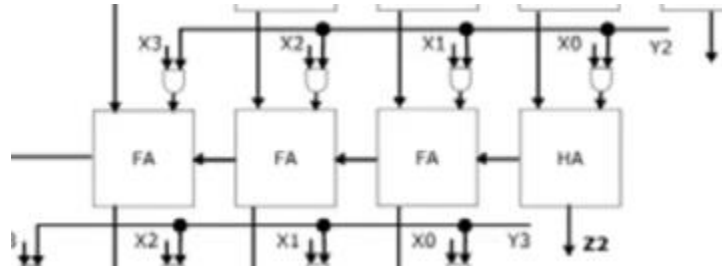


Bloque celda unitaria

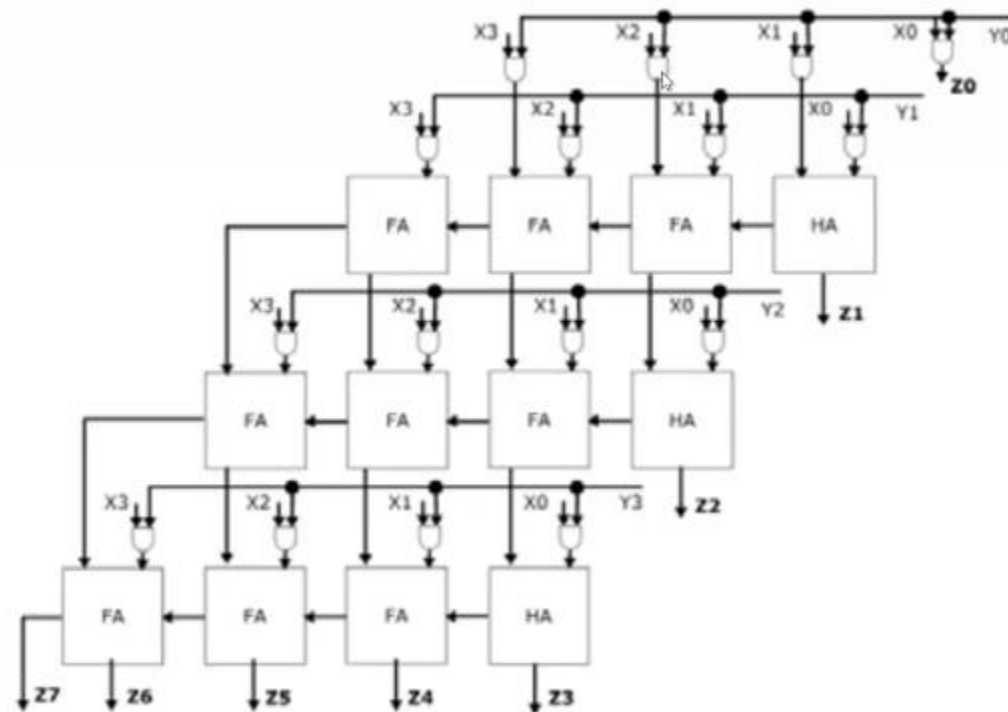
Primeramente, se implentó un generate para generar esta celda, la cual tiene primeramente un half adder, y posteriormente va generando los bloques full adder en horizontal.



```
wire [N_BITS-1 : 0]c_wire;
half_adder
half_adder_1
(
    .a(y & x[0]),
    .b(b[0]),
    .c(c_wire[0]),
    .s(s[0])
);
genvar i;
generate
for(i=1; i< N_BITS; i = i + 1)begin: gen_adders_cell
    full_adder
    full_adder_i
    (
        .a(x[i] & y),
        .b(b[i]),
        .cin(c_wire[i-1]),
        .cout(c_wire[i]),
        .s(s[i])
    );
end
endgenerate
assign s[N_BITS] = c_wire[N_BITS-1];
```

Bloque Top level

Primeramente, se instancia una celda como la anterior para poder colocar las compuertas and a las entradas, posteriormente se utiliza otro generate para poder crear los bloques de celdas como la anterior, estas son las que se generan de forma horizontal.



```

wire [N_BITS*N_BITS-2 : 0]s1_wire;

adders_cell
adders_cell_1
(
    .x(x),
    .b({1'b0, x[N_BITS-1 : 1] & {N_BITS-1{y[0]}}}),
    .y(y[1]),
    .s(s1_wire[N_BITS:0])
);

genvar i;
generate
for(i=1; i< N_BITS-1; i = i + 1)begin: gen_adders_cell
    adders_cell
    adders_cell_i
    (
        .x(x),
        .b(s1_wire[((N_BITS+1)*(i)-1) : ((N_BITS+1)*(i-1)+1)]),
        .y(y[i+1]),
        .s(s1_wire[((N_BITS+1)*(i+1)-1) : ((N_BITS+1)*(i)) ])
    );
end
endgenerate

genvar Asig;
generate
for(Asig=1; Asig< N_BITS-1; Asig = Asig + 1) begin: Asig_values
    assign s[Asig+1] = s1_wire[(N_BITS+1)*Asig];
end
endgenerate

assign s[0] = x[0] & y[0];
assign s[1] = s1_wire[0];
assign s[2*N_BITS-1:N_BITS] = s1_wire[N_BITS*N_BITS-2:(N_BITS*N_BITS-N_BITS)-1];

endmodule

```

Resultado

Se puede observar que el resultado satisface las multiplicaciones formadas en el test bench.

```
initial begin
    #0 x_reg = 8;
    #0 y_reg = 5;
    #10 x_reg = 6;
    #0 y_reg = 12;
    #10 x_reg = 15;
    #0 y_reg = 15;
    #10 x_reg = 12;
    #0 y_reg = 11;
end
```

	Msgs				
+ /multiplier_tb/x_reg	12	8	6	15	12
+ /multiplier_tb/y_reg	11	5	12	15	11
+ /multiplier_tb/s_reg	132	40	72	225	132

Conclusión

El hecho de separar en dos códigos el generate de forma horizontal y el vertical, facilitó más la realización del código obteniendo un resultado satisfactorio. El mayor problema se presentó a la hora de asignar las salidas s, ya que no se podían incluir en el mismo generate, la solución fue crear un generate nuevo en el top level para poder realizar las asignaciones.