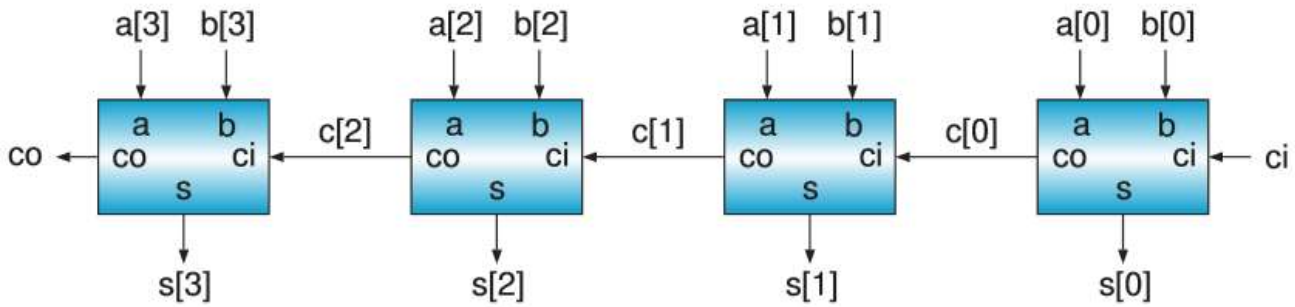


Pin planner နဲ့ FPGA pin တွေသတ်မှတ်ပြီးရင် board ထဲထည့်စမ်းလို့ရပါပြီ။

Ripple Adder



Full adder တွေကိုဆင့်ပွားပြီးတော့ ripple adder ကိုတည်ဆောက်ထားပါတယ်။ စာမျက်နှာ ၃၉ ရဲ့အောက်ဆုံးပုံမှာ ripple adder ကိုပြထားပါတယ်။ အဲဒီပုံမှာ full adder လေးလုံးကို အသုံးပြုထား တဲ့အတွက်ကြောင့် 4-bit ripple adder လို့ခေါ်ပါတယ်။

```
module add1 (a,b,ci,s,co);

input a, b, ci;

output s, co;

assign s = a ^ b ^ ci;

assign co = a & b | b & ci | ci & a;

endmodule
```

အပေါ်က code ဟာ full adder ရဲ့ code ဖြစ်ပါတယ်။ ပြောပြီးသားပါ - ripple adder ဆောက်ဖို့ full adder ကိုယူသုံးရတယ်ဆိုတာ။ အခုလည်း 4-bit ripple adder ဆောက်ဖို့ရန်အတွက် full adder ရဲ့ module

တစ်ခုဖြစ်တဲ့ add1 ဆိုတဲ့ module လေးကိုယူသုံးမှာဖြစ်ပါတယ်။ ripple adder အတွက် Verilog HDL code ကိုဆက်ရေးကြပါမယ်။

```
module rp_adder (a,b,ci,s,co);

input [3:0] a, b;

input      ci;

output [3:0] s;

output      co;

wire [2:0] c;

// add1 (a,  b,  ci,  s,  co);















add1 a0 (a[0], b[0], ci,  s[0], c[0]);

add1 a1 (a[1], b[1], c[0], s[1], c[1]);

add1 a2 (a[2], b[2], c[1], s[2], c[2]);

add1 a3 (a[3], b[3], c[2], s[3], co);

endmodule
```

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location
 a[3]	Input	PIN_C12	7	B7_N0	PIN_C12
 a[2]	Input	PIN_D12	7	B7_N0	PIN_D12
 a[1]	Input	PIN_C11	7	B7_N0	PIN_C11
 a[0]	Input	PIN_C10	7	B7_N0	PIN_C10
 b[3]	Input	PIN_A14	7	B7_N0	PIN_A14
 b[2]	Input	PIN_A13	7	B7_N0	PIN_A13
 b[1]	Input	PIN_B12	7	B7_N0	PIN_B12
 b[0]	Input	PIN_A12	7	B7_N0	PIN_A12
 ci	Input	PIN_B14	7	B7_N0	PIN_B14
 co	Output	PIN_D13	7	B7_N0	PIN_D13
 s[3]	Output	PIN_B10	7	B7_N0	PIN_B10
 s[2]	Output	PIN_A10	7	B7_N0	PIN_A10
 s[1]	Output	PIN_A9	7	B7_N0	PIN_A9
 s[0]	Output	PIN_A8	7	B7_N0	PIN_A8

Pin planner နဲ့ pin assignment ချပြီးရင် compile ပြန်လုပ်ပြီး board ထဲထည့်စမ်းပါတော့မယ်။

Verilog HDL code မှာပါတဲ့ addition algorithms က အပေါင်းအနှုတ်လက္ခဏာတွေကို ထည့်မစဉ်းစားတဲ့ unsigned အတွက်ကော လက္ခဏာထည့်စဉ်းစားတဲ့ signed တွေအတွက် 2's complement representation အတွက်ကောနှစ်ခုစလုံးအတွက်အဆင်ပြေပါတယ်။ 4-bit ကိန်းဂဏန်းနှစ်ခုကိုနမူနာအနေနဲ့ ပေါင်းပြပါမယ်။

Addition on unsigned numbers:

$$\begin{array}{r}
 0111 \quad (7) \\
 + 1011 \quad (11) \\
 \hline
 0010 \quad (2)
 \end{array}$$

Addition on signed numbers:

$$\begin{array}{r}
 0111 \quad (+7) \\
 + 1011 \quad (-5) \\
 \hline
 0010 \quad (+2)
 \end{array}$$

ဘယ်ဘက်မှာ unsigned ကိန်းဂဏန်းနှစ်ခုဖြစ်တဲ့ 7 နဲ့ 11 ကိုပေါင်းထားပါတယ်။ ပေါင်းလဒ်က 18 ဖြစ်ရ မှာပါ။ overflow ဖြစ်သွားတဲ့အတွက်ကြောင့် 4-bit code မဟုတ်နိုင်တော့ပါဘူး။ ညာဘက်က ဥပမာကို ဆက်ကြည့်ပါမယ်။ (+7) နဲ့ (-5) နှစ်ခုပေါင်းရလဒ်က +2 ရပါမယ်။ ရလဒ်မှန်ကန်တာကိုမြင်တွေ့ရမှာဖြစ်ပါတယ်။