avsddac.v (DAC Module) The dac module converts the 10-bit digital signal from the rvmyth core to an analog output.
[avsddac](https://github.com/vsdip/rvmyth_avsddac_interface.git)

Inputs:

- D: A 10-bit digital input from the processor.

- VREFH: Reference voltage for the DAC.

Output:

- OUT: Analog output signal.

Testbench

The testbench.v file is a test module to verify the functionality of vsdbabysoc. It includes signal initialization, clock generation, and waveform dumping for both pre-synthesis and post-synthesis simulations. Waveform Output:

- pre_synth_sim.vcd or post_synth_sim.vcd files generated based on simulation conditions.

Simulation Steps

Pre-Synthesis Simulation

Run the following command to perform a pre-synthesis simulation:

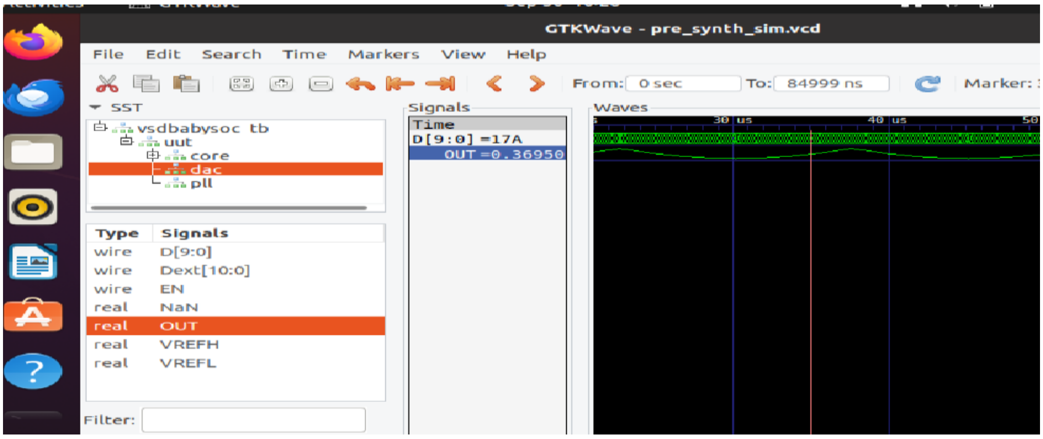
```
iverilog -o output/pre_synth_sim/pre_synth_sim.out -DPRE_SYNTH_SIM \

  -I src/include -I src/module \

  src/module/testbench.v src/module/vsdbabysoc.v

cd output/pre_synth_sim

./pre_synth_sim.out
```



From: 0 sec To: 84999 ns

Type	Signals
wire	D[9:0]
wire	Dext[10:0]
wire	EN
real	NaN
real	OUT
real	VREFH
real	VREFL

```
Time
D[9:0] =17A
    OUT =0.36950
```

```
module avsddac (
  OUT,
  D,
  VREFH,
  VREFL
);

  output     OUT;
  input [9:0] D;
  input      VREFH;
  input      VREFL;


  reg  real OUT;
  wire real VREFL;
  wire real VREFH;

  real NaN;
  wire EN;

  wire [10:0] Dext;      // unsigned extended

  assign Dext = {1'b0, D};
  assign EN = 1;

  initial begin
    NaN = 0.0 / 0.0;
    if (EN == 1'b0) begin
      OUT <= 0.0;
    end
    else if (VREFH == NaN) begin
      OUT <= NaN;
    end
    else if (VREFL == NaN) begin
      OUT <= NaN;
    end
    else if (EN == 1'b1) begin
      OUT <= VREFL + ($itor(Dext) / 1023.0) * (VREFH - VREFL);
    end
    else begin
      OUT <= NaN;
    end
  end

  always @(D or EN or VREFH or VREFL) begin
    if (EN == 1'b0) begin
      OUT <= 0.0;
    end
    else if (VREFH == NaN) begin
      OUT <= NaN;
    end
    else if (VREFL == NaN) begin
      OUT <= NaN;
    end
    else if (EN == 1'b1) begin
      OUT <= VREFL + ($itor(Dext) / 1023.0) * (VREFH - VREFL);
    end
    else begin
      OUT <= NaN;
    end
  end
endmodule
```