Seung Ki Lee

CSE 3381 <Digital Logic Design N14 1172>

Post-Lab 06

1. Write Up & Prelab Code

   In the prelab I've practiced how to effectively convert a logic diagram to Verilog assign statements. In the process I discovered that even without the strict grouping (ex: (a^b)^cin), the simulation still worked fine. However in spirit of making the implicit explicit I followed the diagram exactly. In fixing 4-bit adder code from the lab book, I was able to better understand the bit adders for the demo and top level design was less abstract then k-maps.

   Prelab-Code: 3297

2. Source Code (fulladd.v, fulladd2.v, adder4.v, adder4_top.v)

Fulladd.v

```
module fulladd (
input wire a,
input wire b,
input wire cin,
output cout,
output wire s
) ;
assign s = (a^ b) ^ cin;
assign cout = (a&b) | ((a^b) & cin);
endmodule
```

Fulladd_top.v

```
module fulladd_top (
input wire [2:0] sw,
input wire [0:0] btn,
```

```
output wire [1:0] ld
);
fulladd U1
(
.a(sw[0]),
.b(sw[1]),
.cin(btn[0]),
.s(ld[0]),
.cout(ld[1])
);
endmodule
```

adder4.v

```
module adder4 (
input wire cin,
input wire [3:0] a,
input wire [3:0] b,
output wire cout,
output wire [3:0] s
);

wire c1;
wire c2;
wire c3;

fulladd U1
( .a(a[3]),
.b(b[3]),
.cin(c3),
.cout(cout),
.s(s[3])
```

```
);
fulladd U2
( .a(a[2]),
.b(b[2]),
.cin(c2),
.cout(c3),
.s(s[2])
);
fulladd U3
( .a(a[1]),
.b(b[1]),
.cin(c1),
.cout(c2),
.s(s[1])
);
fulladd U4
( .a(a[0]),
.b(b[0]),
.cin(cin),
.cout(c1),
.s(s[0])
);
endmodule
```

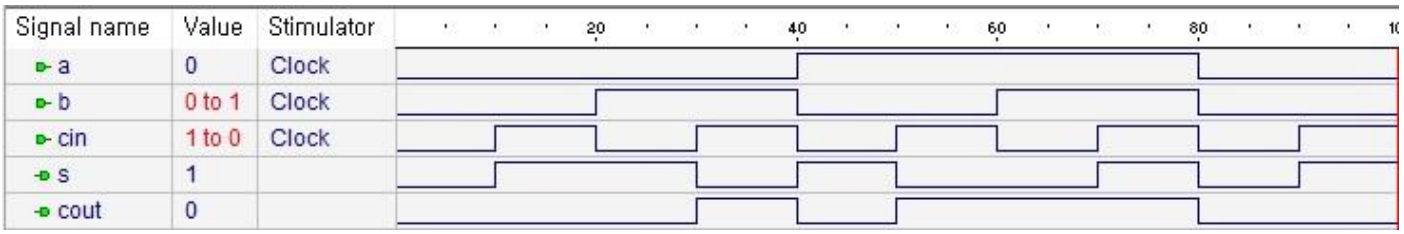
adder4_top.v

```
module adder4_top (
input wire [7:0] sw,
input wire [0:0] btn,
output wire [4:0] ld
);
```

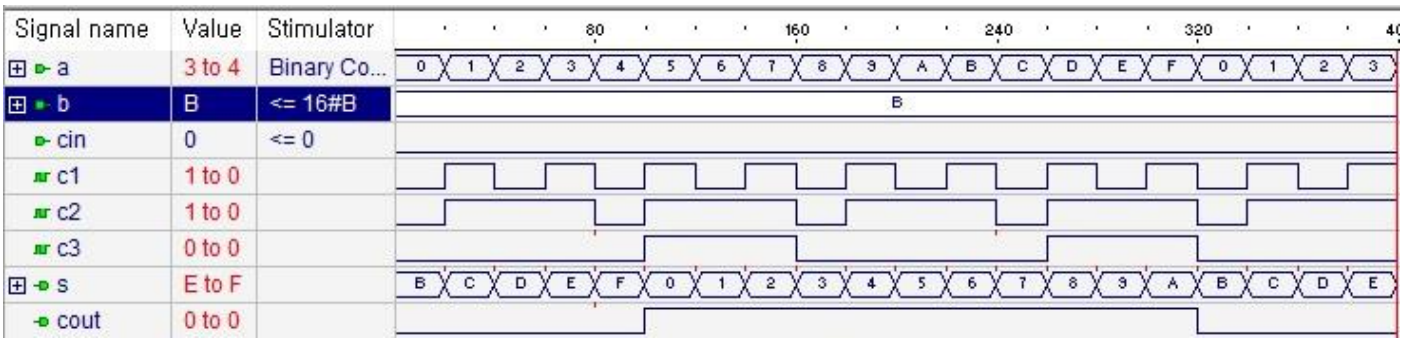
```
wire [3:0] sum;
adder4 U1
(
.cin(btn[0]),
.a(sw[7:4]),
.b(sw[3:0]),
.cout(ld[4]),
.s(ld[3:0])
);
endmodule
```

3. Simulation (Example 12, Example 13)

Example 12



Example 13



4. Demo Code

Demo12-Code: 6879

Demo13-Code: 1765