

New Alliance and Foundation CPLD Fitter Option

The new Alliance Series and Foundation Series software contain a CPLD fitter option called “use advanced fitting.” This new feature allows the software to group equations that have the same input signals into the same function block. You would typically use this option

```
*****Function Block Resource Summary*****
```

Function Block	# of Macro-cells	FB Inputs Used	Signals Used	Total Pt Used	O/I/O Req	I/O Avail
FB1	10	36	36	35	8/0	10
FB2	14	36	40	66	4/0	14
FB3	12	36	37	87	9/0	9
FB4	12	36	36	76	8/0	10
FB5	11	34	34	45	5/0	13
FB6	9	36	36	34	5/0	13

Figure 1: *Function Block Resource Summary from Fitter Report.*

on large designs that cannot fit because of function block input restrictions.

If your design is unable to fit into a particular device, take a look at the top of the report (.RPT) file. Designs which appear to have plenty of I/O pins, product terms, and macrocells when mapped into an XC9500 device may not fit because of the 36-signal input limit to each function block.

In the example in **Figure 1**, we see that almost all of the FB inputs are being used. However, the total number of product terms and macrocells in each function block are not fully utilized. This would be a perfect candidate for trying the “advanced fitting” algorithm.

What exactly does this new algorithm do differently? There are 36 inputs to a function block in the XC9500 CPLD architecture. The CPLD fitter software will default to fitting designs based on a pin-locking algorithm. This pin-locking algorithm tends to spread equations throughout the CPLD, ensuring room for growth and change for all of the equations.

This spreading function, however, tends to use up the number of function block inputs very quickly. By selecting the “advanced fitting” algorithm, the software weights common signal usage higher and tends to group equations with the same signals into the same function block. As a result, it provides a denser fit for larger designs by freeing the otherwise used inputs from other function blocks. ♦