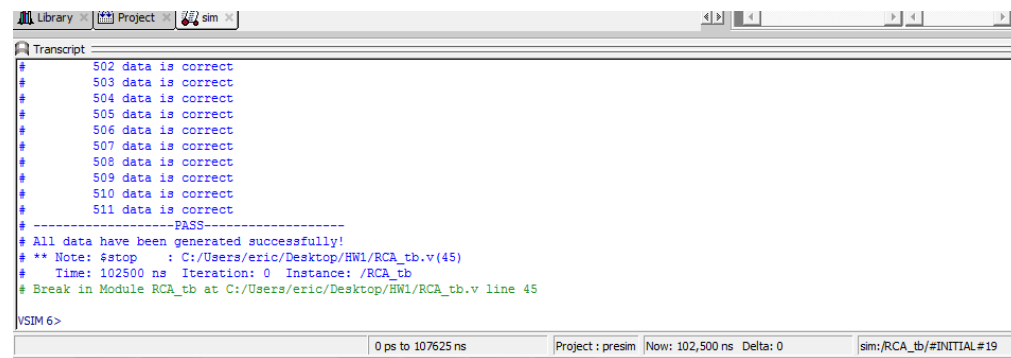


## Functional Simulation:



The screenshot shows the 'Transcript' window of the Quartus Prime IDE. The transcript contains the following text:

```
# 502 data is correct
# 503 data is correct
# 504 data is correct
# 505 data is correct
# 506 data is correct
# 507 data is correct
# 508 data is correct
# 509 data is correct
# 510 data is correct
# 511 data is correct
# -----PASS-----
# All data have been generated successfully!
# ** Note: $stop : C:/Users/eric/Desktop/HW1/RCA_tb.v(45)
# Time: 102500 ns Iteration: 0 Instance: /RCA_tb
# Break in Module RCA_tb at C:/Users/eric/Desktop/HW1/RCA_tb.v line 45
VSIM 6>
```

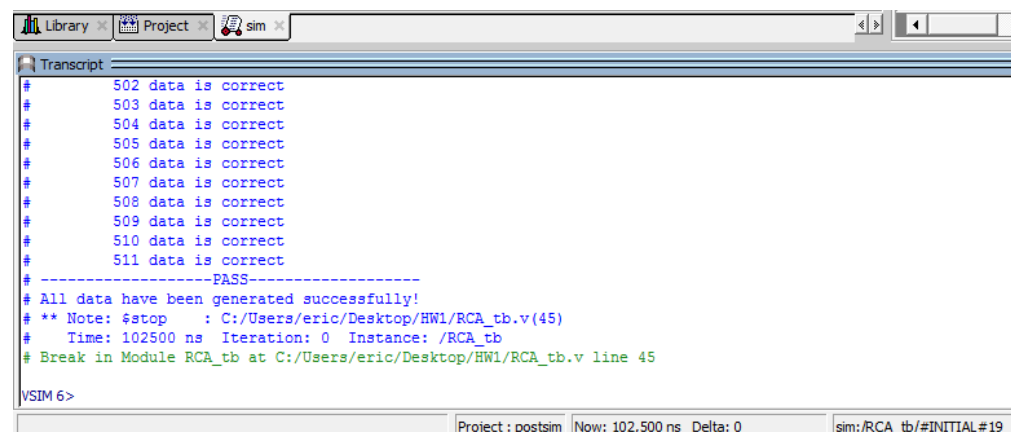
The status bar at the bottom indicates: 0 ps to 107625 ns, Project: presim, Now: 102,500 ns Delta: 0, sim:/RCA\_tb/#INITIAL#19.

## Gate-Level Simulation:

### Synthesis

Flow Summary	
Flow Status	Successful - Sun Nov 05 13:40:10 2017
Quartus Prime Version	16.0.2 Build 222 07/20/2016 SJ Standard Edition
Revision Name	RCA
Top-level Entity Name	RCA
Family	Cyclone V
Device	5CGTFD9E5F35C7
Timing Models	Final
Logic utilization (in ALMs)	4 / 113,560 ( < 1 % )
Total registers	0
Total pins	14 / 616 ( 2 % )
Total virtual pins	0
Total block memory bits	0 / 12,492,800 ( 0 % )
Total DSP Blocks	0 / 342 ( 0 % )
Total HSSI RX PCSs	0 / 12 ( 0 % )
Total HSSI PMA RX Deserializers	0 / 12 ( 0 % )
Total HSSI TX PCSs	0 / 12 ( 0 % )
Total HSSI PMA TX Serializers	0 / 12 ( 0 % )
Total PLLs	0 / 20 ( 0 % )
Total DLLs	0 / 4 ( 0 % )

### Simulation



The screenshot shows the 'Transcript' window of the Quartus Prime IDE. The transcript contains the following text:

```
# 502 data is correct
# 503 data is correct
# 504 data is correct
# 505 data is correct
# 506 data is correct
# 507 data is correct
# 508 data is correct
# 509 data is correct
# 510 data is correct
# 511 data is correct
# -----PASS-----
# All data have been generated successfully!
# ** Note: $stop : C:/Users/eric/Desktop/HW1/RCA_tb.v(45)
# Time: 102500 ns Iteration: 0 Instance: /RCA_tb
# Break in Module RCA_tb at C:/Users/eric/Desktop/HW1/RCA_tb.v line 45
VSIM 6>
```

The status bar at the bottom indicates: Project: postsim, Now: 102,500 ns Delta: 0, sim:/RCA\_tb/#INITIAL#19.

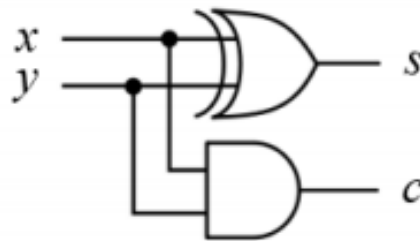
## Half-Adder:

先用 xor 接 x,y output s 達到 sum 的效果

用 and 將 x,y output c 達到 carry 的效果

藉此建構出一個 half-Adder

```
1  `timescale 10ns / 1ps
2  module HA(s, c, x, y);
3  input x, y;
4  output s, c;
5
6  xor (s,x,y);
7  and (c,x,y);
8
9  endmodule
10
```



## Full adder:

使用兩個 half-Adder 及一個 or 做出一個 full adder

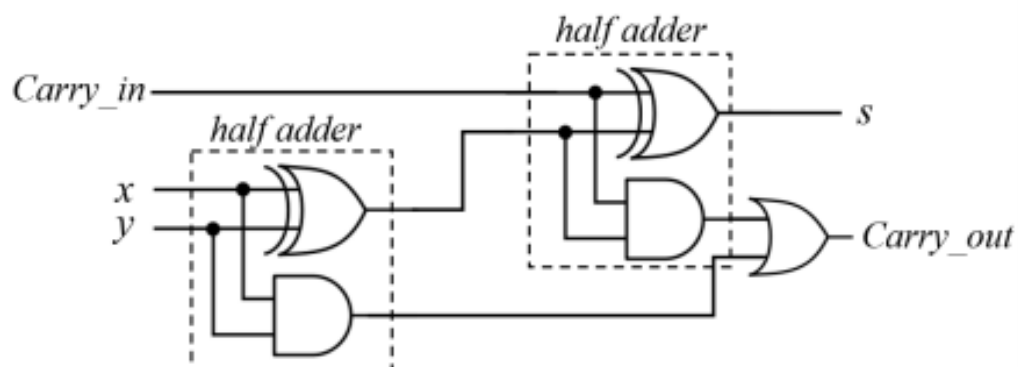
將 x,y 接到 M1 half adder 的 input , output 出 s1 和 c1

將 Carry\_in 和 M1 half adder 的 s1 output 接到 M2 half adder 的 input

output 出 s(即此 full adder 的 sum)和 c2

最後把 c1,c2 接上 or gate , output 即為 full adder 的 Carry\_out

```
1  `timescale 10ns / 1ps
2
3  module FA(s, c_out, x, y, c_in);
4  input x, y, c_in;
5  output s, c_out;
6  wire s1, c1, c2;
7
8  HA M1(s1,c1,x,y);
9  HA M2(s,c2,c_in,s1);
10 or (c_out,c1,c2);
11
12 endmodule
```



## Ripple carry adder:

使用 4 個 full Adder 依照電路圖接線組合出 ripple carry adder

FA1 input x,y 的第 0 個 bit 及 Carry\_in

output 出 sum 的第 0 個 bit 並把進位 c1 給下一個 full Adder FA2

FA2 input x,y 的第 1 個 bit 及上一個 full Adder 的進位 c1

output 出 sum 的第 1 個 bit 並把進位給下一個 full Adder FA3

依此類推使用 4 個 full Adder，以達到 ripple carry adder 的效果

```
1  `timescale 10ns / 1ps
2
3  module RCA(s, cout, x, y, c0);
4  input  [3:0] x, y;
5  output [3:0] s;
6  input  c0;
7  output cout;
8  wire c1, c2, c3;
9
10 FA FA1(s[0],c1,x[0],y[0],c0);
11 FA FA2(s[1],c2,x[1],y[1],c1);
12 FA FA3(s[2],c3,x[2],y[2],c2);
13 FA FA4(s[3],cout,x[3],y[3],c3);
14
15 endmodule
16
```

