Intro. to Synopsys Verdi Debug System

Presenter: Jui-Huang Tsai

Advisor: Prof. Chen-Yi Lee

2023.10.04

Outline

- Startup Verdi with Simulation Tool
- Verdi’s Feature
  - nTrace
  - nWave
  - nMemory
  - nState

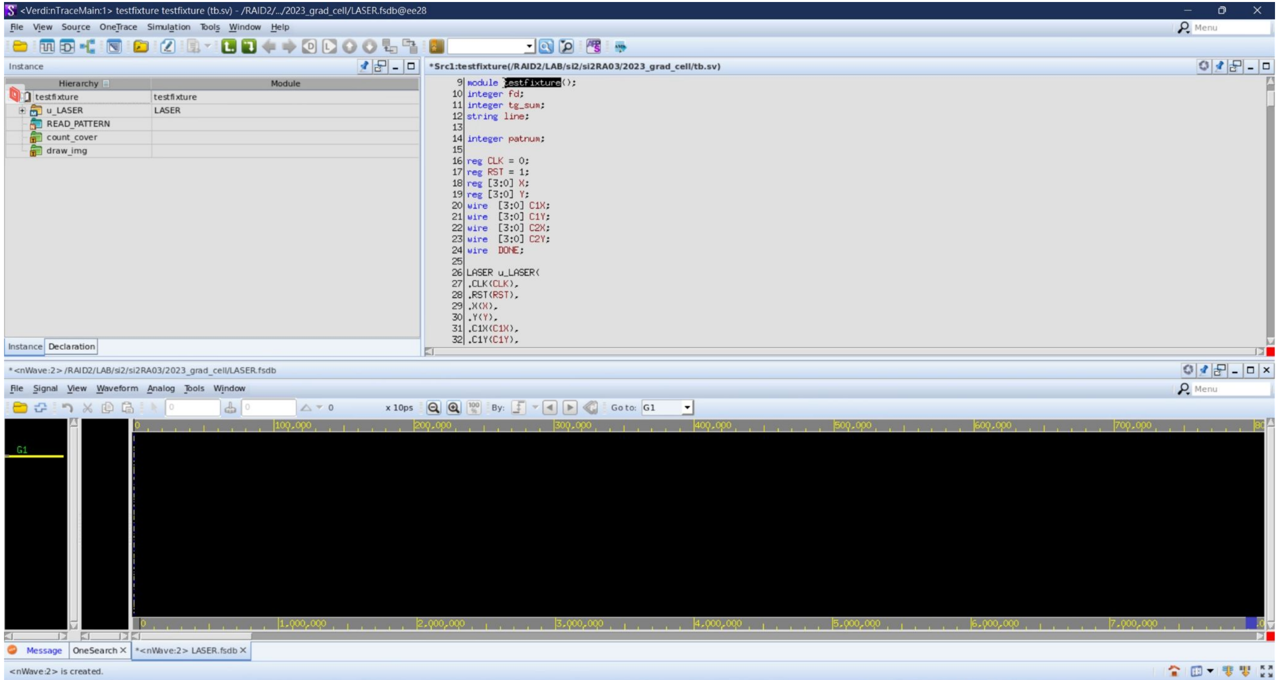
Startup

- ./01_run_vcs_rtl for dumping .fsdb
- ./04_verdi for starting Verdi

Verdi’s Features

- RTL debug tool developed by Synopsys, it supports
    - Verilog
    - SystemVerilog
    - VHDL
- Features
    - nTrace
    - nWave
    - nMemory
    - nState

Startup



<Verdi:nTraceMain:1> testfixture testfixture (tb.sv) - /RAID2/.../2023_grad_cell/LASER.fsdb@ee28

Hierarchy	Module
testfixture	testfixture
u_LASER	LASER
READ_PATTERN	
count_cover	
draw_img	

*Src1:testfixture(/RAID2/LAB/si2/si2RA03/2023_grad_cell/tb.sv)

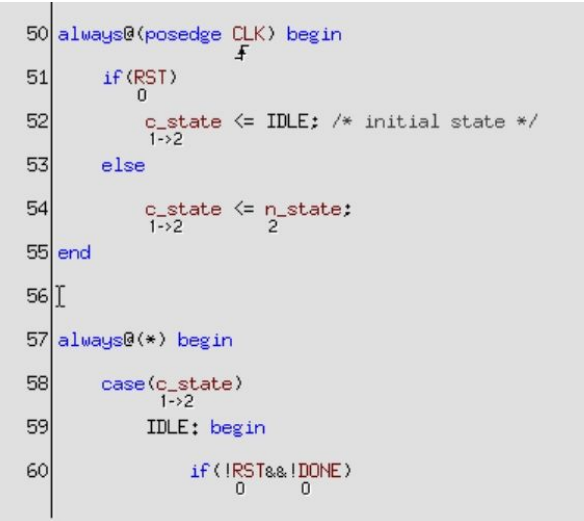
```
module testfixture();
integer fd;
integer tg_sum;
string line;

integer patnum;

reg CLK = 0;
reg RST = 1;
reg [3:0] X;
reg [3:0] Y;
wire  [3:0] C1X;
wire  [3:0] C1Y;
wire  [3:0] C2X;
wire  [3:0] C2Y;
wire  DONE;

LASER u_LASER(
.CLK(CLK),
.RST(RST),
.X(X),
.Y(Y),
.C1X(C1X),
.C1Y(C1Y),
```

*<nWave:2> /RAID2/LAB/si2/si2RA03/2023_grad_cell/LASER.fsdb

<nWave:2> is created.

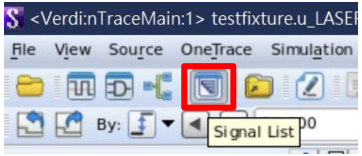
nTrace: Show Signal’s Value

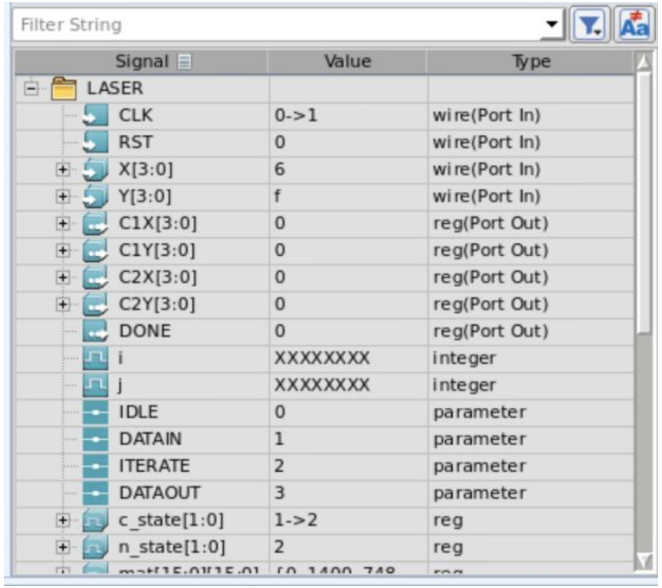
- Show the value of signal (Command: “x”)



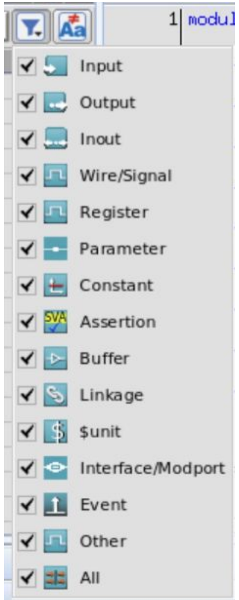
```
always@(posedge CLK) begin
                 ↓
    if(RST)
       0
        c_state <= IDLE; /* initial state */
        1->2
    else

        c_state <= n_state;
        1->2       2
end


always@(*) begin

    case(c_state)
         1->2
        IDLE: begin

            if(!RST&&!DONE)
                0     0
```

nTrace: Signal List



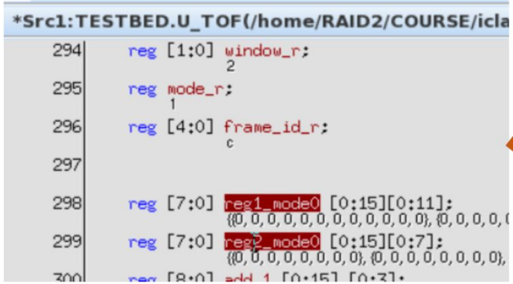


Signal	Value	Type
LASER		
CLK	0->1	wire(Port In)
RST	0	wire(Port In)
X[3:0]	6	wire(Port In)
Y[3:0]	f	wire(Port In)
C1X[3:0]	0	reg(Port Out)
C1Y[3:0]	0	reg(Port Out)
C2X[3:0]	0	reg(Port Out)
C2Y[3:0]	0	reg(Port Out)
DONE	0	reg(Port Out)
i	XXXXXXXX	integer
j	XXXXXXXX	integer
IDLE	0	parameter
DATAIN	1	parameter
ITERATE	2	parameter
DATAOUT	3	parameter
c_state[1:0]	1->2	reg
n_state[1:0]	2	reg
mat[15:0][15:0]	{0 1400 748	reg



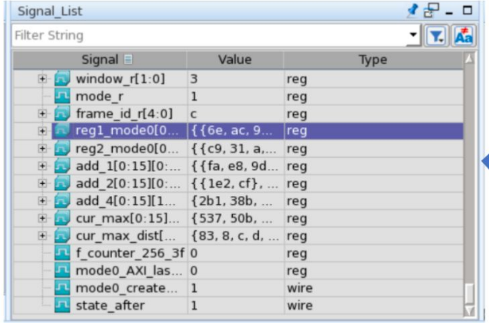
- ☑ Input
- ☑ Output
- ☑ Inout
- ☑ Wire/Signal
- ☑ Register
- ☑ Parameter
- ☑ Constant
- ☑ Assertion
- ☑ Buffer
- ☑ Linkage
- ☑ $unit
- ☑ Interface/Modport
- ☑ Event
- ☑ Other
- ☑ All

nTrace – Add Value Into Waveform



*Src1:TESTBED.U_TOF(/home/RAID2/COURSE/icla

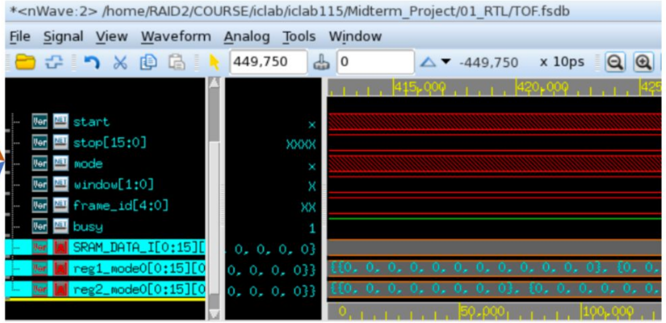
```
    reg [1:0] window_r;
              2
    reg mode_r;
        1
    reg [4:0] frame_id_r;
              c

    reg [7:0] reg1_mode0 [0:15][0:11];
              {{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0
    reg [7:0] reg2_mode0 [0:15][0:7];
              {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
    reg [8:0] add_1 [0:15] [0:3];
```



Signal_List

Signal	Value	Type
window_r[1:0]	3	reg
mode_r	1	reg
frame_id_r[4:0]	c	reg
reg1_mode0[0...	{{6e, ac, 9...	reg
reg2_mode0[0...	{{c9, 31, a,...	reg
add_1[0:15][0:...	{{fa, e8, 9d...	reg
add_2[0:15][0:...	{{1e2, cf}, ...	reg
add_4[0:15][1...	{2b1, 38b, ...	reg
cur_max[0:15]...	{537, 50b, ...	reg
cur_max_dist[...	{83, 8, c, d, ...	reg
f_counter_256_3f	0	reg
mode0_AXI_las...	0	reg
mode0_create...	1	wire
state_after	1	wire

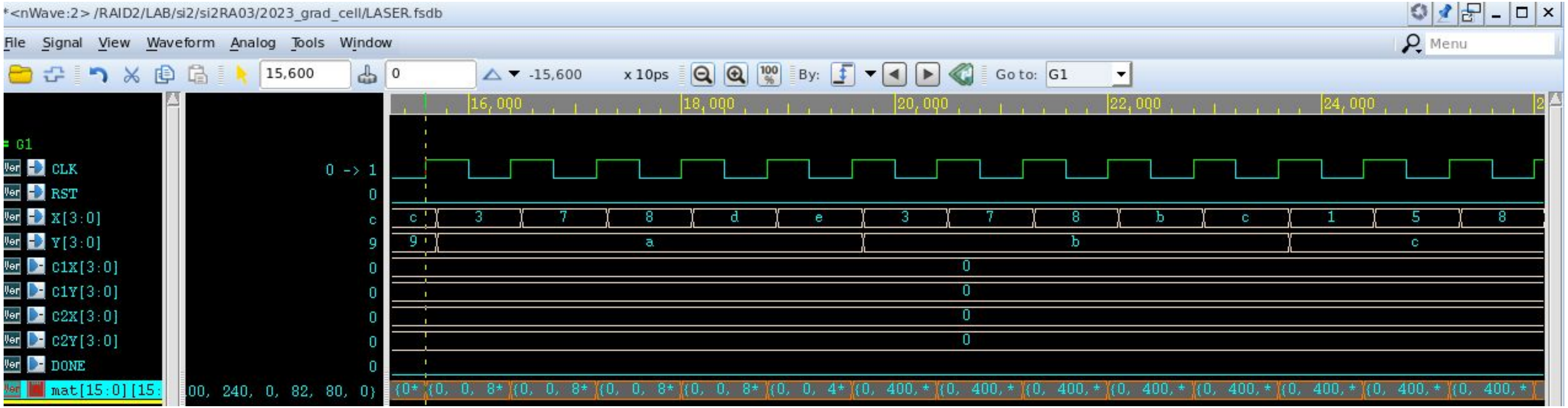
Ctrl + W



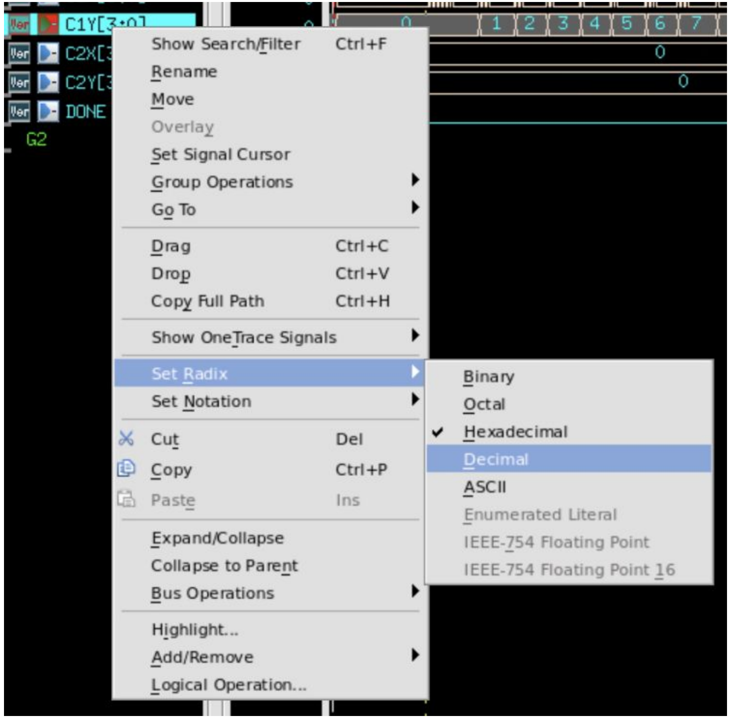
*<nWave:2> /home/RAID2/COURSE/iclab/iclab115/Midterm_Project/01_RTL/TOF.fsdb

```
start                    x
stop[15:0]            XXXX
mode                     x
window[1:0]              X
frame_id[4:0]           XX
busy                     1
SRAM_DATA_I[0:15][  0, 0, 0, 0}
reg1_mode0[0:15][0  0, 0, 0, 0}}   {{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0,
reg2_mode0[0:15][0  0, 0, 0, 0}}   {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0,
```

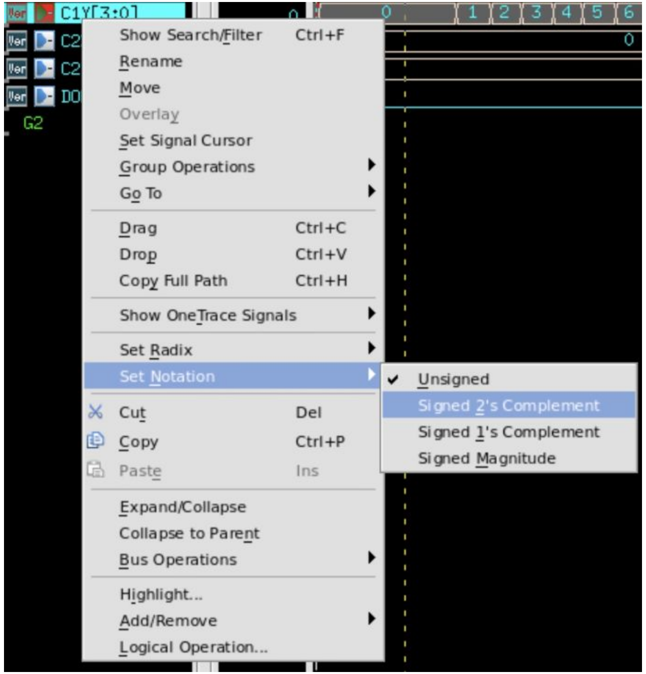
nWave

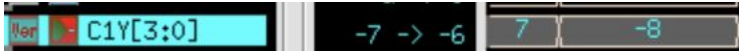

*<nWave:2> /RAID2/LAB/si2/si2RA03/2023_grad_cell/LASER.fsdb
File Signal View Waveform Analog Tools Window
Menu
15,600
0
-15,600
x 10ps
By:
Go to: G1
16,000
18,000
20,000
22,000
24,000
G1
CLK 0 -> 1
RST 0
X[3:0] c
Y[3:0] 9
C1X[3:0] 0
C1Y[3:0] 0
C2X[3:0] 0
C2Y[3:0] 0
DONE 0
mat[15:0][15: 00, 240, 0, 82, 80, 0}
c 3 7 8 d e 3 7 8 b c 1 5 8
9 a b c
0
0
0
0
{0* {0, 0, 8* {0, 0, 8* {0, 0, 8* {0, 0, 8* {0, 0, 4* {0, 400,* {0, 400,* {0, 400,* {0, 400,* {0, 400,* {0, 400,* {0, 400,* {0, 400,*

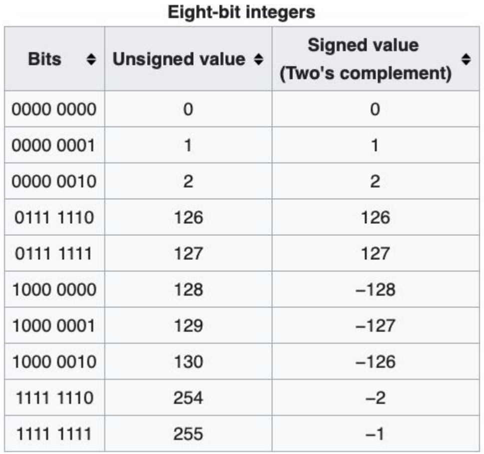
nWave - Change Radix of Signal



nWave - 2's Complement





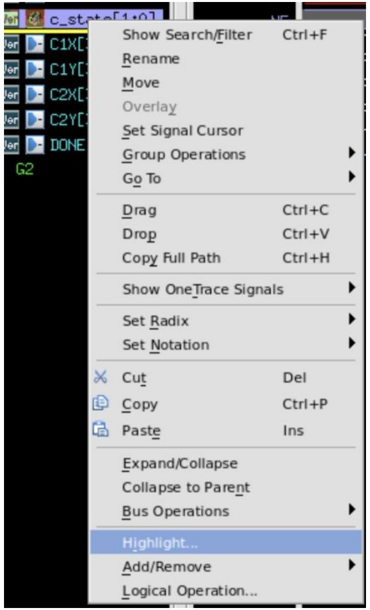


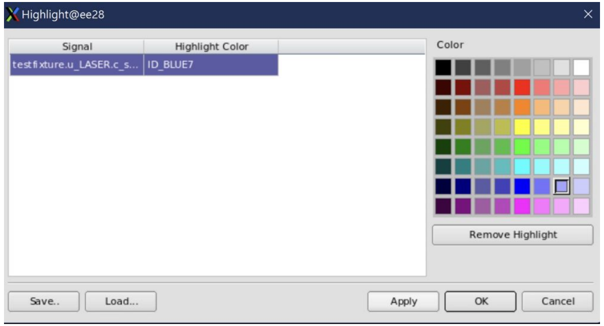
Eight-bit integers

Bits	Unsigned value	Signed value (Two's complement)
0000 0000	0	0
0000 0001	1	1
0000 0010	2	2
0111 1110	126	126
0111 1111	127	127
1000 0000	128	−128
1000 0001	129	−127
1000 0010	130	−126
1111 1110	254	−2
1111 1111	255	−1

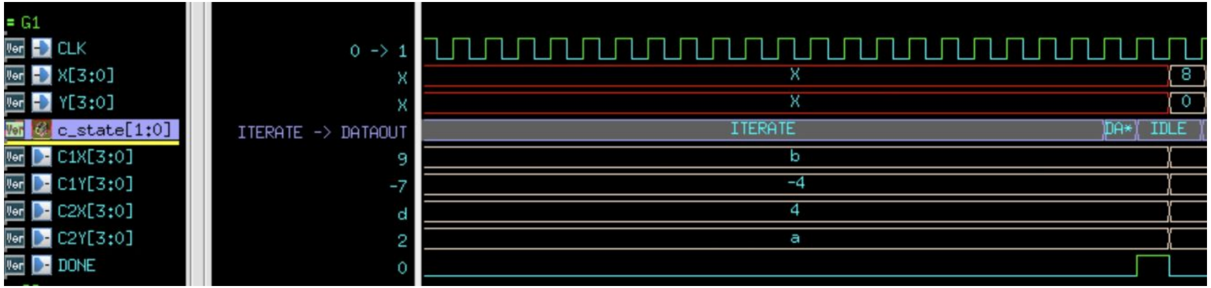
Source:
Two's complement - Wikipedia

nWave - Highlight Signal



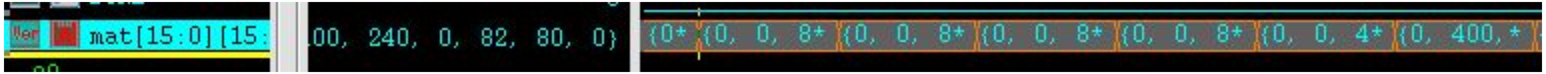


Signal	Highlight Color
testfixture.u_LASER.c_s...	ID_BLUE7

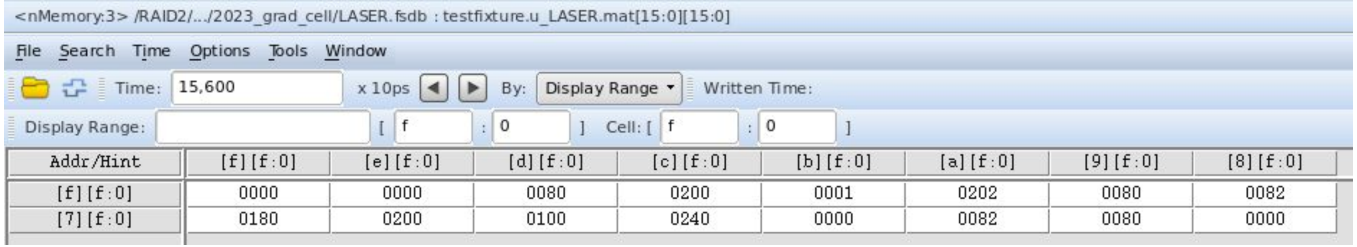


Vector

- How do we debug vector signal easier?



nMemory



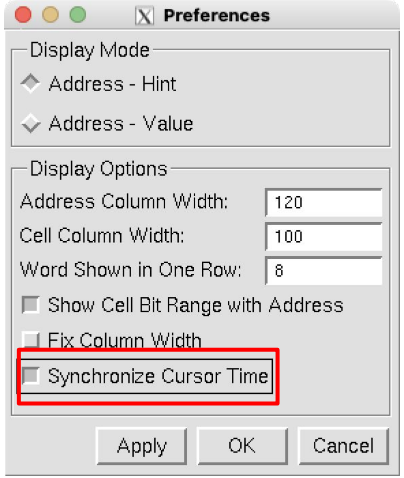
<nMemory:3> /RAID2/.../2023_grad_cell/LASER.fsdb : testfixture.u_LASER.mat[15:0][15:0]

Time: 15,600 x 10ps By: Display Range Written Time:

Display Range: [f : 0] Cell: [f : 0]

Addr/Hint	[f][f:0]	[e][f:0]	[d][f:0]	[c][f:0]	[b][f:0]	[a][f:0]	[9][f:0]	[8][f:0]
[f][f:0]	0000	0000	0080	0200	0001	0202	0080	0082
[7][f:0]	0180	0200	0100	0240	0000	0082	0080	0000

nMemory (Setting)



Preferences

Display Mode

◇ Address - Hint

◇ Address - Value

Display Options

Address Column Width: 120

Cell Column Width: 100

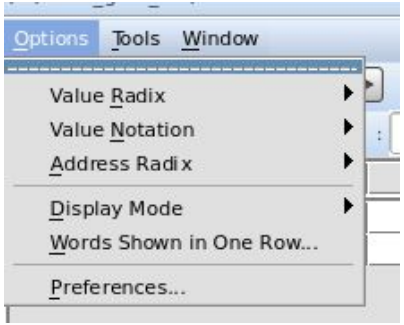
Word Shown in One Row: 8

□ Show Cell Bit Range with Address

□ Fix Column Width

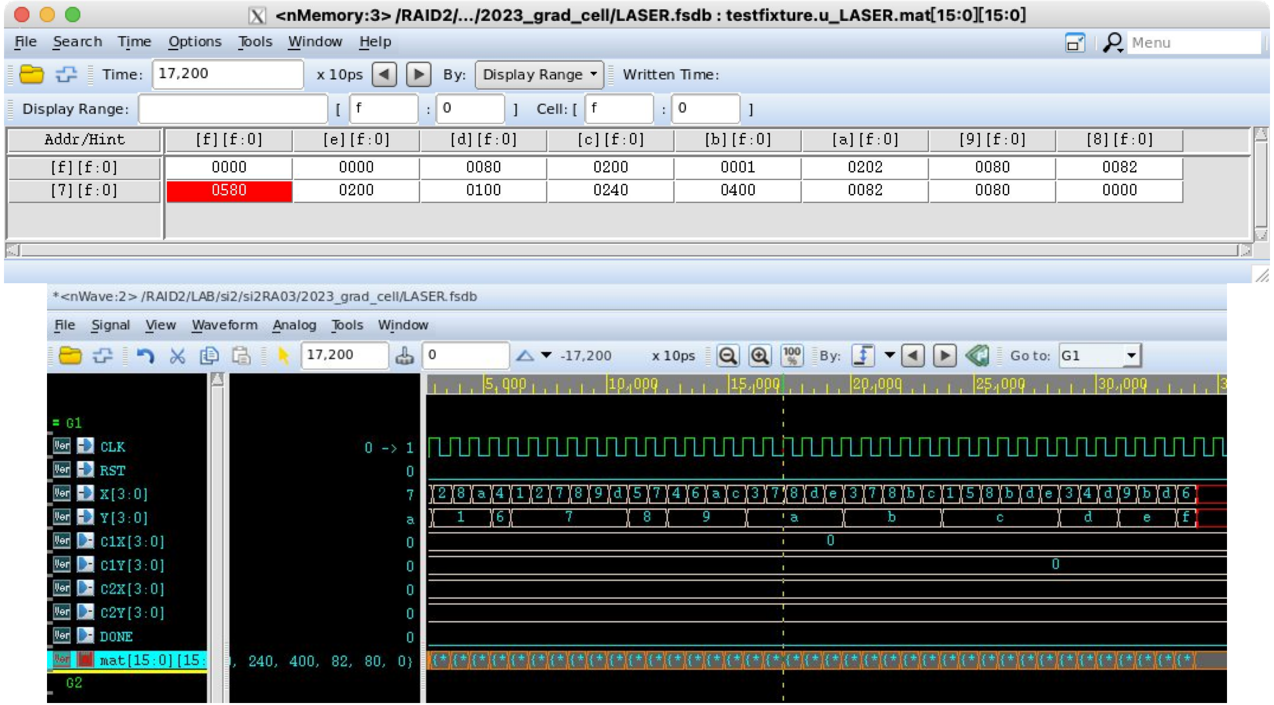
□ Synchronize Cursor Time

Synchronize with waveform



Set value’s radix, notation and address radix

nMemory (Sync with Waveform)



<nMemory:3> /RAID2/.../2023_grad_cell/LASER.fsdb : testfixture.u_LASER.mat[15:0][15:0]

File Search Time Options Tools Window Help
Menu

Time: 17,200 x 10ps By: Display Range Written Time:

Display Range: [f : 0] Cell: [f : 0]

Addr/Hint	[f][f:0]	[e][f:0]	[d][f:0]	[c][f:0]	[b][f:0]	[a][f:0]	[9][f:0]	[8][f:0]
[f][f:0]	0000	0000	0080	0200	0001	0202	0080	0082
[7][f:0]	0580	0200	0100	0240	0400	0082	0080	0000

*<nWave:2> /RAID2/LAB/si2/si2RA03/2023_grad_cell/LASER.fsdb

File Signal View Waveform Analog Tools Window

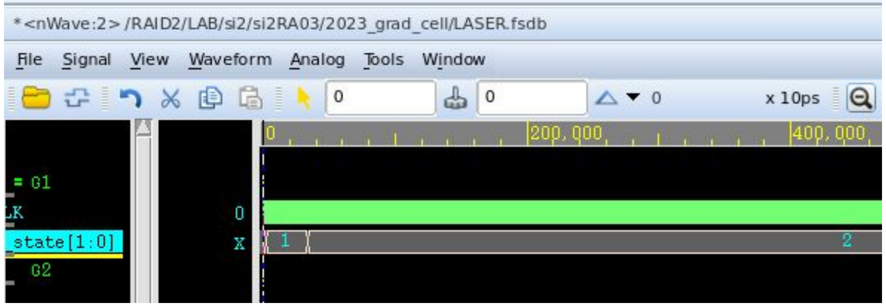
17,200 0 -17,200 x 10ps By: Go to: G1

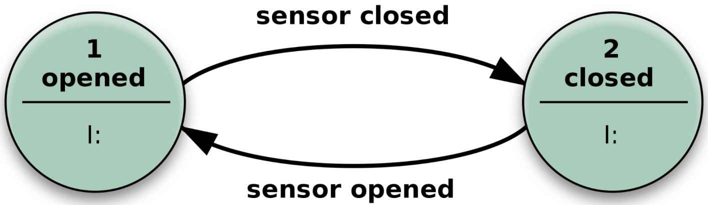
5,000
10,000
15,000
20,000
25,000
30,000
G1
G2
2 8 a 4 1 2 7 8 9 d 5 7 4 6 a c 3 7 8 d e 3 7 8 b c 1 5 8 b d e 3 4 d 9 b d 6
1 6 7 8 9 a b c d e f
0
0

```
G1
CLK          0 -> 1
RST          0
X[3:0]       7
Y[3:0]       a
C1X[3:0]     0
C1Y[3:0]     0
C2X[3:0]     0
C2Y[3:0]     0
DONE         0
mat[15:0][15:   , 240, 400, 82, 80, 0}
G2
```

Finite State Machine

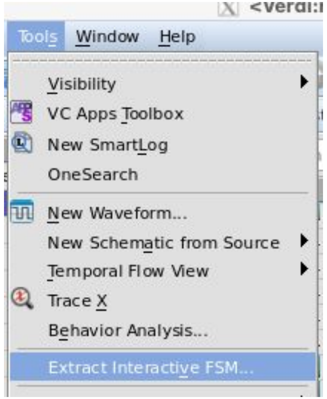
- How do we show the current state in “readable” format of the reg refered to FSM?

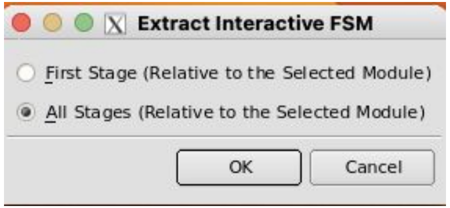

*<nWave:2> /RAID2/LAB/si2/si2RA03/2023_grad_cell/LASER.fsdb
File Signal View Waveform Analog Tools Window
0
0
0
x 10ps
0
200,000
400,000
G1
LK
0
state[1:0]
x
1
2
G2



Source:
Finite-state machine - Wikipedia

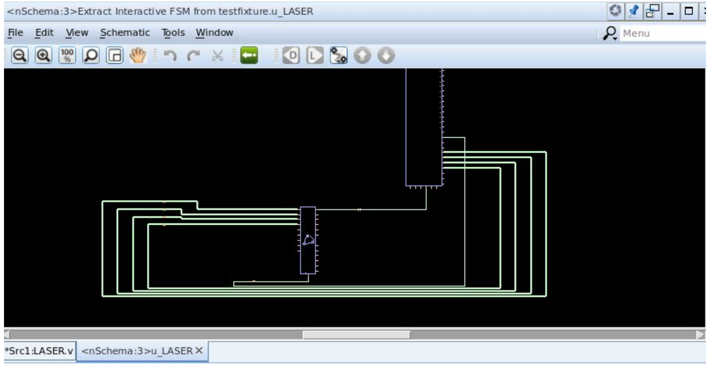
nState (1/2)

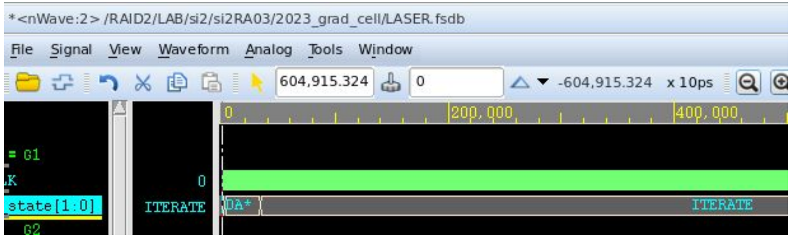




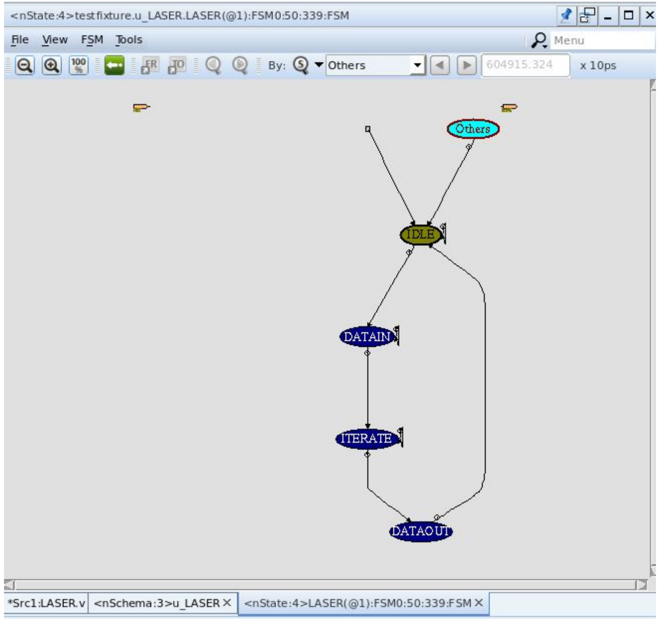
○ First Stage (Relative to the Selected Module)

● All Stages (Relative to the Selected Module)

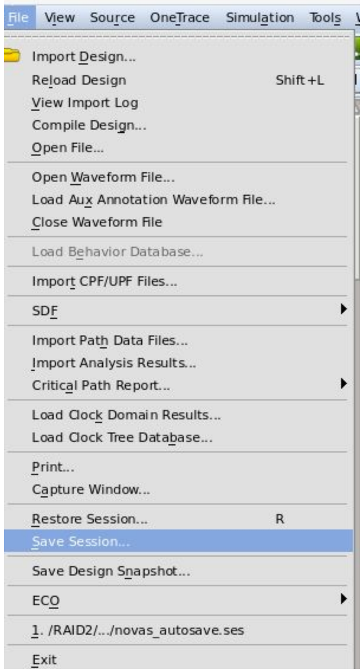




nState (2/2)


<nState:4>testfixture.u_LASER.LASER(@1):FSM0:50:339:FSM
File
View
FSM
Tools
Menu
By:
Others
604915.324
x 10ps
Others
IDLE
DATAIN
ITERATE
DATAOUT
*Src1:LASER.v
<nSchema:3>u_LASER
<nState:4>LASER(@1):FSM0:50:339:FSM

Save Session



Conclusion

- nTrace showed the signal value and signal in list
- nWave displayed the waveform of signal
- nMemory eased the pain on observing value of 2d reg
- nState helped us dealing with FSM issues