

Universidad Rafael Landívar
Facultad de Ingeniería
Ingeniería en Industrial y de Sistemas
Arquitectura del Computador I
Ing. Jefferson Esquivel



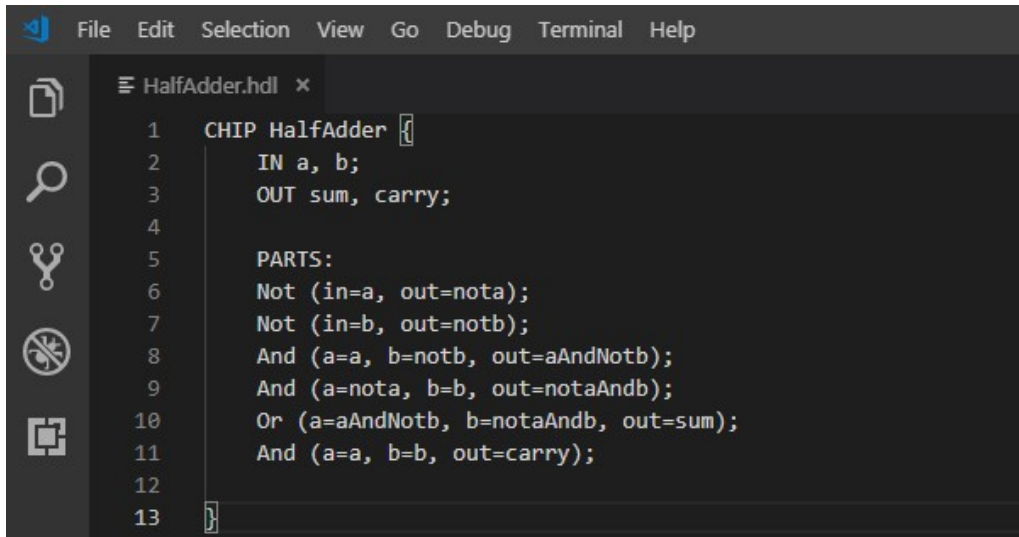
Laboratorio No. 2

Gabriel Molina – 1213416
Axel Rodriguez - 1229715
Julia Pineda - 1020415
Yazmine Sierra - 1174916

Guatemala, 31 de enero del 2019

Screenshots

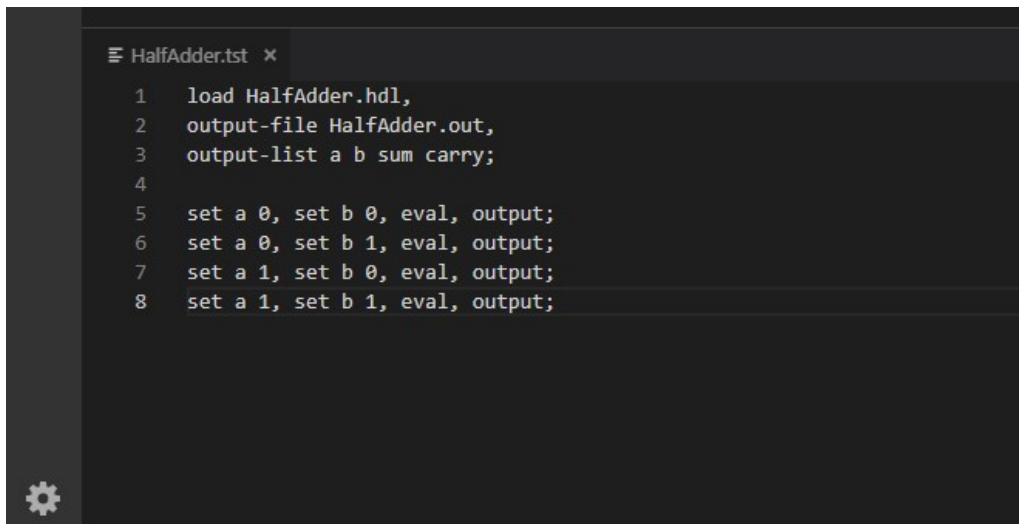
Archivo HDL del medio sumador



The screenshot shows a text editor window titled "HalfAdder.hdl". The code defines a chip named "HalfAdder" with two inputs, "a" and "b", and two outputs, "sum" and "carry". It uses a "PARTS:" block to instantiate logic components: "Not" gates for "nota" and "notb", "And" gates for "aAndNotb", "notaAndb", and "carry", and an "Or" gate for "sum".

```
1  CHIP HalfAdder
2      IN a, b;
3      OUT sum, carry;
4
5      PARTS:
6          Not (in=a, out=nota);
7          Not (in=b, out=notb);
8          And (a=a, b=notb, out=aAndNotb);
9          And (a=nota, b=b, out=notaAndb);
10         Or (a=aAndNotb, b=notaAndb, out=sum);
11         And (a=a, b=b, out=carry);
12
13
```

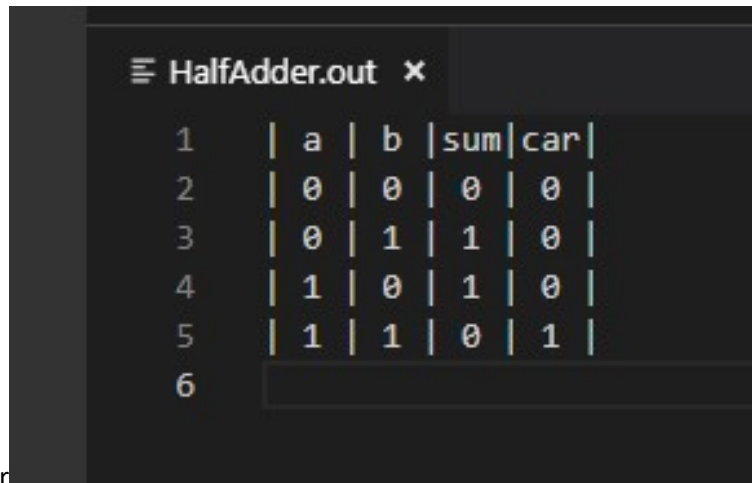
Archivo para pruebas del medio sumador



The screenshot shows a text editor window titled "HalfAdder.tst". The code sets up a test environment by loading "HalfAdder.hdl", specifying the output file "HalfAdder.out", and listing the outputs "a", "b", "sum", and "carry". It then performs four test cases, each setting inputs "a" and "b" to specific values (0 or 1), evaluating the circuit, and outputting the results.

```
1  load HalfAdder.hdl,
2  output-file HalfAdder.out,
3  output-list a b sum carry;
4
5  set a 0, set b 0, eval, output;
6  set a 0, set b 1, eval, output;
7  set a 1, set b 0, eval, output;
8  set a 1, set b 1, eval, output;
```

Salida del medio sumador y la tabla de verdad para



```
≡ HalfAdder.out ×
1 | a | b | sum | car |
2 | 0 | 0 | 0 | 0 |
3 | 0 | 1 | 1 | 0 |
4 | 1 | 0 | 1 | 0 |
5 | 1 | 1 | 0 | 1 |
6 |   |   |   |   |
```

comparar

Truth Table			
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Archivo HDL del sumador completo

```
Adder.tst - Visual Studio Code
Adder.hdl x
CHIP FullAdder
    IN a, b, c;
    OUT sum, carry;

    PARTS:
        Not (in=a, out=nota);
        Not (in=b, out=notb);
        And (a=a, b=notb, out=aAndNotb);
        And (a=nota, b=b, out=notaAndb);
        Or (a=aAndNotb, b=notaAndb, out=sum1);
        And (a=a, b=b, out=carry1);

        Not (in=sum1, out=nota1);
        Not (in=c, out=notb1);
        And (a=sum1, b=notb1, out=aAndNotb1);
        And (a=nota1, b=c, out=notaAndb1);
        Or (a=aAndNotb1, b=notaAndb1, out=sum);
        And (a=sum1, b=c, out=carry2);

        Or (a=carry1, b=carry2, out=carry);
```

Archivo de pruebas para el sumador completo

```

1  load FullAdder.hdl,
2  output-file FullAdder.out,
3  output-list a b c sum carry;
4
5  set a 0, set b 0, set c 0, eval, output;
6  set a 0, set b 0, set c 1, eval, output;
7  set a 0, set b 1, set c 0, eval, output;
8  set a 0, set b 1, set c 1, eval, output;
9  set a 1, set b 0, set c 0, eval, output;
10 set a 1, set b 0, set c 1, eval, output;
11 set a 1, set b 1, set c 0, eval, output;
12 set a 1, set b 1, set c 1, eval, output;

```

a del sumador completo y tabla de verdad para comparar

1	a	b	c	sum	car
2	0	0	0	0	0
3	0	0	1	1	0
4	0	1	0	1	0
5	0	1	1	0	1
6	1	0	0	1	0
7	1	0	1	0	1
8	1	1	0	0	1
9	1	1	1	1	1
10					

A	B	CarryIn	CarryOut	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1