מכון טכנולוגי חולון
Holon Institute of Technology

שפת תכנון חומרה –
Verilog

מטלה מספר 4

<u>מגיש</u> :

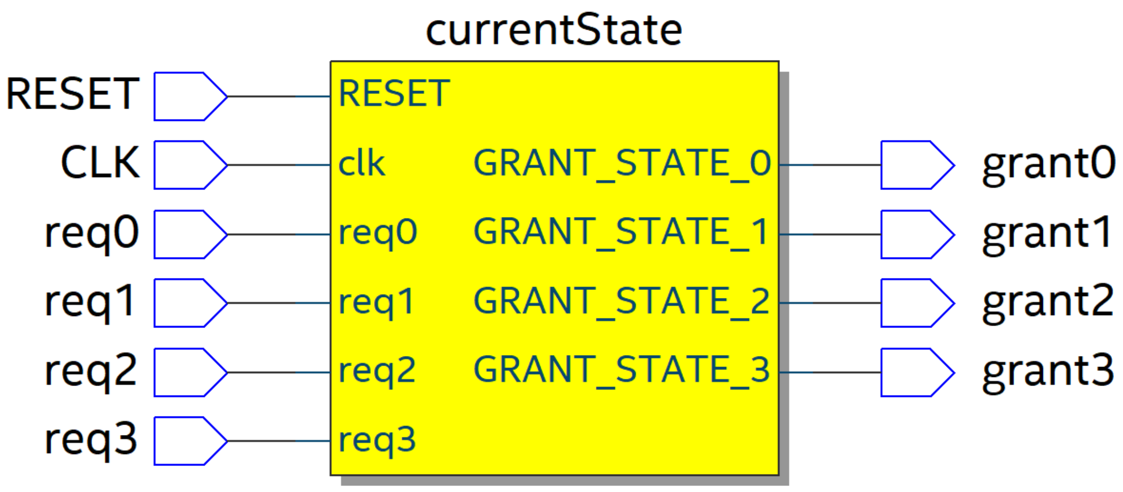
חיים עוזר - 316063569

מכונת מצבים MooreArbiter

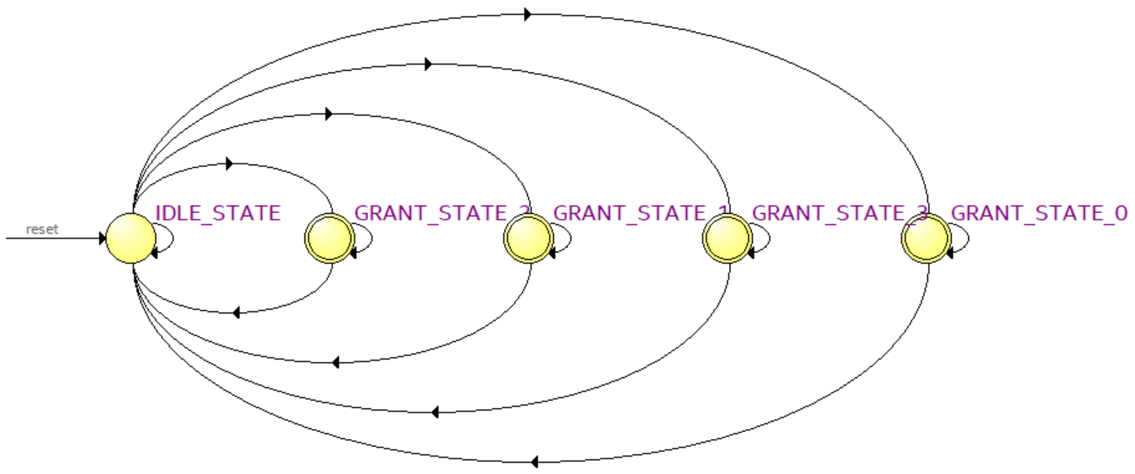
❖ Code:

```
module MooreArbiter (
      input wire CLK,RESET,req0,req1,req2,req3,
  output reg grant0,grant1,grant2,grant3);

  // Define states
  reg [2:0] currentState, nextState;

  // State constants
  parameter IDLE_STATE = 3'b000;
  parameter GRANT_STATE_0 = 3'b001;
  parameter GRANT_STATE_1 = 3'b010;
  parameter GRANT_STATE_2 = 3'b011;
  parameter GRANT_STATE_3 = 3'b100;

  // logic for next state and outputs
  always @(*) begin
    grant0 = (currentState == GRANT_STATE_0);
    grant1 = (currentState == GRANT_STATE_1);
    grant2 = (currentState == GRANT_STATE_2);
    grant3 = (currentState == GRANT_STATE_3);

    // Default next state
    nextState = currentState;

    // State transitions
    case (currentState)
      IDLE_STATE:
        if (req0) begin
          nextState = GRANT_STATE_0;
        end
        else if (req1) begin
          nextState = GRANT_STATE_1;
        end
        else if (req2) begin
          nextState = GRANT_STATE_2;
        end
        else if (req3) begin
          nextState = GRANT_STATE_3;
        end

      GRANT_STATE_0: begin
                if (!req0) begin
          nextState = IDLE_STATE;
        end
             end
      GRANT_STATE_1: begin
                if (!req1) begin
          nextState = IDLE_STATE;
        end
             end
      GRANT_STATE_2:begin
                if (!req2) begin
          nextState = IDLE_STATE;
        end
             end
      GRANT_STATE_3:begin
                if (!req3) begin
          nextState = IDLE_STATE;
        end
             end

      default: nextState = IDLE_STATE;
    endcase
  end

  // Sequential logic for state register and synchronous reset
always @(posedge CLK) begin
    if (RESET) begin
        currentState <= IDLE_STATE;
    end
    else begin
        currentState <= nextState;
    end
end

   endmodule
```

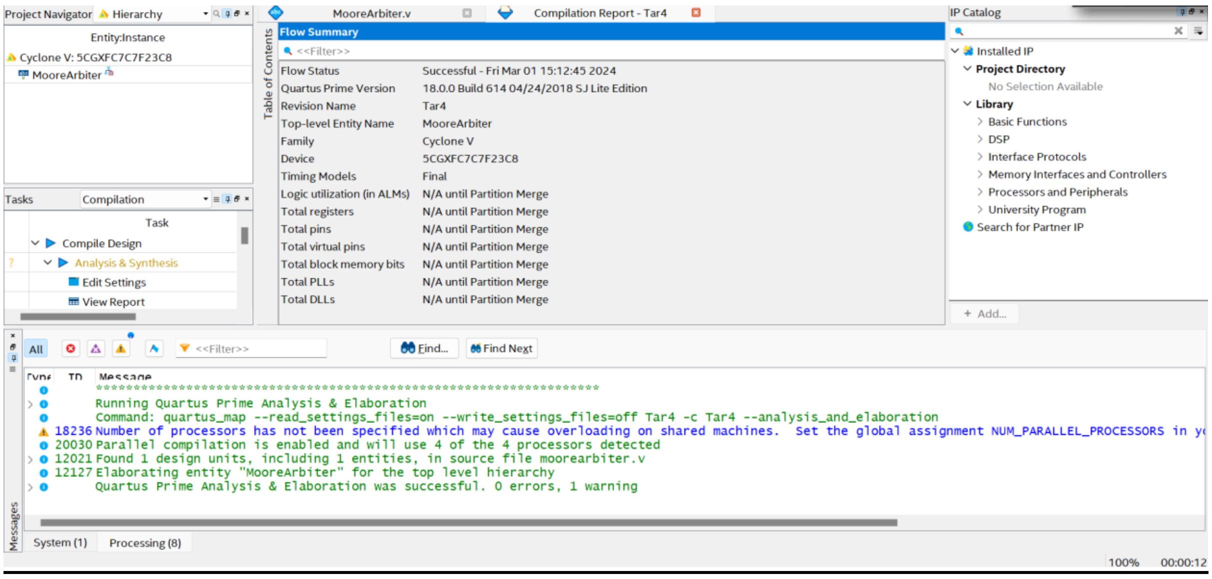
❖ RTL:



❖ State Machine



❖ Compilation:



Flow Summary

Flow Status	Successful - Fri Mar 01 15:12:45 2024
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	Tar4
Top-level Entity Name	MooreArbiter
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	N/A until Partition Merge
Total registers	N/A until Partition Merge
Total pins	N/A until Partition Merge
Total virtual pins	N/A until Partition Merge
Total block memory bits	N/A until Partition Merge
Total PLLs	N/A until Partition Merge
Total DLLs	N/A until Partition Merge

```
Type  ID     Message
             *******************************************************************
             Running Quartus Prime Analysis & Elaboration
             Command: quartus_map --read_settings_files=on --write_settings_files=off Tar4 -c Tar4 --analysis_and_elaboration
      18236  Number of processors has not been specified which may cause overloading on shared machines.  Set the global assignment NUM_PARALLEL_PROCESSORS in y
      20030  Parallel compilation is enabled and will use 4 of the 4 processors detected
      12021  Found 1 design units, including 1 entities, in source file moorearbiter.v
      12127  Elaborating entity "MooreArbiter" for the top level hierarchy
             Quartus Prime Analysis & Elaboration was successful. 0 errors, 1 warning
```

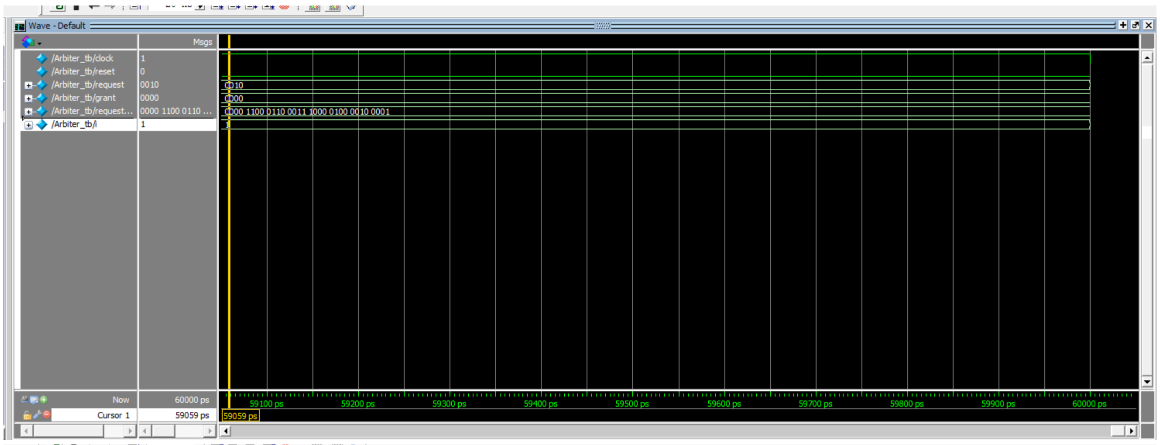
Test Bench:

❖ Code:

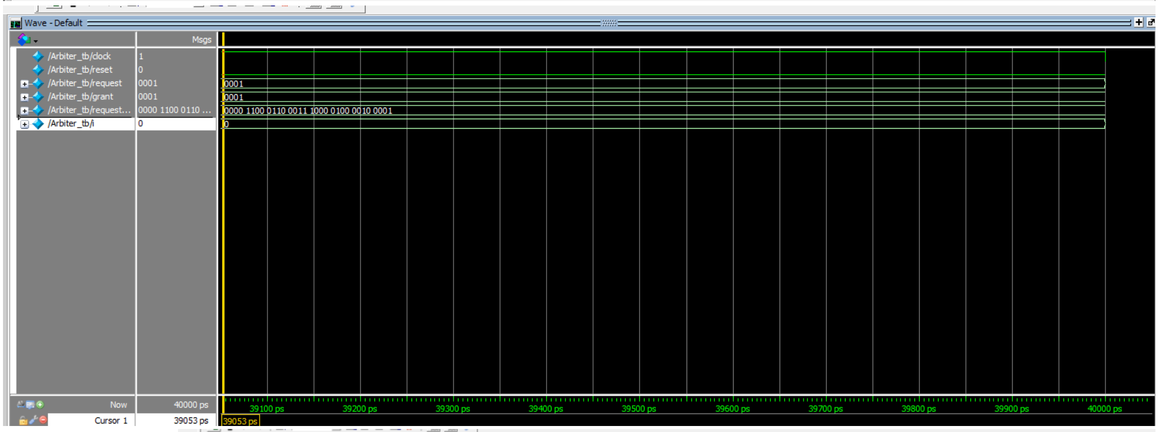
```
`timescale 1ns / 1ps

module MooreArbiter_tb;

reg clock;
reg reset;
reg [3:0] request;
wire [3:0] grant;
Arbiter uut (
    .clock(clock),
    .reset(reset),
    .request(request),
    .grant(grant)
);
reg [3:0] request_sequence[7:0];
integer i;
initial begin
    clock = 0;
    forever #10 clock = ~clock; // 50 MHz clock
end
initial begin
    reset = 1;
    request = 4'b0000;
    #20;
    reset = 0;

    request_sequence[0] = 4'b0001;
    request_sequence[1] = 4'b0010;
    request_sequence[2] = 4'b0100;
    request_sequence[3] = 4'b1000;
    request_sequence[4] = 4'b0011;
    request_sequence[5] = 4'b0110;
    request_sequence[6] = 4'b1100;
    request_sequence[7] = 4'b0000;
    for (i = 0; i < 8; i = i + 1) begin
        request = request_sequence[i];
```

```
            #20;
        end

        $finish;
    end

endmodule
```

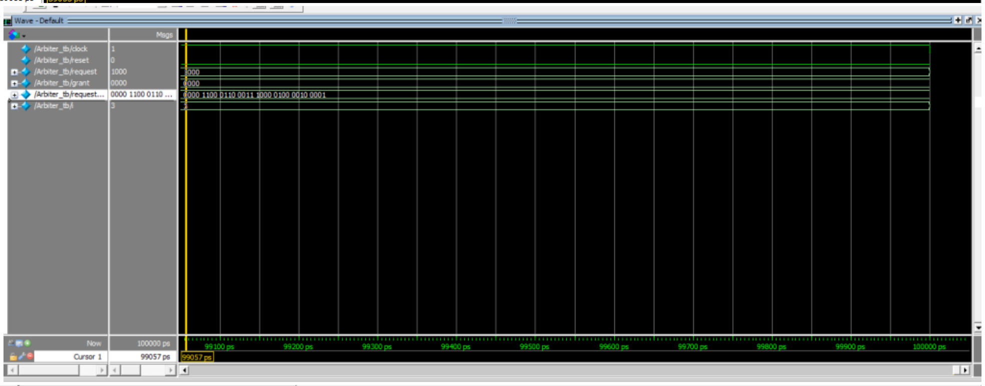
❖ Wave – Model sim:



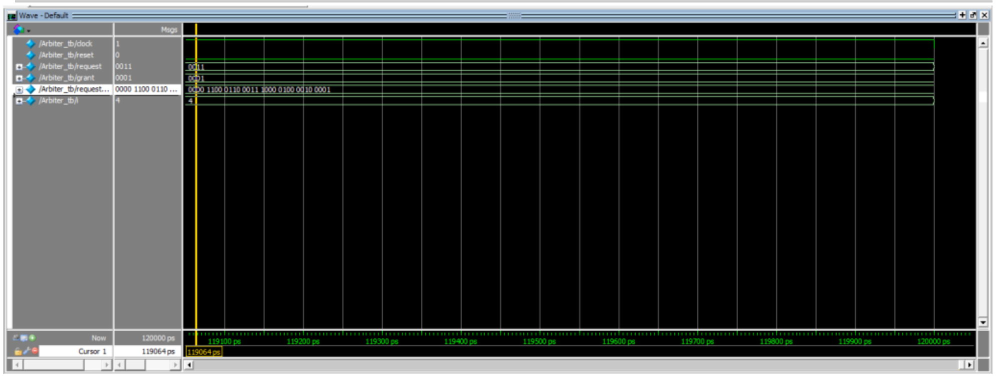
```
Wave - Default
                              Msgs
/Arbiter_tb/clock             1
/Arbiter_tb/reset             0
/Arbiter_tb/request           0010         0010
/Arbiter_tb/grant             0000         0000
/Arbiter_tb/request...        0000 1100 0110 ...   0000 1100 0110 0011 1000 0100 0010 0001
/Arbiter_tb/i                 1            1
Now                           60000 ps     59100 ps  59200 ps  59300 ps  59400 ps  59500 ps  59600 ps  59700 ps  59800 ps  59900 ps  60000 ps
Cursor 1                      59059 ps     59059 ps
```



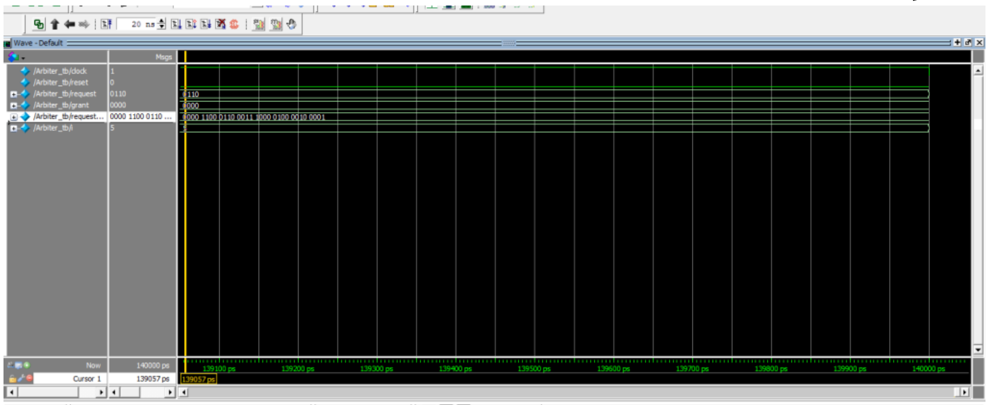
```
Wave - Default
                              Msgs
/Arbiter_tb/clock             1
/Arbiter_tb/reset             0
/Arbiter_tb/request           0001         0001
/Arbiter_tb/grant             0001         0001
/Arbiter_tb/request...        0000 1100 0110 ...   0000 1100 0110 0011 1000 0100 0010 0001
/Arbiter_tb/i                 0            0
Now                           40000 ps     39100 ps  39200 ps  39300 ps  39400 ps  39500 ps  39600 ps  39700 ps  39800 ps  39900 ps  40000 ps
Cursor 1                      39053 ps     39053 ps
```



```
Wave - Default
                              Msgs
/Arbiter_tb/clock             1
/Arbiter_tb/reset             0
/Arbiter_tb/request           1000         1000
/Arbiter_tb/grant             0000         0000
/Arbiter_tb/request...        0000 1100 0110 ...   0000 1100 0110 0011 1000 0100 0010 0001
/Arbiter_tb/i                 3            3
Now                           100000 ps    99100 ps  99200 ps  99300 ps  99400 ps  99500 ps  99600 ps  99700 ps  99800 ps  99900 ps  100000 ps
Cursor 1                      99057 ps     99057 ps
```



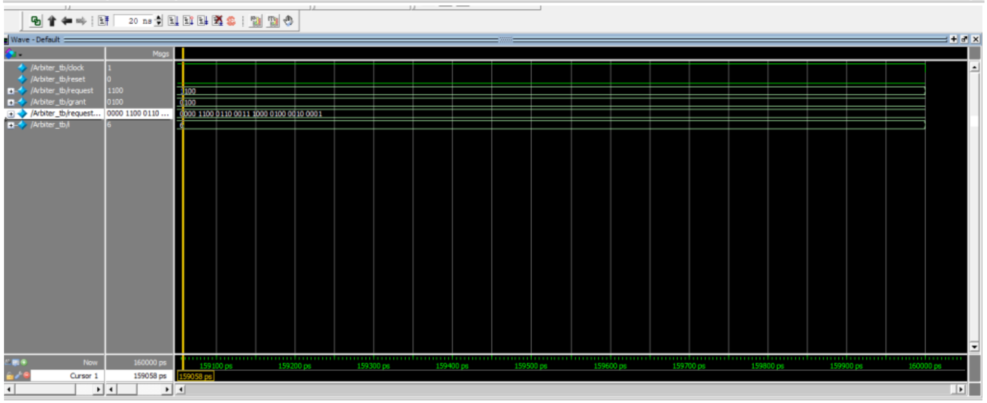
```
Wave - Default
                              Msgs
/Arbiter_tb/clock             1
/Arbiter_tb/reset             0
/Arbiter_tb/request           0011         0011
/Arbiter_tb/grant             0001         0001
/Arbiter_tb/request...        0000 1100 0110 ...   0000 1100 0110 0011 1000 0100 0010 0001
/Arbiter_tb/i                 4            4
Now                           120000 ps    119100 ps  119200 ps  119300 ps  119400 ps  119500 ps  119600 ps  119700 ps  119800 ps  119900 ps  120000 ps
Cursor 1                      119064 ps    119064 ps
```


Wave - Default

```
/Arbiter_tb/clock          1
/Arbiter_tb/reset          0
/Arbiter_tb/request        0110
/Arbiter_tb/grant          0000
/Arbiter_tb/request...     0000 1100 0110 ...
/Arbiter_tb/i              5

Now        140000 ps
Cursor 1   139057 ps
```

0110
0000
0000 1100 0110 0011 1000 0100 0010 0001
5

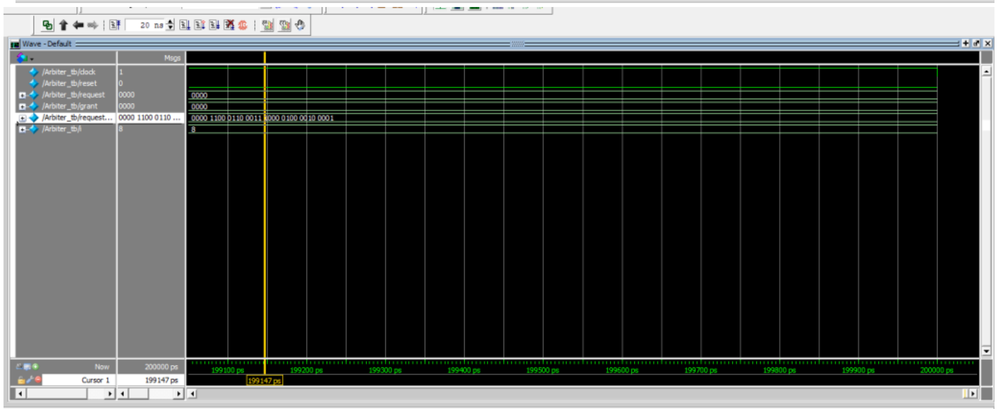
139100 ps 139200 ps 139300 ps 139400 ps 139500 ps 139600 ps 139700 ps 139800 ps 139900 ps 140000 ps
139057 ps


Wave - Default

```
/Arbiter_tb/clock          1
/Arbiter_tb/reset          0
/Arbiter_tb/request        1100
/Arbiter_tb/grant          0100
/Arbiter_tb/request...     0000 1100 0110 ...
/Arbiter_tb/i              6

Now        160000 ps
Cursor 1   159058 ps
```

1100
0100
0000 1100 0110 0011 1000 0100 0010 0001
6

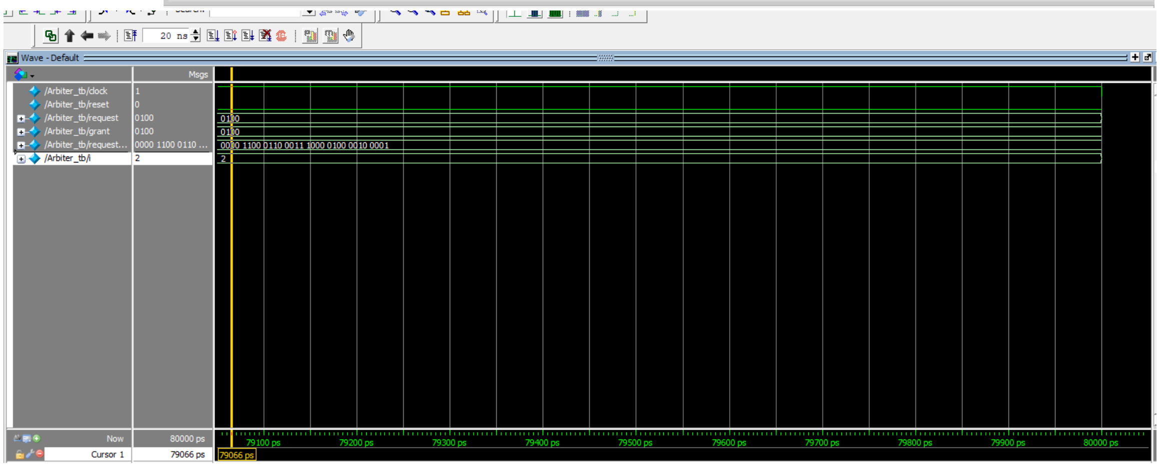
159100 ps 159200 ps 159300 ps 159400 ps 159500 ps 159600 ps 159700 ps 159800 ps 159900 ps 160000 ps
159058 ps


Wave - Default

```
/Arbiter_tb/clock          1
/Arbiter_tb/reset          0
/Arbiter_tb/request        0000
/Arbiter_tb/grant          0000
/Arbiter_tb/request...     0000 1100 0110 ...
/Arbiter_tb/i              8

Now        200000 ps
Cursor 1   199147 ps
```

0000
0000
0000 1100 0110 0011 1000 0100 0010 0001
8

199100 ps 199200 ps 199300 ps 199400 ps 199500 ps 199600 ps 199700 ps 199800 ps 199900 ps 200000 ps
199147 ps


Wave - Default

```
/Arbiter_tb/clock          1
/Arbiter_tb/reset          0
/Arbiter_tb/request        0100
/Arbiter_tb/grant          0100
/Arbiter_tb/request...     0000 1100 0110 ...
/Arbiter_tb/i              2

Now        80000 ps
Cursor 1   79066 ps
```

0100
0100
0000 1100 0110 0011 1000 0100 0010 0001
2

79100 ps 79200 ps 79300 ps 79400 ps 79500 ps 79600 ps 79700 ps 79800 ps 79900 ps 80000 ps
79066 ps