Effectiveness of the Verilog-A Noise Macromodel of Current Feedback Operational Amplifier

Georgi Valkov[1] and Elissaveta Gadjeva[2]

Abstract – **In the present paper a detailed Verilog-A macromodel taking into account the noise effects is developed for the LM6181 current feedback low noise operational amplifier. The efficiency of the model is assessed and compared to the corresponding Spice library model.**

Keywords – **Effectiveness, Verilog-A, Noise Macromodel, Current Feedback Operational Amplifier.**

I. Introduction

The high-level Analog Hardware Description Language (AHDL) Verilog-A describes the structure and behavior of analog systems and their components using modules. Verilog-A combines structural modeling with language based behavioral modeling. An important feature of the language is that it includes coupled algebraic-integro-differential equations, rich set of analog operators, if-then and case statements, as well as implicit and explicit equations [1]. Methodologies and uses of analog behavioral modeling with the Verilog-A language are developed in [2]. Simplified Verilog-A models of operational amplifiers are proposed in [2-4]. A number of research academic and industrial groups create compact models using Verilog-A, which are used to simulate electronic devices and circuits correctly and efficiently. In [5] recommendations are provided for writing and optimization of the compact models in Verilog-A. As a result, the language Verilog-A can be characterized as an extremely efficient language for writing compact models. Recently, the AHDLs are widely used in the areas of analog and RF circuit simulation. The capabilities and limitations of compact models of analog RF circuits are discussed in [6]. Correlated noise models are developed in [7] using Verilog-A language. They are used in the simulation of the noise behavior of MOSFET using correlated noise sources with complex correlation coefficients.

There are many opamp macromodels described in the Verilog-A and VHDL-AMS languages [2-4], but most of them are too simplified. In addition to that, the noise behavior of the actual devices, which is very important for models of high precision low-noise opamps, is not considered by those models [2-6].

In the present paper a detailed macromodel is developed for the LM6181 current feedback operational amplifier, using the Verilog-A language. The model includes noise effects. The model efficiency is investigated and compared to the corresponding PSpice library model.

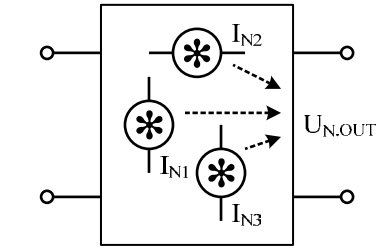
[1]Georgi Valkov is with the Department of Electronics at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, e-mail: gvalkov@abv.bg.

[2]Elissaveta Gadjeva is with the Department of Electronics at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, e-mail: egadjeva@tu-sofia.bg.

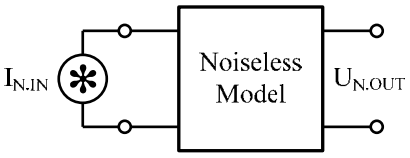
II. Noise Modelling of the Operational Amplifiers

When modeling noisy two-port devices (Fig. 1a), such as operational amplifiers, the output noise is referred to the input port of the device. As a result, a noiseless model is created, and finally a noise block is added to the input (Fig. 1b).

For a noisy operational amplifier (Fig. 2a), the noise block consists of two current noise sources I_{N1} and I_{N2}, connected between the corresponding input and the reference node, and a voltage noise source in series with one of the inputs, as shown in Fig. 2b. The signals from these noise sources are frequency dependent and can be obtained from the datasheet of the op-amp.



a)



b)

Fig. 1. Two-port device with noise (a) and equivalent representation as a noiseless model and referred noise to the input (b)

III. Noise Implementation in the Spice Model

The noise sources I_{N1}, I_{N2} and E_N (Fig. 2b) are realized in the computer realization of the *Spice* macromodel [12] using auxiliary circuits shown in Fig. 3, that create the noise signals for the corresponding noise sources.

The resistor R_{32} creates the white noise current for I_{N2}. The resistor R_{33} creates the white noise current for I_{N1}. The voltage between nodes n_{59} and n_{61} defines E_N. It includes flicker and shot noise from the diodes D_9 and D_{10}, as well as white noise from the resistors R_{30} and R_{31}.

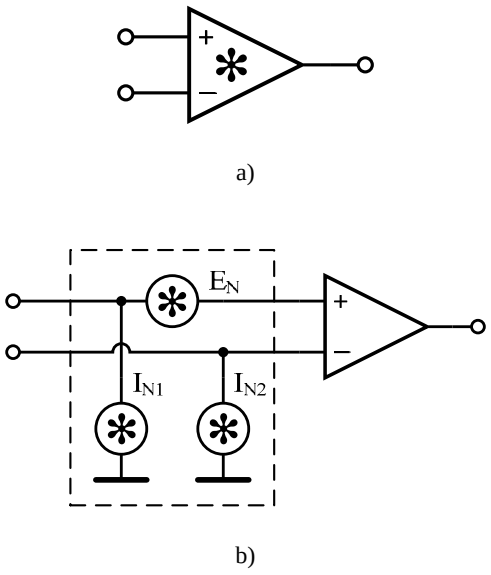


Fig. 2. Symbol of a noisy op-amp (a) and equivalent representation as a noise block added to the inputs of a noiseless op-amp model (b)

The symmetric implementation cancels out the DC component from V_{15} and V_{16}, which are used to set the required operating point of the two diodes.

Taking into account that the Modified Nodal Approach (MNA) is applied in the circuit simulators, the auxiliary circuit in Fig. 3 increases the matrix order of the op-amp model by 10 – six additional nodes (n_{58}, n_{59}, n_{60}, n_{61}, n_{62} and n_{63}) and four additional equations describing the independent voltage sources V_{15}, V_{16}, V_{17} and V_{18}.

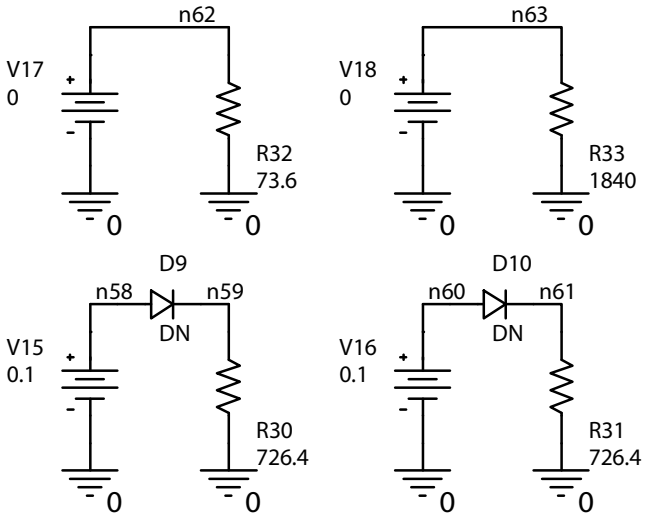


Fig. 3. Noise signals

IV. Noise Modelling in Verilog-A

In Verilog-A the noise is implemented using the *white_noise*, *flicker_noise*, and *noise_table* functions [10]. The parameter of the *white_noise* function specifies the noise level. The first parameter of the *flicker_noise* function is the noise level at 1*Hz*. The second parameter is the frequency exponent. The parameter passed to the *noise_table* function defines the noise levels for a number of frequency points, using the following form: *'{f1, v1, f2, v2…, fn, vn}*. For some simulators, a file name can be specified as the parameter for *noise_table*. Each line of the file should contain a pair of frequency point and noise level. All noise levels described above are in $\frac{V^2}{Hz}$. A sample block of noise voltage source has the following form:

```
module v_noise(n1, n2);
inout n1, n2;
electrical n1, n2;
parameter real NT[0:5]
                  = '{1k,2.5m,10k,0.4m,100k,0.9m};
parameter real Ewn = 1  from (0:inf);
parameter real Efn = 1  from (0:inf);
parameter real AF  = 1  from (0:inf);
analog begin
    V(n1, n2)   <+ white_noise(Ewn*Ewn)
                 + flicker_noise(Efn*Efn, AF*2)
                 + noise_table(NT);
end endmodule
```

An advantage of Verilog-A is that the noise functions can be integrated as a part of the model equations. No auxiliary circuits are needed and the matrix order is not affected.

The Verilog-A implementation of LM6181 current feedback op-amp macromodel with noise is shown in Fig. 4.

V. Effectiveness Assessment

The noise models based on behavioral modeling using Verilog-A are characterized with higher efficiency, due to the fact that they do not increase the order of the circuit matrix. The auxiliary block that generates the noise signals, shown in Fig. 3, which are required by the Spice macromodel, increases the model matrix order *n* by 10 and reduces the efficiency of the model.

The relative speed increasing using the Verilog-A model, compared to the *Spice* model has the form:

$$\varepsilon_C = 100\,\frac{t_S - t_V}{t_S}\ [\%], \qquad (1)$$

where t_S is the simulation time, when using the library *Spice* model for LM6181 and t_V is the simulation time for the Verilog-A model from Fig. 4.

The dependence of ε_C on the matrix order *n* is shown in Fig. 5. The model accuracy of the Verilog-A implementation is the same as for the *Spice* model.

In order to assess the effectiveness of the two models depending on the matrix order, the following circuits are simulated: a single non-inverting amplifier with a matrix order *n*=71 for the *Spice* model; a circuit with three amplifiers (*n*=199); a circuit with five amplifiers (*n*=327) and a circuit with seven amplifiers (*n*=455).

```
// LM6181 current feedback Op-Amp macromodel with noise
//               in+  in-   v+   v-  out
module LM6181_ns(n01, n02, n99, n50, n41);
inout       n01, n02, n99, n50, n41;
electrical  n01, n02, n99, n50, n41;
electrical  n03, n04, n05, n06, n07, n08, n09;
electrical  n10, n11, n12, n13, n15, n16, n17;
electrical  n20, n21, n22, n23;
electrical  n30, n31, n33, n34, n35, n36, n37, n38;
electrical  n40, n44, n45, n46, n47, n49;
electrical  n55, n56, n57, n98, gnd; ground gnd;
branch(n99, n05) GI1, FI1;      branch(n04, n50) GI2, FI2;
branch(n05, n06) GR1;           branch(n06, n99) C1;
branch(n04, n07) GR2;           branch(n07, n50) C2;
branch(n99, n08) GR3;           branch(n99, n10) V1;
branch(n10, n30) RE1;           branch(n50, n09) GR4;
branch(n11, n50) V2;            branch(n11, n31) RE2;
branch(n12, n05) va3;           branch(n04, n13) va4;
branch(n01, n99) GR6, GB1;      branch(n01, n50) GR7;
branch(n01, gnd) FN1, CIN1;     branch(n99, n02) GB2;
branch(n02, gnd) FN2, CIN2;     branch(n03, n01) EOS;
branch(n99, n50) in;            branch(n56, gnd) n56__gnd;
branch(n01, n49) n01__n49;      branch(n55, gnd) n55__gnd;
branch(n45, gnd) n45__gnd;      branch(n47, gnd) n47__gnd;
branch(n57, gnd) n57__gnd;
branch(n99, n50) I3;            branch(n99, n49) R8;
branch(n49, n50) R9;            branch(n99, n16) V3;
branch(n17, n50) V4;            branch(n99, n98) EH;
branch(n98, n15) G1, R5, C3;
branch(n98, n20) G2, R14, C4;   branch(n15, n49) n15__n49;
branch(n98, n21) G3, R15, C5;   branch(n20, n49) n20__n49;
branch(n98, n22) G4, R16, C6;   branch(n21, n49) n21__n49;
branch(n98, n23) G5, R17, C7;   branch(n22, n49) n22__n49;
branch(gnd, n45) G10;           branch(n44, n45) L3;
branch(n44, gnd) R25;           branch(gnd, n47) G11;
branch(n46, n47) L4;            branch(n46, gnd) R26;
branch(n99, gnd) n99__gnd;      branch(n50, gnd) n50__gnd;
branch(gnd, n55) I12, R27;      branch(gnd, n56) I13, R28;
branch(gnd, n57) I14, R29;
branch(n99, n50) F6;            branch(n99, n35) F5;
branch(n99, n36) va7;           branch(n99, n37) E1;
branch(n37, n38) va8;           branch(n38, n40) R35;
branch(n33, n40) V5;            branch(n40, n34) V6;
branch(n41, n02) CF1;           branch(n40, n41) L5, RL5;
branch(n99, n23) n99__n23;
// temperature coeffitient scaling factor
parameter real Tnom=$simparam("tnom")+273.15 from (0:inf);
analog function real Tc;
input TC1, TC2; real TC1, TC2; begin
Tc = 1 + ($temperature-Tnom)*(TC1+($temperature-Tnom)*TC2);
end endfunction
// input stage
analog begin
I(GI1)  <+ 243.75u + V(in)*2.708u;
I(GI2)  <+ 243.75u + V(in)*2.708u;
I(FI2)  <+ I(va4)*100;          I(GR1)  <+ V(GR1)*238u;
I(C1)   <+ ddt(V(C1))*468f;     I(GR2)  <+ V(GR2)*238u;
I(C2)   <+ ddt(V(C2))*468f;     I(GR3)  <+ V(GR3)*1.58m;
V(V1)   <+ 0.3;                 I(RE1)  <+ V(RE1)/130;
I(GR4)  <+ V(GR4)*1.58m;        I(FI1)  <+ I(va3)*100;
V(V2)   <+ 0.3;                 I(RE2)  <+ V(RE2)/150
V(va3)  <+ 0;                   V(va4)  <+ 0;
I(GR6)  <+ V(GR6)*50n;          I(GR7)  <+ V(GR7)*50n;
I(GB1)  <+ -1.2u + V(in)*40n + V(n56__gnd)*1m;
I(FN1)  <+ white_noise((3.00143e-12)*(3.00143e-12));
I(CIN1) <+ ddt(V(CIN1))*2p;
I(GB2)  <+ 18.5u-V(in)*150n-V(n01__n49)*100n-
V(n55__gnd)*1u;
I(FN2)  <+ white_noise((1.50071e-11)*(1.50071e-11));
I(CIN2) <+ ddt(V(CIN2))*5.75p;
V(EOS)  <+ -2.8m + V(in)*93u + V(n45__gnd) + V(n47__gnd)
+V(n57__gnd) + flicker_noise((2.86835e-8 )*(2.86835e-8), 1)
            + white_noise  ((3.97682e-13)*(3.97682e-13)
/* second stage */         + (4.90727e-9 )*(4.90727e-9));
I(I3)   <+ 4.47m;           V(EH)   <+ V(R8);
I(R8)   <+ V(R8)/7.19k;     V(V3)   <+ 1.7;
I(R9)   <+ V(R9)/7.19k;     V(V4)   <+ 2;
I(G1)   <+ V(GR3)*1.58m + V(GR4)*1.58m;
//  Fp1 = 27.96 KHz
I(R5)   <+ V(R5)/2.372e+6;  I(C3)   <+ ddt(V(C3))*2.4p;
I(G2)   <+ V(n15__n49)*1m;  // pole stage, Fp = 250 MHz
I(R14)  <+ V(R14)/1k;       I(C4)   <+ ddt(V(C4))*692f;
I(G3)   <+ V(n20__n49)*1m;  // pole stage, Fp = 250 MHz
I(R15)  <+ V(R15)/1k;       I(C5)   <+ ddt(V(C5))*692f;
I(G4)   <+ V(n21__n49)*1m;  // pole stage, Fp = 275 MHz
I(R16)  <+ V(R16)/1k;       I(C6)   <+ ddt(V(C6))*578.7f;
I(G5)   <+ V(n22__n49)*1m;  // pole stage, Fp = 500 MHz
I(R17)  <+ V(R17)/1k;       I(C7)   <+ ddt(V(C7))*318.3f;
// PSRR stage
I(G10)  <+ V(n99__gnd)*141.3u;
V(L3)   <+ ddt(I(L3))*26.53u;   I(R25)  <+ V(R25)/10;
I(G11)  <+ V(n50__gnd)*141.3u;
V(L4)   <+ ddt(I(L4))*2.27364u; I(R26)  <+ V(R26)/10;
// thermal effects
I(I12)  <+ 1;   I(R27)  <+ V(R27)/(10*Tc(3.453m, 79.3u));
I(I13)  <+ 1m;  I(R28)  <+ V(R28)/(1.5*Tc(930.3u, 80.75u));
I(I14)  <+ 1m;  I(R29)  <+ V(R29)/(3.34*Tc(3.111m, 0));
// output stage
I(F6)   <+ I(va7);              V(va7)  <+ 0;
I(F5)   <+ I(va8);              V(va8)  <+ 0;
V(E1)   <+ V(n99__n23);         V(V5)   <+ 5.3;
I(R35)  <+ V(R35)/50;           V(V6)   <+ 5.3;
I(CF1)  <+ ddt(V(CF1))*2.1p;    V(L5)   <+ ddt(I(L5))*31n;
I(RL5)  <+ V(RL5)/100k;
end
// input stage
QPN Q1  (n50, n03, n05);        DX  D1  (n30, n08);
QNN Q2  (n99, n03, n04);        DX  D2  (n09, n31);
QNI Q3  (n08, n06, n02);        DY  DS1 (n03, n12);
QPI Q4  (n09, n07, n02);        DY  DS2 (n13, n03);
// second stage
DX  D3  (n15, n16);             DX  D4  (n17, n15);
// output stage
DX  D7  (n36, n35);             DX  D8  (n35, n99);
DX  D5  (n23, n33);             DX  D6  (n34, n23);
endmodule
```

Fig. 4. Verilog-A implementation of LM6181 current feedback op-amp macromodel with noise

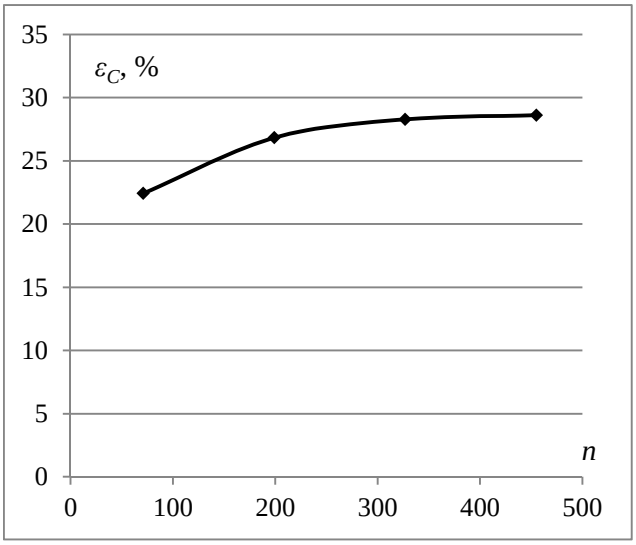


Fig. 5. Relative speed increase for noise simulation in the Verilog-A model, compared to the Spice model

The following simulation settings are used: AC interval [1*Hz*-1*GHz*], 10k *points/decade*, and Print interval for noise contribution table: every 5th point. All circuits are simulated in the environment of Dolphin SMASH [11].

The time measurements are performed by a custom plugin, which hooks the simulation start and end events, and reports the time difference. The plugin uses the RDTSC processor instruction and enables time measurements to be performed with a very-high precision. In addition to that, exclusive mode is enabled in the simulator, the process and thread priorities are set to real-time, and the affinities are locked to a single CPU core. This results in a very low maximum deviation time, less than 0.8% from the mean value. To further enhance the precision, 12 simulations are run in each case and the best 8 times are averaged to calculate the mean time.

ACKNOWLEDGEMENT

The investigations are supported by the project №122PD0026-03.

VI. CONCLUSION

A detailed computer macromodel of the current feedback operational amplifier has been developed using the Verilog-A language. The model includes noise effects. The model accuracy of the Verilog-A implementation is the same as for the *Spice* model. The corresponding code is given. The model efficiency is investigated and compared to the corresponding *Spice* library model.

REFERENCES

[1] I. Miller, T. Cassagnes, “Verilog-AMS Eases Mixed Mode Signal Simulation”, Technical Proceedings of the 2000 International Conference on Modeling and Simulation of Microsystems, pp. 305-308.
[2] Dan FitzPatrick, Ira Miller, “Analog Behavioral Modeling with the Verilog-A Language”, Kluwer Academic Publishers, 2003.
[3] http://www.eda.org/twiki/bin/view.cgi/VerilogAMS
[4] http://w2.cadence.com/appnotes/CDN_Creating_Analog_Behavioral_Models.pdf
[5] R. Frevert, et al. “Modeling and Simulation for RF System Design”, pp. 219-246, ISBN 10 0-387-27585-1, Springer, 2005.
[6] Huabiao Qin, Fei Wang, “Modeling of Operational Amplifier based on VHDL-AMS”, pp. 894-897, IEEE, 2006.
[7] Coram, G.J., “How to (and how not to) write a compact model in Verilog-A”, Proceedings of the IEEE International Behavioral Modeling and Simulation Conference (BMAS 2004), 21-22 Oct. 2004, pp. 97 – 106.
[8] Troyanovsky, B., O'Halloran, P., Mierzwinski, M., “Analog RF model development with Verilog-A”, Radio Frequency integrated Circuits (RFIC) Symposium, 2005, Digest of Papers, 2005 IEEE, 12-14 June 2005, pp. 287-290.
[9] C. McAndrew, G. Coram, A. Blaum, and O. Pilloud, “Correlated noise modeling and simulation,” in Proc. Tech. Workshop Compact Modeling, Anaheim, CA, May 2005, pp. 40-45.
[10] “Verilog-AMS Language Reference Manual”, version 2.3.1, Accellera, June 1, 2009.
[11] Dolphin Integration SMASH overview, www.dolphin.fr/medal/smash/smash_overview.php.
[12] “Development of an Extensive SPICE Macromodel for Current-Feedback Amplifiers”, Texas Instruments, May, 2004, http://www.ti.com/lit/an/snoa247a/snoa247a.pdf