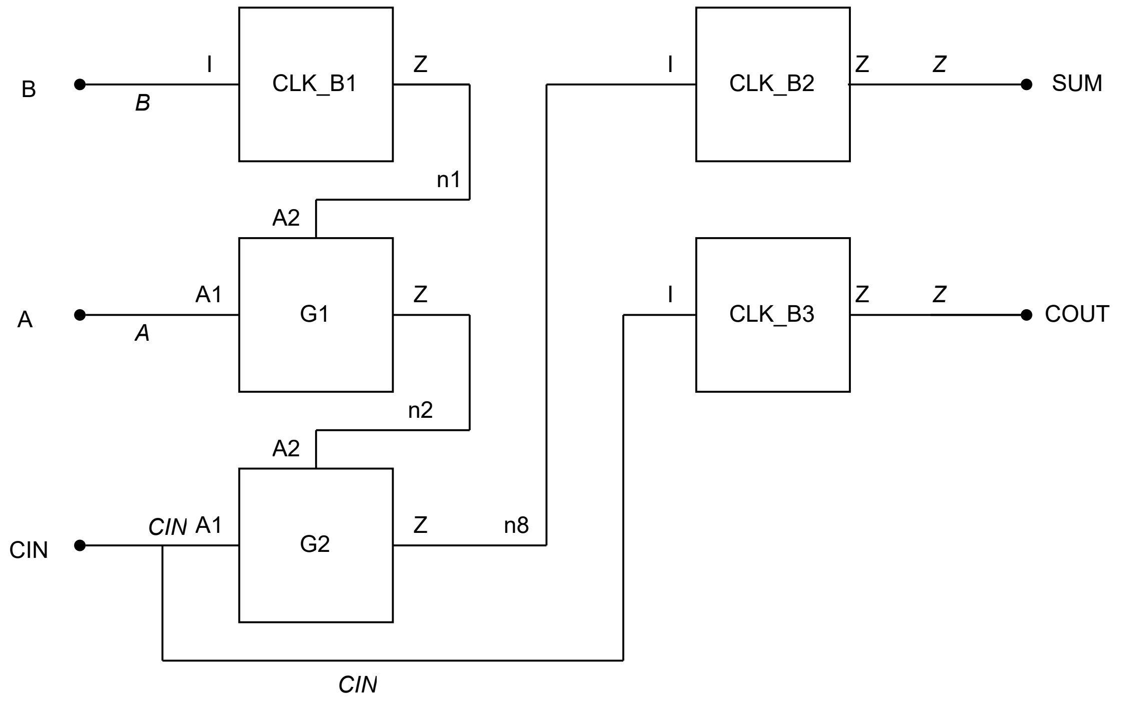
FullAdder

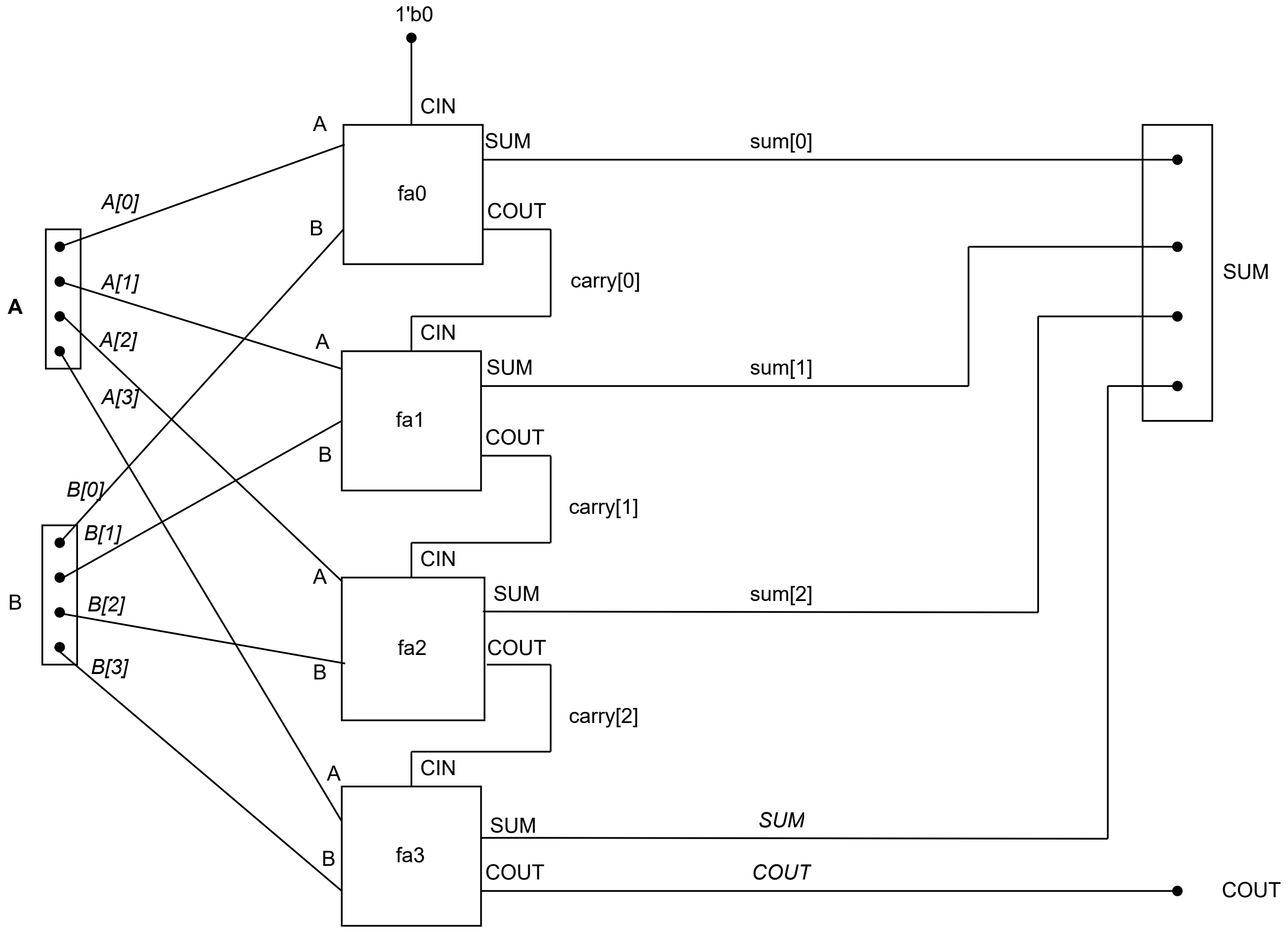
```
module FullAdder (
    input A,
    input B,
    input CIN,
    output SUM,
    output COUT);

    wire n1,n2,n8;

    BUFFSGD3BWP30P140HVT CLK_B1 (.I(B), .Z(n1) );

    XOR2SGD0BWP30P140 G1 (.A1(A), .A2(n1), .Z(n2));
    XOR2SGD0BWP30P140 G2 (.A1(CIN), .A2(n2), .Z(n8));

    BUFFSGD3BWP30P140HVT CLK_B2 (.I(n8), .Z(SUM) );

    BUFFSGD3BWP30P140HVT CLK_B3 (.I(CIN), .Z(COUT) );


endmodule
```



1. **Instance Ref_names** : CLK_B1, CLK_B2, CLK_B3, G1, G2
2. **Ports** : A, B, CIN, SUM, COUT
3. **Pins** : (I,Z) , (I,Z) , (I,Z) , (A1, A2, Z) , (A1, A2, Z)
4. **Nets** : n1, n2, n8
5. **Port derived nets** :
    1. B -> I *CLK_B1 ,* **B**
    2. A -> A1 *G1 ,* **A**
    3. CIN -> A1 *G2 ,* **CIN**
    4. CIN -> I *CLK_B3* **CIN**
    5. *Z* -> SUM *CLK_B2* **Z**
    6. Z -> COUT *CLK_B3* **Z**

FourbitAdder

```
module FourBitAdder (
    input [3:0] A,
    input [3:0] B,
    output [3:0] SUM,
    output COUT);

    wire [3:0] carry;
    wire [3:0] sum;

    // First bit
    FullAdder fa0(.A(A[0]), .B(B[0]), .CIN(1'b0), .SUM(sum[0]), .COUT(carry[0]));

    // Subsequent bits
    FullAdder fa1(.A(A[1]), .B(B[1]), .CIN(carry[0]), .SUM(sum[1]),
.COUT(carry[1]));
    FullAdder fa2(.A(A[2]), .B(B[2]), .CIN(carry[1]), .SUM(sum[2]),
.COUT(carry[2]));
    FullAdder fa3(.A(A[3]), .B(B[3]), .CIN(carry[2]), .SUM(SUM), .COUT(COUT));

    assign SUM = sum;

endmodule
```



1. **Instance Ref_names** : fa0, fa1, fa2, fa3
2. **Ports** : A[3:0], B[3:0], SUM, COUT
3. **Pins** : (A,B,CIN,SUM,COUT), (A,B,CIN,SUM,COUT), (A,B,CIN,SUM,COUT), (A,B,CIN,SUM,COUT)
4. **Nets** : sum[0], sum[1], sum[2], carry[0], carry[1], carry[2]
5. **Port derived nets** :
    1. A[0] -> A *fa0* **A[0]**
    2. A[1] -> A *fa1* **A[1]**
    3. A[2] -> A *fa2* **A[2]**
    4. A[3] -> A *fa3* **A[3]**
    5. B[0] -> A *fa0* **B[0]**
    6. B[1] -> A *fa1* **B[1]**
    7. B[2] -> A *fa2* **B[2]**
    8. B[3] -> A *fa3* **B[3]**
    9. SUM -> SUM[3] *fa3* **SUM**
    10. COUT -> COUT fa3 **COUT**