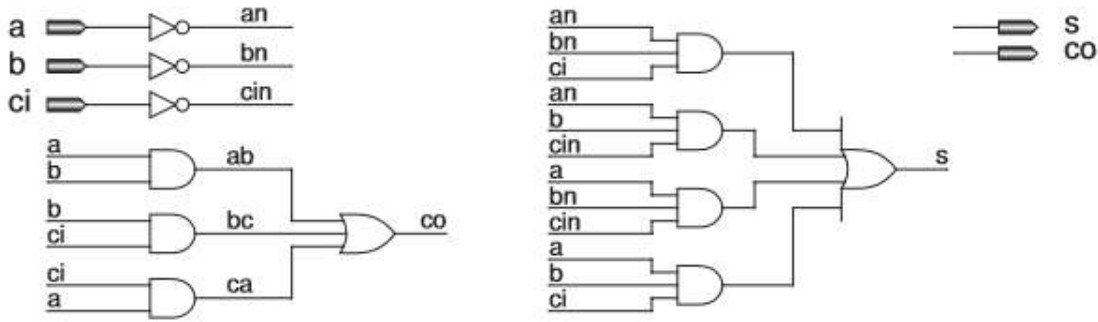


Full Adder

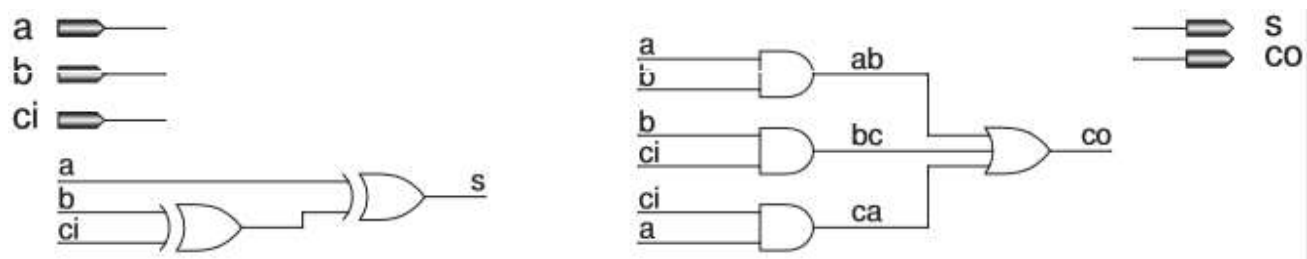


Full adder ရဲ့ တည်ဆောက်ပုံကတော့ စာမျက်နှာ ၃၆ ကအတိုင်းဖြစ်ပါတယ်။ adder ဆိုတော့ ပေါင်းမှာ ပေါ့ဗျာ။ ပေါင်းတဲ့နေရာမှာ carry out နဲ့ sum ဆိုပြီးတော့ရှိတယ်။ 1 နဲ့ 1 ပေါင်းရင် 10 ရပါတယ်။ 10 ဆိုတဲ့နေရာမှာ 1 က carry out (co) ဖြစ်ပြီးတော့ 0 က sum(s) ဖြစ်ပါတယ်။ Carry Out ကိုဘယ်လို တွက်မလဲဆိုတော့ အပေါ်က diagram ကိုသာကြည့်လိုက်ပါ။ a နဲ့ b ကို AND ခံပြီးတော့ OR input ထဲ ထည့်တယ်။ b နဲ့ carry in(ci) ကို AND ခံပြီးတော့ OR input ထဲထည့်တယ်။ ci နဲ့ a ကို AND ခံပြီး တော့ OR input ထဲထည့်တယ်။ ဒီလိုနဲ့ carry out ကို ထုတ်ပါတယ်။ sum မှာ diagram အတိုင်းပါဘဲ။

ကျွန်တော်တို့ Verilog HDL နဲ့ program ရေးပြီး FPGA ထဲထည့်စမ်းတဲ့အခါမှာ digital logic တွေရဲ့ diagram တွေကိုကြည့်ပြီးတော့ ရေးရတာ ပိုပြီးတော့အဆင်ပြေပါတယ်။ Truth table တွေကိုကြည့်ပြီးတော့ လွယ်ကူစွာသဘောပေါက်နိုင်မှာဖြစ်ပါတယ်။ Full adder ရဲ့ truth table ကိုကြည့် လိုက်ရအောင်လားဗျာ။

Input signal			Output signal		Comment
a	b	ci	co	s	
0	0	0	0	0	$0 + 0 + 0 = 0\ 0$
0	0	1	0	1	$0 + 0 + 1 = 0\ 1$
0	1	0	0	1	$0 + 1 + 0 = 0\ 1$
0	1	1	1	0	$0 + 1 + 1 = 1\ 0$
1	0	0	0	1	$1 + 0 + 0 = 0\ 1$
1	0	1	1	0	$1 + 0 + 1 = 1\ 0$
1	1	0	1	0	$1 + 1 + 0 = 1\ 0$
1	1	1	1	1	$1 + 1 + 1 = 1\ 1$

ရှင်းရှင်းလေးပါ။ နောက်က comment ကိုကြည့်ပြီးနားလည်မယ်ထင်ပါတယ်။ Full adder ကို XOR gate တွေသုံးပြီးတော့တည်ဆောက်လို့လည်းရပါတယ်။

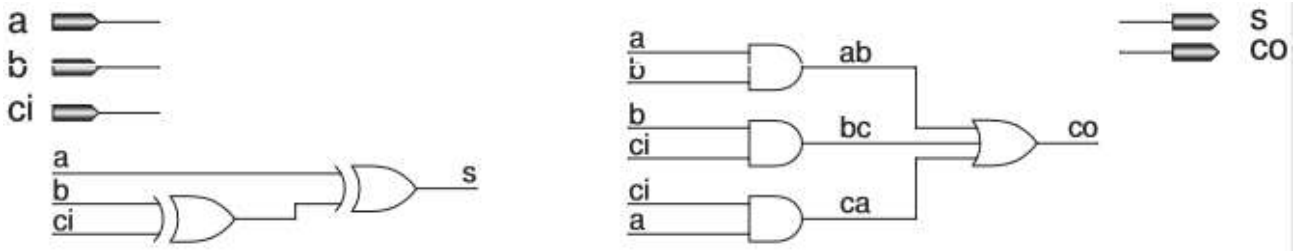


အပေါ်က ပုံကတော့ full adder ကို XOR တွေသုံးပြီးတော့တည်ဆောက်ထားတာဖြစ်ပါတယ်။ ကောင်းပါပြီ။ schematic တွေကြည့်ပြီးတော့ Verilog HDL နဲ့ရေးပါတော့မယ်ခင်ဗျာ။

စာမျက်နှာ ၃၆ ရဲ့အောက်ဆုံးပုံနဲ့အလုပ်လုပ်ဆောင်တူညီတဲ့ Verilog HDL ကိုရေးပါမယ်။ ရေးမယ့်ပုံစံက structural style နဲ့ရေးမှာဖြစ်ပါတယ်။

```
module fa_structural(a,b,ci,s,co):  
  
    input a, b, ci;  
  
    output s, co;  
  
    wire ab, bc, ca;  
  
    xor i1(s, a, b, ci);  
  
    and i2(ab, a, b);  
  
    and i3(bc, b, ci);  
  
    and i4(ca, ci, a);  
  
    or i5(co, ab, bc, ca);
```

endmodule



Code နဲ့ပုံကိုနှိုင်းယှဉ်ကြည့်ပါ။ ပုံမှာ carry out(co) ကိုတွက်ဖို့ရန်အတွက် a နဲ့ b ကို AND လုပ်ထားပြီး တော့ ab ထွက်ထားပါတယ်။ Gate Output ဖြစ်တဲ့အတွက် wire လို့ခေါ်ပါတယ်။ ထိုနည်းတူ b နဲ့ c ကိုလည်း AND လုပ်ထားပြီးတော့ bc ထွက်ထားပါတယ်။ Gate Output ဖြစ်တဲ့အတွက် wire လို့ခေါ်ပါတယ်။ c နဲ့ a ကိုလည်း AND လုပ်ထားပြီးတော့ ca ထွက်ထားပါတယ်။ Gate Output ဖြစ်တဲ့အတွက် wire လို့ခေါ်ပါတယ်။ `or i5(co, ab, bc, ca);` ဆိုတဲ့ code နဲ့ wire သုံးခုဖြစ်တဲ့ ab, bc, ca တို့ကို co(carry out) ထဲကိုထည့်လိုက်တာပါ။ Code ကတော့ရှင်းရှင်းလေးပါ။ နားလည်ကြလိမ့်မယ်ထင်ပါတယ်။

ခုဆက်ရေးမှာကတော့ full adder ကိုဘဲ dataflow style နဲ့ဖြစ်ပါတယ်။

```
module fa_dataflow(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
```

အထက်ပါ code ရဲ့အလုပ်လုပ်ပုံကို schematic ကြည့်ခြင်းဖြင့်နားလည်နိုင်ပါတယ်။ ခုဆက်ရေး

မှာကတော့ full adder ကိုဘဲ behavioral style နဲ့ဖြစ်ပါတယ်။

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

```
input a, b, ci;
```






```
output s, co;
```

```
assign {co,s} = a + b + ci;
```

```
endmodule
```

Named: *

Edit:

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
 a	Input	PIN_C10	7	B7_N0	PIN_C10	2.5 V
 b	Input	PIN_C11	7	B7_N0	PIN_C11	2.5 V
 ci	Input	PIN_D12	7	B7_N0	PIN_D12	2.5 V
 co	Output	PIN_A9	7	B7_N0	PIN_A9	2.5 V
 s	Output	PIN_A8	7	B7_N0	PIN_A8	2.5 V

Pin planner နဲ့ FPGA pin တွေသတ်မှတ်ပြီးရင် board ထဲထည့်စမ်းလို့ရပါပြီ။