

THE ADDER REVISITED

In EE/CPE 219 you studied the full adder. You probably designed one using NAND gates or other gates. You might have even programmed one into a PAL. Here we shall look at the adder in a totally new light - as designed today.

The first look at it will be in the form of a VHDL file.

```
entity full_adder is
  port(ci: in bit;
        a,b: in bit;
        sum,co: out bit);
end full_adder;
architecture full_adder of full_adder is
begin
  sum< = a xor b xor ci;
  co< = ((a or b) and ci) or (a and b);
end;
```

We will consider the details of this file later. We are just looking at an overview for now. This file describes a device we have named “full_adder” (not very imaginative). We have indicated it has three inputs (a, b, and ci) and two outputs (sum and co). Finally, we have given the Boolean equations for the two outputs. **This is not the only way we will be describing logic circuits!**

Using the software provided with the VHDL text, we can gain a lot of information regarding this. We may look at the report file. Excerpts from it are shown in the boxes below

```
Created 5 PLD nodes.
Input File(s): adder. pla
Device       : C22V10
Package      : PALC22V10- 20PC/PI
ReportFile   : adder. rpt
DESIGN EQUATIONS (10: 02: 47)
  sum =  ci * /a * /b + /ci * a * /b + /ci * /a * b + ci * a * b
  co =   a * b + ci * b + ci * a
```



```

entity adder2 is
  port (ci:          in bit;
        a1,a0:      in bit;
        b1,b0:      in bit;
        s1,s0:      out bit;
        co:          out bit);
end adder2;
architecture adder2 of adder2 is
  component full_adder
    port (ci:  in bit;
          a,b: in bit;
          sum,co:out bit);
  end component;
  signal c1: bit;
begin
  u1: full_adder port map(ci,a0,b0,s0,c1);
  u2: full_adder port map(c1,a1,b1,s1,co);
end;

```

This is a 2-bit adder (we could have made it 4-bit or 8-bit but let's stay simple! It uses two instances of the adder we have already described. Again, we shall deal with the VHDL syntax later.

PLD Compiler Software: PLA2JED.EXE 07/APR/96 [v3.20F] 4 IR x61
 DESIGN EQUATIONS (12:06:22)

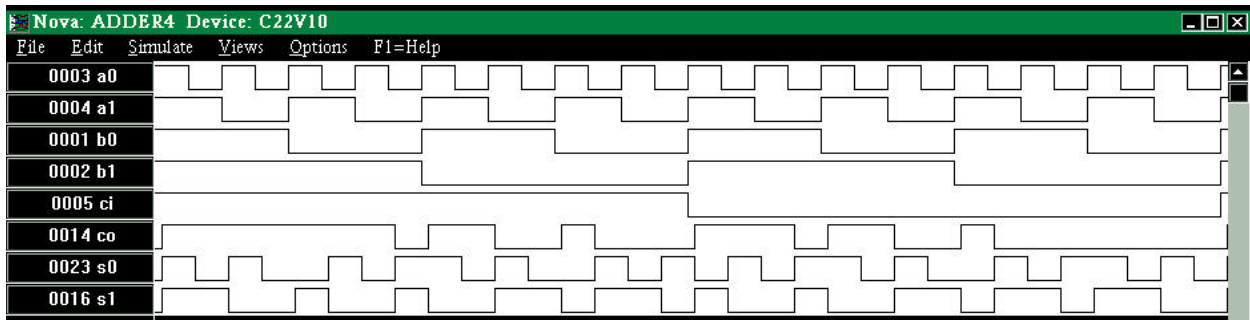
$$\begin{aligned}
 s1 &= a1 * /a0 * /b1 * /b0 + /ci * a1 * /b1 * /b0 + /a1 * /a0 * b1 * /b0 + \\
 & \quad /ci * /a1 * b1 * /b0 + /a1 * a0 * /b1 * b0 + ci * /a1 * /b1 * b0 + \\
 & \quad a1 * a0 * b1 * b0 + ci * a1 * b1 * b0 + /ci * a1 * /a0 * /b1 \\
 & \quad ci * /a1 * a0 * /b1 + /ci * /a1 * /a0 * b1 + ci * a1 * a0 * b1 \\
 s0 &= ci * /a0 * /b0 + /ci * a0 * /b0 + /ci * /a0 * b0 + ci * a0 * b0 \\
 co &= a0 * b1 * b0 + ci * b1 * b0 + a1 * a0 * b0 + ci * a1 * b0 \\
 & \quad ci * a0 * b1 + ci * a1 * a0 + a1 * b1
 \end{aligned}$$

Completed Successfully

C22V10

b0 =	1	24	* not used
b1 =	2	23	= s0
a0 =	3	22	* not used
a1 =	4	21	* not used
ci =	5	20	* not used
not used	* 6	19	* not used
not used	* 7	18	* not used
not used	* 8	17	* not used
not used	* 9	16	= s1
not used	* 10	15	* not used
not used	* 11	14	= co
not used	* 12	13	* not used

The simulation of this is shown below.



Note that this adder is not implemented as two single adders with ripple carry. Why not? This is something we will need to study.