VHDL-AMS Description of Digitally Programmable Gain Amplifiers through SPI

Marieta Kovacheva[1] and Ivailo Pandiev[2]

Abstract – **In this paper a behavioural VHDL-AMS models for digitally programmable gain amplifiers (PGAs) are presented. For creating the models, simplification and build-up techniques known from macromodelling of operational amplifiers have been adapted. The models accurately reflect input impedance, transfer function (amplifier gain in binary steps versus controlling digital code through SPI), small-signal frequency response, large-signal pulse response, output characteristic (voltage and current limitations) and output resistance. Model parameters are extracted for the one channel PGA MCP6S21 and the two channels PGA MCP6S22 from Microchip. The behaviours of the equivalent circuits are created following their structures and operation principle. The modelling of the PGA behaviour is implemented and confirms to the format of the simulation program System Vision 5.5 (from Mentor Graphics). The simulation results show accurate agreement with the theoretical predictions.**

Keywords – **Mixed-signal circuits, Programmable gain amplifiers, SPI, Behavioural models, VHDL-AMS, Mixed-signal simulation.**

I. INTRODUCTION

The programmable gain amplifiers (PGAs) are electronic amplifiers (typically an operational amplifiers), which gain can be controlled by external digital or analogue signals. The gain can be set from less than 1V/V to over 100V/V. They have analogue input and output. For the most PGAs the external controlling digital signals are applied to the specific address inputs by using SPI or I²C standard. Typical applications for the PGAs are mixed-signal processing systems, test equipment and medical instrumentation.

After analyse of the existing model libraries in OrCAD PSpice A/D [1] and SystemVision (from Mentor Graphics) [2] some conclusions are made. In the System Vision libraries, a PGA behavioural model can be found for AD526 [3]. But in this model the external digital signal is parallel passed, nevertheless this model reflects all basic characteristics and parameters of the PSpice model. In the OrCAD PSpice A/D libraries some PGA models can be found, one of them is a macro model of AD526 [4]. In this model two basic modes of operation are reflected – transparent and latch mode. The PSpice macro model is not compatible to VHDL-AMS simulators. In these models is not reflected serial transmission of data (SPI or I^{2}C), which is typical for modern PGA. The existing models with a suitable choice of parameters and elements can be used of a mixed-signal circuit simulations, but not confirm to the architectures of a broad class of the monolithic PGAs. Transformation from OrCAD PSpice A/D to System Vision libraries can be done, but it's quite complicated, requires a lot of resources and additional processing. Since there is no behavioural models for PGAs with SPI data transfer and they are necessary for simulating mixed-signal circuits and systems. The goal of this paper therefore is to develop a behavioural VHDL-AMS model that accurately simulates the basic electrical characteristics of PGAs with synchronous serial data input and output transmission.

[1]Marieta Kovacheva is with the Faculty of Electronics, TU-Sofia, Kl. Ohridski 8, 1000 Sofia, Bulgaria, E-mail: m_kovacheva@tu-sofia.bg.

[2]Ivailo Pandiev is with the Faculty of Electronics, TU-Sofia, Kl. Ohridski 8, 1000 Sofia, Bulgaria, E-mail: ipandiev@tu-sofia.bg.

II. MONOLITHIC PGAS

The monolithic one channel PGA MCP6S21 and the two channel PGA MCP6S22 [5] from Microchip are used as an examples for creating the behavioural models. In fact the digitally programmable ICs MCP6S21 and MCP6S22 are typical representatives of the programmable amplifiers over an SPI™ bus. Thus add gain control and input channel selection (for MCP6S22) to the embedded control system. These PGAs are optimized for high speed, low offset voltage and single-supply operation with rail-to-rail input and output capability. These specifications support single-supply applications needing flexible performance or multiple inputs. Fig. 1 summarizes the external view of MCP6S21. The input voltage is passed to v_{in} pin and the output voltage is obtained by v_{out} pin. These PGAs are configured in a non-inverting circuit with gains of 1, 2, 4, 5, 8, 10, 16 and 32V/V that can be digitally selected using signals, applied to pins SI , SCK and $\overline{CS}$. A daisy chain configuration is possible through SO pin.

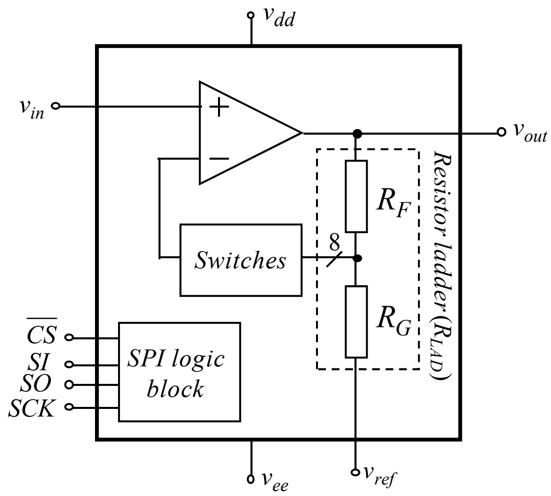

Fig. 1. Monolithic PGA MCP6S21 external view [5].

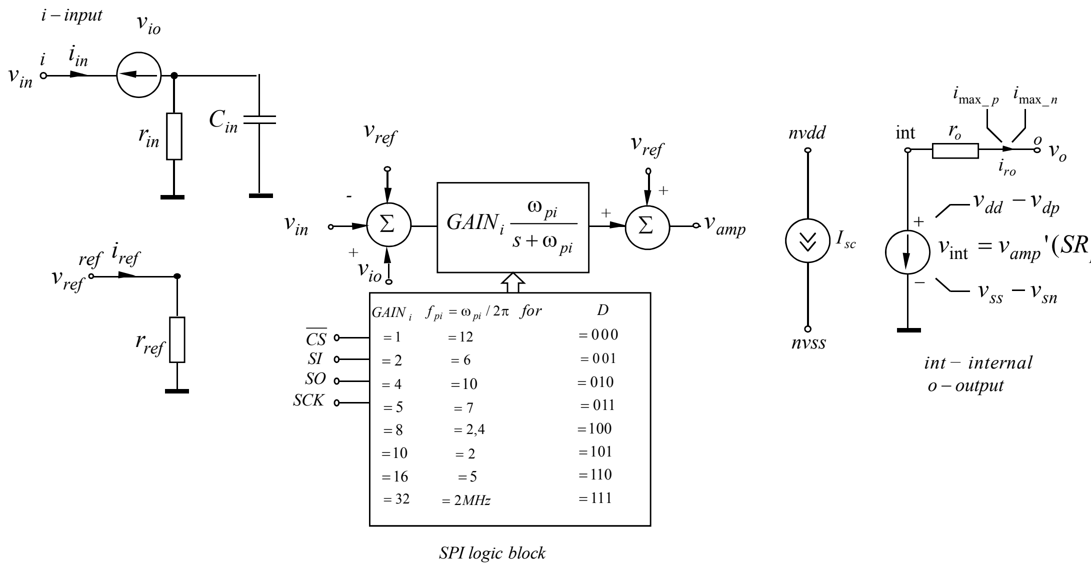

Fig. 2. Circuit diagram of a one channel PGA behavioural model.

These devices come with an internal register that allow user to select gains, channels and shutdown mode of operation. In general the internal structure of the MCP6S21 consists of one non-inverting amplifier, analogue switches with resistor ladder and SPI logic block.

The input signal v_{in} for the MCP6S21 is applied to a terminal *input* referred to ground and the input signals for the MCP6S22 are connected to the *CH0* or *CH1*. All input terminals are high-impedance CMOS with very low bias currents ($< 0{,}5nA$). For the two channel MCP6S22, the internal multiplexer 2x1 selects which one is amplified to the output. The output pin (*VOUT*) is a low-impedance ($< 1\Omega$) voltage source. The selected gain (*G*), selected input (*input / CH0* or *CH1*) and voltage at *VREF* determine its value. The SPI interface inputs are: Chip Select ($\overline{CS}$), Serial Input (SI) and Serial Clock (SCK). These are Schmitt triggered, CMOS logic inputs. These devices have a SPI interface serial output (SO) pin. This is a CMOS push-pull output and does not ever go High-Z. Once the device is deselected ($\overline{CS}$ goes high), SO is forced low. This feature supports daisy chaining configuration.

III. Behaviour Modelling with VHDL-AMS

The created behavioural model of digitally PGA is developed by using a style combining structural and mathematical description. The structural description is the netlist of the model and the behavioural description consists of simultaneous statements to describe the continuous behaviour. The behaviour of the proposed PGA is described using the structure given on Fig. 1.

A. A behavioural language: VHDL-AMS

VHDL-AMS is a comparatively new standard 1076.1 of VHDL that supports hierarchical description and simulation of analogue, digital and mixed-signal applications with conservative and non-conservative equations [6, 7]. On the mixed-signal side a variety of abstraction levels is supported. The VHDL-AMS modelling is not restricted to mixed-signal applications but also supports thermal and mechatronic systems.

B. A behavioural PGA VHDL-AMS models

The behavioural models of the PGAs are built using the results obtained by analyses of the ICs MCP6S21 and MCP6S22 [5]. The circuit diagram of a one channel PGA model is shown in Fig. 2, where the different stages are presented with structural and behavioural elements. The model includes the following elements and parameters with numerical values: $r_{in} = 100T\Omega$ and $C_{in} = 15pF$ – input resistance and capacitance; $r_{lad} = 4.9k\Omega$ – internal resistance; $V_{io} = 275\mu V$ – input offset voltage; $I_{io} = 250pA$ – input offset current; $I_{ib} = 1pA$ –input bias current; $I_{sc} = 1mA$ – dc supply current; $GAIN_i = 1$, 2, 4, 5, 8, 10, 16 and 32; $f_{pi} = \omega_{pi}/2\pi =$ 12, 6, 10, 7, 2,4, 2, 5 and $2MHz$ are the cut-off frequencies (at $-3dB$) for gains 1, 2, 4, 5, 8, 10, 16 and 32, respectively; $SR_{p1} = -SR_{n1} = 4V/\mu s$ – positive and negative slew rates at gains 1 and 2; $SR_{p2} = -SR_{n2} = 11V/\mu s$ – positive and negative slew rates at gains 4, 5, 8 and 10; $SR_{p3} = -SR_{n3} = 22V/\mu s$ – positive and negative slew rates at gains 16 and 32; v_{int} – output voltage-controlled voltage source; $v_{dp} = 30mV$ – positive and $v_{sn} = 20mV$ – negative voltage drops for the output voltage limitation; $i_{max\_p} = -i_{max\_n} = 30mA$ – maximum output currents; $r_{out} = 0{,}01\Omega$ – output resistance.

The proposed model includes small- and large- signal effects such as (1) accurate input impedance, (2) amplifier gain in binary steps versus controlling digital code, (3) SPI transmitting of the control data, (4) ac small-signal frequency responses, (5) slew rates, (6) dc supply current, (7) voltage and current limitations and (8) output resistance.

```
library IEEE; library ieee_proposed; use IEEE.std_logic_1164.all; use IEEE.std_logic_arith.all; use
IEEE.math_real.all; use ieee.std_logic_unsigned.all; use ieee_proposed.electrical_systems.all;
entity mcp6s21 is
   generic ( --  generic parameters here);
   port (  terminal input, ref, nvdd, nvss, output: electrical;
           signal  SI, CS, SCK : in  std_logic;
           signal  SO : out std_logic := '0');
end entity mcp6s21;
architecture default of mcp6s21 is
-- constant declaration here
terminal internal:electrical;
quantity vin across iin,icin,ii through input to electrical_ref;
quantity vref across iref through ref to electrical_ref;
quantity vdd across nvdd;
quantity vss across nvss;
quantity isc through nvdd to nvss;
quantity vout across output;
quantity vrout across irout through internal to output;
quantity vint across iintern,i_ib through internal;
quantity v0, vamp, v_io: voltage;
quantity irout_h : current;
signal sh_down:real:=0.0;
signal gain:real:=1.0;
signal SI_int : std_logic_vector(0 to 31) ;
signal SI_int3 : std_logic_vector(0 to 2);   -- output of comparators
signal SO_int : std_logic_vector(0 to 31):="00000000000000000000000000000000";
begin
v_io == vio;
ii == iio / 2.0;
i_ib == iib;
isc==supply_current;
iin==vin/rin;
iref==vref/rlad;
v0==vref;
icin==cin*vin'dot;
irout_h==vrout/rout;
if  gain=1.0 and sh_down=0.0 use
        vamp==vin'ltf(NUM1,DEN1) - v0'ltf(NUM1,DEN1)+v0+v_io'ltf(NUM1,DEN1);
 elsif gain=2.0 and sh_down=0.0 use
        vamp==vin'ltf(NUM2,DEN2)  - v0'ltf(NUM2,DEN2)+v0 +v_io'ltf(NUM2,DEN2);
 elsif gain=4.0 and sh_down=0.0 use
        vamp==vin'ltf(NUM4,DEN4)  - v0'ltf(NUM4,DEN4)+v0+v_io'ltf(NUM4,DEN4);
 elsif gain=5.0 and sh_down=0.0 use
        vamp==vin'ltf(NUM5,DEN5) - v0'ltf(NUM5,DEN5)+v0+v_io'ltf(NUM5,DEN5);
 elsif gain=8.0 and sh_down=0.0 use
        vamp==vin'ltf(NUM8,DEN8)  - v0'ltf(NUM8,DEN8)+v0+v_io'ltf(NUM8,DEN8);
 elsif gain=10.0 and sh_down=0.0 use
        vamp==vin'ltf(NUM10,DEN10)  - v0'ltf(NUM10,DEN10)+v0+v_io'ltf(NUM10,DEN10);
 elsif gain=16.0 and sh_down=0.0 use
        vamp==vin'ltf(NUM16,DEN16) - v0'ltf(NUM16,DEN16)+v0+v_io'ltf(NUM16,DEN16);
 elsif gain=32.0 and sh_down=0.0 use
        vamp==vin'ltf(NUM32,DEN32) - v0'ltf(NUM32,DEN32)+v0+v_io'ltf(NUM32,DEN32);
 else
      vamp==v0;
end use;
--limitation of output voltage
if vamp'above(vdd-vdp) use
        vint==vdd-vdp;
   elsif not vamp'above(vss+vsn) use
        vint==vss+vsn;
else
--slew rate at diff gain values
if gain=16.0 or gain=32.0 use
   vint==vamp'slew(SRp3, SRn3);
elsif gain=4.0 or gain=5.0 or gain=8.0 or gain=10.0  use
        vint==vamp'slew(SRp2,SRn2);
else
        vint==vamp'slew(SRp1,SRn1);
end use;
end use;
--limitation of output current
   if irout_h'above(imax_p) use
        irout==imax_p;
   elsif not irout_h'above(imax_n) use
        irout==imax_n;
   else
        irout==irout_h;
   end use;
control_proc : process is
procedure gain_change (signal SI_int1 : std_logic_vector(0 to 2);
signal s:out real
 ) is
   begin
     case SI_int1 (0 to 2) is
   when b"111" =>  s <=32.0;
   when b"001" => s <=2.0;
   when b"010" => s <=4.0;
   when b"011" => s <=5.0;
   when b"100" => s <=8.0;
   when b"101" => s <=10.0;
   when b"110" => s <=16.0;
   when others => s <=1.0;
  end case;
end procedure gain_change;
variable index, I, i1 : integer := 0;
begin
if ( SCK = '1' and SCK'event ) then
      SI_int(index) <= SI;
      index := index+1;
     SI_int3(0)<=  SI_int(29) ;
     SI_int3(1)<=  SI_int(30) ;
     SI_int3(2)<=  SI_int(31) ;
   SO_int(0 to 15) <="0000000000000000" ;
   SO_int(16 to 31)<=SI_int(0 to 15);
    if index = 32 then
       index := 0;
    end if;
end if;
if ( SI_int(16)= '0' and SI_int(17)= '0' and SI_int(18)= '1' ) then
sh_down <= 1.0;
elsif ( SI_int(16)= '0' and SI_int(17)= '1' and SI_int(18)= '0' ) then
sh_down <= 0.0;
else
sh_down <= 0.0;
end if;
--counter 32 bits
if ( CS = '0') then
      gain_change(SI_int3,gain);
   else
      wait until  ( CS = '0');
      gain_change(SI_int3,gain);
end if;
 if ( SCK = '0') then
 SO <= SO_int (i1);
 i1:=i1+1;
 if i1=32 then
 i1:=0;
 end if;
 end if;
   wait on  SCK, SI, CS, gain;
end process ;
end architecture default;
```

Fig. 3. A one channel PGA behavioural VHDL-AMS model.

Fig. 3 shows the behavioural VHDL-AMS model of PGA. The library clause and the use clause make all declarations in the packages electrical_systems, math_real and std_logic_1164 visible in the model. This is necessary, because the model uses nature electrical from package electrical_system and constant math_2_pi for the value of 2π from package math_real. The signals SI, CS, SCK, SO are of std_logic type, which is defined in package std_logic_1164. The proposed PGA model is composed by an *entity* and an *architecture*, where bold text indicates reserved words and upper-case text indicates predefined concepts. The *entity* declares the generic model parameters, as well as specifies interface terminals of nature electrical and logical ports of std_logic type. The generic parameters and constants, used in the simultaneous statements, are not given with their concrete numerical values in the model description. The proposed PGA model includes the following electrical terminals: input port – input, reference port – ref, output port – output, port for the positive supply voltage – nvdd and port for the negative supply voltage – nvss. The model has one inner terminal internal. It's used to specify the controlled source vint.

The *architecture* contains the implementation of the model. It is coded by combining structural and behavioural elements.

Also in the model the operation of two registers is reflected – instruction register and gain register. The significant bits for the instruction register are bit_{16}, bit_{17} and bit_{18} from SI, for gain register the significant bits are bit_{29}, bit_{30} and bit_{31} from SI. In the two channel MCP6S22, another register is added in the code - address register, where the significant bits within SI are bit_{23} and bit_{28}. These bits define the state of the digital signal addr. The addr defines whether the vamp is taken from the electrical terminals ch0 or ch1. The ch0 or ch1 replace the electrical terminal input in the code, given in Fig. 3.

IV. Model Performance

The verification check of the created behavioural PGA model is performed by comparing simulation results with the manufacturer's data for the IC MCP6S21. The simulations of the model are performed within System Vision 5.5 program (from Mentor Graphics). The test circuits are created following the test conditions, given in the semiconductor data book of the corresponding PGA.

The simulation testing is made for two values of the gain with values +2 and +8. The input sine-wave signal is chosen with amplitude $0.1V$, offset voltage $+2.5V$ and frequency $100kHz$. The external reference voltage is set to $+2.5V$ and the IC MCP6S21 is biased with $+5V$ power supply. The SPI data is defined with the states of the *SI, CS* and *SCK* signals. Fig. 4 shows the simulation results for the two values, upper results are for gain 2 and the results given below are for gain 8. The comparison gives a very good correspondence between the behavioural of the proposed PGA model and the real amplifier. The maximum error is not higher than 0.1%, which guarantee the sufficient degree of accuracy.

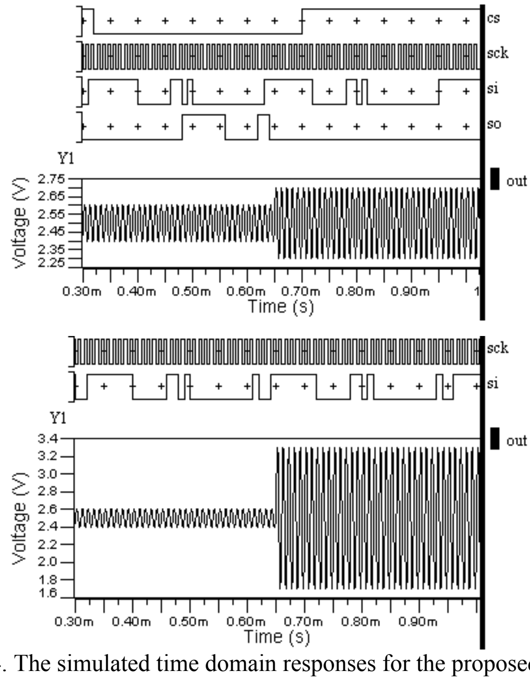

Fig. 4. The simulated time domain responses for the proposed PGA model at gain 2 and 8, respectively.

In order the workability of the PGA to be proved, the model is simulated in daisy chain configuration. It is realized following the specified way, given in technical documentation. Simulation results are shown on Fig. 5. The input sine-wave signal is with amplitude $0.1V$ for the first IC MCP6S21 and $0.05V$ for the second IC. For the two input sources the offset voltage is equal to $2.5V$ and the frequency is $100kHz$. The reference voltage is set to $2.5V$. SI signal for the first device sets the gain to 10, and for the second device the gain is 2. The *SI2* signal is *SO* of the first device that controls the second device in the daisy chain configuration. Simulation results show the proper work of the created model. The error is not higher than 0.1%.

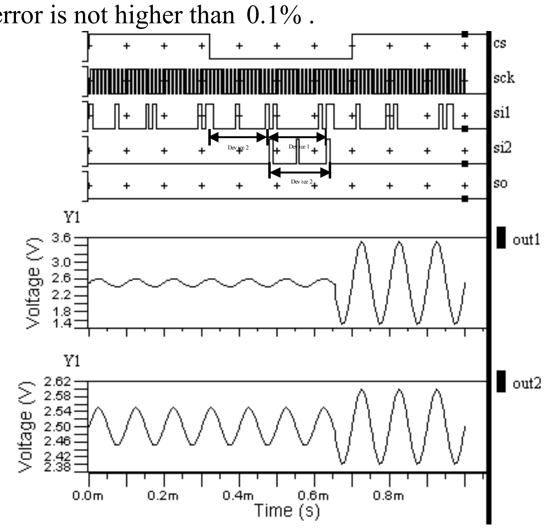

Fig. 5. The simulated time domain responses for the PGA model, connected in daisy chain configuration at gain 10 and 2, respectively.

V. Conclusion

In this paper a generalized behavioural VHDL-AMS model of monolithic PGAs over an SPI™ bus, based on the data sheet characteristics, has been presented. The proposed model accurately describes the dc, ac and transient behaviour of monolithic PGAs with binary voltage gains. The created model can be used for analysis and design of wide range of mixed-signal circuits and systems.

Acknowledgement

This paper is a part of a project under contract № 132pd0001-03/2013, which is funded by the research program of the TU-Sofia, Bulgaria.

References

[1] *PSpice A/D library list*, Release 9.2, Cadence Design Systems, 2000.
[2] System Vision mixed-signal model library (ver. 5.5), Mentor Graphics, 2007.
[3] D. Martev, I. Panayotov, I. Pandiev, "Behavioural VHDL-AMS Model for Monolithic Programmable Gain Amplifiers", ICEST vol. 2, pp. 835-838, 2010.
[4] I. Pandiev, P. Yakimov, T. Todorov, "Macromodeling of programmable gain amplifiers". E+E, No 7-8, pp. 69-76, 2009.
[5] Single-Ended, Rail-to-Rail I/O, Low Gain PGA, MCP6S21/2/6/8 - Datasheet, Microchip Technology Inc., 2003.
[6] E. Christen, K. Bakalar, "VHDL-AMS - A hardware description language for analog and mixed-signal applications, IEEE Trans. on cir. and syst. - II, vol. 46 (10), pp. 1263-1272, 1999.
[7] Definition of analog and mixed signal extensions to IEEE standard VHDL, IEEE Standard 1076.1, 1999.