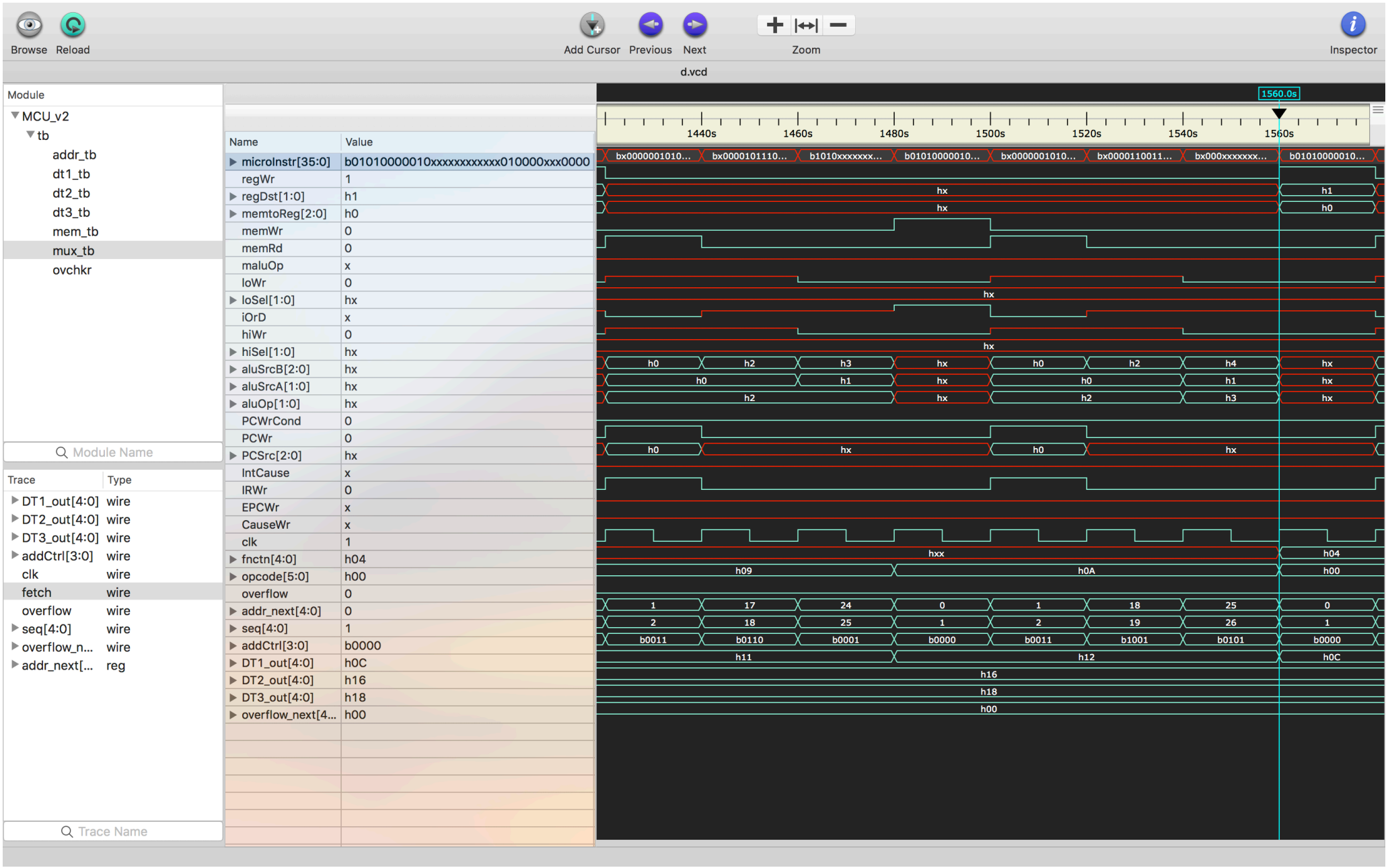


ORI

State : 18 ORI EX



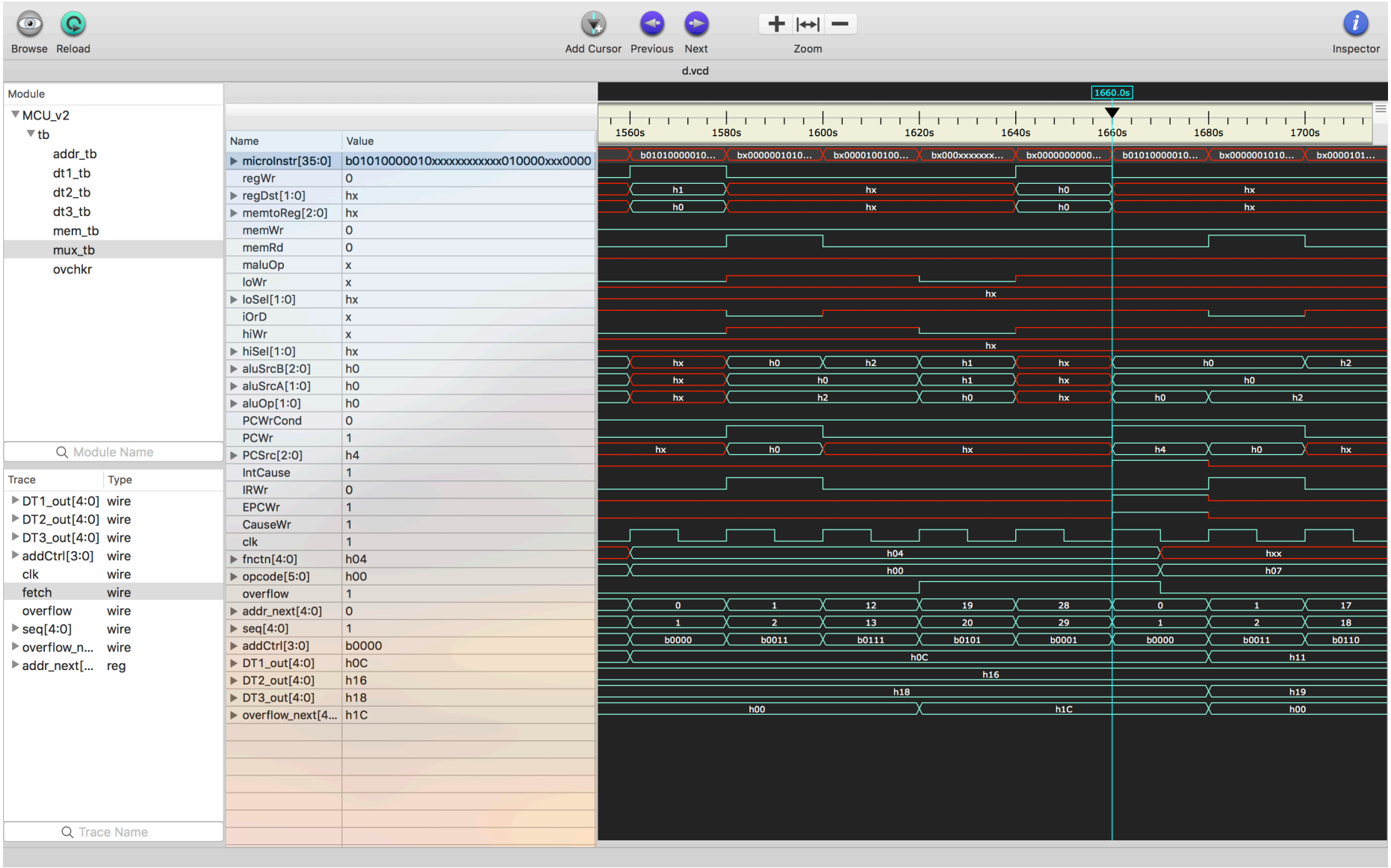
State : 25 ORI WB



ADD with OVERFLOW

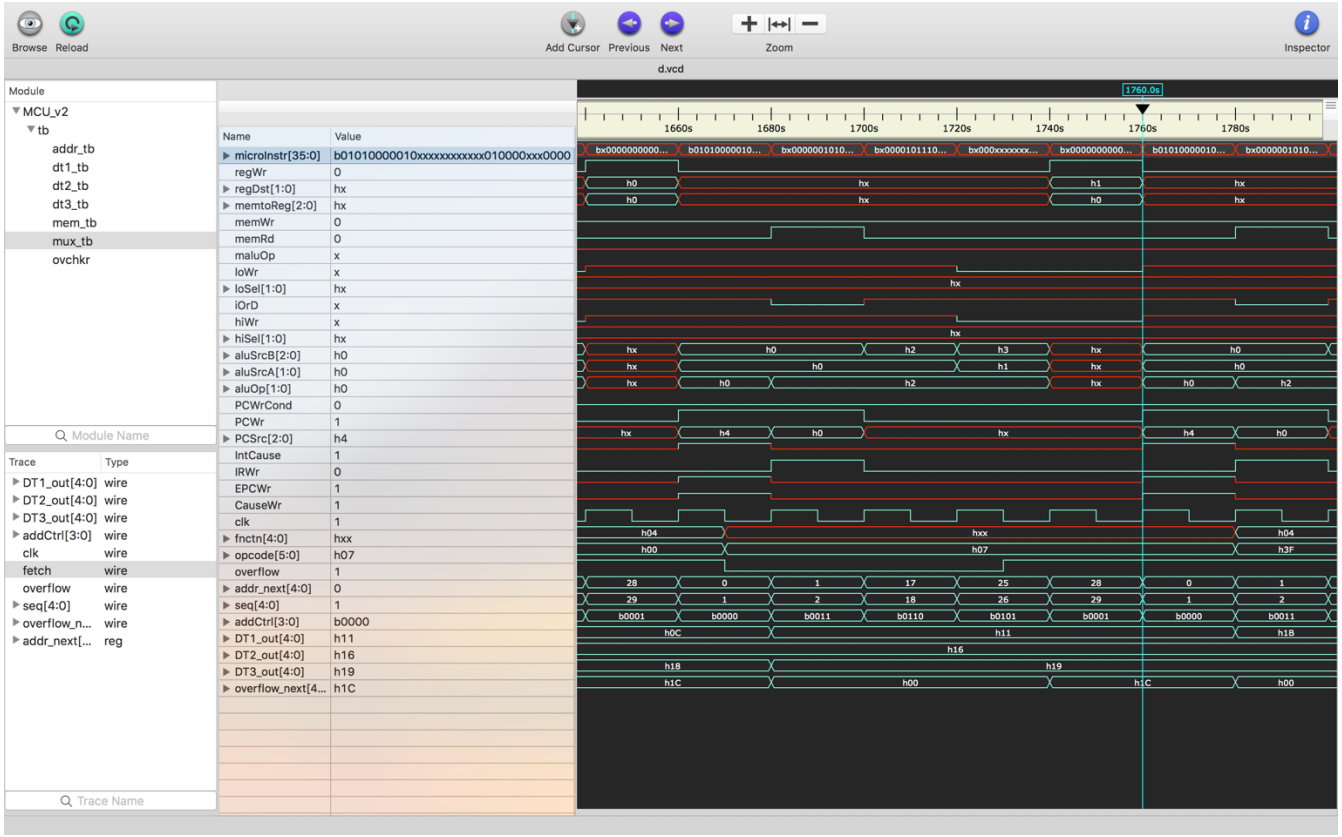
State : 19 ADD WB

In State 12 , ALU sets overflow flag . The control goes to WB state 19 , from there the FSM state is changed to state 28 (ARITHMETIC OVERFLOW) This overflow can occur with add and ,addi instructions.



ADDI with OVERFLOW

Write Back stage of ADDi 25 to Exception Stage 28.



State : 28 OVERFLOW EXCEPTION STATE

This state is created to set

IntCause : Interruption Cause = 1 i.e. Due to Overflow.

CauseWr: ControlSignal to write in Cause Register the IntCause, is set to 1.

EPCWr : To Write the address of instruction where interrupt happened.

AluSrcA = 00 (current PC)

AluSRcB= 000 (4)

AluOP = 00 (Subtract)

PCSRC= 04 = {address for entry of OS to handle interrupt/exception}

Result is Stored in EPC Register

INSTRUCTION UNDEFINED EXCEPTION

After ID stage , if OPCODE of given instruction does not matches any of the given instructions, the next stage is ***STATE 27.***

This state is created to set

IntCause : Interruption Cause = 0 i.e. Due to no instruction defined.

CauseWr: ControlSignal to write in Cause Register the IntCause, is set to 1.

EPCWr : To Write the address of instruction where interrupt happened.

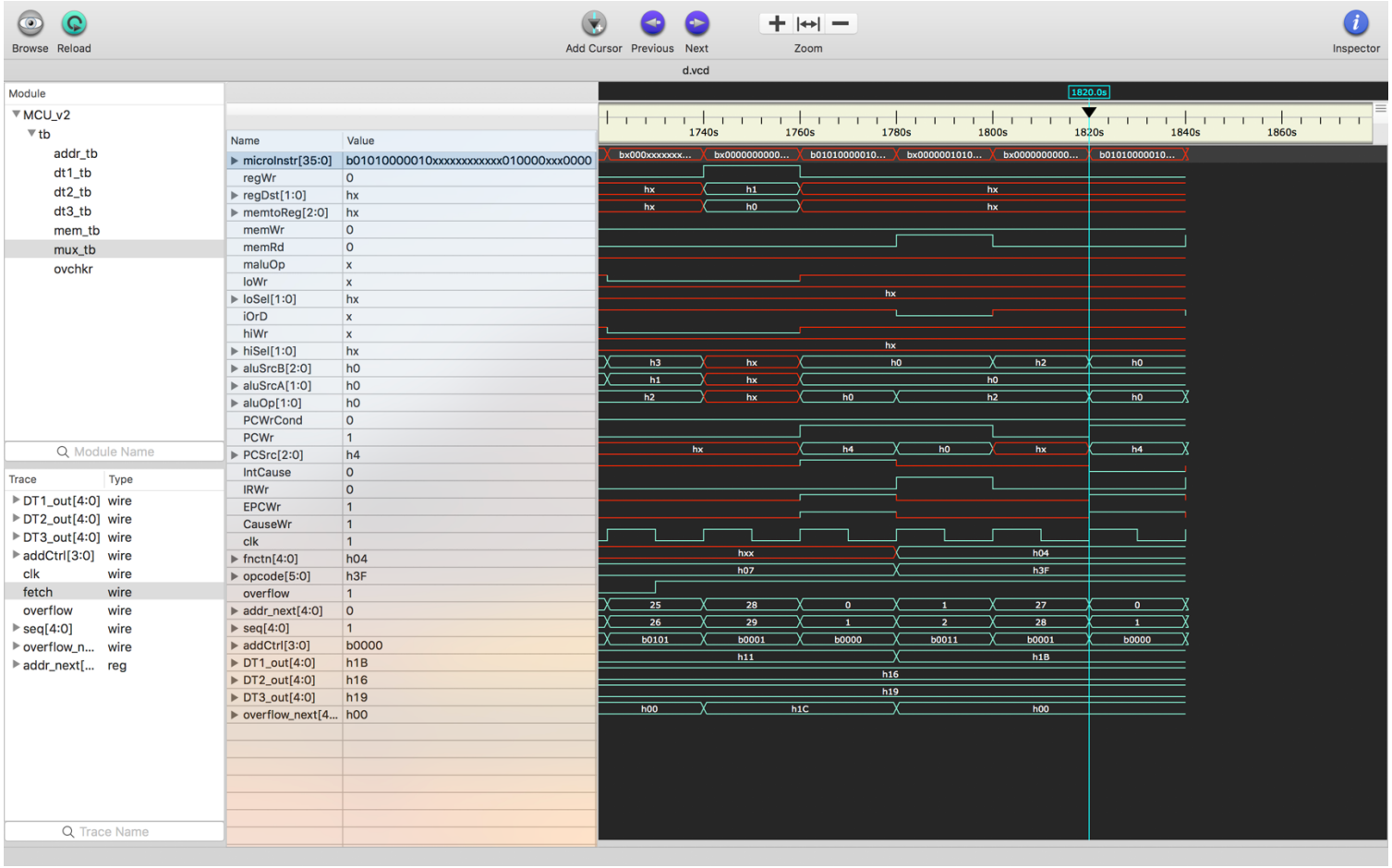
AluSrcA = 00 (current PC)

AluSRcB= 000 (4)

AluOP = 00 (Subtract)

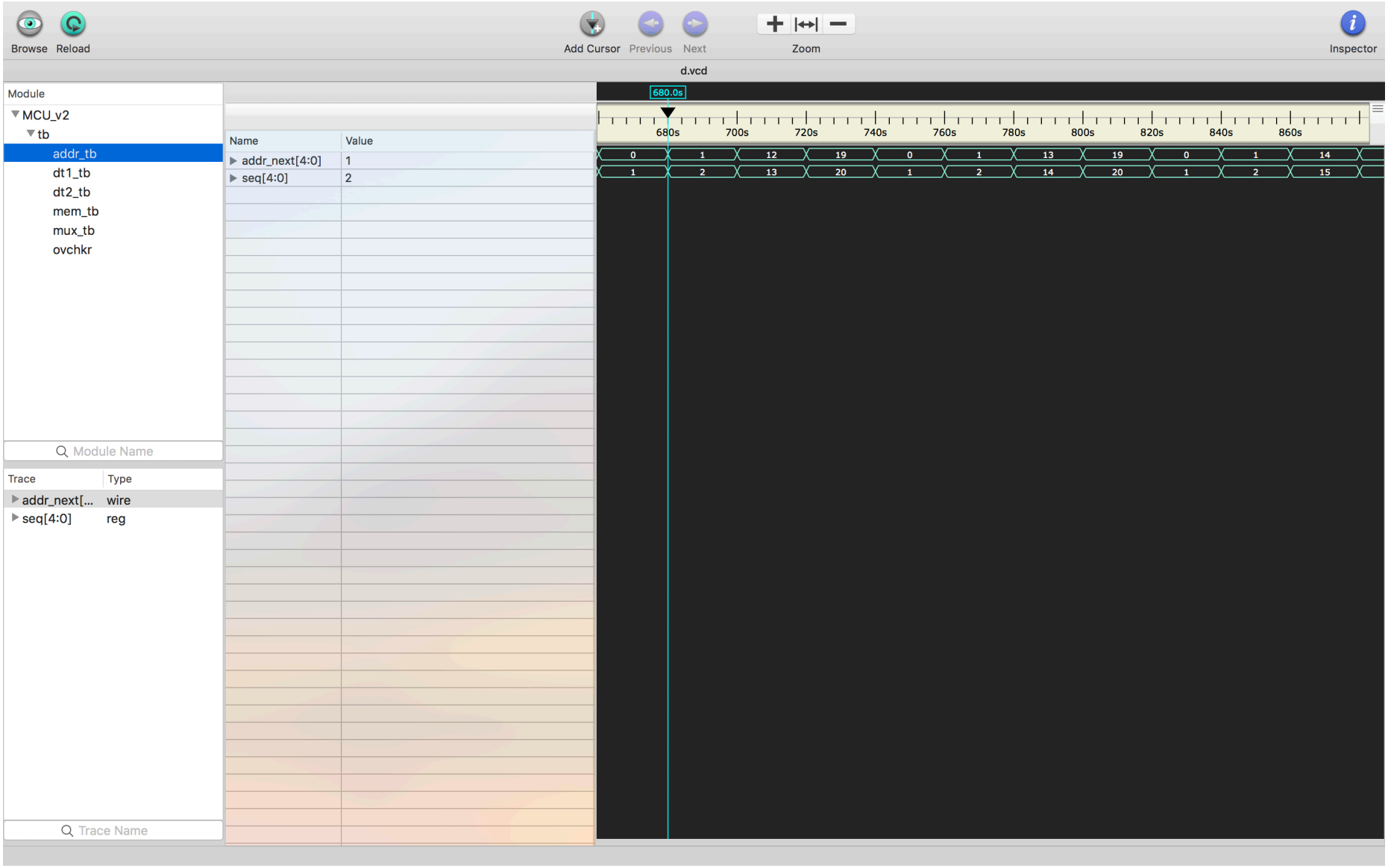
PCSRC= 04 = {address for entry of OS to handle interrupt/exception}

Result is Stored in EPC Register.

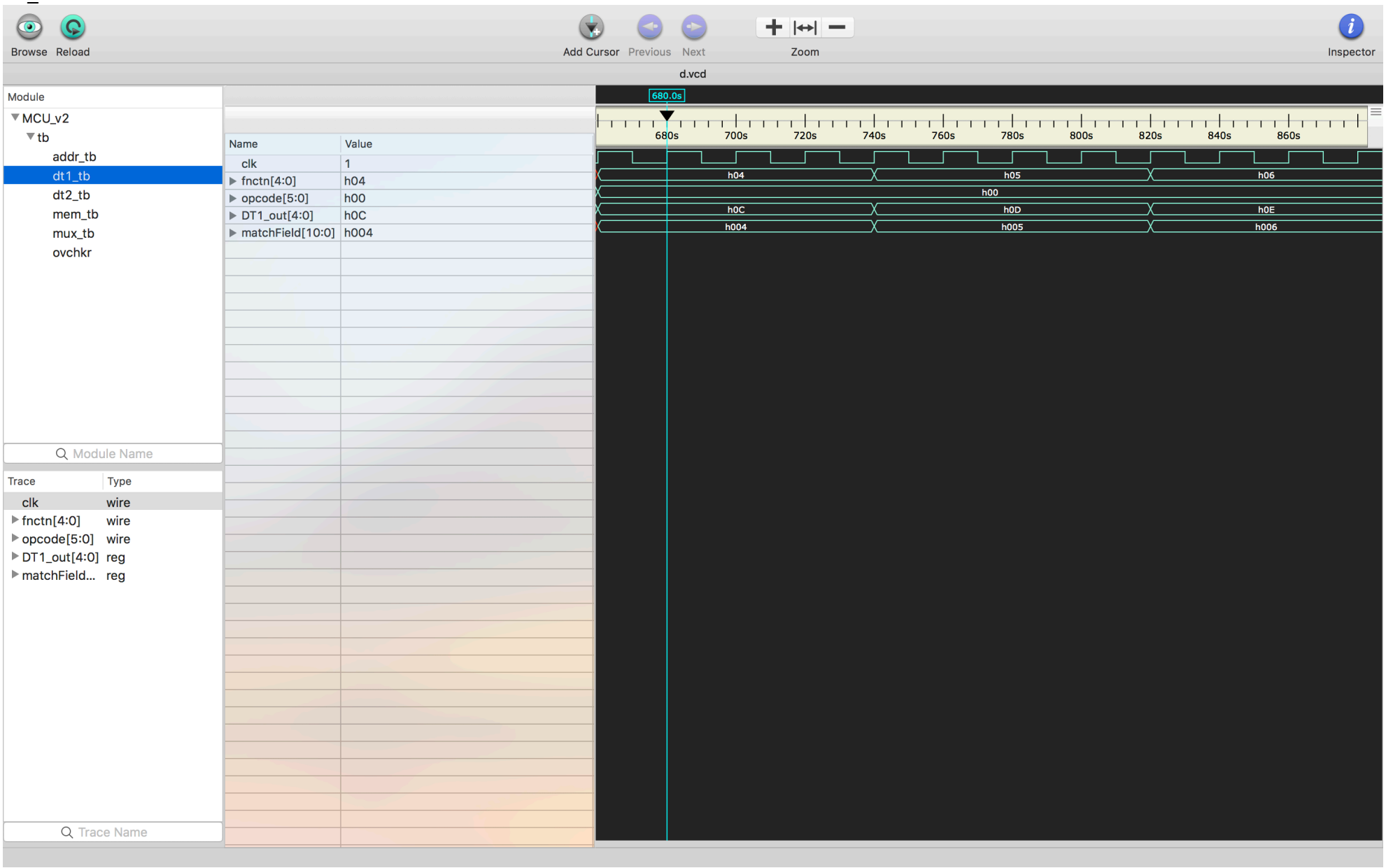


Individual Modules Waveforms

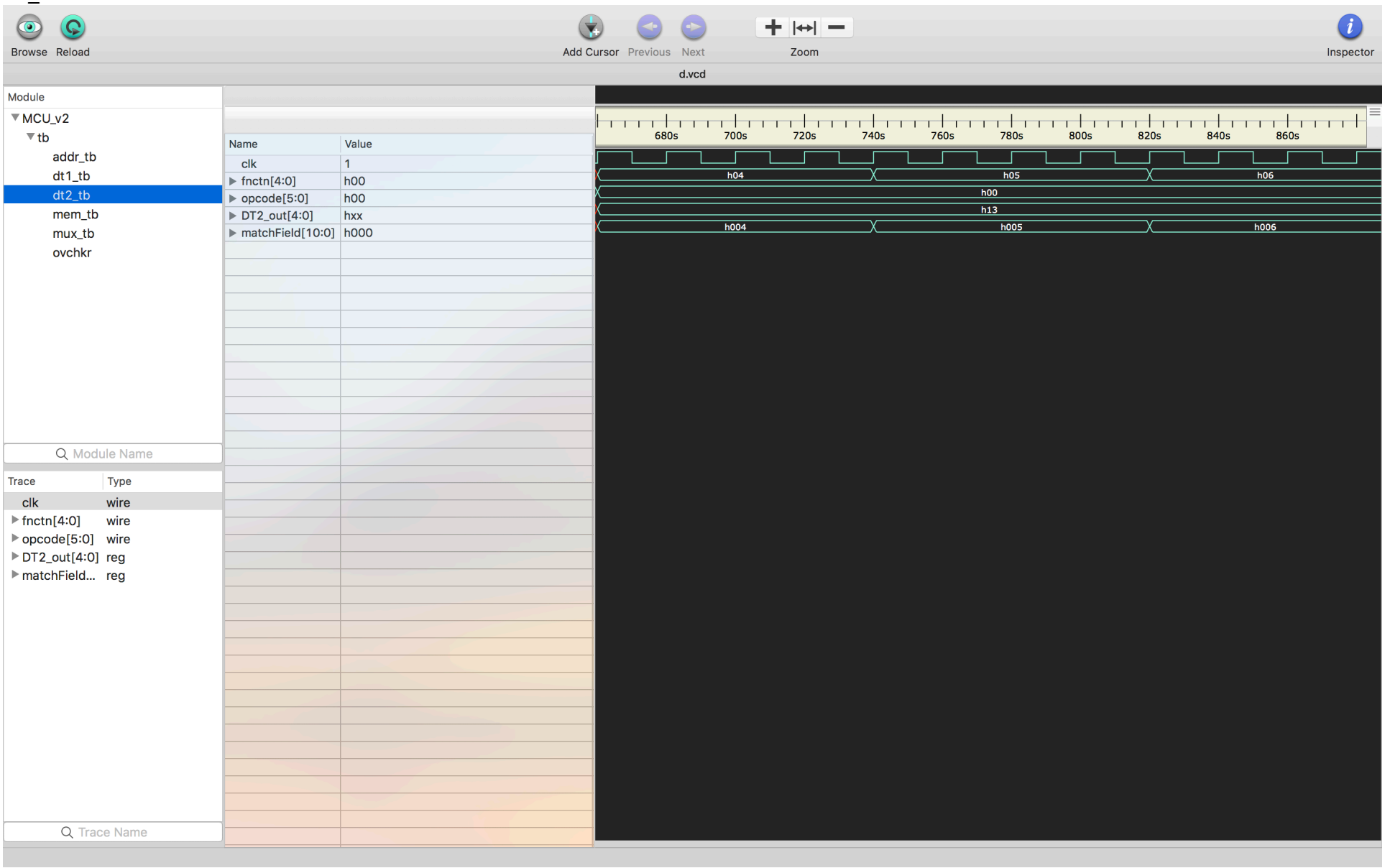
Adder



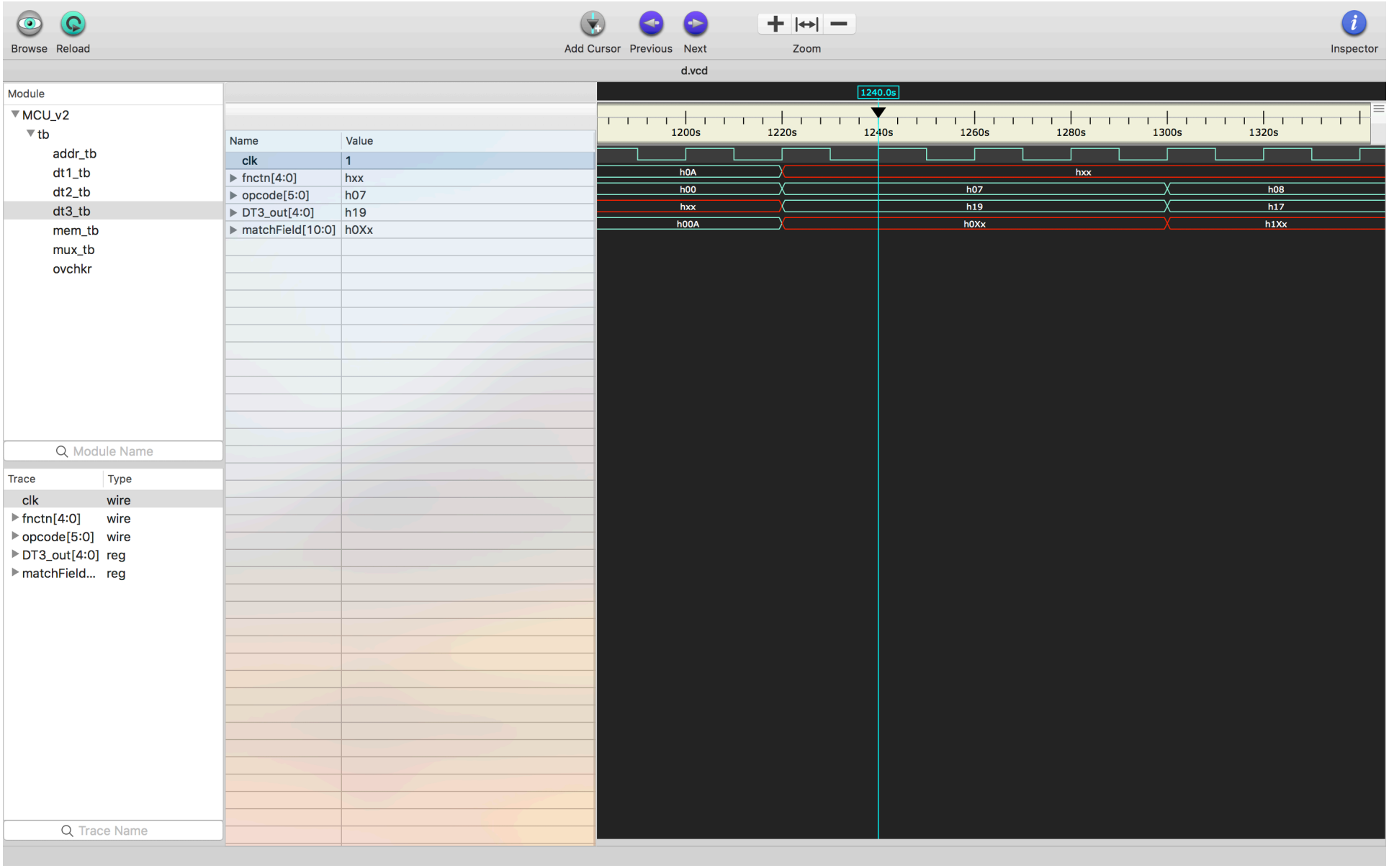
DT_1


Browse
Reload
Add Cursor
Previous
Next
Zoom
Inspector
d.vcd
Module
MCU_v2
tb
addr_tb
dt1_tb
dt2_tb
mem_tb
mux_tb
ovchkr
Module Name
Trace
Type
clk
wire
fnctn[4:0]
wire
opcode[5:0]
wire
DT1_out[4:0]
reg
matchField...
reg
Trace Name
Name
Value
clk
1
fnctn[4:0]
h04
opcode[5:0]
h00
DT1_out[4:0]
h0C
matchField[10:0]
h004
680.0s
680s
700s
720s
740s
760s
780s
800s
820s
840s
860s
h04
h05
h06
h00
h0C
h0D
h0E
h004
h005
h006

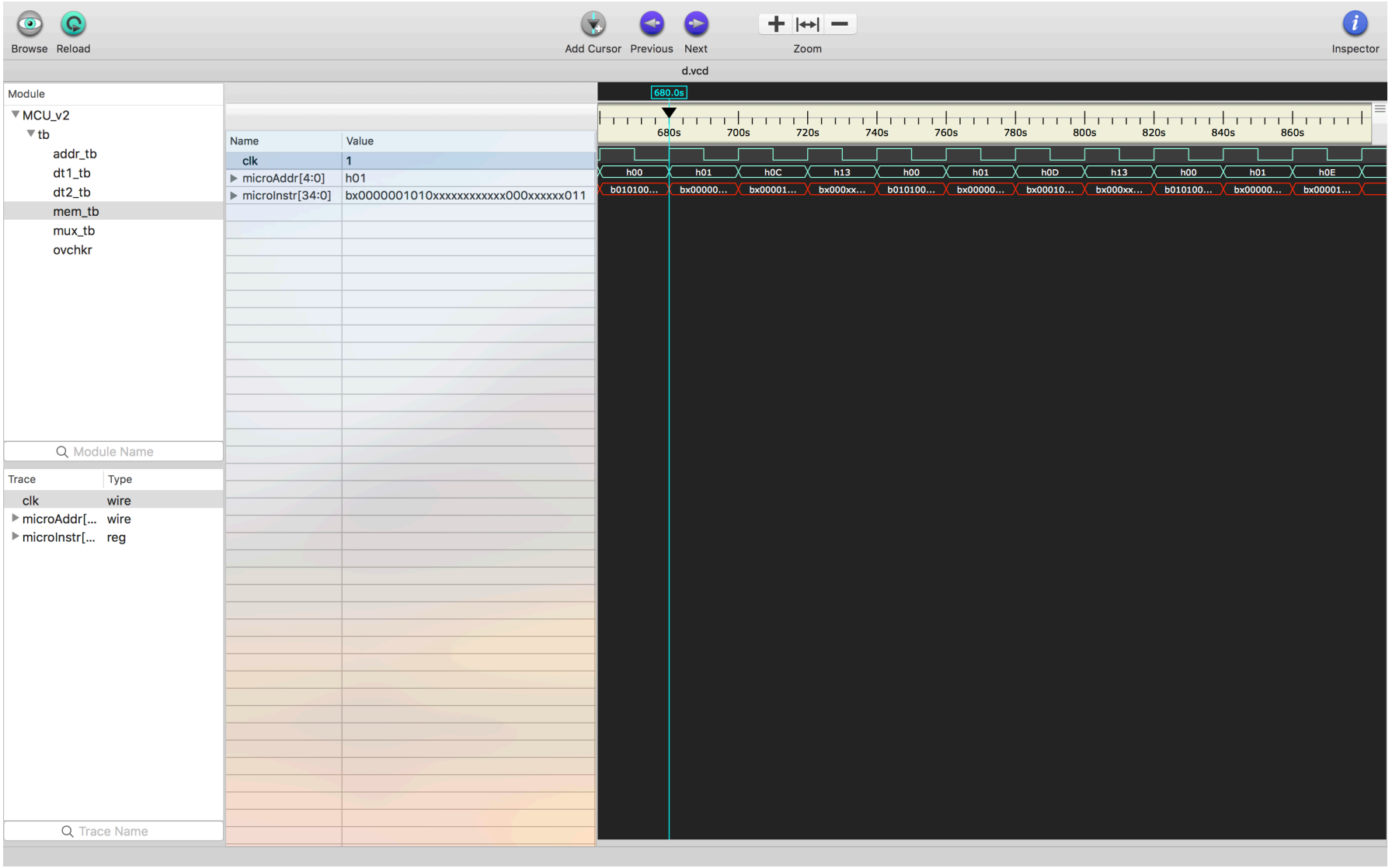
DT_2


Browse
Reload
Add Cursor
Previous
Next
Zoom
Inspector
d.vcd
Module
MCU_v2
tb
addr_tb
dt1_tb
dt2_tb
mem_tb
mux_tb
ovchkr
Module Name
Trace
Type
clk
wire
fnctn[4:0]
wire
opcode[5:0]
wire
DT2_out[4:0]
reg
matchField...
reg
Trace Name
Name
Value
clk
1
fnctn[4:0]
h00
opcode[5:0]
h00
DT2_out[4:0]
hxx
matchField[10:0]
h000
680s
700s
720s
740s
760s
780s
800s
820s
840s
860s
h04
h05
h06
h00
h13
h004
h005
h006

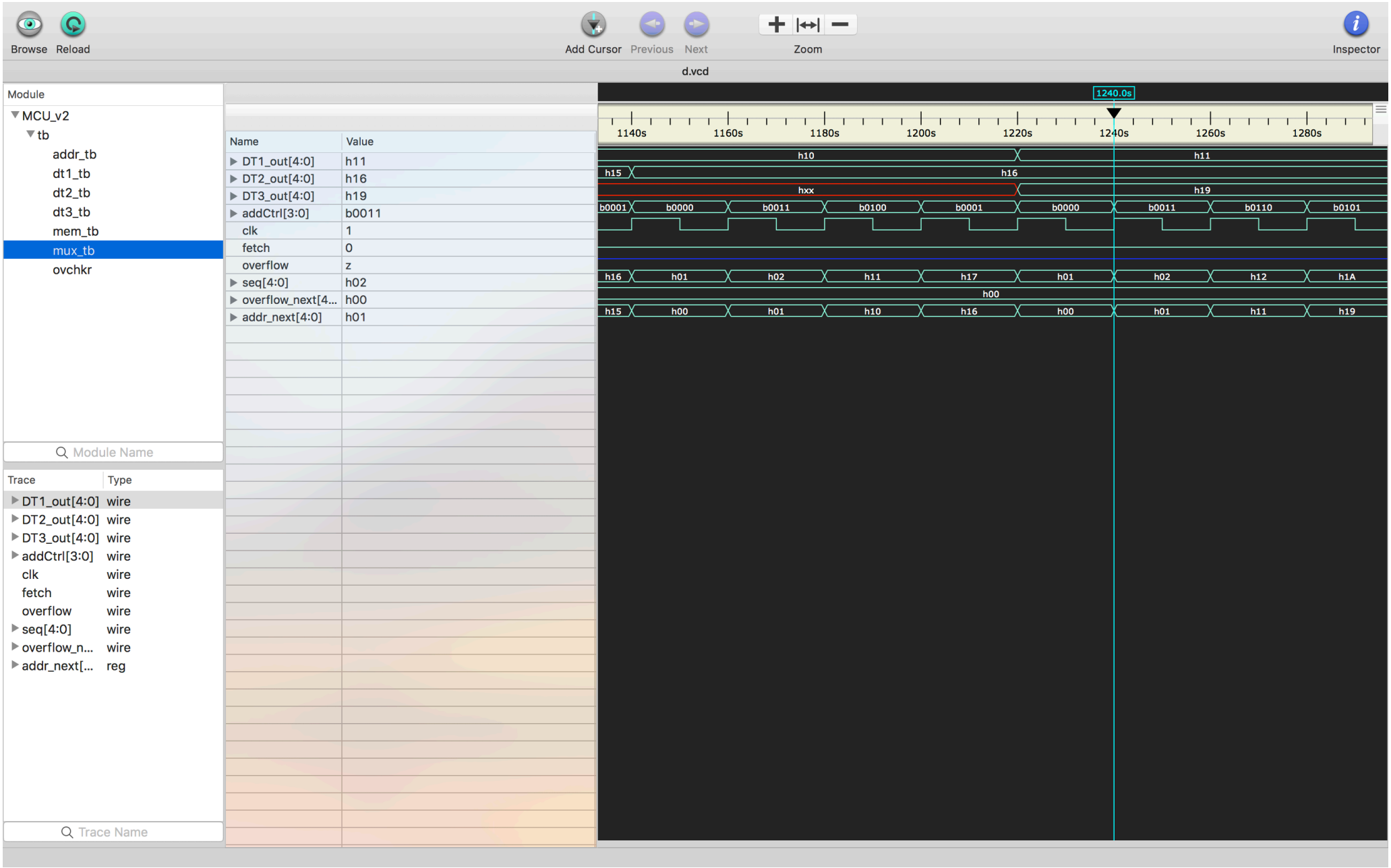
DT_3


Browse
Reload
Add Cursor
Previous
Next
Zoom
Inspector
d.vcd
Module
MCU_v2
tb
addr_tb
dt1_tb
dt2_tb
dt3_tb
mem_tb
mux_tb
ovchkr
Module Name
Trace
Type
clk
wire
fnctn[4:0]
wire
opcode[5:0]
wire
DT3_out[4:0]
reg
matchField...
reg
Trace Name
Name
Value
clk
1
fnctn[4:0]
hxx
opcode[5:0]
h07
DT3_out[4:0]
h19
matchField[10:0]
h0Xx
1240.0s
1200s
1220s
1240s
1260s
1280s
1300s
1320s
h0A
hxx
h00
h07
h08
hxx
h19
h17
h00A
h0Xx
h1Xx

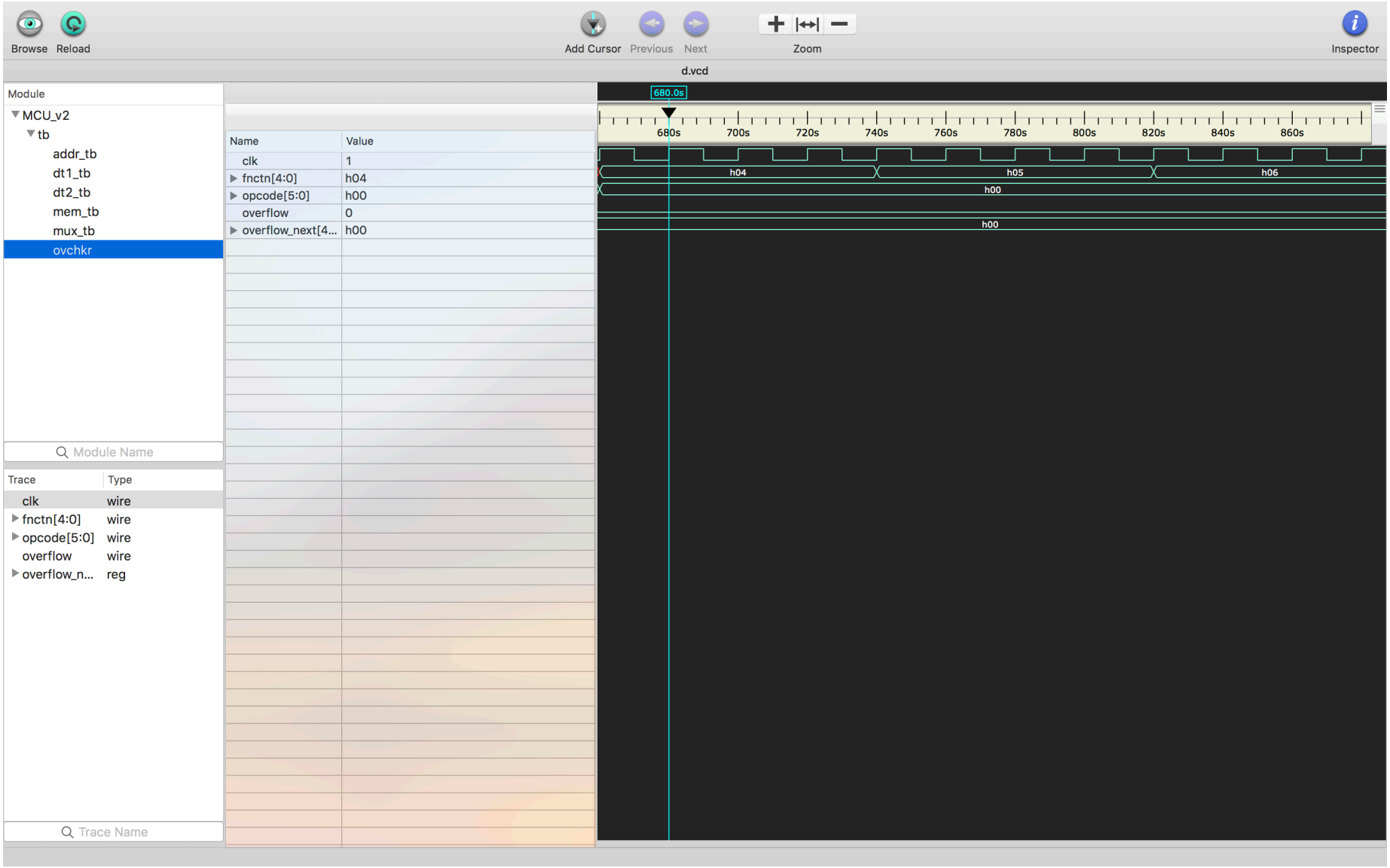
Microinstruction Memory



Multiplexer



Overflow Checker



Microcontroller