INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE COMPUTO

Std Logic Arith & Unsigned

Practica 7

Materia:

Arquitectura de computadoras

Profesor:

Castillo Cabrera Gelacio

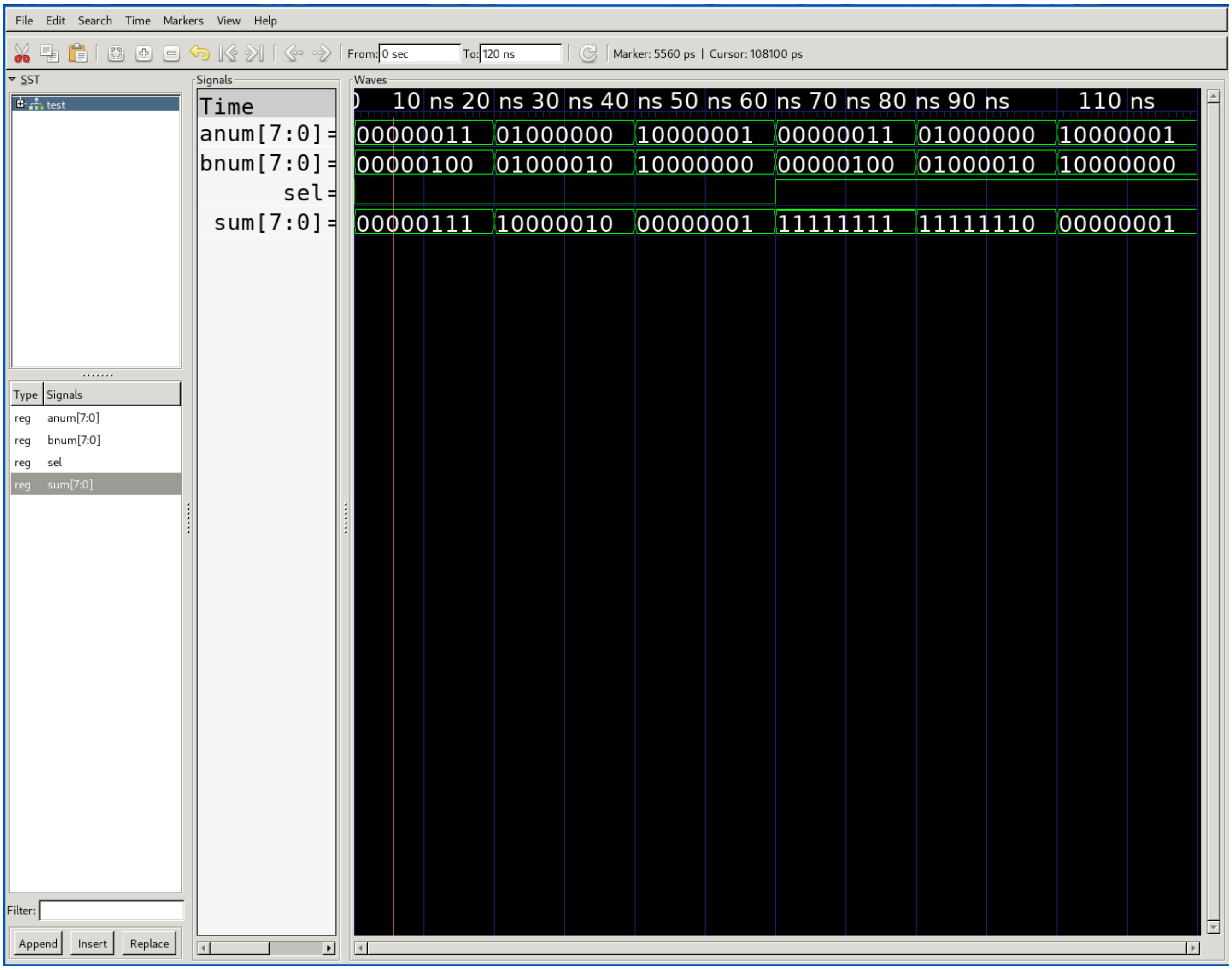
Alumno:

Cortés Piña Oziel

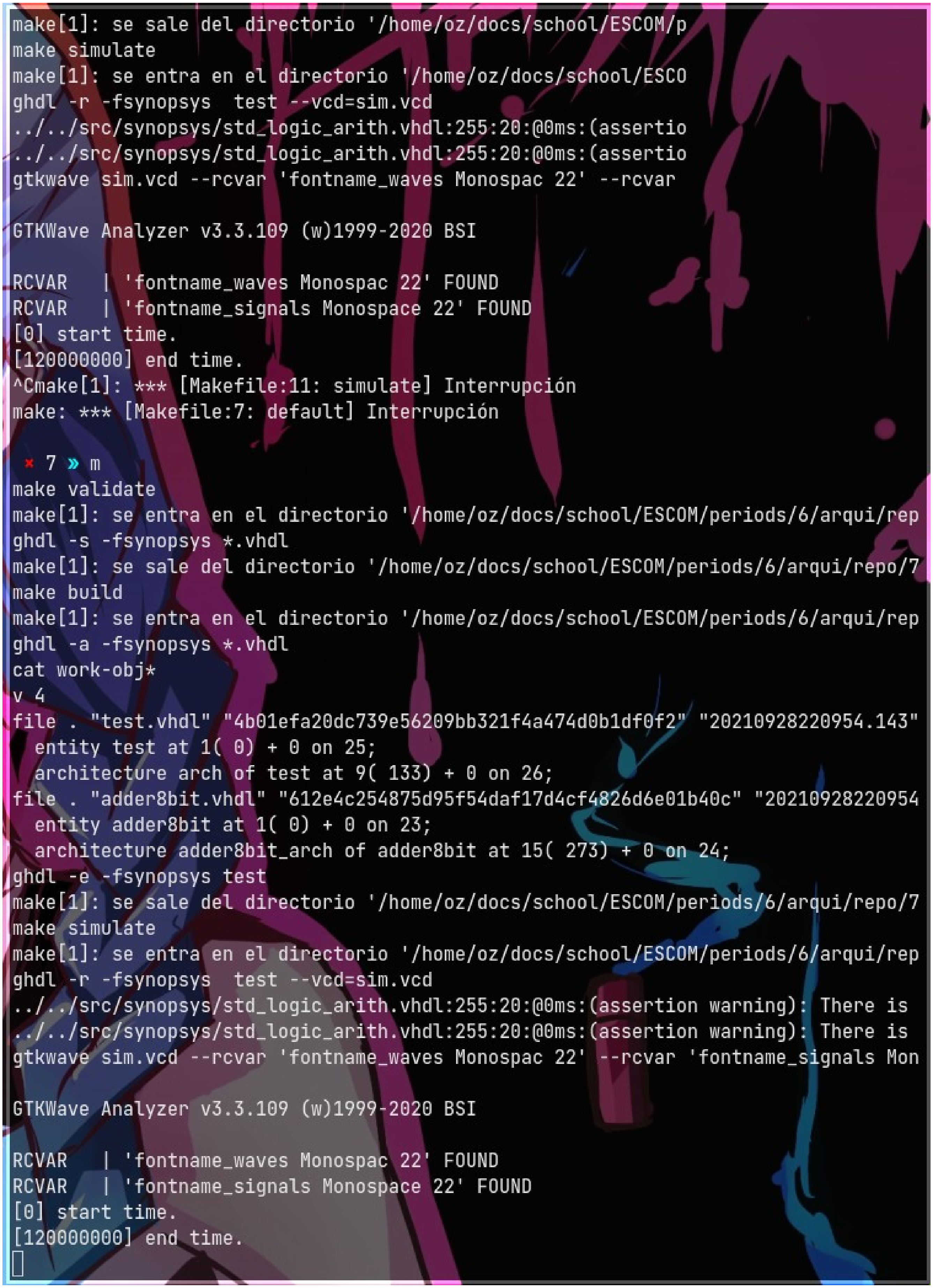
Grupo:

3CM12

Simulación en GHDL y GTLWAVE



```
Time       0  10 ns 20 ns 30 ns 40 ns 50 ns 60 ns 70 ns 80 ns 90 ns    110 ns
anum[7:0]= 00000011  01000000  10000001  00000011  01000000  10000001
bnum[7:0]= 00000100  01000010  10000000  00000100  01000010  10000000
      sel=
 sum[7:0]= 00000111  10000010  00000001  11111111  11111110  00000001
```



```
make[1]: se sale del directorio '/home/oz/docs/school/ESCOM/p
make simulate
make[1]: se entra en el directorio '/home/oz/docs/school/ESCO
ghdl -r -fsynopsys  test --vcd=sim.vcd
../../src/synopsys/std_logic_arith.vhdl:255:20:@0ms:(assertio
../../src/synopsys/std_logic_arith.vhdl:255:20:@0ms:(assertio
gtkwave sim.vcd --rcvar 'fontname_waves Monospac 22' --rcvar

GTKWave Analyzer v3.3.109 (w)1999-2020 BSI

RCVAR   | 'fontname_waves Monospac 22' FOUND
RCVAR   | 'fontname_signals Monospace 22' FOUND
[0] start time.
[120000000] end time.
^Cmake[1]: *** [Makefile:11: simulate] Interrupción
make: *** [Makefile:7: default] Interrupción

 × 7 » m
make validate
make[1]: se entra en el directorio '/home/oz/docs/school/ESCOM/periods/6/arqui/rep
ghdl -s -fsynopsys *.vhdl
make[1]: se sale del directorio '/home/oz/docs/school/ESCOM/periods/6/arqui/repo/7
make build
make[1]: se entra en el directorio '/home/oz/docs/school/ESCOM/periods/6/arqui/rep
ghdl -a -fsynopsys *.vhdl
cat work-obj*
v 4
file . "test.vhdl" "4b01efa20dc739e56209bb321f4a474d0b1df0f2" "20210928220954.143"
  entity test at 1( 0) + 0 on 25;
  architecture arch of test at 9( 133) + 0 on 26;
file . "adder8bit.vhdl" "612e4c254875d95f54daf17d4cf4826d6e01b40c" "20210928220954
  entity adder8bit at 1( 0) + 0 on 23;
  architecture adder8bit_arch of adder8bit at 15( 273) + 0 on 24;
ghdl -e -fsynopsys test
make[1]: se sale del directorio '/home/oz/docs/school/ESCOM/periods/6/arqui/repo/7
make simulate
make[1]: se entra en el directorio '/home/oz/docs/school/ESCOM/periods/6/arqui/rep
ghdl -r -fsynopsys  test --vcd=sim.vcd
../../src/synopsys/std_logic_arith.vhdl:255:20:@0ms:(assertion warning): There is
../../src/synopsys/std_logic_arith.vhdl:255:20:@0ms:(assertion warning): There is
gtkwave sim.vcd --rcvar 'fontname_waves Monospac 22' --rcvar 'fontname_signals Mon

GTKWave Analyzer v3.3.109 (w)1999-2020 BSI

RCVAR   | 'fontname_waves Monospac 22' FOUND
RCVAR   | 'fontname_signals Monospace 22' FOUND
[0] start time.
[120000000] end time.
```

Código VHDL

Adder8bit

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity adder8bit is
      port(
            sel:  in    std_logic;
            anum: in    unsigned(7 downto 0);
            bnum: in    unsigned(7 downto 0);
            sum:  out   unsigned(7 downto 0)
      );
end adder8bit;

architecture adder8bit_arch of adder8bit is
begin
      sum <= anum + bnum when sel = '0'
               else anum - bnum;
end adder8bit_arch;
```

Test

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity test is
end test;

architecture arch of test is
      component adder8bit is
            port(
                  sel:   in    std_logic;
                  anum:  in    unsigned(7 downto 0);
                  bnum:  in    unsigned(7 downto 0);
                  sum:   out   unsigned(7 downto 0)
            );
      end component;
      signal sel:   std_logic;
      signal anum:  unsigned(7 downto 0);
      signal bnum:  unsigned(7 downto 0);
      signal sum:   unsigned(7 downto 0);
begin
      adder: adder8bit port map( sel => sel, anum => anum, bnum => bnum, sum => sum );
      process begin
            sel <= '0';
            anum <= "00000011";
            bnum <= "00000100";
            wait for 20 ns;
            anum <= "01000000";
            bnum <= "01000010";
            wait for 20 ns;
            anum <= "10000001";
            bnum <= "10000000";
```

```
             wait for 20 ns;
             sel <= '1';
             anum <= "00000011";
             bnum <= "00000100";
             wait for 20 ns;
             anum <= "01000000";
             bnum <= "01000010";
             wait for 20 ns;
             anum <= "10000001";
             bnum <= "10000000";
             wait for 20 ns;
             wait;
      end process;
end arch;
```

Análisis de vectores

Practica 6

Vectores de entrada

A		B	
Binario	Hexadecimal	Binario	Hexadecimal
00000011	3	00000100	4
01000000	40	01000010	42
10000001	81	10000000	80

Valores de salida

A		B		Selector		
Binario	Hexa	Binario	Hexa	Binario	Binario	Hexa
00000011	3	00000100	4	0	00000111	7
01000000	40	01000010	42	0	00000000	0
10000001	81	10000000	80	0	00000000	0
00000011	3	00000100	4	1	11111111	FF
01000000	40	01000010	42	1	11111110	FE
10000001	81	10000000	80	1	00000001	1

Practica 7

Vectores de entrada

A		B	
Binario	Hexadecimal	Binario	Hexadecimal
00000011	3	00000100	4
01000000	40	01000010	42
10000001	81	10000000	80

Valores de salida

A		B		Selector		
Binario	Hexa	Binario	Hexa	Binario	Binario	Hexa
00000011	3	00000100	4	0	00000111	7
01000000	40	01000010	42	0	00000000	82
10000001	81	10000000	80	0	00000000	01
00000011	3	00000100	4	1	11111111	FF
01000000	40	01000010	42	1	11111110	FE
10000001	81	10000000	80	1	00000001	1

Conclusión

La biblioteca de arith en VHDL permite realizar circuitos más fáciles y ahorra mucho tiempo más sin en-cambio arroja valores distintos al momento de realizar las operaciones de suma y resta haciendo que se aproveche el número a sumarse y restarte a diferencia del circuito anterior en el que teníamos desbordes. Pareciera que usa los positivos y negativos a su favor del número para representarlo.