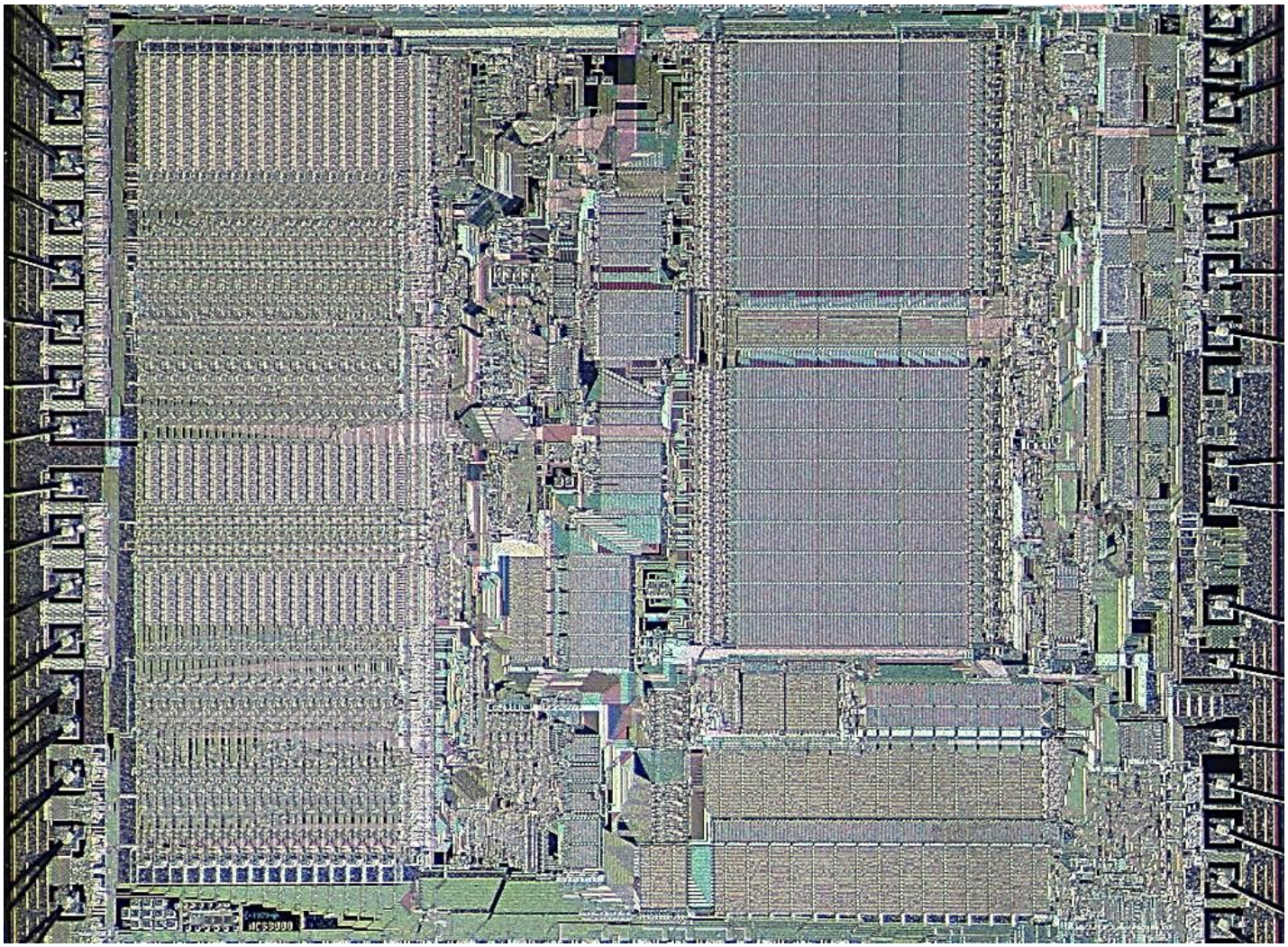
SISTEMAS DIGITALES

FIRST LAB

Victor Moreno Arribas

a180070



Integrated circuit from MC68000

CODE
SCREENSHOTS

1. Half Adder
2. Half Adder Test
3. Full Adder (with logic gates)
4. Full Adder (with half adders)
5. Full Adder Test (logic gates)
6. Full Adder Test (half adders)

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity HalfAdder is
5     generic (delay : time := 0 ns);
6     port(a, b : in  std_logic;
7          s, c : out std_logic);
8   end entity;
9
10  architecture Flow of HalfAdder is
11
12  begin
13          s <= a xor b after delay;
14          c <= a and b after delay;
15
16  end architecture;
17
```

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity HalfAdderTest is
5  end entity;
6
7  architecture SimpleTest of HalfAdderTest is
8    signal inta, intb: std_logic:='0';
9    signal ints, intc: std_logic;
10   constant lag: time:= 1 ns;
11   begin
12     TheHalfAdder: entity work.HalfAdder(Flow)
13       generic map(delay => lag)
14       port map(a => inta, b => intb,
15                s => ints, c => intc);
16
17     inta <= not inta after 10*lag;
18     intb <= not intb after 20*lag;
19
20   end architecture SimpleTest;
21
22
23
```

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity FullAdder is
5    generic (delay : time := 0 ns);
6    port (a, b, c     : in  std_logic;
7          sum, carry : out std_logic);
8  end entity;
9
10 architecture WithGates of FullAdder is
11
12 signal intA, intB, intC: std_logic;
13
14 begin
15         intA <= a xor b after delay;
16         intB <= c and intA after delay;
17         intC <= b and a after delay;
18         sum <= intA xor c after delay;
19         carry <= intB or intC after delay;
20
21 end WithGates;
```

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity FullAdder is
5     generic (delay : time := 0 ns);
6     port (a, b, c     : in  std_logic;
7           sum, carry : out std_logic);
8   end entity;
9
10  architecture WithHalfAdders of FullAdder is
11
12  signal aux1, aux2, aux3: std_logic;
13
14  begin
15  First: entity work.HalfAdder(Flow)
16           generic map(delay => delay)
17           port map(b => a, a => b,
18                    s => aux2, c => aux1);
19
20  Second: entity work.HalfAdder(Flow)
21           generic map(delay => delay)
22           port map(a => c, b => aux2,
23                    s => sum, c =>aux3);
24
25           carry <= aux3 or aux1 after delay;
26  end WithHalfAdders;
27
```

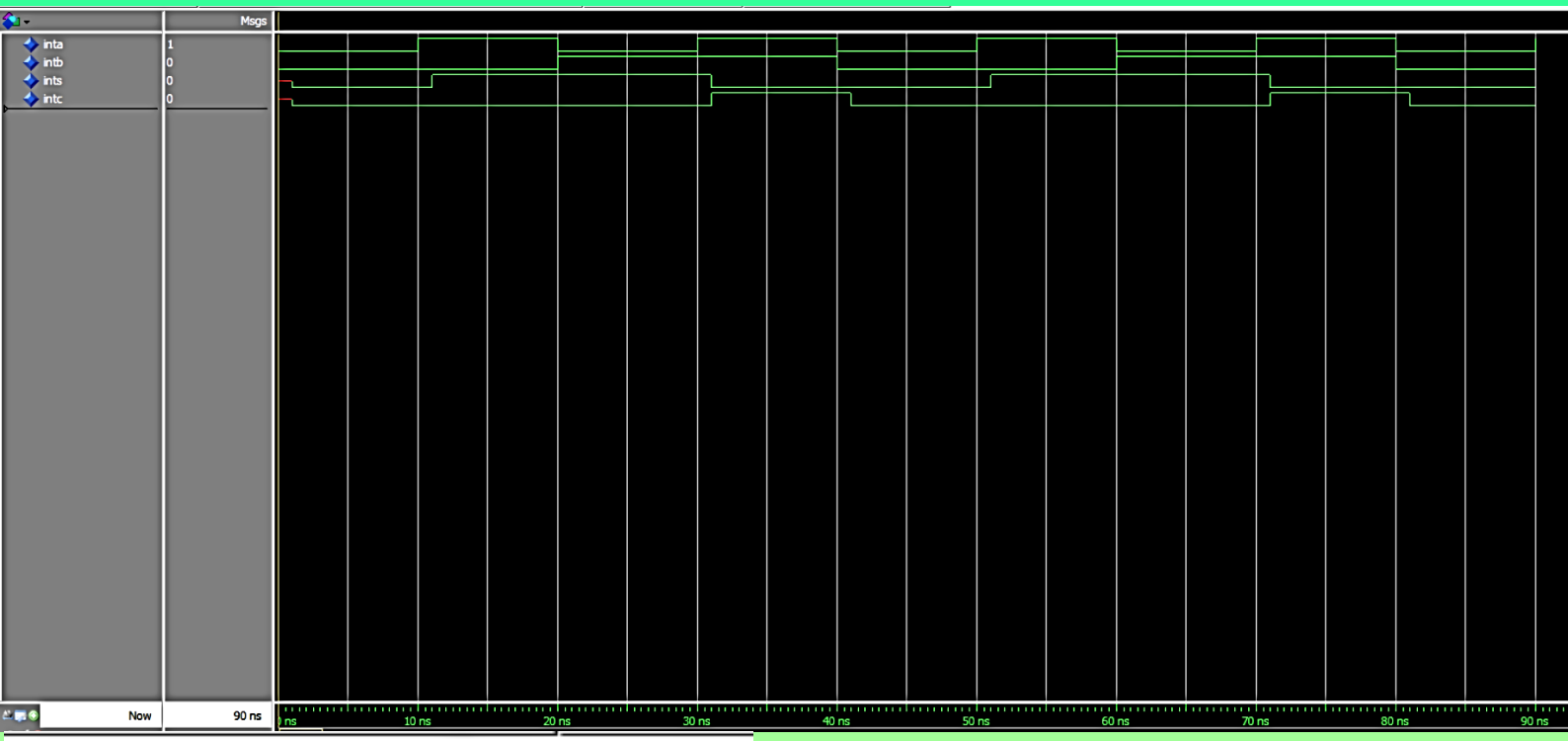
```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity FullAdderTest is
5   end entity;
6
7   architecture TestSums of FulladderTest is
8     signal a, b, c: std_logic:='0';
9     signal sum, carry: std_logic;
10    constant lag: time:= 1 ns;
11    begin
12      TheFullAdder: entity work.FullAdder(WithGates)
13        generic map(delay => lag)
14        port map(a => a, b => b, c=> c,
15                 sum => sum, carry => carry);
16
17      a <= not a after 10*lag;
18      b <= not b after 20*lag;
19      c <= not c after 40*lag;
20
21      end architecture testSums;
22
23
24
```

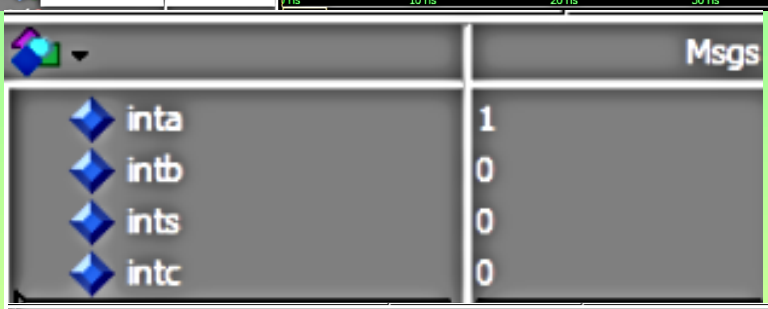
```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity FullAdderTest is
5   end entity;
6
7   architecture TestSums of FulladderTest is
8     signal a, b, c: std_logic:='0';
9     signal sum, carry: std_logic;
10    constant lag: time:= 1 ns;
11    begin
12      TheFullAdder: entity work.FullAdder(WithHalfAdders)
13        generic map(delay => lag)
14        port map(a => a, b => b, c=> c,
15                 sum => sum, carry => carry);
16
17      a <= not a after 10*lag;
18      b <= not b after 20*lag;
19      c <= not c after 40*lag;
20
21      end architecture testSums;
22
23
24
```

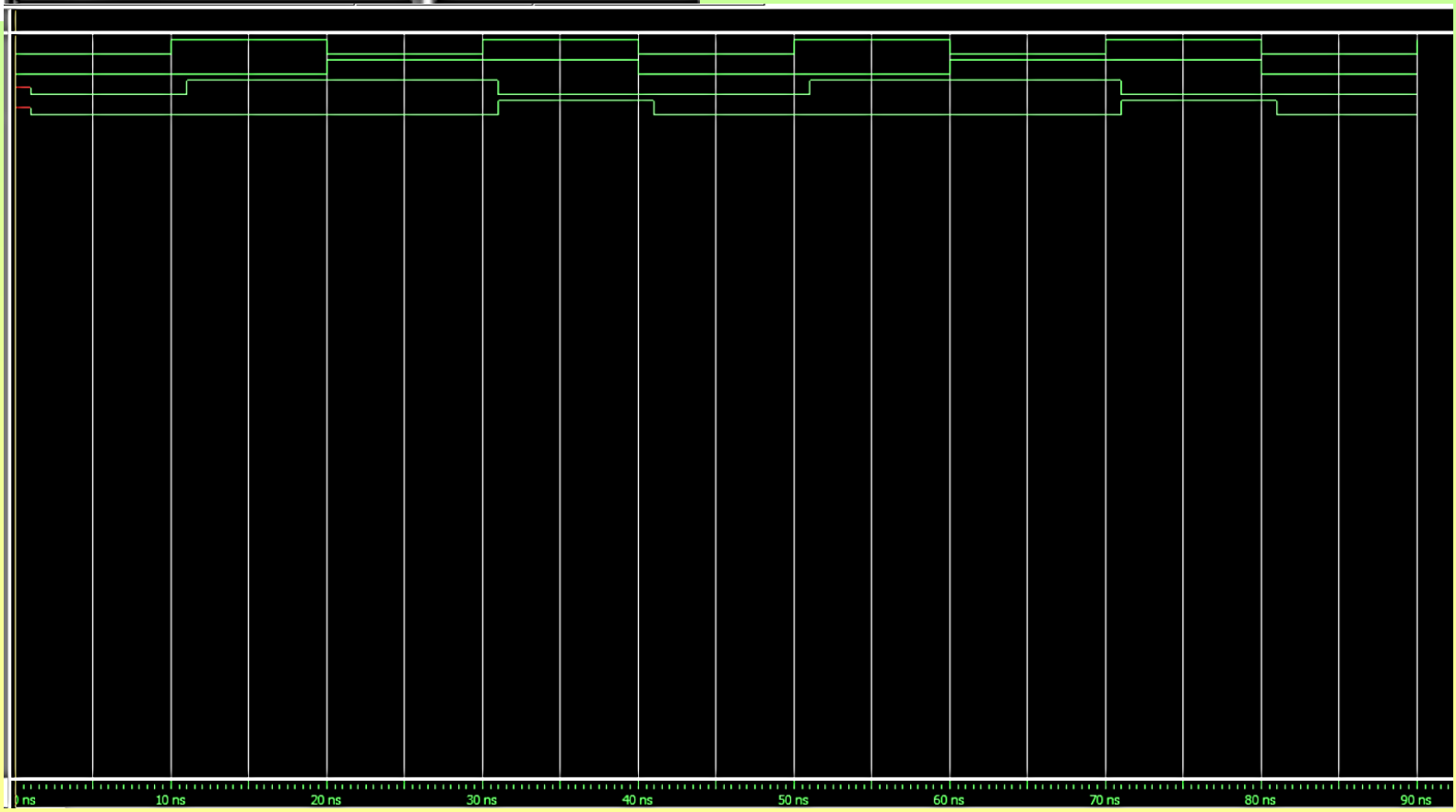
SIMULATION SCREENSHOTS

1. Half Adder Test
2. Full Adder Test (logic gates)
3. Full Adder Test (half adders)

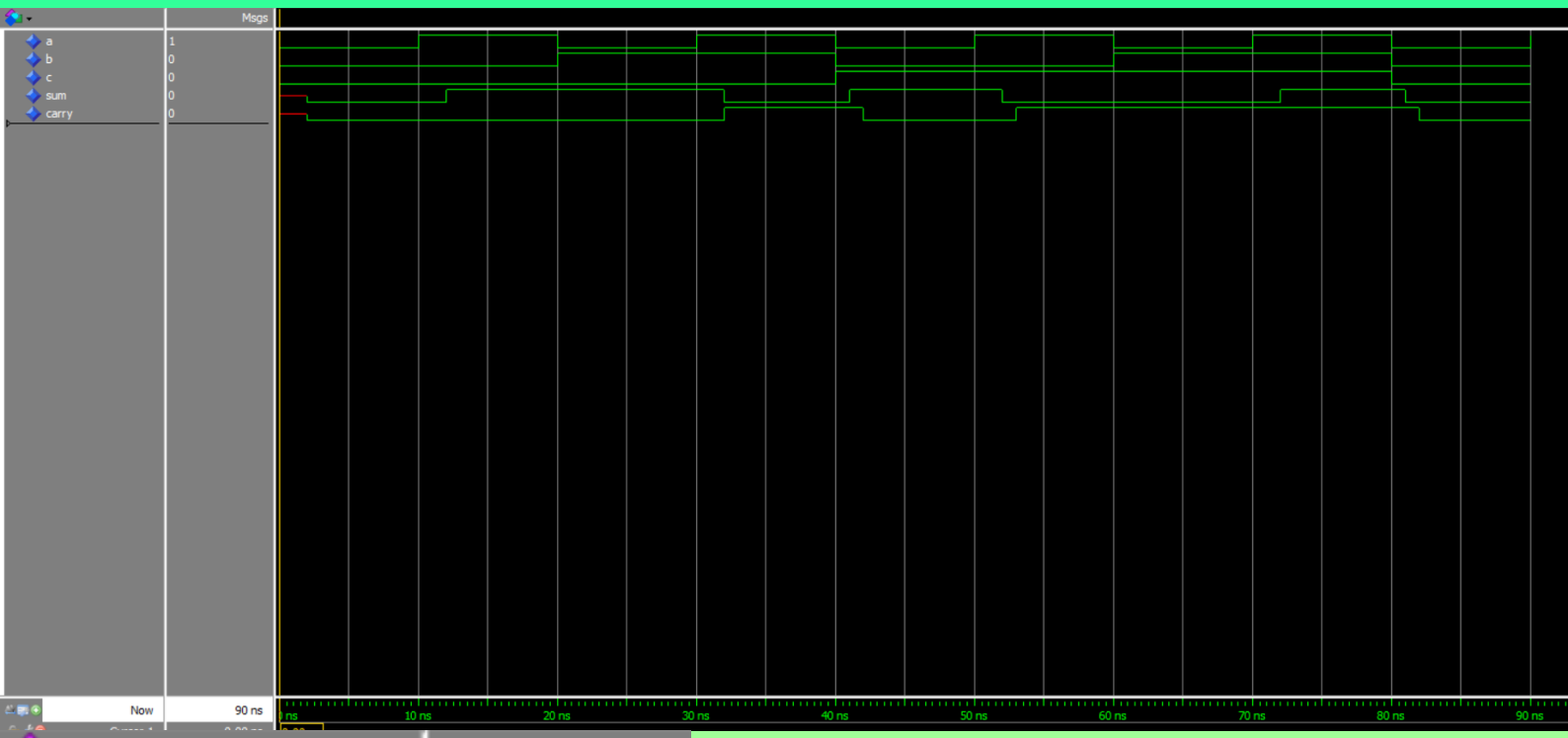
1 2 3


Msgs
inta
intb
ints
intc
1
0
0
0
Now
90 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns

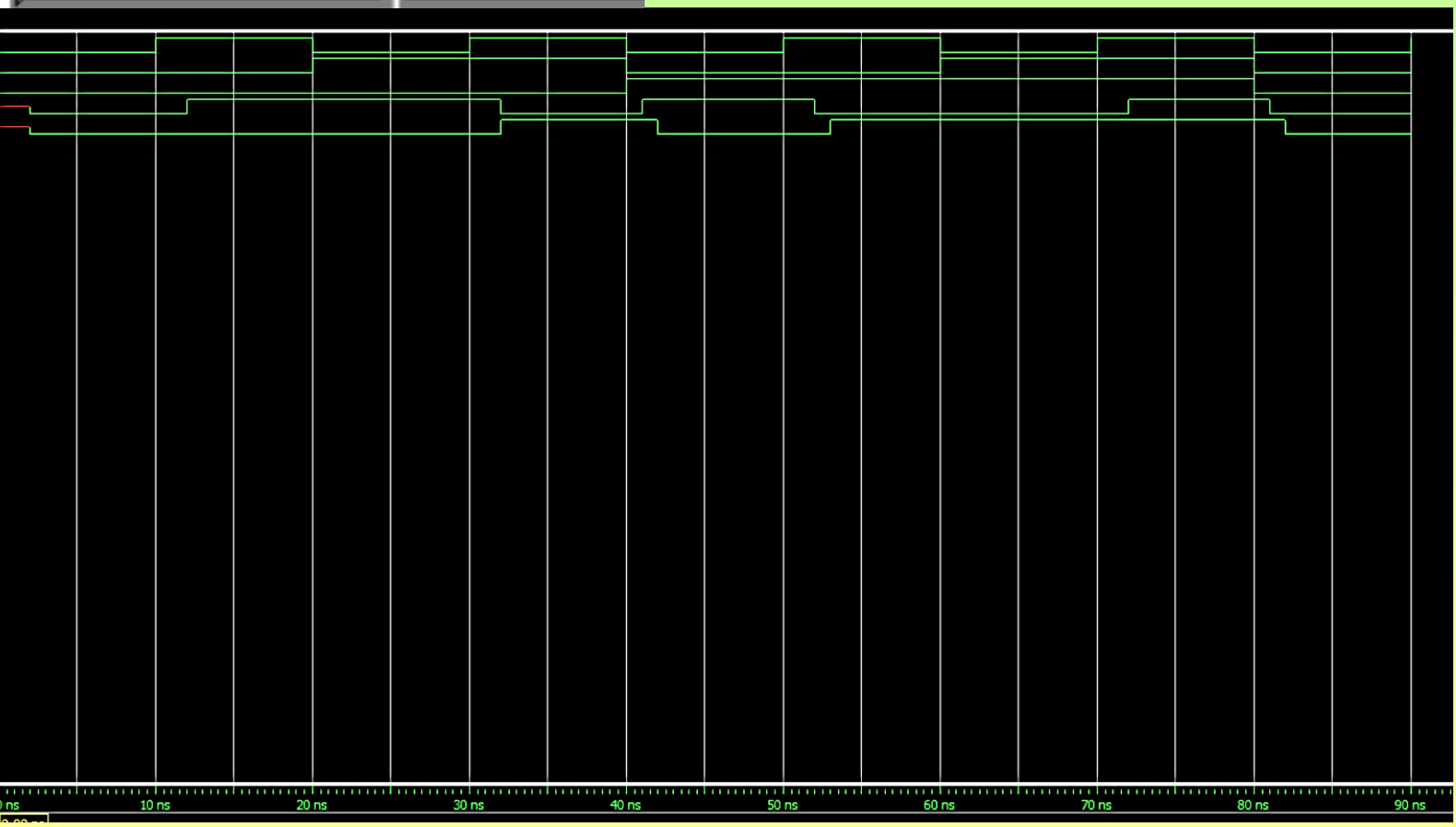

Msgs
inta
intb
ints
intc
1
0
0
0


10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns

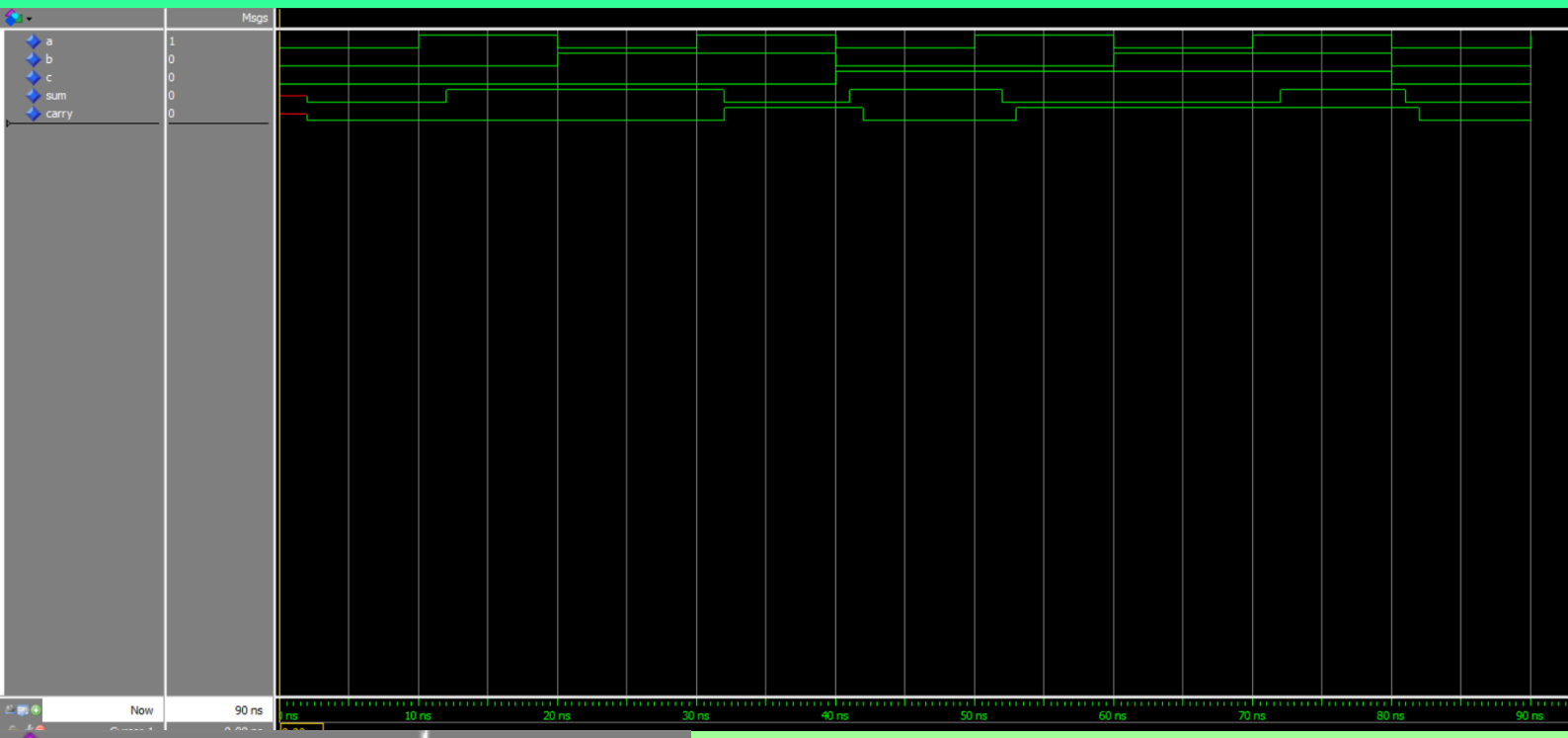
1 2 3


Msgs
a
b
c
sum
carry
1
0
0
0
0
Now
90 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns

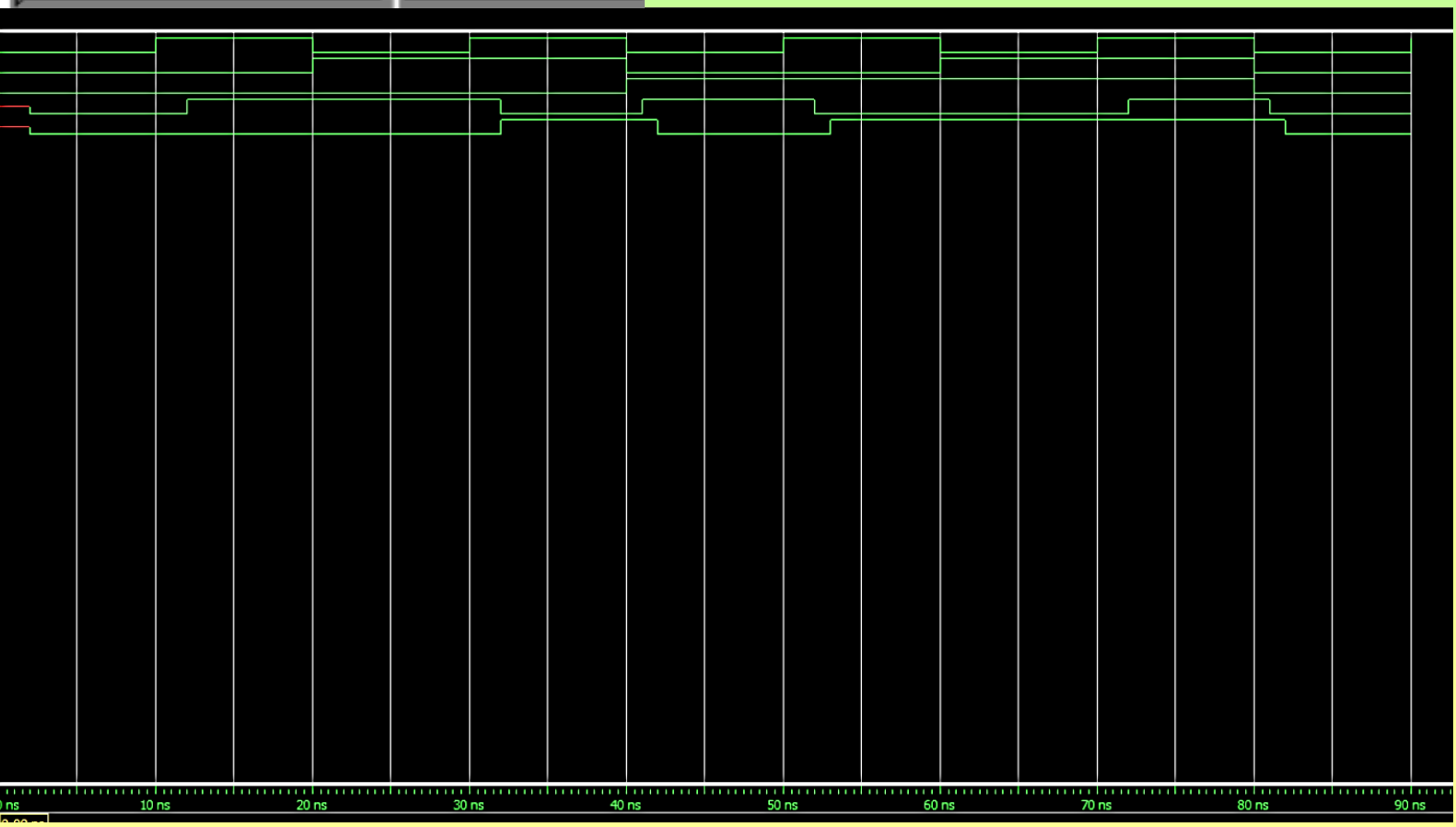

Msgs
a
b
c
sum
carry
1
0
0
0
0


10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns

1 2 3


Msgs
a
b
c
sum
carry
1
0
0
0
0
Now
90 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns


Msgs
a
b
c
sum
carry
1
0
0
0
0


10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns

Q & A

1. Question 1
2. Question2
3. Question 3
4. Question 4
5. Question 5

Which is the biggest delay present in the first simulation of the full-adder?

By the “first simulation” I understand that is the simulation for the Full Adder that was designed only with logic gates.
For my specific design of my Full Adder (WithGates) the highest delay showed in the simulation was of 3ns.
Which means that there is a special case in which it takes 3ns to go from the input all the way to the output.

Which is the biggest delay present in the second simulation of the full-adder?

By the “second simulation” I understand that is the simulation for the Full Adder that was designed only with half adders.
For my specific design of my Full Adder (WithHalfAdders) the highest delay showed in the simulation was also 3ns.
Which means that there is a special case in which it takes 3ns to go from the input all the way to the output.

Explain what's the reason for this difference.

In my case, both delays had the same value, which means there is no difference.

Explain what could you do to change the frequency of the signals of the simulation of the full-adder, without altering the delays of the circuit.

What's the value finally used for the delay of gates of the half-adder in the simulation of the “WithHalfAdders" architecture of the full-adder?

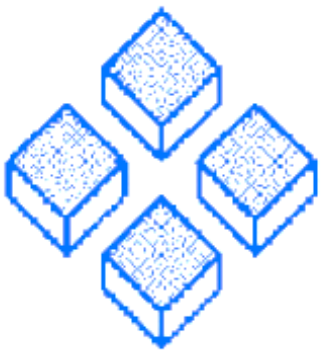
Where does that value come from, and how do these gates finally get it?

POLITÉCNICA

"Ingeniamos el futuro"

CAMPUS
DE EXCELENCIA
INTERNACIONAL

E.T.S. de Ingenieros
Informaticos



datsi