ECSE 325 Lab 1: Register Transfer Level Design of a Modulo 33401 Digital Circuit

Cédric Barré (260792994)
Andy Li ()
Yuankang Wei(260787802)

g40_modulo33401 Digital Circuit

The g40_modulo33401 digital circuit was designed to calculate the modulo 33401 of any 32 bit input in the format of a 16 bit output. In fact, the circuit calculated the remainder of the 32 bit output divided by 33401. To do so, it utilizes some clever tricks to avoid calculating lengthy multiplications and divisions. In the end, the circuit implements the modulo as the following equations:

$$A \, mod \, 33401 \; = A \; - \; (floor(A/33401) \; * \; 33401)$$

$$floor(A/33401) = (A \; * \; 32147) > 30$$

Here, *A* is the 32 bit input and the 16 bit output corresponds to the *A mod 33401* signal. Moreover, we have an extra 17 bit output that corresponds to the *floor(A/33401)* signal. The *floor(A/33401)* signal is calculated by approximating the division through a multiplication with a constant and a 30 bit shift to the right. This is an approximation of the division which automatically rounds down the result of the division calculation.

VHDL g40_modulo33401 File

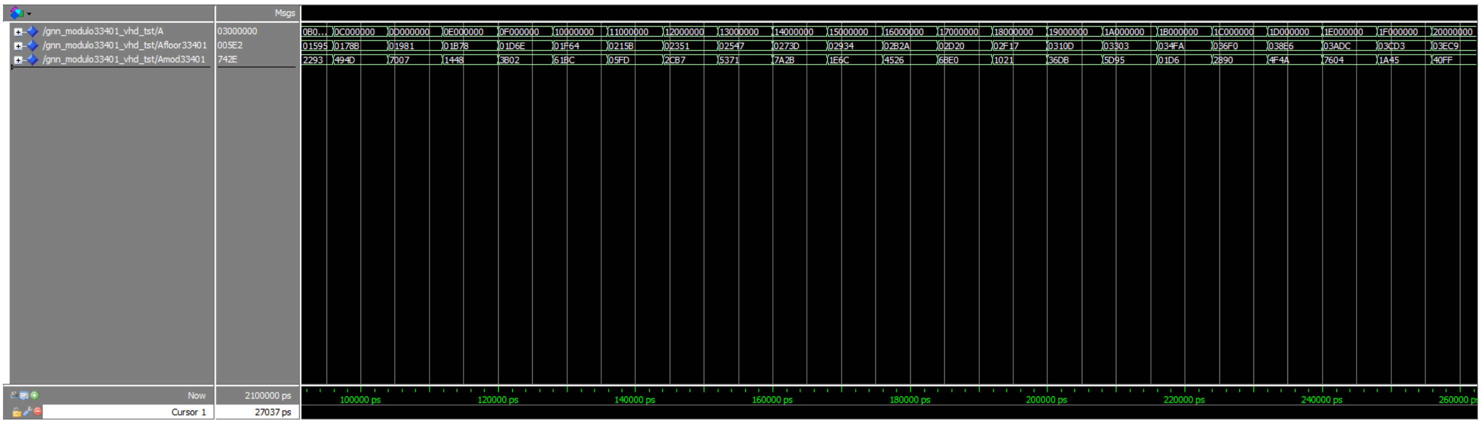
```
--
-- entity name: g40_modulo33401
--
-- Version 1.0
-- Authors: Cedric Barre, Andy Li and Yuankang Wei
-- Date: March 18, 2022
library ieee; -- allows use of the std_logic_vector type
use ieee.std_logic_1164.all;
use ieee.numeric_std.all; -- needed if you are using unsigned numbers

entity g40_modulo33401 is
      port (A :                     in std_logic_vector(31 downto 0);
                  Amod33401 :       out std_logic_vector(15 downto 0);
                  Afloor33401 :     out std_logic_vector(16 downto 0));
end g40_modulo33401;

architecture g40_modulo33401_arch of g40_modulo33401 is
      signal B : unsigned(46 downto 0);
      signal C : unsigned(40 downto 0);
      signal D : signed(46 downto 0);
      signal E : unsigned(31 downto 0);
      signal F : unsigned(19 downto 0);
      signal G : signed(31 downto 0);
      signal H : signed(16 downto 0);
      begin
            B <= (unsigned(A) & "000000000000000") +
                        ("00000000000" & unsigned(A) & "0000") +
                        ("00000000000000" & unsigned(A) & "0") +
                        ("000000000000000" & unsigned(A));
            C <= (unsigned(A) & "000000000") +
                        ("00" & unsigned(A) & "0000000");
            D <= signed(B) - signed("000000" + C);
            H <= signed(D(46) & D(46 downto 31));
            E <= (unsigned(H) & "000000000000000") +
                        ("000000" & unsigned(H) & "000000000") +
                        ("00000000" & unsigned(H) & "0000000") +
                        ("000000000000000" & unsigned(H));
            F <= (unsigned(H) & "000");
            G <= (signed(A)) - (signed(E) - signed("000000000000" & F));
            Afloor33401 <= std_logic_vector(H);
            Amod33401 <= std_logic_vector(G(15 downto 0));


end g40_modulo33401_arch;
```

VHDL Testbench File

```
-- Copyright (C) 2016  Intel Corporation. All rights reserved.
-- Your use of Intel Corporation's design tools, logic functions
-- and other software and tools, and its AMPP partner logic
-- functions, and any output files from any of the foregoing
-- (including device programming or simulation files), and any
-- associated documentation or information are expressly subject
-- to the terms and conditions of the Intel Program License
--  Subscription  Agreement,  the  Intel  Quartus  Prime  License
Agreement,
-- the Intel MegaCore Function License Agreement, or other
-- applicable license agreement, including, without limitation,
-- that your use is for the sole purpose of programming logic
-- devices manufactured by Intel and sold by Intel or its
-- authorized distributors.  Please refer to the applicable
-- agreement for further details.

--
*****************************************************************
*******
-- This file contains a Vhdl test bench template that is freely
editable to
-- suit user's needs .Comments are provided in each section to help
the user
-- fill out necessary details.
--
*****************************************************************
*******
-- Generated on "03/10/2022 16:17:55"

-- Vhdl Test Bench template for design  :  gNN_modulo33401
--
-- Simulation tool : ModelSim (VHDL)
--

LIBRARY ieee;
USE ieee.std_logic_1164.all;
use ieee.numeric_std.all;

ENTITY gNN_modulo33401_vhd_tst IS
END gNN_modulo33401_vhd_tst;
ARCHITECTURE gNN_modulo33401_arch OF gNN_modulo33401_vhd_tst IS
-- constants
-- signals
SIGNAL A : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL Afloor33401 : STD_LOGIC_VECTOR(16 DOWNTO 0);
SIGNAL Amod33401 : STD_LOGIC_VECTOR(15 DOWNTO 0);
COMPONENT gNN_modulo33401
      PORT (
      A : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
      Afloor33401 : OUT STD_LOGIC_VECTOR(16 DOWNTO 0);
      Amod33401 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0)
      );
END COMPONENT;
BEGIN
      i1 : gNN_modulo33401
      PORT MAP (
-- list connections between master ports and signals
      A => A,
      Afloor33401 => Afloor33401,
      Amod33401 => Amod33401
      );
always : PROCESS
-- optional sensitivity list
-- (        )
-- variable declarations
BEGIN
      for MSB in 0 to 255 loop
            A <= std_logic_vector(to_unsigned(MSB, 8) & x"000000");
            wait for 8 ns;
      end loop;
      WAIT;
END PROCESS always;
END gNN_modulo33401_arch;
```

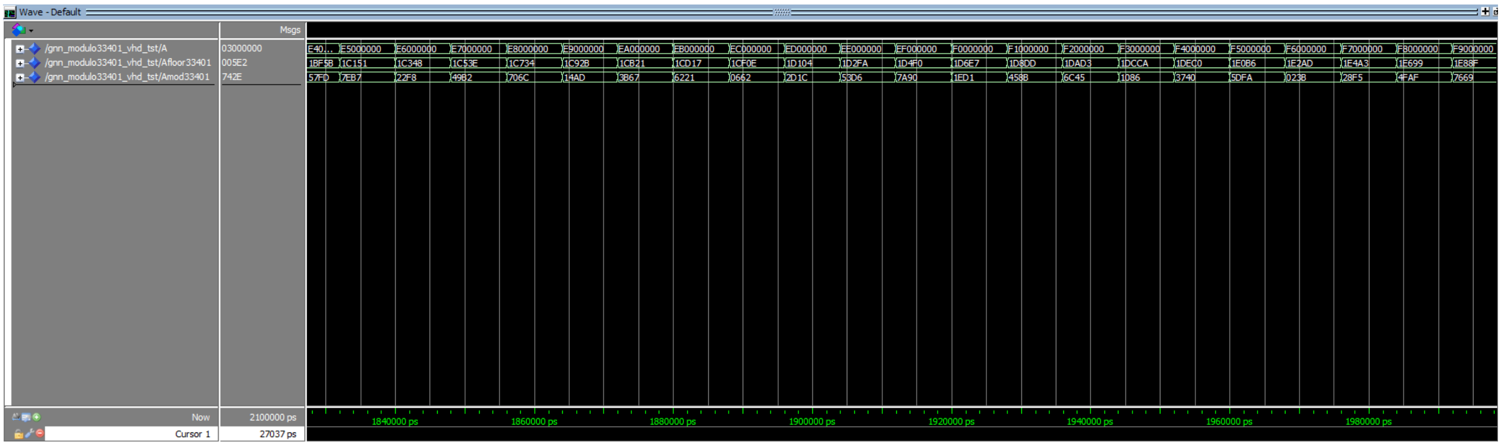
Testbench results

Since the testbench simulation tested for many values, it was impossible for us to fit all the evaluated inputs along with their corresponding outputs on the screen and take a screenshot. Therefore, we supply 2 screenshots at different times during the simulation displaying a considerable amount of inputs with their corresponding outputs. In the end, we cross-checked many of the outputs with calculations carried out by calculator and all of the outputs we checked matched the calculations.



```
                                          Msgs
/gnn_modulo33401_vhd_tst/A                03000000  0B0... 0C000000 0D000000 0E000000 0F000000 10000000 11000000 12000000 13000000 14000000 15000000 16000000 17000000 18000000 19000000 1A000000 1B000000 1C000000 1D000000 1E000000 1F000000 20000000
/gnn_modulo33401_vhd_tst/Afloor33401      005E2     01595 0178B 01981 01B78 01D6E 01F64 0215B 02351 02547 0273D 02934 02B2A 02D20 02F17 0310D 03303 034FA 036F0 038E6 03ADC 03CD3 03EC9
/gnn_modulo33401_vhd_tst/Amod33401        742E      2293 494D 7007 1448 3B02 61BC 05FD 2CB7 5371 7A2B 1E6C 4526 6BE0 1021 36DB 5D95 01D6 2890 4F4A 7604 1A45 40FF

Now       2100000 ps     100000 ps  120000 ps  140000 ps  160000 ps  180000 ps  200000 ps  220000 ps  240000 ps  260000 ps
Cursor 1  27037 ps
```

Figure 1: First screenshot of the testbench simulation



```
Wave - Default
                                          Msgs
/gnn_modulo33401_vhd_tst/A                03000000  E40... E5000000 E6000000 E7000000 E8000000 E9000000 EA000000 EB000000 EC000000 ED000000 EE000000 EF000000 F0000000 F1000000 F2000000 F3000000 F4000000 F5000000 F6000000 F7000000 F8000000 F9000000
/gnn_modulo33401_vhd_tst/Afloor33401      005E2     1BF5B 1C151 1C348 1C53E 1C734 1C92B 1CB21 1CD17 1CF0E 1D104 1D2FA 1D4F0 1D6E7 1D8DD 1DAD3 1DCCA 1DEC0 1E0B6 1E2AD 1E4A3 1E699 1E88F
/gnn_modulo33401_vhd_tst/Amod33401        742E      57FD 7EB7 22F8 49B2 706C 14AD 3B67 6221 0662 2D1C 53D6 7A90 1ED1 458B 6C45 1086 3740 5DFA 023B 28F5 4FAF 7669

Now       2100000 ps     1840000 ps  1860000 ps  1880000 ps  1900000 ps  1920000 ps  1940000 ps  1960000 ps  1980000 ps
Cursor 1  27037 ps
```

Figure 2: Second screenshot of the testbench simulation