Laboratory 2 Report
Verilog-A

Lukas Krampitz
1192170

2026–06–07

Date performed:	2025–09–30
Course identifier:	ENGG*4080
Course section:	0101
Course title:	Micro and Nano-Scale Electronics
Instructor:	Professor Stefano Gregori

1 Introduction

Modern day electronic systems and integrated circuits (IC) rely heavily on hardware descriptive languages (HDL). It would be an immense understatement to claim designing complex a IC with a million or more transistors at the schematic level as impractical. Luckily with HDLs like Verilog, and extensions that support analogue behaviour like Verilog-A this task becomes realistic. This laboratory activity will include the critical component definition and entry at the behavioural level. It will also revist schematic layout and how to build up circuits hierarchically. The end goal will be to design and implement a 4-bit DAC and ADC.

2 Analog to Digital Converter

2.1 Entry

The design for the 4-bit ADC was entered into Virtuoso using Verilog-A. This 4-bit ADC is a successive approximation converter as described in the text for ENGG*4080. The resulting symbol for the 4-bit ADC is seen in Fig. 1. All the component values can be read off the symbol or Verilog-A source code. Please note the documentation and justification of the model can be read as in-line comments in the Verilog-A source and such comments begin with "D&J".

```
1  // VerilogA for engg4080_lab2, adc_succ, veriloga
2  //
3  // 4-bit Analog to Digital Converter (ADC)
4
5  `include "constants.vams" // Default import
6  `include "disciplines.vams" // Default import
7
8  module adc_succ (out, in, clk, vref, vdd);
9    parameter real td = 0; // Sets the output transition delay
10   parameter real tt = 0; // Sets the output transition time (rise or fall
       )
11
12   input in, clk, vref, vdd;
13   output [0:3] out;
14
15   voltage   in, clk, vref, vdd;
16   voltage [0:3] out;
17
18   integer i;
19   real sample, fullscale, supply;
20   integer result[0:4]; //D&J: 5-bit internal result, 1st bit is the sign,
         this gets discarded.
21
22   analog begin
23     fullscale = V(vref);
24     supply = V(vdd);
25
```

```
26      @(cross(V(clk) - V(vdd) / 2, +1) or initial_step) begin //D&J: Run on
            every rising edge of clk
27        sample = V(in); //D&J: Read VIN directly to scratch pad
28
29        for (i = 0; i < 5; i = i + 1) begin //D&J: increment i to match
             chart, hard code 5 bits
30          if (sample > 0) begin //D&J: Checks for over/undershoot
31            result[i] = 1; //D&J: Anything above 0, is positive and needs
                 to contribute to digital result.
32            sample = sample - (fullscale / pow(2, i+1)); //D&J: As i
                 increases take smaller steps until LSB
33          end else begin
34            result[i] = 0; //D&J: Anything below 0, has too much undershoot
                 and the sample needs to come up again.
35            sample = sample + (fullscale / pow(2, i+1));
36          end
37        end
38      end
39
40      //D&J: Page 653 & 654 of the ENGG*4080 text is for signed input.
           Perform a shift to discard the sign bit
41      V(out[0]) <+ transition(result[1] * supply, td, tt);
42      V(out[1]) <+ transition(result[2] * supply, td, tt);
43      V(out[2]) <+ transition(result[3] * supply, td, tt);
44      V(out[3]) <+ transition(result[4] * supply, td, tt);
45    end
46  endmodule
```

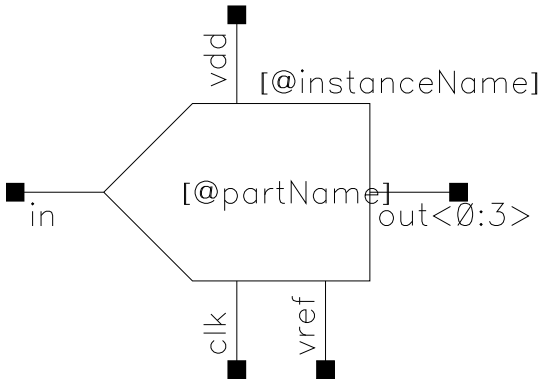

Figure 1: Symbol of the 4-bit successive approximation ADC

2.2 Test Bench

A new schematic was entered to act as a test bench for a simulation of the ADC. All the component values can be derived from the Verilog-A code of the ADC, or from the provided test bench in

the laboratory handout. Any other changes can be read of the schematic in Fig. 2 (eg. bus width increased).

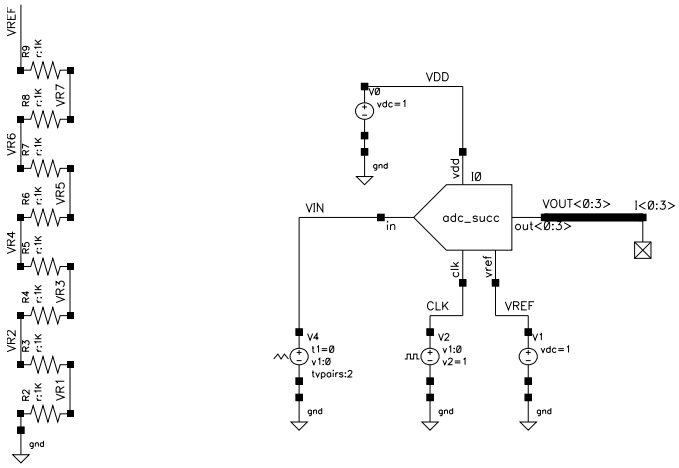

Figure 2: Schematic view of test bench for the ADC

2.3 Simulation

The final plot of the ADC simulation can be seen in Fig. 3. In the final waveform one can see all of the $2^4 = 16$ output combinations for the ADC on the input-output transfer curve. For any possible analogue input the ADC responds with a corresponding digital output value. This result can be seen to hit all three thresholds of the reference voltages (VR2, VR4, VR6). In the simulation VREF and VDD supply voltage was kept a 1V so the ADC handles the input accordingly to the main reference voltage. This simulation used transient analysis with a fixed time of $17\mu s$ for the TSMC 65nm process. All simulation was at ambient temperature with no corner cases.

3 Digital to Analog Converter

3.1 Entry

The design for the 4-bit DAC was entered into Virtuoso using Verilog-A. This 4-bit DAC is a modified verion of the 2-bit DAC used in the demonstration. The resulting symbol for the 4-bit DAC is seen in Fig. 4. All the component values can be read off the symbol or Verilog-A source code. Please note the justifications of the modifications can be read as in-line comments in the Verilog-A source.

```
1 // VerilogA for engg4080_lab2, dac_4bit, veriloga
2 //
3 // 4-bit Digital to Analog Converter (DAC)
4
5 `include "constants.vams" // Default import
6 `include "disciplines.vams" // Default import
```

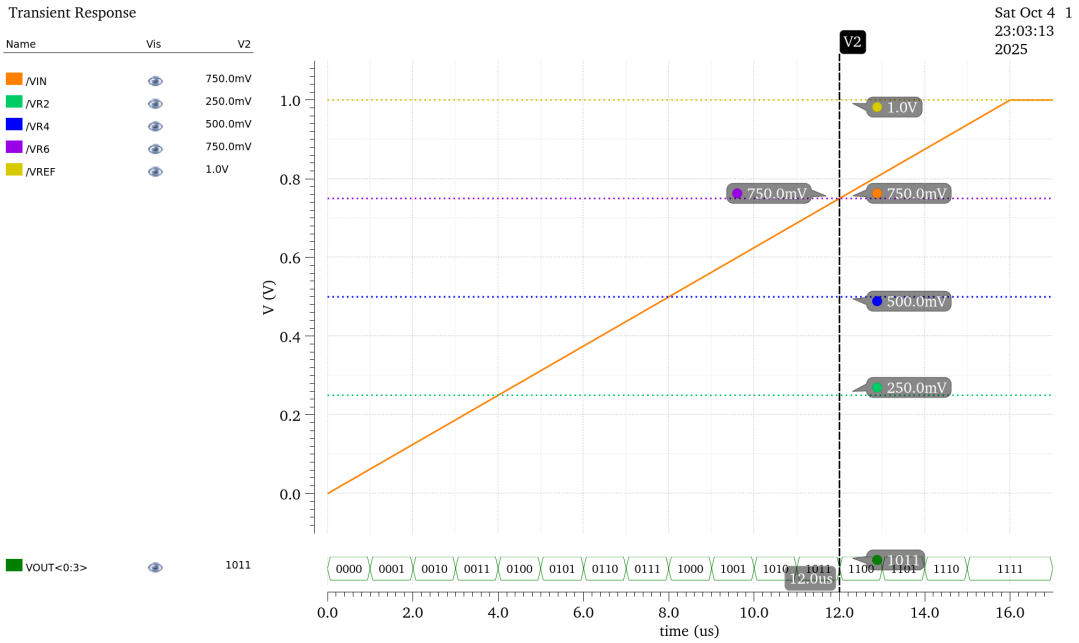

Figure 3: ADC simulation plot

```
7
8  module dac_4bit (out, in, clk, vref, vdd);
9    parameter real td = 0; // Sets the output transition delay
10   parameter real tt = 0; // Sets the output transition time (rise or fall
       )
11
12   input [0:3] in; // MOD: Need four bits of input
13   input clk, vref, vdd;
14   output out;
15
16   voltage [0:3] in; // MOD: Need four bits of input
17   voltage   out, clk, vref, vdd;
18
19   real result, fullscale, supply;
20   integer bit[0:3]; // MOD: Need four bits of temp bit storage
21   integer i;
22
23   analog begin
24     fullscale = V(vref);
25     supply = V(vdd);
26
27     @(cross(V(clk) - supply / 2, +1) or initial_step) begin
28       result = 0;
29
30       bit[0] = floor(V(in[0]) / supply + 0.5);
31       bit[1] = floor(V(in[1]) / supply + 0.5);
32       bit[2] = floor(V(in[2]) / supply + 0.5); // MOD: Added two more
           noise
```

```
33        bit[3] = floor(V(in[3]) / supply + 0.5); // correction terms
34
35        for (i = 0; i < 4; i = i + 1) begin // MOD: Make loop iterate four
              times
36          result = result + fullscale / 16 * pow(2, i) * bit[i]; //MOD:
                Divide fullscale by 16 since input is 4-bit
37        end
38      end
39
40      V(out) <+ transition(result, td, tt);
41    end
42  endmodule
```

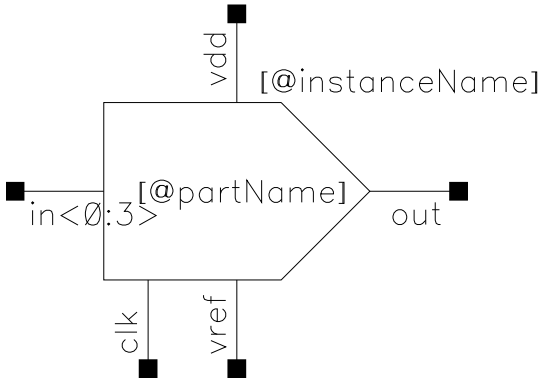

Figure 4: Symbol of the 4-bit DAC

3.2 Test Bench

Another new schematic was entered to act as a test bench for a simulation of the DAC. All the component values can be derived from the Verilog-A code of the DAC, or from the provided test bench in the laboratory handout. Any other changes can be read of the schematic in Fig. 5 (eg. bus width increased).

3.3 Simulation

The final plot of the DAC simulation can be seen in Fig. 6. In the final waveform one can see the $2^4 = 16$ output combinations for the DAC on the input-output transfer curve. For each of the 16 possible 4-bit binary inputs the DAC responds with a distinct analogue output voltage. The step size of the output is uniform and results in a consisted stair-step pattern. This result can be seen to hit all three thresholds of the reference voltages (VR2, VR4, VR6). In the simulation VREF and VDD supply voltage was kept a 1V so the DAC scales the output accordingly to the main reference voltage. This simulation used transient analysis with a fixed time of $400\mu s$ for the TSMC 65nm process. All simulation was at ambient temperature with no corner cases.

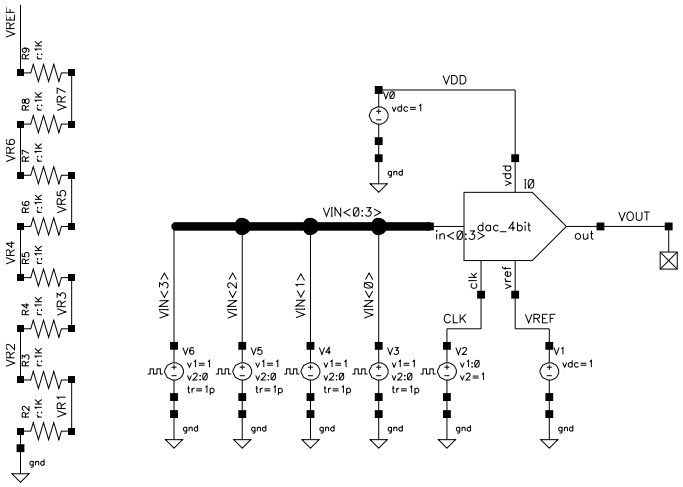

Figure 5: Schematic view of test bench for the DAC

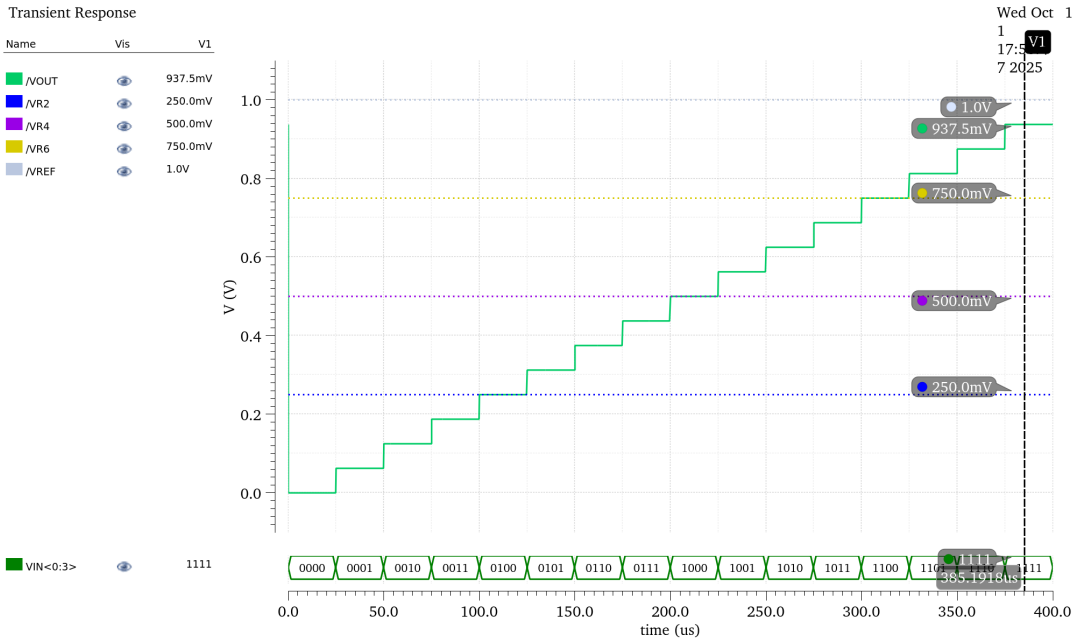

Figure 6: DAC simulation plot

4 Observations and Conclusions

4.1 Parting Thoughts

This laboratory activity again helped build a stronger understanding of the Cadence Virtuoso design tools and provided insight into the subject matter of Verilog-A and behavioural entry. The obtained results were able to be explained based on the theory, and align with the expected behaviour of ADCs and DACs. It will be important to remember the fundamental principals of Verilog-A, as this will be a critical skill for laboratory activities and future courses like VLSI design. Getting to use Verilog-A was very enjoyable, and this laboritory activity prooved to be quite fun.