

1 Introduction

The project 3 DTMF receiver module, TDSP versions, DSP behavior algorithm are implemented using Verilog and TDSP assembly language. Finite State Machine is implemented using Verilog behaviorally. The output dma_grant is 1 in GRANTD_DMA and tdsp_grant is 1 in GRANT_TDSP and CLEAR with GRANT_TDSP being priority except in DMA_PRA state. The finite state machine test bench is created to test the TDSP_GRANT and DMA_GRANT being 1 within 1 clock period of TDSP_BREQ or DMA_BREQ. The assembly language TDSP test is implemented for # spectrum with quiet tones at the start and end.

2 Memory Access Arbiter ARB modules

(A)A Brief Module Description

Memory Access Bus Arbiter ARB module allows DMA and TDSP access to Data Sample memory. The ARB state machine diagram is implemented with IDLE, GRANT_TDSP, DMA_PRI, CELAR and GRANT_DMA states. TDSP priority is granted if both devices of tdsp_breq and dma_breq happen at the same time except in DMA_PRI state. Tdsp_grant is 1 in CLEAR and GRANT_TDSP, dma_grant is 1 in GRANT_DMA and they are zeros in other states. The bus arbiter has 25 MHz frequency, 25 MHz positive edge triggered clock and active high reset which is asynchronous.

(B)Arb .v RTL source code

```
/*  
*  
* Author: Zeyar Win  
*  
*  
*/
```

```
module arb (  
    reset,  
    clk,  
    dma_breq,  
    dma_grant,  
    tdsp_breq,  
    tdsp_grant,  
    scan_in0,  
    scan_en,  
    scan_out0  
);
```

```
/*
```

```

*
* DMA/ TDSP bus arbiter
*
*/

```

```

input
    reset,          // system reset
    clk,            // system clock
    dma_breq,       // dma controller bus request
    tdsp_breq ;     // tdsp bus request

```

```

output
    dma_grant,      // dma controller bus grant
    tdsp_grant ;    // tdsp bus grant

```

```

input
    scan_in0,       // test scan mode data input
    scan_en;        // test scan mode enable

```

```

output
    scan_out0;      // test scan mode data output

```

```

`include "./include/arb.h"

```

```

reg [2:0] state;
reg [2:0] nextstate;
reg  dma_grant;
reg  tdsp_grant;

```

```

//clock state change
always @(state or tdsp_breq or dma_breq)
    case(state)
        `ARB_IDLE: if (tdsp_breq == 1) begin

            nextstate = `ARB_GRANT_TDSP;
            end
        //
            tdsp_grant = 1;
            else if(dma_breq == 1 && tdsp_breq == 0)
            begin
                nextstate = `ARB_GRANT_DMA;

            end else begin
                nextstate = `ARB_IDLE;
            end

        `ARB_GRANT_TDSP:
            if (tdsp_breq == 0)begin

```

```

        nextstate = `ARB_CLEAR;
    end else begin
        nextstate = `ARB_GRANT_TDSP;
    end
`ARB_GRANT_DMA:
    if (dma_breq == 0) begin
        nextstate = `ARB_CLEAR;
    end else begin
        nextstate = `ARB_GRANT_DMA;
    end
`ARB_CLEAR:
    if (dma_breq == 1) begin
        nextstate = `ARB_DMA_PRI;
    end else if (tdsp_breq == 1) begin
        nextstate = `ARB_GRANT_TDSP;
    end else begin
        nextstate = `ARB_CLEAR;
    end
`ARB_DMA_PRI:
    if (dma_breq == 1) begin
        nextstate = `ARB_GRANT_DMA;
    end
    else if (tdsp_breq == 1 && dma_breq == 0) begin
        nextstate = `ARB_GRANT_TDSP;
    end
    else if (dma_breq != 1 && tdsp_breq != 1) begin
        nextstate = `ARB_IDLE;
    end
    else begin
        nextstate = `ARB_DMA_PRI;
    end

default: nextstate = `ARB_IDLE;
endcase

```

```

// seq
always @(posedge clk or posedge reset)
begin: FSM_SEQ
    if (reset == 1'b1)
    begin
        state <= `ARB_IDLE;
    end
    else
    begin

```

```

        state <= nextstate;
    end

end

// output
always @(posedge clk or posedge reset)
begin
    if (reset == 1'b1)
    begin
        tdsp_grant = 1'b0;
        dma_grant = 1'b0;

    end
    else
    begin
        case(nextstate)

            `ARB_IDLE:begin
                tdsp_grant = 1'b0;
                dma_grant = 1'b0;

            end

            `ARB_GRANT_TDSP:begin
                tdsp_grant = 1'b1;
                dma_grant = 1'b0;

            end

            `ARB_GRANT_DMA:begin
                tdsp_grant = 1'b0;
                dma_grant = 1'b1;

            end

            `ARB_CLEAR:begin
                tdsp_grant = 1'b1;
                dma_grant = 1'b0;

            end

            `ARB_DMA_PRI:begin
                tdsp_grant = 1'b0;
                dma_grant = 1'b0;

            end

        endcase
    end
end

```

```
        end
    end

endmodule
```

(c)arb_test.v test bench source code

```
/*
 *
 * Author: zeyar win
 *      Rochester, NY, USA
 */

`timescale 1ns / 1ns

module test;

    wire dma_grant, tdsp_grant;
    wire scan_out0;

    reg clk, dma_breq, reset, tdsp_breq;
    reg scan_in0, scan_en;

    arb top(
        .reset(reset),
        .clk(clk),
        .dma_breq(dma_breq),
        .dma_grant(dma_grant),
        .tdsp_breq(tdsp_breq),
        .tdsp_grant(tdsp_grant),
        .scan_in0(scan_in0),
        .scan_en(scan_en),
        .scan_out0(scan_out0)
    );

    reg [4: 0]
        dma_wait,
        tdsp_wait ;

    integer
```

```

i,
j,
k,
l,
dma_error1bit,
tdsp_error1bit,
dma_cnt2,
tdsp_cnt2,
dma_cnt,
tdsp_cnt ;

wire
grant = dma_grant | tdsp_grant ;

initial
begin
    $timeformat( -9, 2, "ns", 16);
`ifdef SDFSCAN

    $sdf_annotate("sdf/arb_tsmc18_scan.sdf", test.top);
`endif

    clk = 1'b0;
    dma_breq = 1'b0;
    reset = 1'b0;
    tdsp_breq = 1'b0;
    scan_in0 = 1'b0;
    scan_en = 1'b0;
    dma_cnt = 0 ;
    tdsp_cnt = 0 ;
    dma_cnt2 = 0;
    tdsp_cnt2=0;
    dma_wait = $random ;
    tdsp_wait = $random ;

    @(negedge clk)
    reset = 1'b1 ;
    repeat (2)
        @(negedge clk) ;
    @(negedge clk)
    reset = 1'b0 ;
    repeat (2)
        @(posedge clk) ;

    repeat (256)
    begin
        @(posedge clk)

```

```

        dma_wait <= $random ;
        tdsp_wait <= $random ;
        fork
        // if output changes > one clock cycle #40ns of brequest high and another output changes to
zero,flag an error
        // use counter

        dma_request ;
        tdsp_request ;
        dma_check;
        tdsp_check;
    join
    repeat (4)
        @(posedge clk) ;
end
repeat (4)
    @(posedge clk) ;
if (dma_cnt != tdsp_cnt)
begin
    $display(" ** Fails simulation!");
    $display(" ** 256 Individual Bus request cycles generated,");
    $display(" ** (#tdsp grants == %d) != (#dma grants == %d)", tdsp_cnt, dma_cnt);
end
else
begin
    $display(" ** Passes simulation!");
    $display(" ** 256 Individual Bus request cycles generated,");
    $display(" ** (#tdsp grants == %d) == (#dma grants == %d)", tdsp_cnt, dma_cnt);
end
$stop ;
end

always #20
    clk = ~clk ;

task dma_request ;
begin
    repeat (dma_wait)
        @(posedge clk) ;
    dma_breq <= 1 ;
    $display("%t DMA Bus Request", $time);
    for (i = 0 ; i < (dma_wait + tdsp_wait + 10) ; i = i + 1)
        @(posedge clk)
        if (dma_grant)
            begin
                dma_cnt = dma_cnt + 1 ;
                i = (dma_wait + tdsp_wait + 10) ;
            end
        end
end

```

```

        end
        @(posedge clk)
        dma_breq <= 0 ;
        @(posedge clk);
    end
endtask

task tdsp_request ;
begin
    repeat (tdsp_wait)
        @(posedge clk) ;
    tdsp_breq <= 1 ;
    $display("%t TDSP Bus Request", $time);
    for (j = 0 ; j < (dma_wait + tdsp_wait + 10) ; j = j + 1)
        @(posedge clk)
        if (tdsp_grant)
            begin
                tdsp_cnt = tdsp_cnt + 1 ;
                j = (dma_wait + tdsp_wait + 10) ;
            end
        @(posedge clk)
        tdsp_breq <= 0 ;
        @(posedge clk);
    end
endtask

task dma_check;
begin
    @(negedge tdsp_grant)
    begin
        dma_cnt2 =0;
        for (k=0 ; k<50 ; k=k+1 )
            @(posedge clk)
            begin
                if(dma_grant != 1 )
                    begin
                        dma_cnt2 = dma_cnt2 + 1;
                        // $display("dma counter ", dma_cnt2);
                    end
                else if(dma_grant == 1)
                    begin
                        k <= 50;
                    end
                if ( dma_cnt2 > 2)
                    begin
                        dma_error1bit <= 1;
                        $display( "%t DMA_GRANT ERROR", $time);
                    end
            end
        end
    end
endtask

```



```

                                $stop;
                                end
                                @(posedge clk);
                                end
                                end
                                end
                                endtask

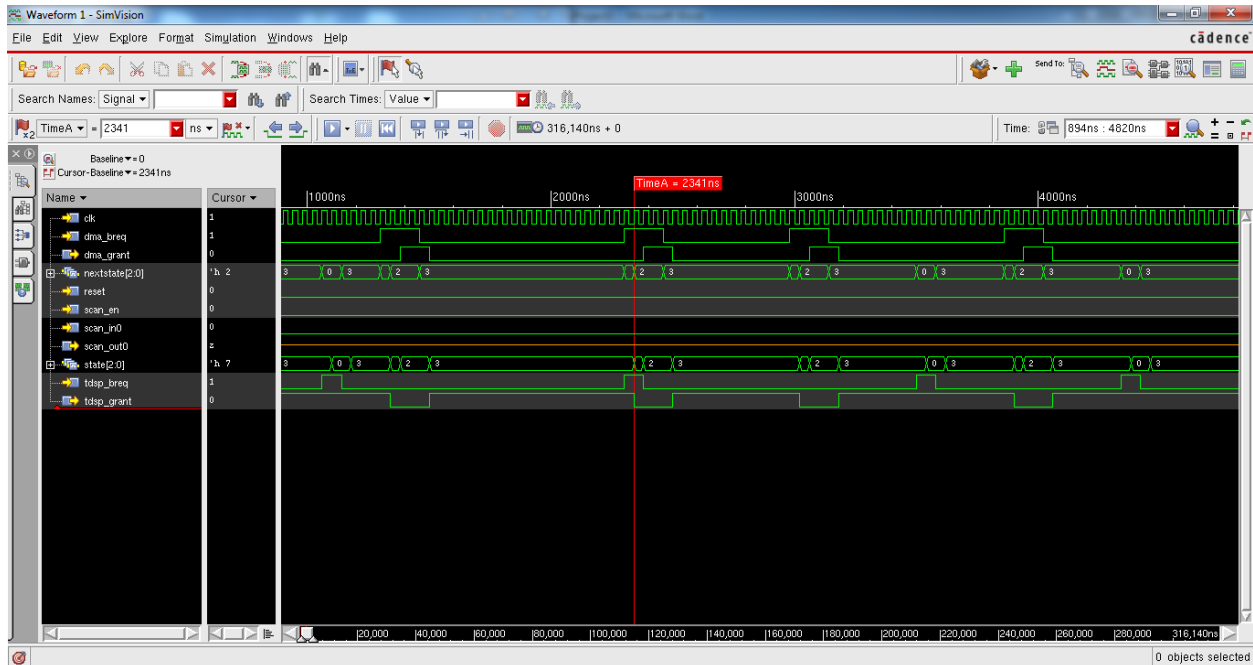
task tdsp_check;
begin
    @(negedge dma_grant)
    begin
        for (l=0 ; l<50 ; l=l+1 )
            @(posedge clk)
            begin
                if(tdsp_grant != 1 ) //&& dma_breq ==1)
                    begin
                        tdsp_cnt2 = tdsp_cnt2 + 1;
                        // $display ("tdsp counter", tdsp_cnt2);
                    end
                else if(tdsp_grant == 1)
                    begin
                        l <= 50;
                    end
                if ( tdsp_cnt2 > 1)
                    begin
                        tdsp_error1bit <= 1;
                        $display( "%t TDSP_GRANT ERROR", $time);

                        $stop;
                    end
                    @(posedge clk);
                end
            end
        end
    end
endtask

endmodule

```

(D) RTL level simulation waveform

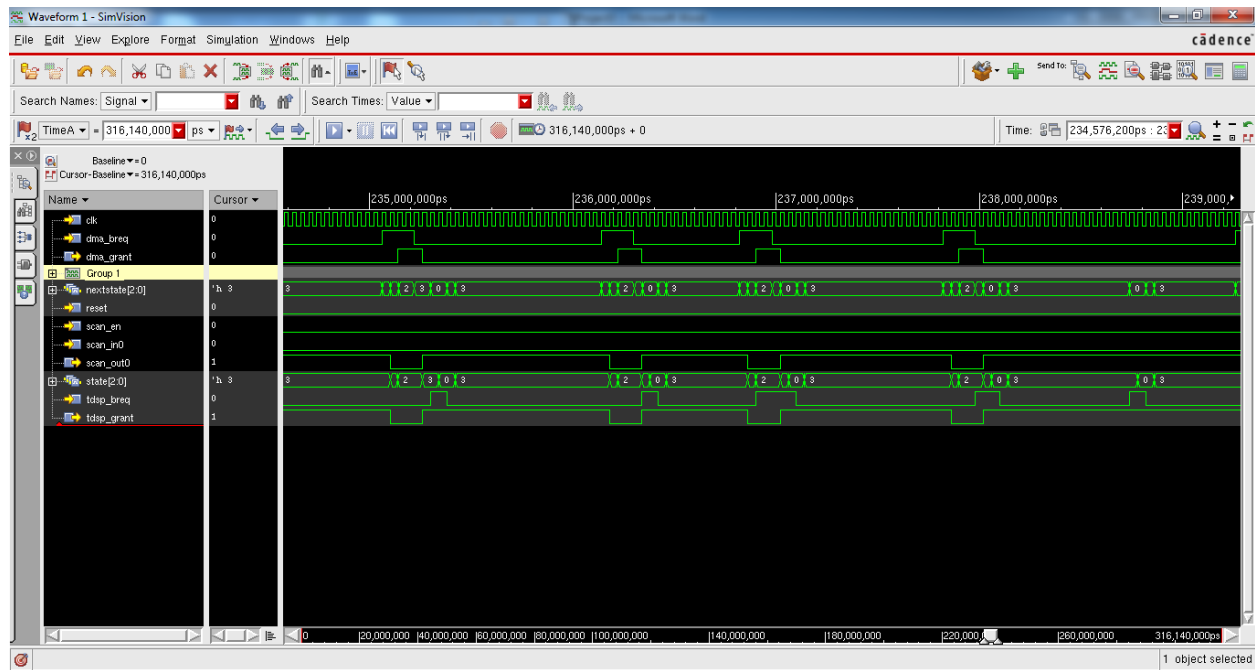


```

297300.00ns TDSP Bus Request
297340.00ns DMA Bus Request
298140.00ns DMA Bus Request
298500.00ns TDSP Bus Request
299180.00ns DMA Bus Request
299780.00ns TDSP Bus Request
300340.00ns DMA Bus Request
300700.00ns TDSP Bus Request
301060.00ns DMA Bus Request
301860.00ns TDSP Bus Request
302180.00ns TDSP Bus Request
302740.00ns DMA Bus Request
303420.00ns DMA Bus Request
303540.00ns TDSP Bus Request
304100.00ns DMA Bus Request
304300.00ns TDSP Bus Request
304660.00ns DMA Bus Request
305540.00ns TDSP Bus Request
305900.00ns TDSP Bus Request
306420.00ns DMA Bus Request
307220.00ns TDSP Bus Request
307740.00ns DMA Bus Request
308620.00ns TDSP Bus Request
308860.00ns DMA Bus Request
309940.00ns DMA Bus Request
310020.00ns TDSP Bus Request
311020.00ns TDSP Bus Request
311620.00ns DMA Bus Request
312180.00ns DMA Bus Request
313220.00ns TDSP Bus Request
313900.00ns DMA Bus Request
314140.00ns TDSP Bus Request
315060.00ns TDSP Bus Request
315540.00ns DMA Bus Request
** Passed simulation!
** 256 Individual Bus request cycles generated.
** (#tdsp grants == 256) == (#dma grants == 256)
Simulation stopped via $stop(1) at time 316140 NS + 0
ncsim>

```

(E) Net list Simulation



(F) Documents of logic synthesis report

/*

*

* Author: zeyar win

* Rochester, NY, USA

*

*/

`timescale 1ns / 1ns

module test;

```
wire dma_grant, tdsp_grant;
```

```
wire scan_out0;
```

```
reg clk, dma_breq, reset, tdsp_breq;
```

```
reg scan_in0, scan_en;
```

```
arb top(
```

```
    .reset(reset),
```

```
    .clk(clk),
```

```
    .dma_breq(dma_breq),
```

```
    .dma_grant(dma_grant),
```

```
    .tdsp_breq(tdsp_breq),
```

```
    .tdsp_grant(tdsp_grant),
```

```
    .scan_in0(scan_in0),
```

```
    .scan_en(scan_en),
```

```
    .scan_out0(scan_out0)
```

```
);
```

```
reg [4: 0]
```

```
    dma_wait,
```

```
    tdsp_wait ;
```

```
integer
```

```
    i,
```

```

j,
k,
l,
dma_error1bit,
tdsp_error1bit,
dma_cnt2,
tdsp_cnt2,
dma_cnt,
tdsp_cnt ;

wire

grant = dma_grant | tdsp_grant ;

initial

begin

$timeformat( -9, 2, "ns", 16);

`ifdef SDFSCAN

$sdf_annotate("sdf/arb_tsmc18_scan.sdf", test.top);

`endif

clk = 1'b0;

dma_breq = 1'b0;

reset = 1'b0;

tdsp_breq = 1'b0;

```

```
scan_in0 = 1'b0;
scan_en = 1'b0;
dma_cnt = 0 ;
tdsp_cnt = 0 ;
dma_cnt2 = 0;
tdsp_cnt2=0;
dma_wait = $random ;
tdsp_wait = $random ;
```

```
@(negedge clk)
reset = 1'b1 ;
repeat (2)
    @(negedge clk) ;
@(negedge clk)
reset = 1'b0 ;
repeat (2)
    @(posedge clk) ;
```

```
repeat (256)
begin
    @(posedge clk)
    dma_wait <= $random ;
    tdsp_wait <= $random ;
fork
```

```
// if output changes > one clock cycle #40ns of brequest high and another output changes to zero, flag an error
```

```
// use counter
```

```
    dma_request ;
```

```
    tdsp_request ;
```

```
    dma_check;
```

```
    tdsp_check;
```

```
join
```

```
repeat (4)
```

```
    @(posedge clk) ;
```

```
end
```

```
repeat (4)
```

```
    @(posedge clk) ;
```

```
if (dma_cnt != tdsp_cnt)
```

```
begin
```

```
    $display(" ** Fails simulation!");
```

```
    $display(" ** 256 Individual Bus request cycles generated,");
```

```
    $display(" ** (#tdsp grants == %d) != (#dma grants == %d)", tdsp_cnt, dma_cnt);
```

```
end
```

```
else
```

```
begin
```

```
    $display(" ** Passes simulation!");
```

```
    $display(" ** 256 Individual Bus request cycles generated,");
```

```
    $display(" ** (#tdsp grants == %d) == (#dma grants == %d)", tdsp_cnt, dma_cnt);
```

```

    end

    $stop ;

end

always #20

    clk = ~clk ;

task dma_request ;

    begin

        repeat (dma_wait)

            @(posedge clk) ;

            dma_breq <= 1 ;

            $display("%t DMA Bus Request", $time);

            for (i = 0 ; i < (dma_wait + tdsp_wait + 10) ; i = i + 1)

                @(posedge clk)

                if (dma_grant)

                    begin

                        dma_cnt = dma_cnt + 1 ;

                        i = (dma_wait + tdsp_wait + 10) ;

                    end

                @(posedge clk)

                dma_breq <= 0 ;

                @(posedge clk);

            end

        endtask

```



```

task tdsp_request ;
    begin
        repeat (tdsp_wait)
            @(posedge clk) ;
        tdsp_breq <= 1 ;
        $display("%t TDSP Bus Request", $time);
        for (j = 0 ; j < (dma_wait + tdsp_wait + 10) ; j = j + 1)
            @(posedge clk)
            if (tdsp_grant)
                begin
                    tdsp_cnt = tdsp_cnt + 1 ;
                    j = (dma_wait + tdsp_wait + 10) ;
                end
            @(posedge clk)
            tdsp_breq <= 0 ;
            @(posedge clk);
        end
    endtask

```

```

task dma_check;
    begin
        @(negedge tdsp_grant)
        begin
            dma_cnt2 =0;

```

```

for (k=0 ; k<50 ; k=k+1 )
  @(posedge clk)
  begin
    if(dma_grant != 1 )
      begin
        dma_cnt2 = dma_cnt2 + 1;

        //      $display("dma counter ", dma_cnt2);
      end
    else if(dma_grant == 1)
      begin
        k <= 50;
      end
    if ( dma_cnt2 > 2)
      begin
        dma_error1bit <= 1;

        $display( "%t DMA_GRANT ERROR", $time);

        $stop;
      end
    @(posedge clk);
  end
end
end
endtask

```

```

task tdsp_check;

begin

@(negedge dma_grant)

begin

for (l=0 ; l<50 ; l=l+1 )

@(posedge clk)

begin

if(tdsp_grant != 1 ) //&& dma_breq ==1)

begin

tdsp_cnt2 = tdsp_cnt2 + 1;

// $display ("tdsp counter", tdsp_cnt2);

end

else if(tdsp_grant == 1)

begin

l <= 50;

end

if ( tdsp_cnt2 > 1)

begin

tdsp_error1bit <= 1;

$display( "%t TDSP_GRANT ERROR", $time);

$stop;

end

@(posedge clk);

end

end

```

end

end

endtask

endmodule

F Synthesis report

```
arb_tsmc18_dc.rpt (~ee620/proj3/tdsp/arb/report/dc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
arb_tsmc18_scan_dc.rpt arb_tsmc18_dc.rpt
1
dc_shell>
*****
Report : area
Design : arb
Version: 0-2010.03-SP3
Date : Tue Dec 9 20:25:14 2014
*****

Library(s) Used:
    typical (File: /classes/ee620/maiee/lib/tsmc-0.18/synopsys/typical.db)

Number of ports:      9
Number of nets:       27
Number of cells:      22
Number of references: 12

Combinational area:   199.584002
Noncombinational area: 342.619202
Net Interconnect area: undefined (Wire load has zero net area)

Total cell area:      542.203204
Total area:           undefined

1
dc_shell>
*****
Report : reference
Design : arb
Version: 0-2010.03-SP3
Date : Tue Dec 9 20:25:14 2014
*****

Attributes:
b - black box (unknown)
bo - allows boundary optimization
d - dont_touch
mo - map_only
h - hierarchical
n - noncombinational
r - removable
s - synthetic operator
u - contains unmapped logic

Reference      Library      Unit Area      Count      Total Area      Attributes
-----
A0I21X1        typical      13.305600      1          13.305600
A0I22X1        typical      16.632000      1          16.632000
A0I31X1        typical      16.632000      1          16.632000
A0I211X1       typical      16.632000      1          16.632000
DFFRHX1        typical      69.854401      4          279.417603  n
DFFSX1         typical      63.201599      1          63.201599  n
JNWX1          typical      6.652800      7          46.569601
NAND3X1        typical      13.305600      1          13.305600
NOR2X1         typical      9.979200      2          19.958401
NOR3X1         typical      13.305600      1          13.305600
OAI22X1        typical      19.958401      1          19.958401

Plain Text Tab Width: 8 Ln 83, Col 13 INS
```

```
arb_tsmc18_dc.rpt (~ee620/proj3/tdsp/arb/report/dc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
arb_tsmc18_scan_dc.rpt arb_tsmc18_dc.rpt

Des/Clust/Port Wire Load Model Library
arb TSMC18_Conservative typical

Point Incr Path
-----
clock clk (rise edge) 0.0000 0.0000
clock network delay (ideal) 0.0000 0.0000
dma_grant_reg/CK (DFFRQX1) 0.0000 0.0000 r
dma_grant_reg/Q (DFFRQX1) 0.2225 0.2225 r
dma_grant (out) 0.0000 0.2225 r
data arrival time 0.2225
-----
clock clk (rise edge) 20.0000 20.0000
clock network delay (ideal) 0.0000 20.0000
clock uncertainty -0.2500 19.7500
output external delay -1.0000 18.7500
data required time 18.7500
-----
data required time 18.7500
data arrival time -0.2225
-----
slack (MET) 18.5275

Startpoint: tdsp_grant_reg
(rising edge-triggered flip-flop clocked by clk)
Endpoint: tdsp_grant (output port clocked by clk)
Path Group: clk
Path Type: max

Des/Clust/Port Wire Load Model Library
arb TSMC18_Conservative typical

Point Incr Path
-----
clock clk (rise edge) 0.0000 0.0000
clock network delay (ideal) 0.0000 0.0000
tdsp_grant_reg/CK (DFFRQX1) 0.0000 0.0000 r
tdsp_grant_reg/Q (DFFRQX1) 0.2225 0.2225 r
tdsp_grant (out) 0.0000 0.2225 r
data arrival time 0.2225
-----
clock clk (rise edge) 20.0000 20.0000
clock network delay (ideal) 0.0000 20.0000
clock uncertainty -0.2500 19.7500
output external delay -1.0000 18.7500
data required time 18.7500
-----
data required time 18.7500
data arrival time -0.2225
-----
slack (MET) 18.5275

tdsp Plain Text Tab Width: 8 Ln 92, Col 17 INS
```

```
arb_tsmc18_scan_dc.rpt (~ee620/proj3/tdsp/arb/report/dc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
arb_tsmc18_scan_dc.rpt

Pattern Summary Report
-----
#internal patterns 0
-----

Uncollapsed Stuck Fault Summary Report
-----
fault class code #faults
-----
Detected DT 196
Possibly detected PT 0
Undetectable UD 0
ATPG untestable AU 0
Not detected ND 0
-----
total faults 196
test coverage 100.00%
-----
Information: The test coverage above may be inferior
than the real test coverage with customized
protocol and test simulation library.

1
dc_shell> Post-compile check design
dc_shell> Warning: In design 'arb', output port 'tdsp_grant' is connected directly to output port 'scan_out0'. (LINT-31)
1
dc_shell> Information: Updating graph... (UID-83)
-----
Report : area
Design : arb
Version: D-2010.03-SP3
Date : Tue Dec 9 20:25:33 2014
-----
Information: Updating design information... (UID-85)
Library(s) Used:
typical (File: /classes/ee620/maiee/lib/tsmc-0.18/synopsys/typical.db)

Number of ports: 9
Number of nets: 29
Number of cells: 22
Number of references: 12

Combinational area: 199.584002
Noncombinational area: 412.473618
Net Interconnect area: undefined (Wire load has zero net area)

Total cell area: 612.057620
Total area: undefined
1
dc_shell>
-----
Report : reference
Design : arb

Plain Text Tab Width: 8 Ln 103, Col 69 INS
```

arb_tsmc18_dc.rpt [~/ee620/proj3/tdsp/arb/report/dc] - gedit

File Edit View Search Tools Documents Help

Open Save Undo

arb_tsmc18_scan_dc.rpt arb_tsmc18_dc.rpt

Des/Clust/Port	Wire Load Model	Library
arb	TSMC18_Conservative	typical

Point

	Incr	Path
clock clk (rise edge)	0.0000	0.0000
clock network delay (ideal)	0.0000	0.0000
dma_grant_reg/CK (DFFR4QX1)	0.0000	0.0000 r
dma_grant_reg/0 (DFFR4QX1)	0.2225	0.2225 r
dma_grant (out)	0.0000	0.2225 r
data arrival time		0.2225
clock clk (rise edge)	20.0000	20.0000
clock network delay (ideal)	0.0000	20.0000
clock uncertainty	-0.2500	19.7500
output external delay	-1.0000	18.7500
data required time		18.7500
data arrival time		-0.2225
slack (MET)		18.5275

Startpoint: tdsp_grant_reg
(rising edge-triggered flip-flop clocked by clk)
Endpoint: tdsp_grant (output port clocked by clk)
Path Group: clk
Path Type: max

Des/Clust/Port	Wire Load Model	Library
arb	TSMC18_Conservative	typical

Point

	Incr	Path
clock clk (rise edge)	0.0000	0.0000
clock network delay (ideal)	0.0000	0.0000
tdsp_grant_reg/CK (DFFR4QX1)	0.0000	0.0000 r
tdsp_grant_reg/0 (DFFR4QX1)	0.2225	0.2225 r
tdsp_grant (out)	0.0000	0.2225 r
data arrival time		0.2225
clock clk (rise edge)	20.0000	20.0000
clock network delay (ideal)	0.0000	20.0000
clock uncertainty	-0.2500	19.7500
output external delay	-1.0000	18.7500
data required time		18.7500
data arrival time		-0.2225
slack (MET)		18.5275

tdsp Plain Text Tab Width: 8 Ln 92, Col 17 INS

G Various timing analyzer

zxw6805@vlsi-legacy:~/ee620/proj3/tdsp/arb

File Edit View Search Terminal Help

G - Generated clock
I - Inactive clock

Clock	Period	Waveform	Attrs	Sources
clk	20.00	{0 10}	p	{clk}

1
pt shell> pt shell> *****
Report : analysis coverage
-status details (untested violated)
-sort by slack
Design : arb
Version: D-2010.06-SP1
Date : Tue Dec 9 21:25:54 2014

Type of Check	Total	Met	Violated	Untested
setup	15	10 (67%)	0 (0%)	5 (33%)
hold	15	10 (67%)	0 (0%)	5 (33%)
recovery	5	0 (0%)	0 (0%)	5 (100%)
min_pulse_width	15	10 (67%)	0 (0%)	5 (33%)
out_setup	3	3 (100%)	0 (0%)	0 (0%)
out_hold	3	3 (100%)	0 (0%)	0 (0%)
All Checks	56	36 (64%)	0 (0%)	20 (36%)

Constrained Pin	Related Pin	Check Type	Slack	Reason
dma_grant_reg/RN(low)		min_pulse_width		
dma_grant_reg/RN(rise)	OK(rise)	recovery	untested	no clock false_paths
dma_grant_reg/SE	OK(rise)	hold	untested	false_paths
dma_grant_reg/SE	OK(rise)	setup	untested	false_paths
state_reg_0/SE	OK(rise)	hold	untested	false_paths
state_reg_0/SE	OK(rise)	setup	untested	false_paths
state_reg_0/SN(low)		min_pulse_width		
state_reg_0/SN(rise)	OK(rise)	recovery	untested	no clock false_paths
state_reg_1/RN(low)		min_pulse_width		
state_reg_1/RN(rise)	OK(rise)	recovery	untested	no clock false_paths
state_reg_1/SE	OK(rise)	hold	untested	false_paths
state_reg_1/SE	OK(rise)	setup	untested	false_paths
state_reg_2/RN(low)		min_pulse_width		
state_reg_2/RN(rise)	OK(rise)	recovery	untested	no clock false_paths
state_reg_2/SE	OK(rise)	hold	untested	false_paths
state_reg_2/SE	OK(rise)	setup	untested	false_paths
tdsp_grant_reg/RN(low)		min_pulse_width		
tdsp_grant_reg/RN(rise)	OK(rise)	recovery	untested	no clock false_paths
tdsp_grant_reg/SE	OK(rise)	hold	untested	false_paths
tdsp_grant_reg/SE	OK(rise)	setup	untested	false_paths

1
pt shell> pt shell> *****
Report : case_analysis
-all
Design : arb
Version: D-2010.06-SP1

arb_tsmc18_scan_pt_rac.rpt (~ee620/proj3/tdsp/arb/report/pt) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

arb_tsmc18_scan_pt_rac.rpt

Report : analysis.coverage
-status_details (untested violated)
-sort_by slack
Design : arb
Version: D-2010.06-SP1
Date : Tue Dec 9 21:25:54 2014

Type of Check	Total	Met	Violated	Untested
setup	15	10 (67%)	0 (0%)	5 (33%)
hold	15	10 (67%)	0 (0%)	5 (33%)
recovery	5	0 (0%)	0 (0%)	5 (100%)
min_pulse_width	15	10 (67%)	0 (0%)	5 (33%)
out_setup	3	3 (100%)	0 (0%)	0 (0%)
out_hold	3	3 (100%)	0 (0%)	0 (0%)
All Checks	56	36 (64%)	0 (0%)	20 (36%)

Constrained Pin	Related Pin	Check Type	Slack	Reason
dma_grant_reg/RN(low)		min_pulse_width		no clock
dma_grant_reg/RN(rise)	CK(rise)	recovery	untested	false_paths
dma_grant_reg/SE	CK(rise)	hold	untested	false_paths
dma_grant_reg/SE	CK(rise)	setup	untested	false_paths
state_reg_0/SE	CK(rise)	hold	untested	false_paths
state_reg_0/SE	CK(rise)	setup	untested	false_paths
state_reg_0/SN(low)		min_pulse_width		no clock
state_reg_0/SN(rise)	CK(rise)	recovery	untested	false_paths
state_reg_1/RN(low)		min_pulse_width		no clock
state_reg_1/RN(rise)	CK(rise)	recovery	untested	false_paths
state_reg_1/SE	CK(rise)	hold	untested	false_paths
state_reg_1/SE	CK(rise)	setup	untested	false_paths
state_reg_2/RN(low)		min_pulse_width		no clock
state_reg_2/RN(rise)	CK(rise)	recovery	untested	false_paths
state_reg_2/SE	CK(rise)	hold	untested	false_paths
state_reg_2/SE	CK(rise)	setup	untested	false_paths
tdsp_grant_reg/RN(low)		min_pulse_width		no clock
tdsp_grant_reg/RN(rise)	CK(rise)	recovery	untested	false_paths
tdsp_grant_reg/SE	CK(rise)	hold	untested	false_paths
tdsp_grant_reg/SE	CK(rise)	setup	untested	false_paths

Plain Text Tab Width: 8 Ln 50, Col 2 INS

arb_tsmc18_scan_pt_rpt.rpt (~ee620/proj3/tdsp/arb/report/pt) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

arb_tsmc18_scan_pt_rpt.rpt

Point	Incr	Path
clock clk (rise edge)	0.0000	0.0000
clock network delay (propagated)	0.0000	0.0000
input external delay	1.0000	1.0000 f
dma_breq (in)	0.0262	1.0262 f
U49/Y (INVX1)	0.2250 *	1.2512 r
U48/Y (NOR3X1)	0.1510 *	1.4022 f
U47/Y (AOI31X1)	0.1660 *	1.5682 r
U44/Y (AOI221X1)	0.1760 *	1.7442 f
state_reg_0/D (SDFFSXL)	0.0000 *	1.7442 f
data arrival time		1.7442
clock clk (rise edge)	20.0000	20.0000
clock network delay (propagated)	0.0000	20.0000
clock uncertainty	-0.2500	19.7500
state_reg_0/CK (SDFFSXL)		19.7500 r
library setup time	-0.2590 *	19.4910
data required time		19.4910
data arrival time		-1.7442
slack (MET)		17.7468

Startpoint: dma_breq (input port clocked by clk)
Endpoint: state_reg_1 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

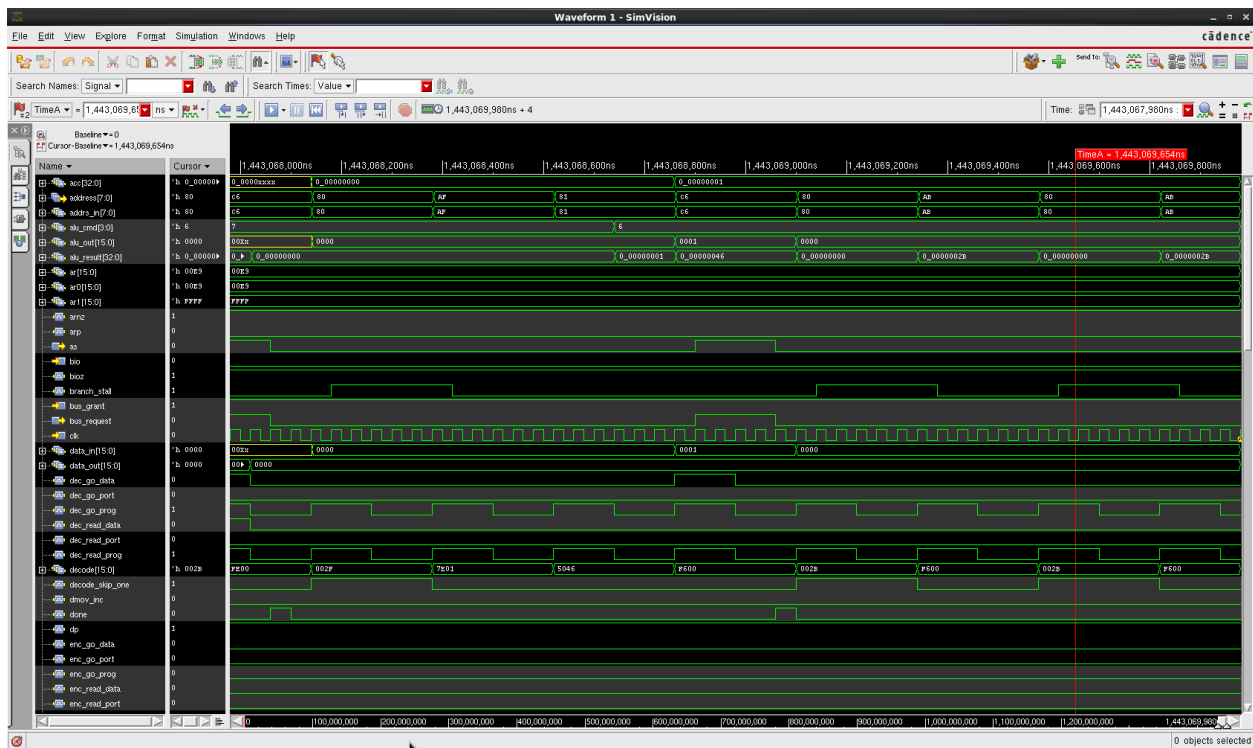
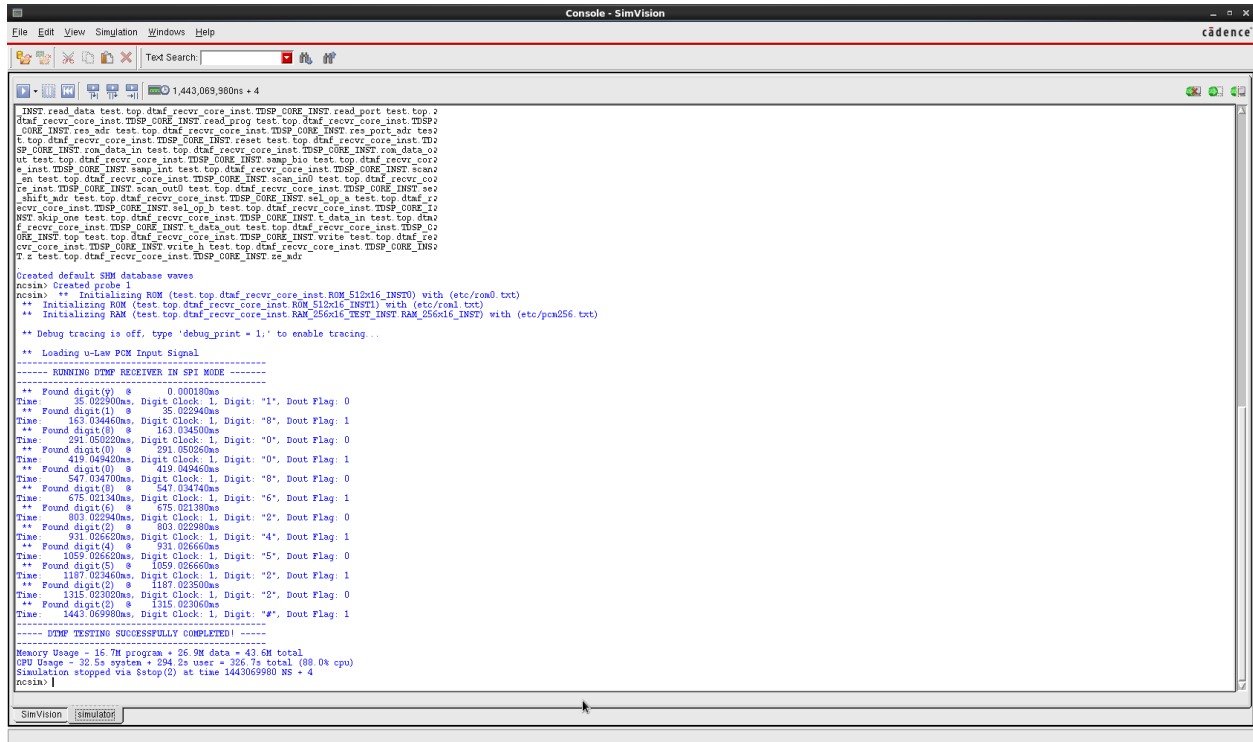
Point	Incr	Path
clock clk (rise edge)	0.0000	0.0000
clock network delay (propagated)	0.0000	0.0000
input external delay	1.0000	1.0000 f
dma_breq (in)	0.0262	1.0262 f
U49/Y (INVX1)	0.2250 *	1.2512 r
U53/Y (NAND3X1)	0.0600 *	1.3202 f
U51/Y (AOI22X1)	0.1380 *	1.4582 r
U50/Y (AOI22X1)	0.0900 *	1.5562 f
state_reg_1/D (SDFFRHQXL)	0.0000 *	1.5562 f
data arrival time		1.5562
clock clk (rise edge)	20.0000	20.0000
clock network delay (propagated)	0.0000	20.0000
clock uncertainty	-0.2500	19.7500
state_reg_1/CK (SDFFRHQXL)		19.7500 r
library setup time	-0.2880 *	19.4620
data required time		19.4620
data arrival time		-1.5562
slack (MET)		17.9058

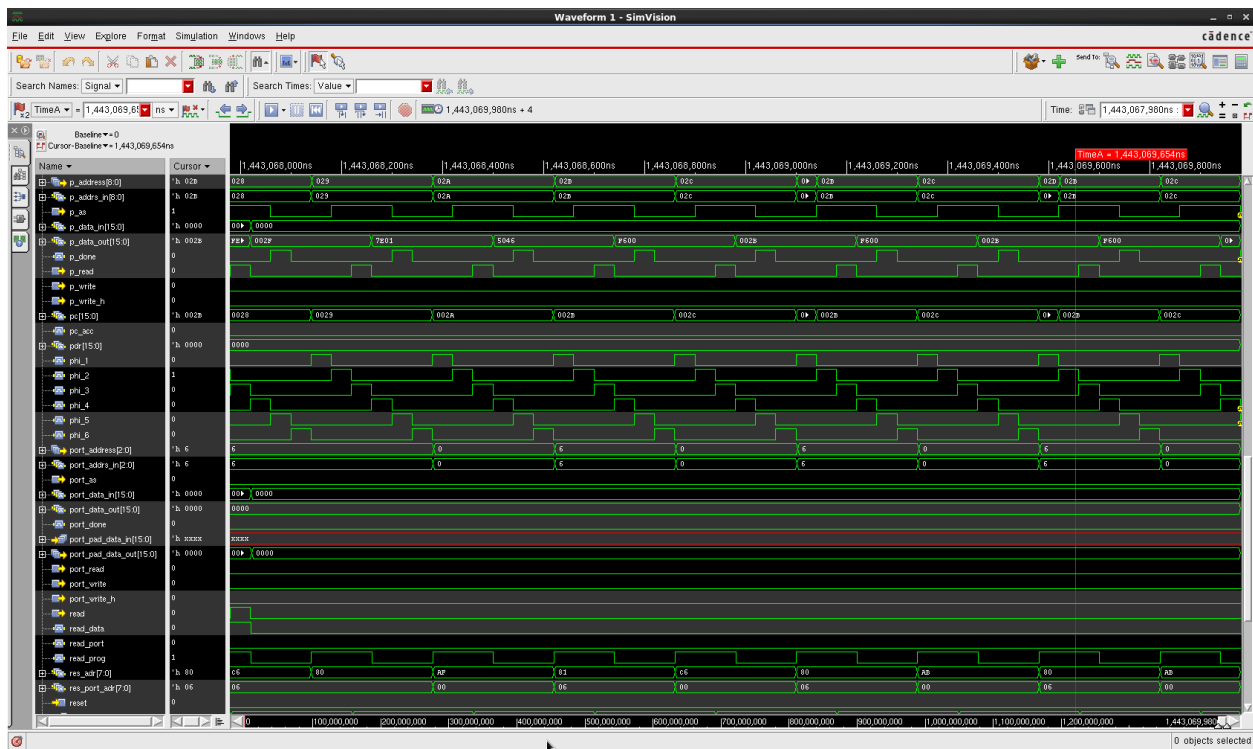
Plain Text Tab Width: 8 Ln 121, Col 64 INS

3 DTMF receiver core RTL new memory access bus arbiter (ARB) modules

DTMF simulation RTL

A : screenshots of RTL level simulation log and waveforms





B: Screen shots of Netlist level simulation log and waveforms

```

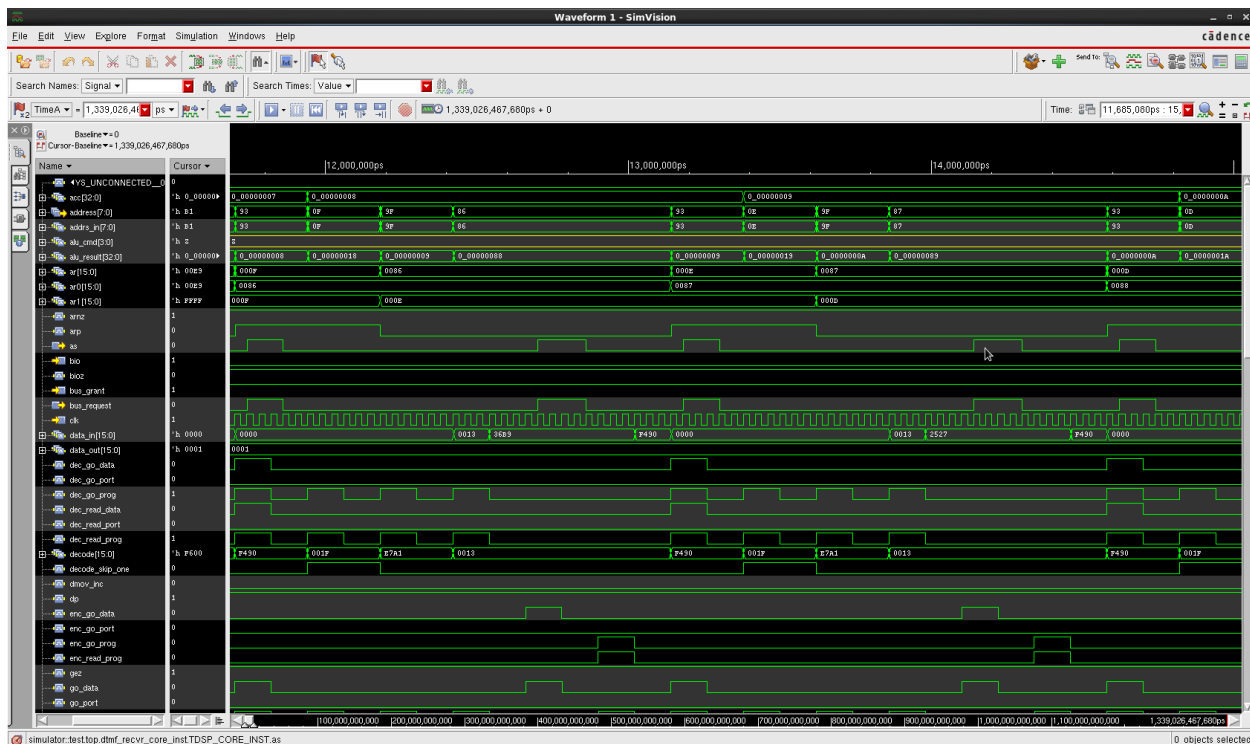
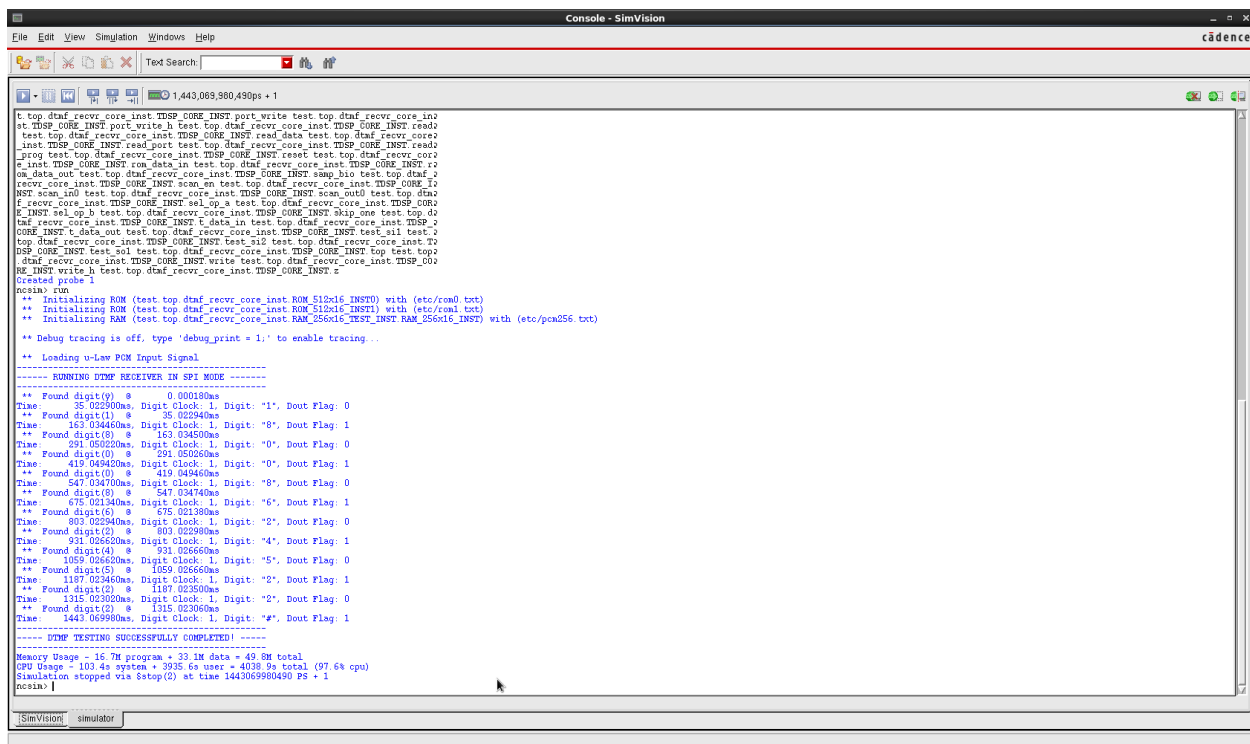
zxw6805@vlsi-legacy: ~/ee620/proj3/tdsp/dtmf+hummm
File Edit View Search Terminal Help

tsc18.SDFFRQX1.v <0x4455a9c>
  streams: 3, words: 290
tsc18.SDFFRQX2.v <0x6ae4fc8>
  streams: 3, words: 290
tsc18.SDFFRQX4.v <0x26ad33c>
  streams: 3, words: 292
tsc18.SDFFRQX1.v <0x396f23ec>
  streams: 3, words: 290
tsc18.SDFFSXI.v <0x38c2995c>
  streams: 5, words: 450
tsc18.SDFFSXI.v <0x70849fc>
  streams: 5, words: 446
tsc18.SDFFX1.v <0x282fa44c>
  streams: 5, words: 450
tsc18.SDFFX1.v <0x578915b>
  streams: 5, words: 594
tsc18.SDFFX1.v <0x63a2519b>
  streams: 5, words: 594
tsc18.XORX1.v <0x7f13386d>
  streams: 4, words: 426
tsc18.XORX1.v <0x2f1e049>
  streams: 4, words: 426
tsc18.XORX2.v <0x5773ae32>
  streams: 10, words: 1858
worklib.dtmf_humm.v <0x7586d69a>
  streams: 1, words: 448
worklib.ram_128x16.v <0x1a7a33fd>
  streams: 7, words: 5732
worklib.ram_128x16_test.v <0x2f93c14>
  streams: 1, words: 650
worklib.ram_256x16_test.v <0x5c9fdeb5>
  streams: 1, words: 650
worklib.ram_512x16_0.v <0x7bd2775d>
  streams: 6, words: 5120
worklib.ram_512x16_1.v <0x73138037>
  streams: 6, words: 5120
worklib.test.v <0x9fb8e141>
  streams: 46, words: 48438

Loading native compiled code: ..... Done
Building instance specific data structures.
Design hierarchy summary:
  Instances Unique
Modules: 4448 123
UDPs: 1262 6
Primitives: 12416 8
Timing outputs: 5974 72
Registers: 666 63
Scalar wires: 7896 -
Expanded wires: 80 5
Vectorized wires: 20 -
Named events: 2 2
Always blocks: 15 15
Initial blocks: 10 10
Cont. assignments: 261 261
Pseudo assignments: 652 46
Timing checks: 8592 -
Interconnect: 13555 -
Simulation timescale: 10ps

Writing initial simulation snapshot: worklib.test.v
ee620

```



C logic synthesis report

Pre scan total cell area and total number of cells

dtmf_recvr_core_tsmc18_dc.rpt (~ee620/proj3/tdsp/dtmf+hummm/report/dc) - gedit						
File Edit View Search Tools Documents Help						
Open Save Undo						
dtmf_recvr_core_tsmc18_dc.rpt X						
typical (File: /classes/ee620/maiee/lib/tsmc-0.18/synopsys/typical.db) ram_256x16_typical (File: /classes/ee620/maiee/lib/tsmc-0.18/model/ram_256x16_typical.db) ram_128x16_typical (File: /classes/ee620/maiee/lib/tsmc-0.18/model/ram_128x16_typical.db) rom_512x16_typical (File: /classes/ee620/maiee/lib/tsmc-0.18/model/rom_512x16_typical.db)						
Number of ports: 56 Number of nets: 297 Number of cells: 15 Number of references: 15						
Combinational area: 75389.538375 Noncombinational area: 254370.532417 Net Interconnect area: undefined (Wire load has zero net area)						
Total cell area: 329760.062792 Total area: undefined						
dc_shell> ***** Report : reference Design : dtmf_recvr_core Version: D-2010.03-SP3 Date : Tue Dec 9 22:03:38 2014 *****						
Attributes: b - black box (unknown) bo - allows boundary optimization d - dont_touch mo - map_only h - hierarchical n - noncombinational r - removable s - synthetic operator u - contains unmapped logic						
Reference	Library	Unit Area	Count	Total Area	Attributes	

arb		598.752004	1	598.752004	h, n	
data_sample_mux		595.425601	1	595.425601	h	
digit_reg		575.467192	1	575.467192	h, n	
dma		2039.083219	1	2039.083219	h, n	
ram_128x16_test		49387.015201	1	49387.015201	b, h	
ram_256x16_test		69484.933626	1	69484.933626	b, h	
results_conv		33789.571625	1	33789.571625	h, n	
rom_512x16_0		47528.105925	1	47528.105925	b, h	
rom_512x16_1		47528.105925	1	47528.105925	b, h	
spl		2095.632043	1	2095.632043	h, n	
tdsp_core		72918.015205	1	72918.015205	h, n	
tdsp_data_mux		379.209601	1	379.209601	h	
tdsp_ds_cs		395.841607	1	395.841607	h, n	
test_control		512.265601	1	512.265601	h	
ulaw_lin_conv		1932.638417	1	1932.638417	h	

Total 15 references				329760.062792		
1						
Plain Text Tab Width: 8 Ln 1570, Col 68 INS						

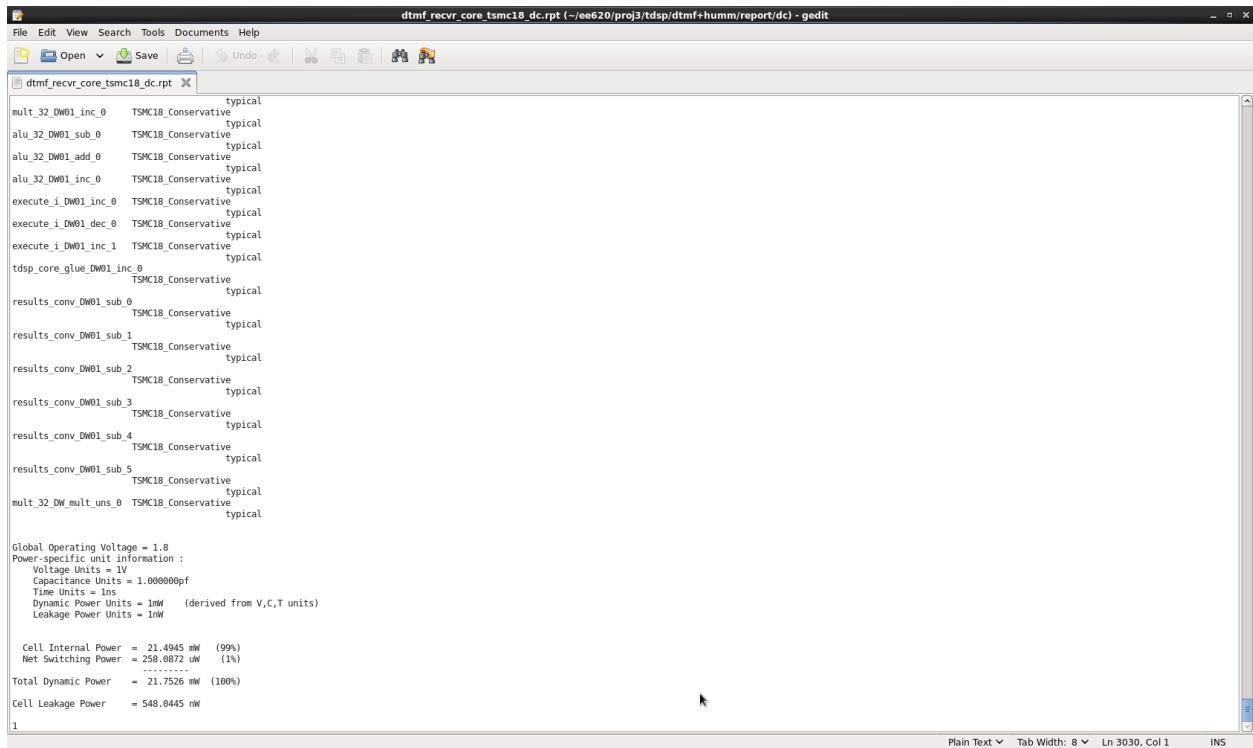
Worst time

dtmf_recvr_core_tsmc18_dc.rpt (~ee620/proj3/tdsp/dtmf+hummm/report/dc) - gedit			
File Edit View Search Tools Documents Help			
Open Save Undo			
dtmf_recvr_core_tsmc18_dc.rpt X			
TSP CORE INST/ALU_32 INST/sub 04/U2_07/CO (A0DFX2) 0.2033 4.40310 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_7/CO (A0DFX2) 0.2033 4.6552 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_8/CO (A0DFX2) 0.2033 4.8585 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_9/CO (A0DFX2) 0.2033 5.0619 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_10/CO (A0DFX2) 0.2033 5.2652 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_11/CO (A0DFX2) 0.2033 5.4685 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_12/CO (A0DFX2) 0.2033 5.6719 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_13/CO (A0DFX2) 0.2037 5.8756 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_14/CO (A0DFX2) 0.2037 6.0793 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_15/CO (A0DFX2) 0.2037 6.2830 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_16/CO (A0DFX2) 0.2039 6.4858 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_17/CO (A0DFX2) 0.2038 6.6897 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_18/CO (A0DFX2) 0.2038 6.8935 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_19/CO (A0DFX2) 0.2038 7.0973 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_20/CO (A0DFX2) 0.2038 7.3011 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_21/CO (A0DFX2) 0.2038 7.5049 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_22/CO (A0DFX2) 0.2038 7.7087 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_23/CO (A0DFX2) 0.2038 7.9125 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_24/CO (A0DFX2) 0.2038 8.1164 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_25/CO (A0DFX2) 0.2038 8.3202 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_26/CO (A0DFX2) 0.2038 8.5240 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_27/CO (A0DFX2) 0.2038 8.7278 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_28/CO (A0DFX2) 0.2038 8.9316 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_29/CO (A0DFX2) 0.2038 9.1355 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_30/CO (A0DFX2) 0.2104 9.3458 f			
TSP CORE INST/ALU_32 INST/sub 04/U2_31/Y (X0R3X2) 0.1941 9.5399 r			
TSP CORE INST/ALU_32 INST/sub 04/DIFF[31] (alu_32_0001 sub 0) 0.0000 9.5399 r			
TSP CORE INST/ALU_32 INST/U35/Y (A0122X1) 0.0557 9.5957 f			
TSP CORE INST/ALU_32 INST/U37/Y (NAND3BX1) 0.1721 9.7607 r			
TSP CORE INST/ALU_32 INST/U36/Y (NAND2X1) 0.0785 9.8472 f			
TSP CORE INST/ALU_32 INST/U11/Y (INWX1) 0.1637 10.0109 r			
TSP CORE INST/ALU_32 INST/U32/Y (A0133X1) 0.1063 10.1172 f			
TSP CORE INST/ALU_32 INST/U193/Y (AND2X2) 0.1014 10.2086 f			
TSP CORE INST/ALU_32 INST/U267/Y (NAND2X1) 0.1502 10.4487 r			
TSP CORE INST/ALU_32 INST/U149/Y (NAND2BX1) 1.3058 11.7545 r			
TSP CORE INST/ALU_32 INST/U55/Y (A0121X1) 0.1040 11.8585 f			
TSP CORE INST/ALU_32 INST/result[1] (alu_32) 0.0000 11.8585 f			
TSP CORE INST/EXECUTE_INST/alu_result[1] (execute_i) 0.0000 11.8585 f			
TSP CORE INST/EXECUTE_INST/acc_reg[1]/D (EDFFXL) 0.0000 11.8585 f			
data arrival time 11.8585			
clock clk (rise edge) 20.0000 20.0000			
clock network delay (ideal) 0.0000 20.0000			
clock uncertainty -0.2500 19.7500			
TSP CORE INST/EXECUTE_INST/acc_reg[1]/CK (EDFFXL) 0.0000 19.7500 r			
library setup time -0.3234 19.4266			
data required time 19.4266			

data required time 19.4266			
data arrival time -11.8585			

slack (MET) 7.5681			
1			
Plain Text Tab Width: 8 Ln 1562, Col 25 INS			

Power consumption



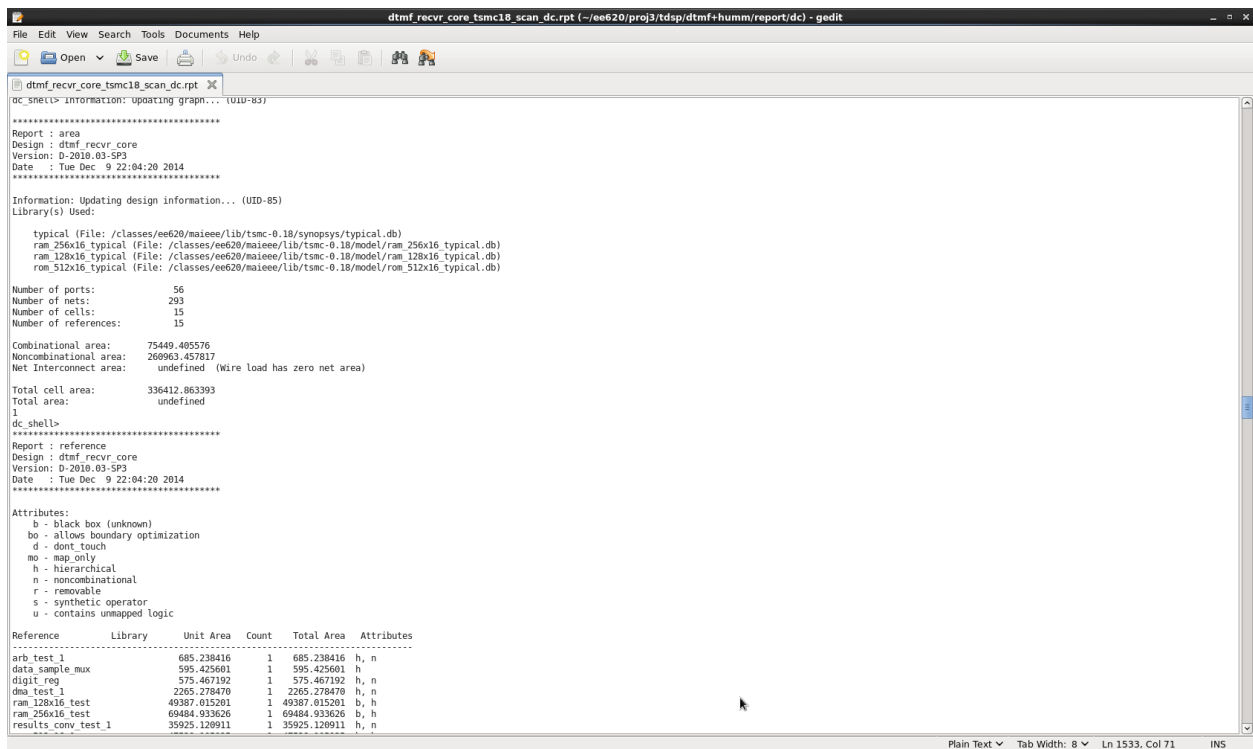
```
dtmf_recvr_core_tsmc18_dc.rpt (~ee620/proj3/tdsp/dtmf+hummm/report/dc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
dtmf_recvr_core_tsmc18_dc.rpt X
mult_32_DW01_inc_0 TSMC18_Conversative typical
alu_32_DW01_sub_0 TSMC18_Conversative typical
alu_32_DW01_add_0 TSMC18_Conversative typical
alu_32_DW01_inc_0 TSMC18_Conversative typical
execute_i_DW01_inc_0 TSMC18_Conversative typical
execute_i_DW01_dec_0 TSMC18_Conversative typical
execute_i_DW01_inc_1 TSMC18_Conversative typical
tdsp_core_glue_DW01_inc_0 TSMC18_Conversative typical
results_conv_DW01_sub_0 TSMC18_Conversative typical
results_conv_DW01_sub_1 TSMC18_Conversative typical
results_conv_DW01_sub_2 TSMC18_Conversative typical
results_conv_DW01_sub_3 TSMC18_Conversative typical
results_conv_DW01_sub_4 TSMC18_Conversative typical
results_conv_DW01_sub_5 TSMC18_Conversative typical
mult_32_DW_mult_0 TSMC18_Conversative typical

Global Operating Voltage = 1.8
Power-specific unit information :
Voltage Units = IV
Capacitance Units = 1.000000pf
Time Units = ns
Dynamic Power Units = mW (derived from V,C,T units)
Leakage Power Units = mW

Cell Internal Power = 21.4945 mW (99%)
Net Switching Power = 258.8872 mW (1%)
-----
Total Dynamic Power = 21.7526 mW (100%)
Cell Leakage Power = 548.8445 mW

1
Plain Text Tab Width: 8 Ln 3030, Col 1 INS
```

Posts scan total cell area and number of cells



```
dtmf_recvr_core_tsmc18_scan_dc.rpt (~ee620/proj3/tdsp/dtmf+hummm/report/dc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
dtmf_recvr_core_tsmc18_scan_dc.rpt X
oc_snet> information: updating graph... (UID-83)
*****
Report : area
Design : dtmf_recvr_core
Version: D-2018.03-SP3
Date : Tue Dec 9 22:04:20 2014
*****
Information: Updating design information... (UID-85)
Library(s) Used:
typical (File: /classes/ee620/maiee/lib/tsmc-0.18/synopsys/typical.db)
ram_256x16_typical (File: /classes/ee620/maiee/lib/tsmc-0.18/model/ram_256x16_typical.db)
ram_128x16_typical (File: /classes/ee620/maiee/lib/tsmc-0.18/model/ram_128x16_typical.db)
rom_512x16_typical (File: /classes/ee620/maiee/lib/tsmc-0.18/model/rom_512x16_typical.db)

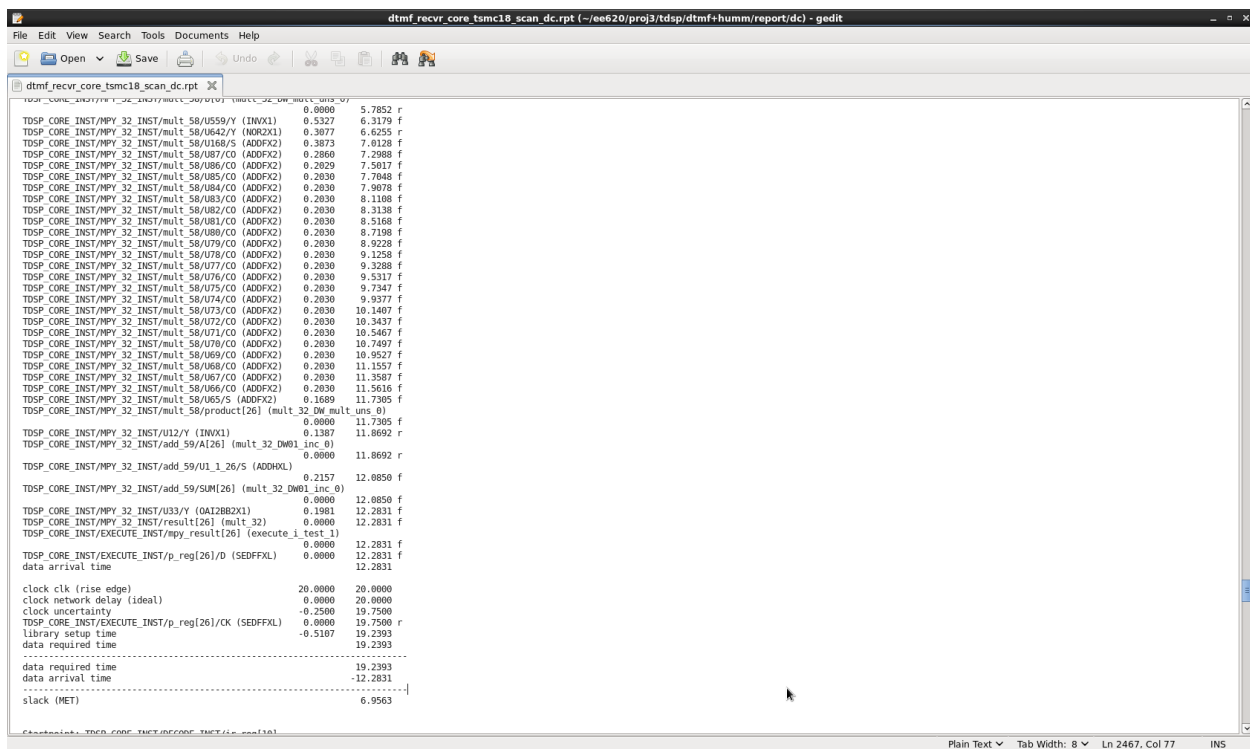
Number of ports: 56
Number of nets: 293
Number of cells: 15
Number of references: 15

Combinational area: 75449.495576
Noncombinational area: 260963.457817
Net Interconnect area: undefined (Wire load has zero net area)

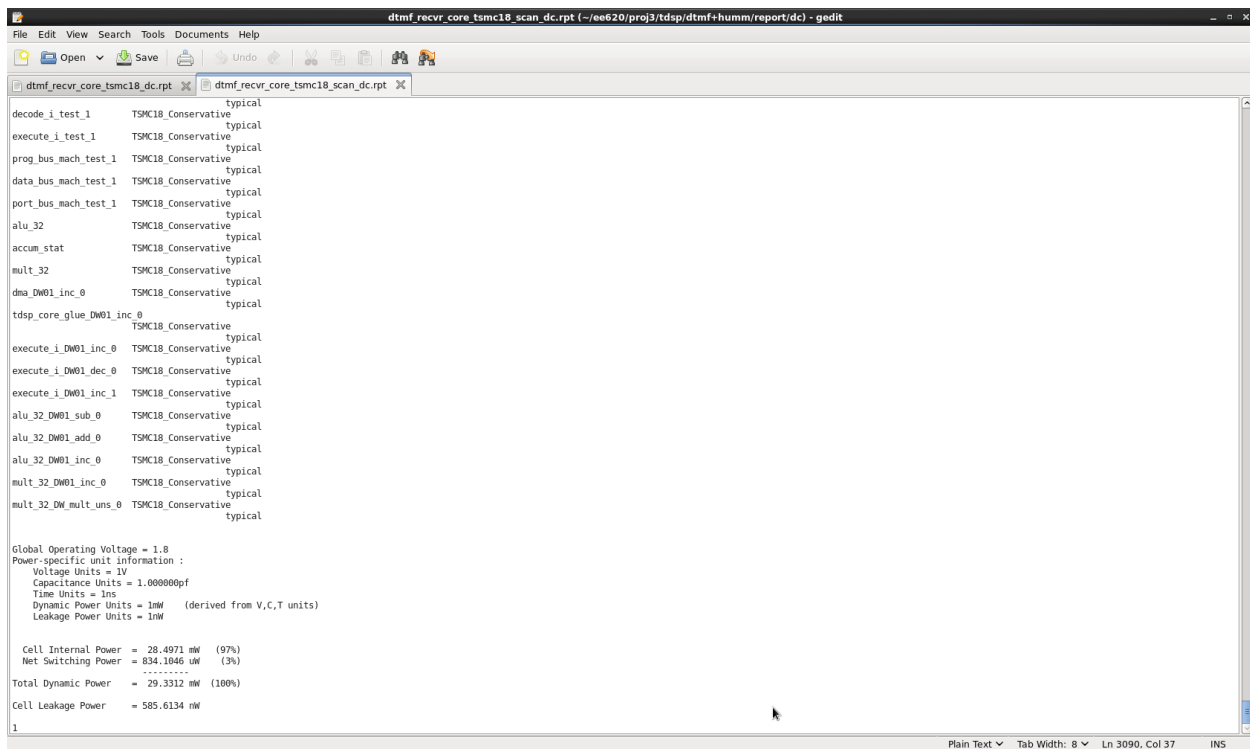
Total cell area: 336412.863393
Total area: undefined
1
dc_shell>
*****
Report : reference
Design : dtmf_recvr_core
Version: D-2018.03-SP3
Date : Tue Dec 9 22:04:20 2014
*****
Attributes:
b - black box (unknown)
bo - allows boundary optimization
d - dont_touch
mo - map_only
h - hierarchical
n - noncombinational
r - removable
s - synthetic operator
u - contains unmapped logic

Reference Library Unit Area Count Total Area Attributes
-----
arb_test_1 685.238416 1 685.238416 h, n
data_sample_mux 595.425601 1 595.425601 h
digit_rom 575.467192 1 575.467192 h, n
dma_test_1 2265.278470 1 2265.278470 h, n
ram_128x16_test 49387.015201 1 49387.015201 b, h
ram_256x16_test 69484.933626 1 69484.933626 b, h
results_conv_test_1 35925.120911 1 35925.120911 h, n
-----
Plain Text Tab Width: 8 Ln 1533, Col 71 INS
```

Worst timing path



Power consumption



D various timing analyzer reports

Worst case for post-scan

dtmf_recvr_core_tsmc18_scan_pt_rt.rpt (-/ee620/proj3/tdsp/dtmf+hummm/report/pt) - gedit

dtmf_recvr_core_tsmc18_scan_pt_rac.rpt

dtmf_recvr_core_tsmc18_scan_pt_rt.rpt

Point

Incr

Path

clock clk (rise edge)

0.0000

0.0000

clock network delay (propagated)

0.0000

0.0000

input external delay

1.0000

1.0000 f

spl fs (in)

0.0079

1.0079 f

SPI_INST/spl.fs (spi_test_1)

0.0000 *

1.0079 f

SPI_INST/U20/Y (N0R2X1)

0.2700 *

1.3579 r

SPI_INST/bit_cnt_reg[3]/RN (DFFRHQX1)

0.0000 *

1.3579 r

data arrival time

1.3579

clock clk (rise edge)

20.0000

20.0000

clock network delay (propagated)

0.5460 *

20.5460

clock uncertainty

-0.2500

20.2960

SPI_INST/bit_cnt_reg[3]/CK (DFFRHQX1)

20.2960 r

library recovery time

-0.1300 *

20.1660

data required time

20.1660

data required time

20.1660

data arrival time

-1.3579

slack (MET)

18.8081

Startpoint: test mode (input port clocked by clk)

Endpoint: TEST_CONTROL_INST/U3

(falling clock gating-check end-point clocked by clk)

Path Group: **clock_gating_default**

Path Type: max

Point

Incr

Path

clock clk (rise edge)

0.0000

0.0000

clock network delay (propagated)

0.0000

0.0000

input external delay

1.0000

1.0000 r

test mode (in)

1.3072

2.3072 r

TEST_CONTROL_INST/test_mode (test_control)

0.0000 *

2.3072 r

TEST_CONTROL_INST/U3/B (NAND2BX1)

0.0000 *

2.3072 r

data arrival time

2.3072

clock clk (fall edge)

10.0000

10.0000

clock network delay (propagated)

0.0000

10.0000

clock uncertainty

-0.2500

9.7500

TEST_CONTROL_INST/U3/AN (NAND2BX1)

9.7500 f

clock gating setup time

0.0000

9.7500

data required time

9.7500

data required time

9.7500

data arrival time

-2.3072

slack (MET)

7.4428

Startpoint: TDSP_CORE_INST/EXECUTE_INST/arp_reg

(rising edge-triggered flip-flop clocked by clk)

Endpoint: TEST_CONTROL_INST/U19

(rising clock gating-check end-point clocked by clk)

Plain Text Tab Width: 8 Ln 168, Col 72 INS

Total timing analysis for post-scan

dtmf_recvr_core_tsmc18_scan_pt_rac.rpt (-/ee620/proj3/tdsp/dtmf+hummm/report/pt) - gedit

dtmf_recvr_core_tsmc18_scan_pt_rac.rpt

Date : Tue Dec 9 22:38:47 2014

Type of Check

Total

Met

Violated

Untested

setup

1731

1259 (73%)

0 (0%)

472 (27%)

hold

1731

1242 (72%)

17 (1%)

472 (27%)

recovery

354

6 (2%)

0 (0%)

348 (98%)

min_period

4

4 (100%)

0 (0%)

0 (0%)

min_pulse_width

1606

1252 (78%)

0 (0%)

354 (22%)

clock_gating_setup

3

3 (100%)

0 (0%)

0 (0%)

clock_gating_hold

3

1 (33%)

2 (67%)

0 (0%)

out_setup

28

28 (100%)

0 (0%)

0 (0%)

out_hold

28

28 (100%)

0 (0%)

0 (0%)

All Checks

5488

3823 (70%)

19 (0%)

1646 (30%)

Constrained Pin

Related Pin

Check Type

Slack

Reason

ARB_INST/dma_grant_reg/RN(low)

min_pulse_width

untested

no clock

ARB_INST/dma_grant_reg/RN(rise)

CK(rise)

recovery

untested

false_paths

ARB_INST/dma_grant_reg/SE

CK(rise)

hold

untested

false_paths

ARB_INST/dma_grant_reg/SE

CK(rise)

setup

untested

false_paths

ARB_INST/state_reg[0]/SE

CK(rise)

hold

untested

false_paths

ARB_INST/state_reg[0]/SE

CK(rise)

setup

untested

false_paths

ARB_INST/state_reg[0]/SN(low)

min_pulse_width

untested

no clock

ARB_INST/state_reg[0]/SN(rise)

CK(rise)

recovery

untested

false_paths

ARB_INST/state_reg[1]/RN(low)

min_pulse_width

untested

no clock

ARB_INST/state_reg[1]/RN(rise)

CK(rise)

recovery

untested

false_paths

ARB_INST/state_reg[1]/SE

CK(rise)

hold

untested

false_paths

ARB_INST/state_reg[1]/SE

CK(rise)

setup

untested

false_paths

ARB_INST/state_reg[2]/RN(low)

min_pulse_width

untested

no clock

ARB_INST/state_reg[2]/RN(rise)

CK(rise)

recovery

untested

false_paths

ARB_INST/state_reg[2]/SE

CK(rise)

hold

untested

false_paths

ARB_INST/state_reg[2]/SE

CK(rise)

setup

untested

false_paths

ARB_INST/tdsp_grant_reg/RN(low)

min_pulse_width

untested

no clock

ARB_INST/tdsp_grant_reg/RN(rise)

CK(rise)

recovery

untested

false_paths

ARB_INST/tdsp_grant_reg/SE

CK(rise)

hold

untested

false_paths

ARB_INST/tdsp_grant_reg/SE

CK(rise)

setup

untested

false_paths

DIGIT_REG_INST/flag_out_reg/SN(low)

min_pulse_width

untested

no clock

DIGIT_REG_INST/flag_out_reg/SN(rise)

CK(rise)

recovery

untested

false_paths

DIGIT_REG_INST/out_reg[0]/SN(low)

min_pulse_width

untested

no clock

DIGIT_REG_INST/out_reg[0]/SN(rise)

CK(rise)

recovery

untested

false_paths

DIGIT_REG_INST/out_reg[1]/SN(low)

min_pulse_width

untested

no clock

DIGIT_REG_INST/out_reg[1]/SN(rise)

CK(rise)

recovery

untested

false_paths

DIGIT_REG_INST/out_reg[2]/SN(low)

min_pulse_width

untested

no clock

Plain Text Tab Width: 8 Ln 1, Col 1 INS

4 TDSP assembly language test program

A Program Description

TDSP assembly language test program is for testing with X, Y and Z pointing to specific memory locations. In this project, specific bit codes are assigned. $Y = X^2 + (2 * Z) - 1$, $X = X + 1$ and $Z = Y$ operations are performed in assembly test program. If finite state machine Verilog and the test bench of assembly are implemented correctly, the # spectrum should be seen in simulation console. The DTMF_TDSP net list simulation takes about 1 hour to finish.

B Assembly language test program source code

```
X = 0x6;
```

```
LACK 1;
```

```
SACL X;
```

```
Y = 0x7;
```

```
LACK 0;
```

```
SACL Y;
```

```
Z = 0x8;
```

```
LACK 0;
```

```
SACL Z;
```

```
A= 0x9
```

```
LACK 1;
```

```
SACL A;
```


two =0x2;

;larp ar0, 0x40;

waitTime =0x1E;

;REGA =0;

;LACK 0x40; 64 decmial value is stored in register A

;SACL REGA;

page0 = 0 ;

base_page0 = 0x000;

base_page1 = 0x080 ; memory page 1

ptr =21

rcc_ptr =22

rcc_697Hz = (0x00e0 & 0x07f) ;

rcc_770Hz = (0x00e1 & 0x07f) ;

rcc_852Hz = (0x00e2 & 0x07f) ;

rcc_941Hz = (0x00e3 & 0x07f) ;

rcc_1209Hz = (0x00e4 & 0x07f) ;

rcc_1336Hz = (0x00e5 & 0x07f) ;

rcc_1477Hz = (0x00e6 & 0x07f) ;

rcc_1633Hz = (0x00e7 & 0x07f) ;

rcc_kick= (0x00e8 & 0x07f) ;

rcc_len = (rcc_kick - rcc_697Hz) ;

;

;pointer

;rc_697Hz = 0x7fff * fp_rc_697Hz ;

;rc_770Hz = 0x7fff * fp_rc_770Hz ;

;rc_852Hz = 0x7fff * fp_rc_852Hz ;

;rc_941Hz = 0x7fff * fp_rc_941Hz ;

;rc_1209Hz = 0x7fff * fp_rc_1209Hz ;

;rc_1336Hz = 0x7fff * fp_rc_1336Hz ;

;rc_1477Hz = 0x7fff * fp_rc_1477Hz ;

;rc_1633Hz = 0x7fff * fp_rc_1633Hz ;

powquiet1 = 60

powquiet2 = 61

powquiet3 = 62

powquiet4 = 63

powquiet5 = 64

powquiet6 = 65

powquiet7 = 66

powquiet8 = 67

pow1 = 68

pow2 = 69

pow3 = 70

pow4 = 71

pow5 = 72

pow6 = 73

pow7 = 74

pow8 = 75

LACK 0

sac1 powquiet1

sac1 powquiet2

sac1 powquiet3

sac1 powquiet4

sac1 powquiet5

sac1 powquiet6

sac1 powquiet7

sac1 powquiet8

sac1 pow1

sac1 pow2

sac1 pow3

sac1 pow5

sac1 pow6

sac1 pow8

bss= 0x3

csa= 0x4

lack 0x07

sac1 bss

lack 0xff

sac1 csa

ZALH bss

OR csa

sac1 pow4

sac1 pow7

larp ar0

lark ar0,0x40

loop:

lt X ;

mpy X

pac

lt Z

mpy two

apac ;loading X^2+2Z into accumualtor

sub A

; $Y = X^2 + (2 * Z) - 1$

SACL Y;

LAC X;

ADD A;

SACL X; X=X+1

LAC Y;

SACL Z; Z= Y

banz loop,*-,ar0; ; use auxiliar register to loop

spectrum:

```
quiet: lark    ar1,rcc_len          ; length of rcc register block
        lark    ar0,(rcc_697Hz+base_page1) ; starting address
        sar          ar0,rcc_ptr
        lark    ar0,(powquiet1+base_page0)
        sar          ar0,ptr
quiet_l: lar    ar0,ptr
        zals     *+,
        sar          ar0,ptr
        lar    ar0,rcc_ptr
        sacl     *+,ar1
        sar          ar0,rcc_ptr
        banz     quiet_l,*-,ar0    ; make quiet tone
```

```
quiet2: lark    ar1,rcc_len          ; length of rcc register block
        lark    ar0,(rcc_697Hz+base_page1) ; starting address
        sar          ar0,rcc_ptr
        lark    ar0,(powquiet1+base_page0)
```



```

        banz    poundsign_1,*-,ar0    ; make quiet tone

poundsign2:  lark    ar1,rcc_len        ; length of rcc register block

        lark    ar0,(rcc_697Hz+base_page1)    ; starting address
        sar                    ar0,rcc_ptr
        lark    ar0,(pow1+base_page0)
        sar                    ar0,ptr

poundsign_2:  lar                    ar0,ptr
        zals    *+
        sar                    ar0,ptr
        lar    ar0,rcc_ptr
        sacl    *+,ar1
        sar                    ar0,rcc_ptr
        banz    poundsign_2,*-,ar0    ; make #spectrum twice

        lark ar0,waitTime

wait1:

        banz wait1,*-,ar0    ;wait for 30 cycles

quiet3:  lark    ar1,rcc_len        ; length of rcc register block

        lark    ar0,(rcc_697Hz+base_page1)    ; starting address
        sar                    ar0,rcc_ptr
        lark    ar0,(powquiet1+base_page0)
        sar                    ar0,ptr

```

```

quiet_3:    lar        ar0,ptr

            zals      *+,
            sar        ar0,ptr

            lar    ar0,rcc_ptr

            sac1      *+,ar1

            sar        ar0,rcc_ptr

            banz     quiet_3,*-,ar0    ; make quiet tone

```

```

quiet4: lark    ar1,rcc_len            ; length of rcc register block

            lark    ar0,(rcc_697Hz+base_page1)    ; starting address

            sar        ar0,rcc_ptr

            lark    ar0,(powquiet1+base_page0)

            sar        ar0,ptr

```

```

quiet_4:    lar        ar0,ptr

            zals      *+,
            sar        ar0,ptr

            lar    ar0,rcc_ptr

            sac1      *+,ar1

            sar        ar0,rcc_ptr

            banz     quiet_4,*-,ar0    ; make quiet tone twice

```

```

stop:      b stop

```

C Assembly Language Test Program Merged Listing

tdspasm, RCS v1.1.1.1

Listing for module: "zxw6805_test"

Prepared: Mon Dec 8 17:43:18 EST 2014

0000 7e01	LACK	1
0001 5006	SACL	X
0002 7e00	LACK	0
0003 5007	SACL	Y
0004 7e00	LACK	0
0005 5008	SACL	Z
0006 7e01	LACK	1
0007 5009	SACL	A
0008 7e00	LACK	0
0009 503c	sac1	powquiet1
000a 503d	sac1	powquiet2
000b 503e	sac1	powquiet3
000c 503f	sac1	powquiet4
000d 5040	sac1	powquiet5
000e 5041	sac1	powquiet6
000f 5042	sac1	powquiet7
0010 5043	sac1	powquiet8
0011 5044	sac1	pow1
0012 5045	sac1	pow2
0013 5046	sac1	pow3
0014 5048	sac1	pow5
0015 5049	sac1	pow6
0016 504b	sac1	pow8
0017 7e07	lack	0x07
0018 5003	sac1	bss
0019 7eff	lack	0xff
001a 5004	sac1	csa
001b 6503	ZALH	bss
001c 7a04	OR	csa
001d 5047	sac1	pow4
001e 504a	sac1	pow7
001f 6880	larp	ar0
0020 7040	lark	ar0,0x40
0021	loop:	
0021 4006	lt	X
0022 6d06	mpy	X
0023 7f8e	pac	
0024 4008	lt	Z
0025 6d02	mpy	two
0026 7f8f	apac	
0027 1009	sub	A
0028 5007	SACL	Y
0029 2006	LAC	X
002a 0009	ADD	A
002b 5006	SACL	X
002c 2007	LAC	Y
002d 5008	SACL	Z
002e f490	banz	loop,*-,ar0
002f 0021		

```

0030      spectrum:
0030 7108      quiet: lark      ar1,rcc_len
0031 70e0      lark      ar0,(rcc_697Hz+base_page1)
0032 3016      sar      ar0,rcc_ptr
0033 703c      lark      ar0,(powquiet1+base_page0)
0034 3015      sar      ar0,ptr
0035 3815      quiet_l: lar      ar0,ptr
0036 66a8      zals      *+
0037 3015      sar      ar0,ptr
0038 3816      lar      ar0,rcc_ptr
0039 50a1      sacl      *+,ar1
003a 3016      sar      ar0,rcc_ptr
003b f490      banz      quiet_l,*-,ar0
003c 0035
003d 7108      quiet2: lark      ar1,rcc_len
003e 70e0      lark      ar0,(rcc_697Hz+base_page1)
003f 3016      sar      ar0,rcc_ptr
0040 703c      lark      ar0,(powquiet1+base_page0)
0041 3015      sar      ar0,ptr
0042 3815      quiet_2: lar      ar0,ptr
0043 66a8      zals      *+
0044 3015      sar      ar0,ptr
0045 3816      lar      ar0,rcc_ptr
0046 50a1      sacl      *+,ar1
0047 3016      sar      ar0,rcc_ptr
0048 f490      banz      quiet_2,*-,ar0
0049 0042
004a 701e      lark      ar0,waitTime
004b f490      wait0: banz      wait0,*-,ar0
004c 004b
004d 7108      poundsign1: lark      ar1,rcc_len
004e 70e0      lark      ar0,(rcc_697Hz+base_page1)
004f 3016      sar      ar0,rcc_ptr
0050 7044      lark      ar0,(pow1+base_page0)
0051 3015      sar      ar0,ptr
0052 3815      poundsign_1: lar      ar0,ptr
0053 66a8      zals      *+
0054 3015      sar      ar0,ptr
0055 3816      lar      ar0,rcc_ptr
0056 50a1      sacl      *+,ar1
0057 3016      sar      ar0,rcc_ptr
0058 f490      banz      poundsign_1,*-,ar0
0059 0052
005a 7108      poundsign2: lark      ar1,rcc_len
005b 70e0      lark      ar0,(rcc_697Hz+base_page1)
005c 3016      sar      ar0,rcc_ptr
005d 7044      lark      ar0,(pow1+base_page0)
005e 3015      sar      ar0,ptr
005f 3815      poundsign_2: lar      ar0,ptr

```

```

0060 66a8      zals      *+
0061 3015      sar       ar0,ptr
0062 3816      lar       ar0,rcc_ptr
0063 50a1      sac1      *+,ar1
0064 3016      sar       ar0,rcc_ptr
0065 f490      banz      poundsign_2,*-,ar0
0066 005f
0067 701e      lark      ar0,waitTime
0068          wait1:
0068 f490      banz      wait1,*-,ar0
0069 0068
006a 7108      quiet3: lark  ar1,rcc_len
006b 70e0      lark      ar0,(rcc_697Hz+base_page1)
006c 3016      sar       ar0,rcc_ptr
006d 703c      lark      ar0,(powquiet1+base_page0)
006e 3015      sar       ar0,ptr
006f 3815      quiet_3: lar  ar0,ptr
0070 66a8      zals      *+
0071 3015      sar       ar0,ptr
0072 3816      lar       ar0,rcc_ptr
0073 50a1      sac1      *+,ar1
0074 3016      sar       ar0,rcc_ptr
0075 f490      banz      quiet_3,*-,ar0
0076 006f
0077 7108      quiet4: lark  ar1,rcc_len
0078 70e0      lark      ar0,(rcc_697Hz+base_page1)
0079 3016      sar       ar0,rcc_ptr
007a 703c      lark      ar0,(powquiet1+base_page0)
007b 3015      sar       ar0,ptr
007c 3815      quiet_4: lar  ar0,ptr
007d 66a8      zals      *+
007e 3015      sar       ar0,ptr
007f 3816      lar       ar0,rcc_ptr
0080 50a1      sac1      *+,ar1
0081 3016      sar       ar0,rcc_ptr
0082 f490      banz      quiet_4,*-,ar0
0083 007c
0084 f500      stop: b      stop
0085 0084

```

D. Assembly Language Test Program Symbol Table

tdspasm, RCS v1.1.1.1

Symbol listing for module: "zxw6805_test"

Prepared: Mon Dec 8 17:43:18 EST 2014

* <symbol> = <hex> (<octal>) (<decimal>) <R,A>

* where: R == relocatable, A == absolute

```
A = 0x0009 (0000011) ( 9) A
AR0 = 0x0000 (0000000) ( 0) A -- predefined symbol
AR1 = 0x0001 (0000001) ( 1) A -- predefined symbol
PA0 = 0x0000 (0000000) ( 0) A -- predefined symbol
PA1 = 0x0001 (0000001) ( 1) A -- predefined symbol
PA2 = 0x0002 (0000002) ( 2) A -- predefined symbol
PA3 = 0x0003 (0000003) ( 3) A -- predefined symbol
PA4 = 0x0004 (0000004) ( 4) A -- predefined symbol
PA5 = 0x0005 (0000005) ( 5) A -- predefined symbol
PA6 = 0x0006 (0000006) ( 6) A -- predefined symbol
PA7 = 0x0007 (0000007) ( 7) A -- predefined symbol
X = 0x0006 (0000006) ( 6) A
Y = 0x0007 (0000007) ( 7) A
Z = 0x0008 (0000010) ( 8) A
ar0 = 0x0000 (0000000) ( 0) A -- predefined symbol
ar1 = 0x0001 (0000001) ( 1) A -- predefined symbol
base_page0 = 0x0000 (0000000) ( 0) A
base_page1 = 0x0080 (0000200) ( 128) A
bss = 0x0003 (0000003) ( 3) A
csa = 0x0004 (0000004) ( 4) A
loop = 0x0021 (0000041) ( 33) R
pa0 = 0x0000 (0000000) ( 0) A -- predefined symbol
pa1 = 0x0001 (0000001) ( 1) A -- predefined symbol
pa2 = 0x0002 (0000002) ( 2) A -- predefined symbol
pa3 = 0x0003 (0000003) ( 3) A -- predefined symbol
pa4 = 0x0004 (0000004) ( 4) A -- predefined symbol
pa5 = 0x0005 (0000005) ( 5) A -- predefined symbol
pa6 = 0x0006 (0000006) ( 6) A -- predefined symbol
pa7 = 0x0007 (0000007) ( 7) A -- predefined symbol
page0 = 0x0000 (0000000) ( 0) A
poundsign1 = 0x004d (0000115) ( 77) R
poundsign2 = 0x005a (0000132) ( 90) R
poundsign_1 = 0x0052 (0000122) ( 82) R
poundsign_2 = 0x005f (0000137) ( 95) R
pow1 = 0x0044 (0000104) ( 68) A
pow2 = 0x0045 (0000105) ( 69) A
pow3 = 0x0046 (0000106) ( 70) A
pow4 = 0x0047 (0000107) ( 71) A
pow5 = 0x0048 (0000110) ( 72) A
pow6 = 0x0049 (0000111) ( 73) A
pow7 = 0x004a (0000112) ( 74) A
pow8 = 0x004b (0000113) ( 75) A
powquiet1 = 0x003c (0000074) ( 60) A
powquiet2 = 0x003d (0000075) ( 61) A
powquiet3 = 0x003e (0000076) ( 62) A
powquiet4 = 0x003f (0000077) ( 63) A
powquiet5 = 0x0040 (0000100) ( 64) A
powquiet6 = 0x0041 (0000101) ( 65) A
```

```

powquiet7 = 0x0042 (0000102) ( 66) A
powquiet8 = 0x0043 (0000103) ( 67) A
  ptr = 0x0015 (0000025) ( 21) A
  quiet = 0x0030 (0000060) ( 48) R
  quiet2 = 0x003d (0000075) ( 61) R
  quiet3 = 0x006a (0000152) ( 106) R
  quiet4 = 0x0077 (0000167) ( 119) R
  quiet_2 = 0x0042 (0000102) ( 66) R
  quiet_3 = 0x006f (0000157) ( 111) R
  quiet_4 = 0x007c (0000174) ( 124) R
  quiet_l = 0x0035 (0000065) ( 53) R
rcc_1209Hz = 0x0064 (0000144) ( 100) A
rcc_1336Hz = 0x0065 (0000145) ( 101) A
rcc_1477Hz = 0x0066 (0000146) ( 102) A
rcc_1633Hz = 0x0067 (0000147) ( 103) A
rcc_697Hz = 0x0060 (0000140) ( 96) A
rcc_770Hz = 0x0061 (0000141) ( 97) A
rcc_852Hz = 0x0062 (0000142) ( 98) A
rcc_941Hz = 0x0063 (0000143) ( 99) A
rcc_kick = 0x0068 (0000150) ( 104) A
rcc_len = 0x0008 (0000010) ( 8) A
rcc_ptr = 0x0016 (0000026) ( 22) A
spectrum = 0x0030 (0000060) ( 48) R
  stop = 0x0084 (0000204) ( 132) R
  two = 0x0002 (0000002) ( 2) A
  wait0 = 0x004b (0000113) ( 75) R
  wait1 = 0x0068 (0000150) ( 104) R
  waitTime = 0x001e (0000036) ( 30) A

```

E. Assembly Language Test Program Object File

```

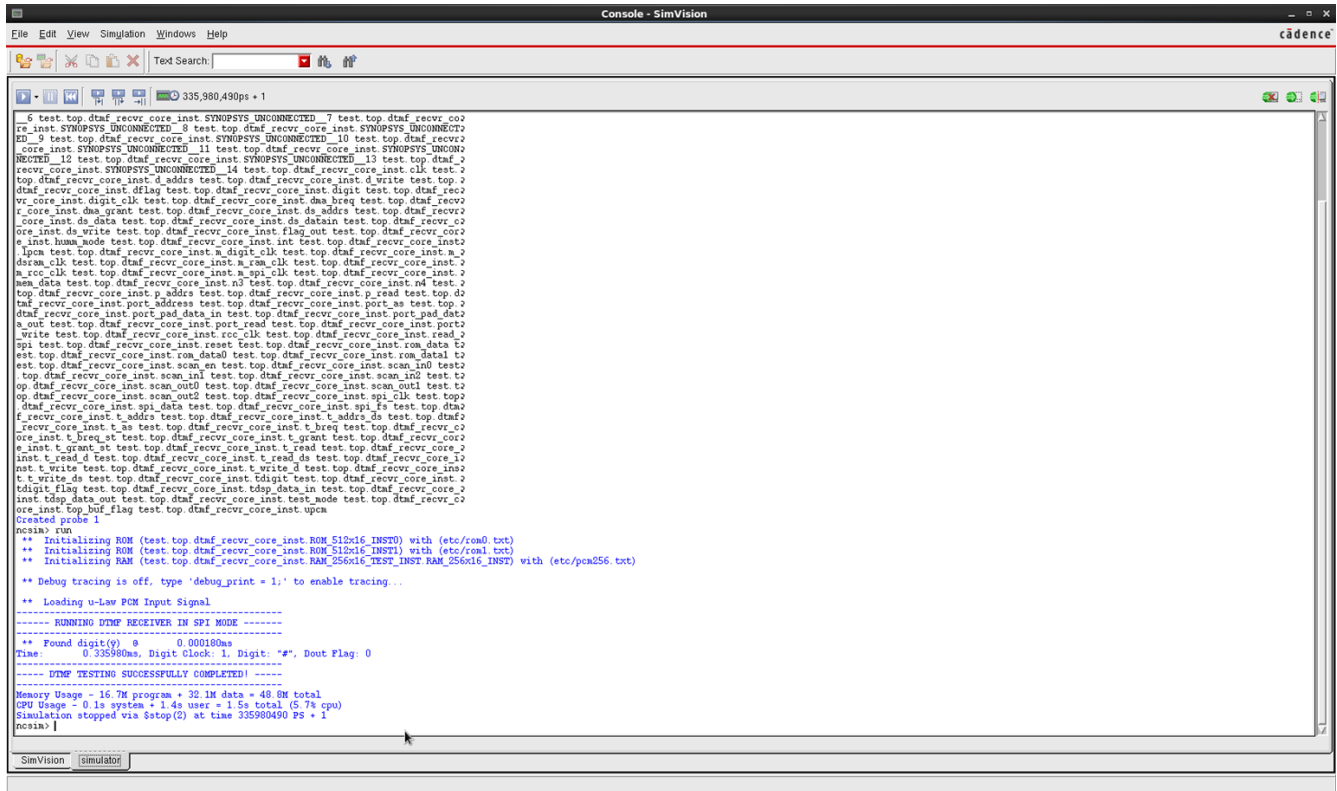
// tdpasm, RCS v1.1.1.1
// Object for module: "zvw6805 test"
// Prepared: Mon Dec 8 17:43:18 EST 201
//*****
0000 7e01
0001 5006
0002 7e00
0003 5007
0004 7e00
0005 5008
0006 7e01
0007 5009
0008 7e00
0009 503c
000a 503d
000b 503e
000c 503f
000d 5040
000e 5041
000f 5042
0010 5043
0011 5044
0012 5045
0013 5046
0014 5048
0015 5049
0016 504b
0017 7e07
0018 5003
0019 7eff
001a 5004
001b 6503
001c 7a04
001d 5047
001e 504a
001f 6880
0020 7040
0021 4006
0022 6d06
0023 7f8e
0024 4008
0025 6d02
0026 7f8f
0027 1009
0028 5007
0029 2006
002a 0009
002b 5006

```

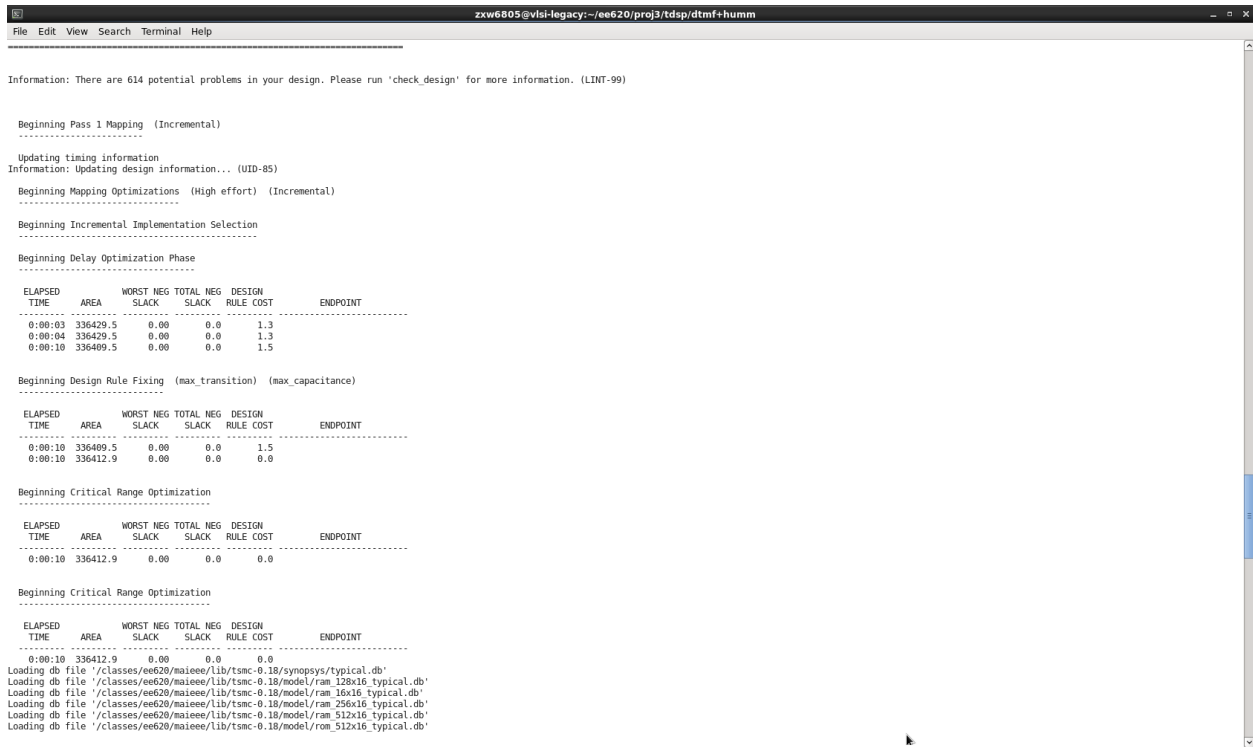
```
obj.txt (-~/Desktop) - GVIEW
File Edit Tools Syntax Buffers Window Help
[Icons]
@0032 3016
@0033 703c
@0034 3015
@0035 3815
@0036 66a8
@0037 3015
@0038 3816
@0039 50a1
@003a 3016
@003b f490
@003c 0035
@003d 7108
@003e 70e0
@003f 3016
@0040 703c
@0041 3015
@0042 3815
@0043 66a8
@0044 3015
@0045 3816
@0046 50a1
@0047 3016
@0048 f490
@0049 0042
@004a 701e
@004b f490
@004c 004b
@004d 7108
@004e 70e0
@004f 3016
@0050 7044
@0051 3015
@0052 3815
@0053 66a8
@0054 3015
@0055 3816
@0056 50a1
@0057 3016
@0058 f490
@0059 0052
@005a 7108
@005b 70e0
@005c 3016
@005d 7044
@005e 3015
@005f 3815
@0060 66a8
@0061 3015
55,10 60%
```

```
obj.txt (-~/Desktop) - GVIEW
File Edit Tools Syntax Buffers Window Help
[Icons]
@005c 3016
@005d 7044
@005e 3015
@005f 3815
@0060 66a8
@0061 3016
@0062 3816
@0063 50a1
@0064 3016
@0065 f490
@0066 005f
@0067 701e
@0068 f490
@0069 0068
@006a 7108
@006b 70e1
@006c 3016
@006d 703c
@006e 3015
@006f 3815
@0070 66a8
@0071 3015
@0072 3816
@0073 50a1
@0074 3016
@0075 f490
@0076 006f
@0077 7108
@0078 70e0
@0079 3016
@007a 703c
@007b 3015
@007c 3815
@007d 66a8
@007e 3015
@007f 3816
@0080 50a1
@0081 3016
@0082 f490
@0083 007c
@0084 f500
@0085 0084
~
~
~
~
~
-- VISUAL --
102,10 Bot
```

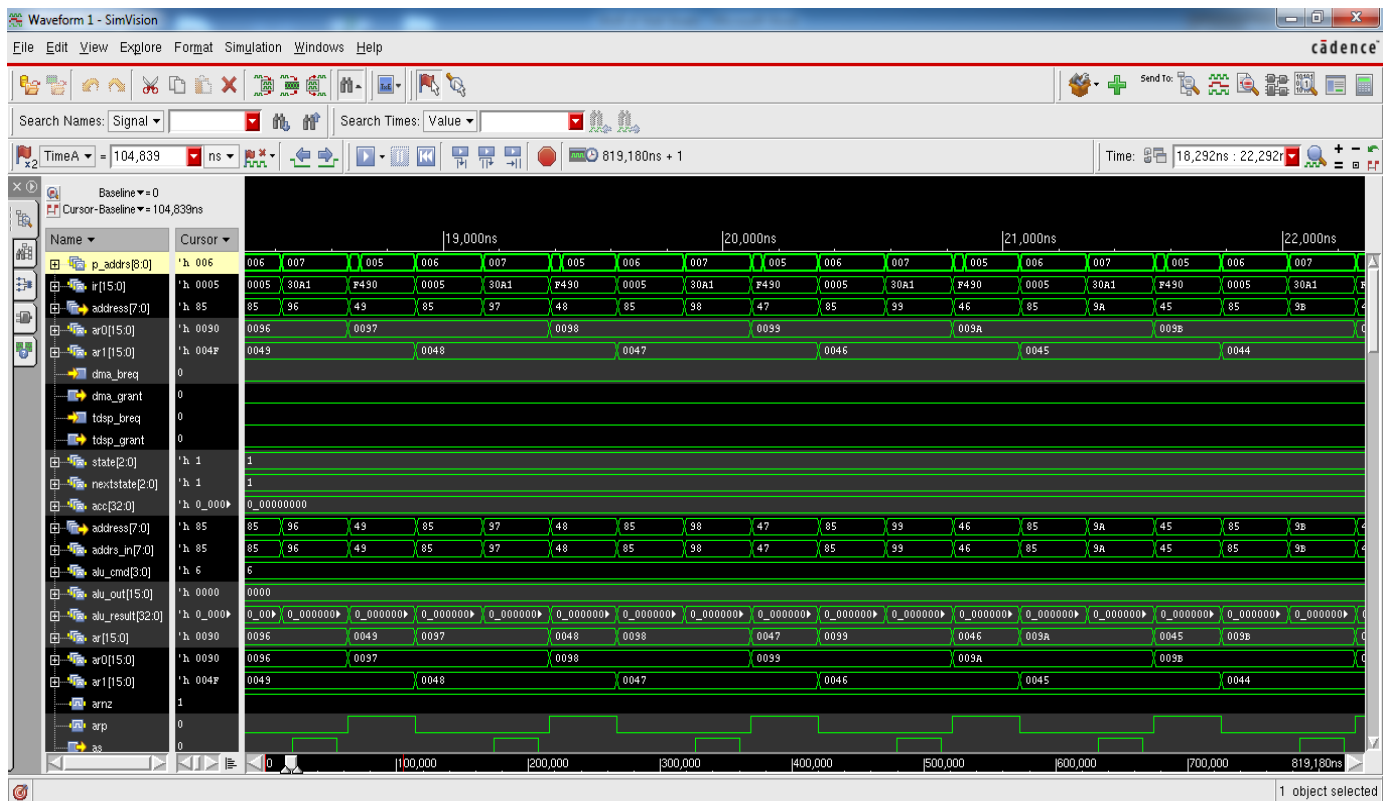
F Simulator Log files

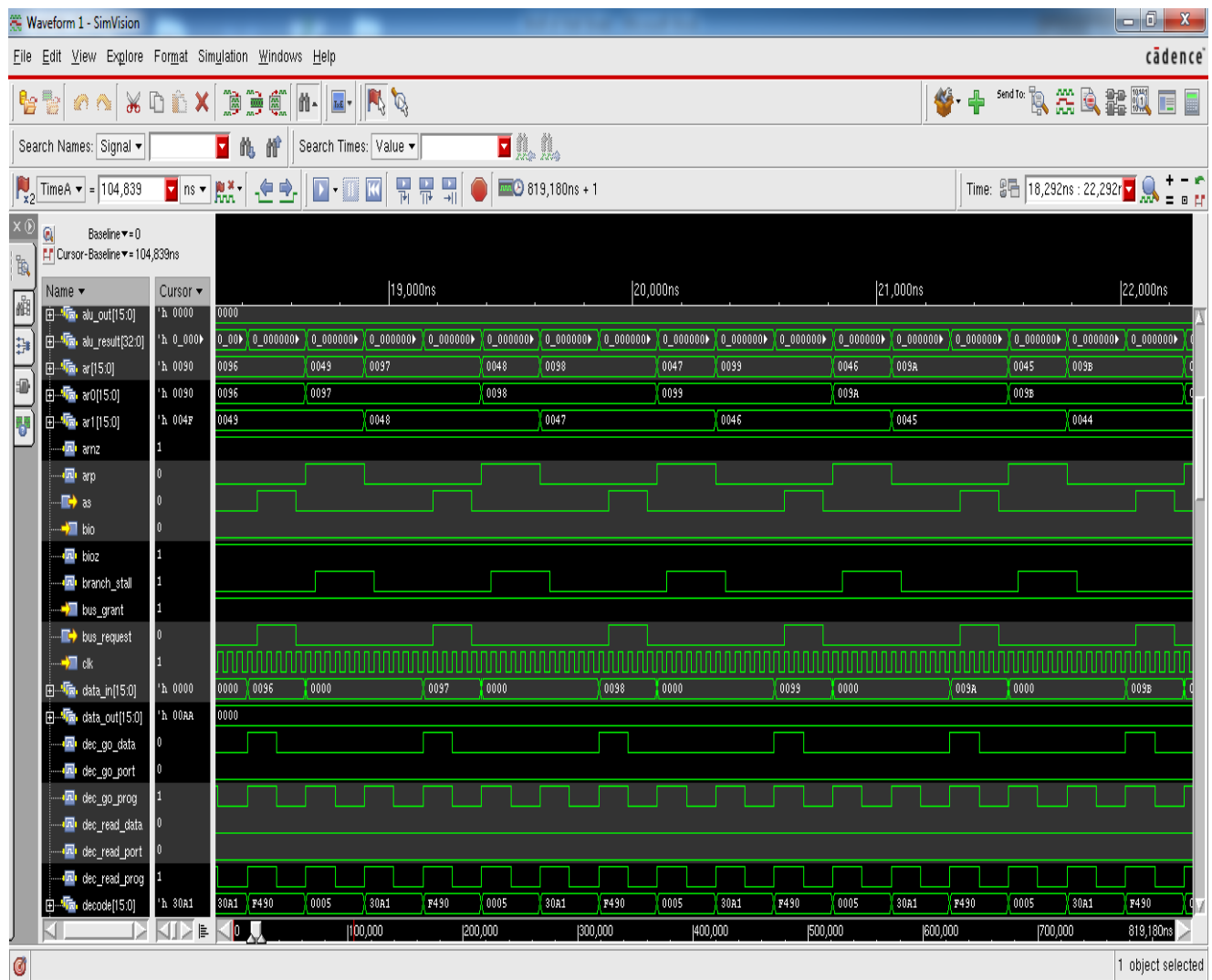


```
6 test.top.dtmf_recvr_core_inst.SYNOPSIS_UNCONNECTED_7 test.top.dtmf_recvr_core_inst.SYNOPSIS_UNCONNECTED_8 test.top.dtmf_recvr_core_inst.SYNOPSIS_UNCONNECTED_9 test.top.dtmf_recvr_core_inst.SYNOPSIS_UNCONNECTED_10 test.top.dtmf_recvr_core_inst.SYNOPSIS_UNCONNECTED_11 test.top.dtmf_recvr_core_inst.SYNOPSIS_UNCONNECTED_12 test.top.dtmf_recvr_core_inst.SYNOPSIS_UNCONNECTED_13 test.top.dtmf_recvr_core_inst.SYNOPSIS_UNCONNECTED_14 test.top.dtmf_recvr_core_inst.clk test.top.dtmf_recvr_core_inst.d_addr test.top.dtmf_recvr_core_inst.d_write test.top.dtmf_recvr_core_inst.dflag test.top.dtmf_recvr_core_inst.digit test.top.dtmf_recvr_core_inst.digit_clk test.top.dtmf_recvr_core_inst.dma_breq test.top.dtmf_recvr_core_inst.dma_grant test.top.dtmf_recvr_core_inst.ds_addr test.top.dtmf_recvr_core_inst.ds_data test.top.dtmf_recvr_core_inst.ds_datain test.top.dtmf_recvr_core_inst.ds_write test.top.dtmf_recvr_core_inst.flag_out test.top.dtmf_recvr_core_inst.huma_mode test.top.dtmf_recvr_core_inst.inst.inst test.top.dtmf_recvr_core_inst.lpcn test.top.dtmf_recvr_core_inst.m_digit_clk test.top.dtmf_recvr_core_inst.m_dram_clk test.top.dtmf_recvr_core_inst.m_ram_clk test.top.dtmf_recvr_core_inst.m_data test.top.dtmf_recvr_core_inst.m3 test.top.dtmf_recvr_core_inst.m4 test.top.dtmf_recvr_core_inst.p_addr test.top.dtmf_recvr_core_inst.p_read test.top.dtmf_recvr_core_inst.port_addr test.top.dtmf_recvr_core_inst.port_as test.top.dtmf_recvr_core_inst.port_pad_data_in test.top.dtmf_recvr_core_inst.port_pad_data_out test.top.dtmf_recvr_core_inst.port_read test.top.dtmf_recvr_core_inst.port_write test.top.dtmf_recvr_core_inst.rst_clk test.top.dtmf_recvr_core_inst.read_spi test.top.dtmf_recvr_core_inst.reset test.top.dtmf_recvr_core_inst.rom_data_in test.top.dtmf_recvr_core_inst.rom_data0 test.top.dtmf_recvr_core_inst.rom_data1 test.top.dtmf_recvr_core_inst.scan_en test.top.dtmf_recvr_core_inst.scan_in0 test.top.dtmf_recvr_core_inst.scan_in1 test.top.dtmf_recvr_core_inst.scan_in2 test.top.dtmf_recvr_core_inst.scan_out0 test.top.dtmf_recvr_core_inst.scan_out1 test.top.dtmf_recvr_core_inst.scan_out2 test.top.dtmf_recvr_core_inst.spi_clk test.top.dtmf_recvr_core_inst.spi_data test.top.dtmf_recvr_core_inst.spi_fs test.top.dtmf_recvr_core_inst.t_addr test.top.dtmf_recvr_core_inst.t_addr_ds test.top.dtmf_recvr_core_inst.t_as test.top.dtmf_recvr_core_inst.t_breq test.top.dtmf_recvr_core_inst.t_breq_st test.top.dtmf_recvr_core_inst.t_grant test.top.dtmf_recvr_core_inst.t_read test.top.dtmf_recvr_core_inst.t_read_ds test.top.dtmf_recvr_core_inst.t_read_spi test.top.dtmf_recvr_core_inst.t_write test.top.dtmf_recvr_core_inst.t_write_d test.top.dtmf_recvr_core_inst.t_write_ds test.top.dtmf_recvr_core_inst.tdigit test.top.dtmf_recvr_core_inst.tdigit_flag test.top.dtmf_recvr_core_inst.tdsp_data_in test.top.dtmf_recvr_core_inst.tdsp_data_out test.top.dtmf_recvr_core_inst.test_mode test.top.dtmf_recvr_core_inst.top_buf_flag test.top.dtmf_recvr_core_inst.upn
Created probe 1
ncsim> run
** Initializing ROM (test.top.dtmf_recvr_core_inst.ROM_512x16_INST0) with (etc/rom0.txt)
** Initializing ROM (test.top.dtmf_recvr_core_inst.ROM_512x16_INST1) with (etc/rom1.txt)
** Initializing RAM (test.top.dtmf_recvr_core_inst.RAM_256x16_TEST_INST_RAM_256x16_INST) with (etc/pcn256.txt)
** Debug tracing is off, type 'debug_print = 1;' to enable tracing...
** Loading u-Law PCM Input Signal
----- RUNNING DTMF RECEIVER IN SPI MODE -----
** Found digit(y) 0 0.000180ms
Time 0.335980ms, Digit Clock: 1, Digit: "*", Dout Flag: 0
----- DTMF TESTING SUCCESSFULLY COMPLETED! -----
Memory Usage - 16.7M program + 32.1M data = 48.8M total
CPU Usage - 0.1s system + 1.4s user = 1.5s total (5.7% cpu)
Simulation stopped via $stop($?) at time 33590490 ps + 1
ncsim>
```



G screen shots of gate level simulation waveforms showing program execution





5 summary and conclusions

Summary

Overall, the finite state machine Verilog implementation is not difficult since it is simple and straight forward implementation except a few minor issues with reset and always@ code. The test bench takes time to implement since I thought about the output TDSP_GRANT and DMA_GRANT should be checked for being 1 or 0 when they are supposed to be instead of just checking them being 1 within the correct clock period cycle. One combinational and two sequential registers are created in FSM Verilog program.

The Verilog file is copied into assembly file location for simulating TDSP assembly language test program. The assembly TDSP was created and used twenty years ago. Thus, it is hard to find the example code online. DTMF_TDSP.asm is used as reference for coding. After figuring out how to add and multiply number using registers and accumulators, how to make a loop using auxiliary register and how to make spectrum with minimum time delay, the assembly program is finished with # spectrum being displayed in simulation console.

Conclusion

The project overall is about understanding rather than writing codes since most of the reference materials are available for use. Finite State Machine is commonly used in digital world. The Verilog code implementation for finite state machine is easy to find online. The test bench logic is used for making sure that TDSP_GRANT and DMA_GRANT occurring within the 1 correct clock cycle instead of them being 1 wrongly 1000 clock cycles after. The assembly language test program coding is not a lot with DTMF_TDSP assembly language reference being provided. The understanding of dual tone multiple frequency register implementation and timing delay is necessary to finish the assembly language test program.