XAPP376 (v1.0) January 3, 2002

Understanding the CoolRunner-II Logic Engine

Summary

CoolRunner™-II is the Xilinx CPLD Family that raises the standard for Complex Programmable Logic Devices. CoolRunner-II delivers unmatched performance with the industry's lowest power at highly competitive price points in an aggressive spectrum of packages. This application note details how CoolRunner-II CPLDs create logic within their CMOS fabric. In all likelihood, you will never need to know these details as the design software will automatically complete your design giving highest speed and lowest power with very little user direction. In the event that you would like to understand the inside details of how CoolRunner-II does its magic, this application note should help serve that need. For general CoolRunner-II information, also refer to the CoolRunner-II Family Data Sheet and individual device data sheets.

High Level Architecture

Figure 1 shows the overall CoolRunner-II architecture. It resembles most CPLDs at this level. I/O pins surround the chip which has multiple programmable Function Blocks (FBs) that are connectable to each other and the outside world with the Advanced Interconnect Matrix (AIM). The pins and AIM deliver input variables to the FBs, which produce logic results that ultimately go to the outside world.

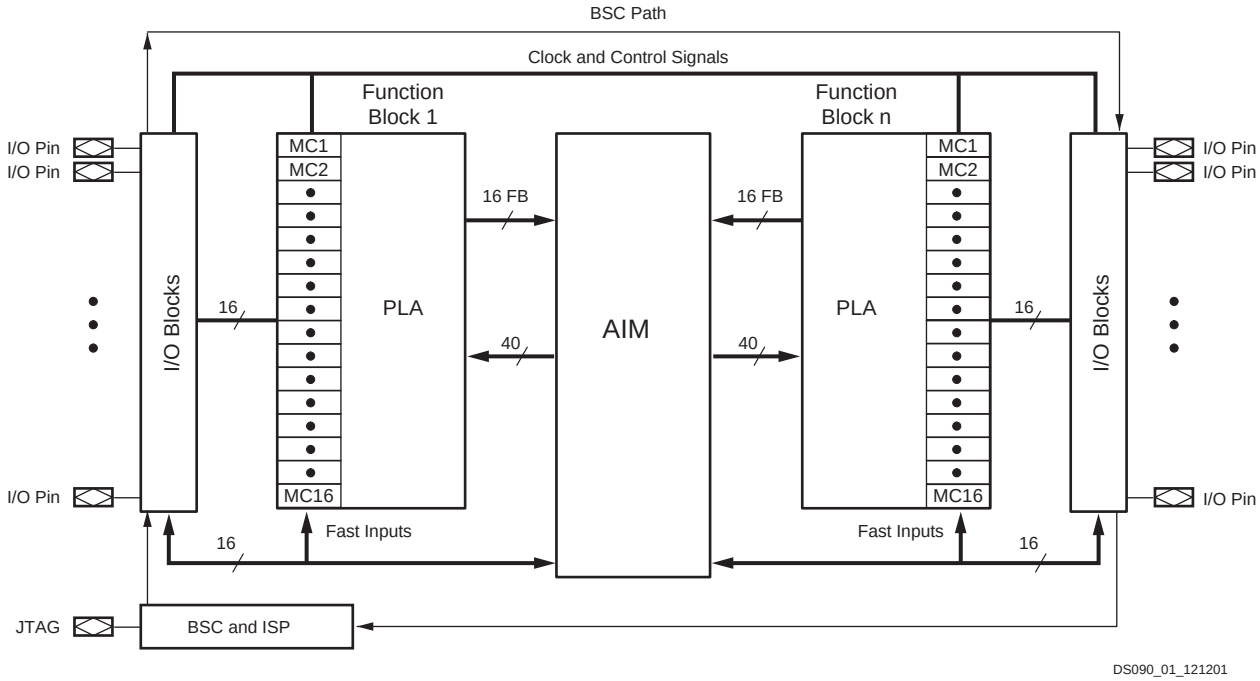

Figure 1: **CoolRunner-II Architecture**

Function Block Architecture

The CoolRunner-II Function Block (Figure 2) accepts 40 input signals from the AIM and directly attaches them to a Programmable Logic Array which operates over the whole FB. Sixteen macrocells take the PLA logic combinations and assign signals to various components of the macrocells (flip-flops, bypass multiplexors, clocks, resets, etc.). The resulting signals are then routed back to the AIM for additional connection to other FBs or to exit the chip at the output pins. Note that Global Resources enter the FB and are made individually available to each macrocell as needed. The design software handles assignment of signal variables to the various FBs and manages internal FB connections to logic resources as well as Global Resources.

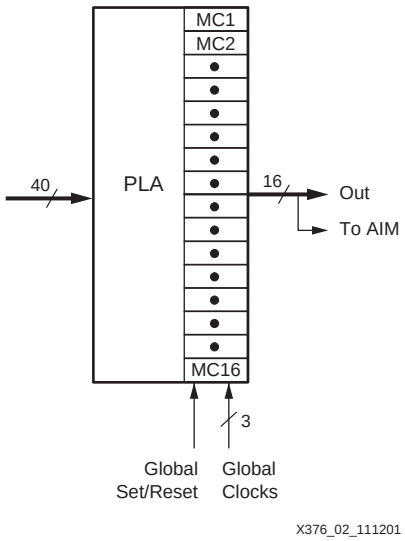

Figure 2: **CoolRunner-II Function Block**

Macrocell

Stepping inside the FB we see the 16 macrocells. Unlike other CPLD architectures, the logic boundaries are a little fuzzy within the FB. It is difficult to isolate specific logic resources that are dedicated to individual macrocells due to the connection capability of the Programmable Logic Array (PLA). Figure 3 details the CoolRunner-II macrocell, which will be referred to frequently in this discussion. Note in general, that the PLA is located on the left side of the drawing. Moving to the right, we see a flip-flop which has a clock enable and multitude of trapezoidal multiplexors. The muxes do not explicitly show their select lines as the bits doing the selection are held in the configuration memory and are changed by programming the internal nonvolatile memory cells. The flop can be configured as a D, T or transparent latch. The clocking can occur on either edge, and can source individually or through the DualEDGE mode. The flop set and reset circuits are very similar. The set can select from no set condition (GND), a product term set signal (PTA), a Control Term Set (CTS) or the Global Set/Reset signal (GSR). Global signals pass through the entire chip. Control Term signals are common within an FB. Product Term signals are local to a specific macrocell within a FB.

The approach we will take is to dissect and connect sections of Figure 3 to illustrate how logic is created within the FB. Figure 3 will be a common reference point.

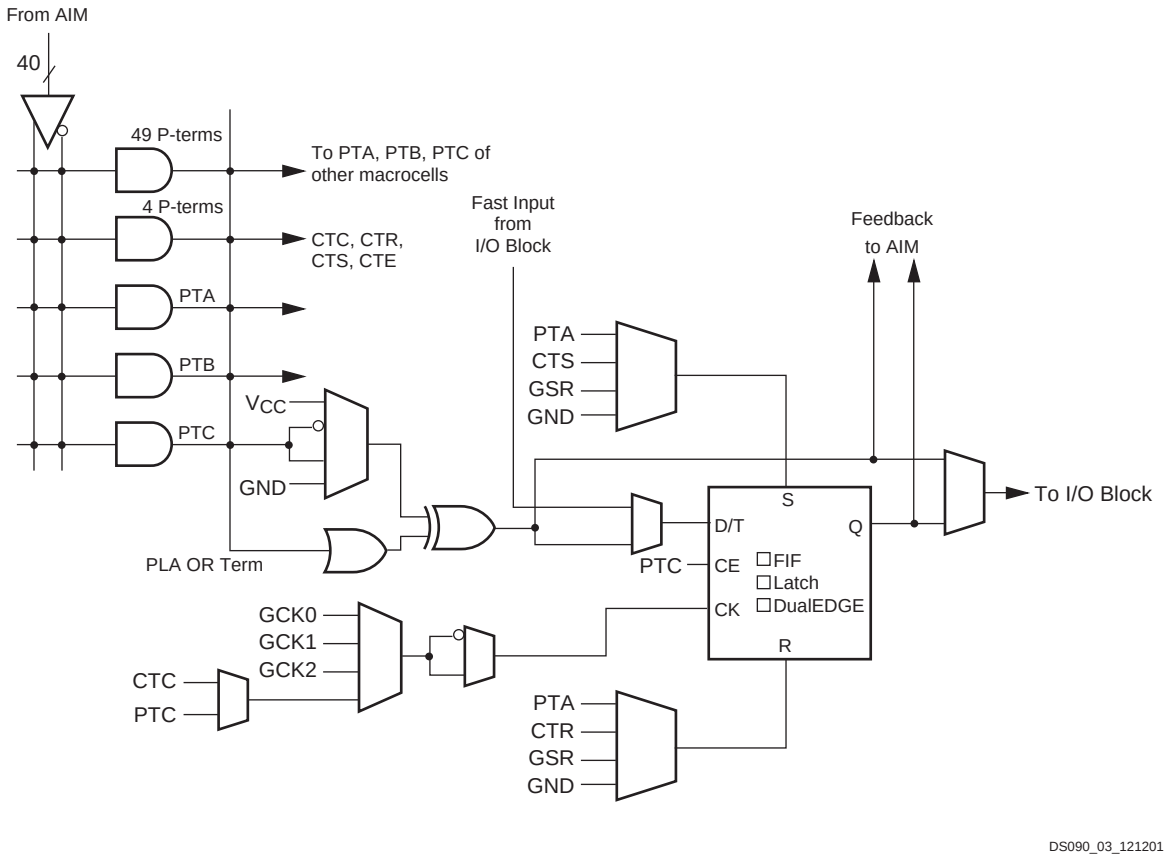

Figure 3: **CoolRunner-II Macrocell**

How a Programmable Logic Array Works

Before looking at the CoolRunner-II PLA, lets discuss the general operation of a PLA. Figure 4 shows one. In this case, we show 40 input signals and their corresponding complementing buffers. The signals and complements enter a programmable AND array.

The programmable AND array may appear strange at first in that the AND gates appear to have only one long input. The notation here is that where vertical lines cross horizontal lines, there is a potential connection site. Unprogrammed sites are shown with an X and programmed ones have a Disk. AND gates are called their mathematical name — product terms or p-terms. CoolRunner-II has a 56 product term set of AND gates in each FB. The p-term outputs head to the right, where they enter a second programming array called the OR array.

Again, we show vertical lines crossing horizontal lines with Xs and Discs. The same idea works in the OR array. An X is a potential connection and a Disc is a connection that has been made. A key idea for understanding the value of a PLA is that of product term sharing. Note in Figure 4, the output labeled mc1 ANDs together input 1 with two in p-term 1. Subsequently, mc1 ORs in input 40. The output labeled mc2 takes the pt1 result and ORs in p-term 3. Hence, the pt1 AND result is used twice. There are no specifically dedicated product terms to any specific “mc” output. Just for completion, mc56 is totally unused in Figure 4.

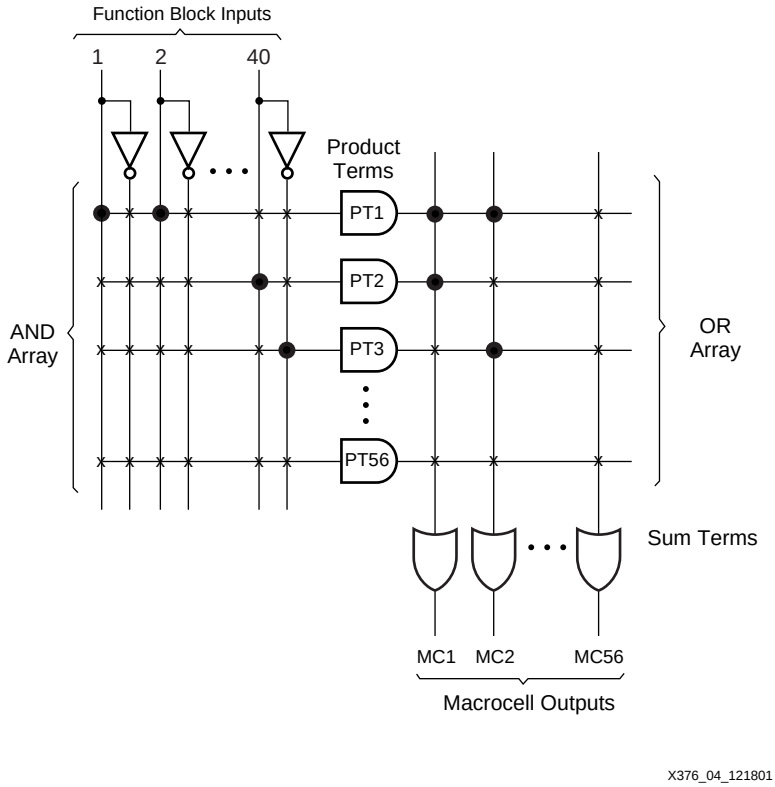

Figure 4: **Basic PLA Architecture**

How the CoolRunner-II PLA Works

Now, we will see how this general idea maps onto the CoolRunner-II FB. Figure 5 has much of the macrocell detail stripped away to focus on the PLA. Note that in this version of the diagram, we show all 40 inputs collapsed down to a single input buffer with its complement driven over a set of product terms on the left. The top AND is actually 49 identical copies of a product term. Below that are four more product terms followed by three single product terms. The four product terms correspond to the ones flagged out as four CT (Control Terms) in Figure 3. The three individual product terms were labeled PTA, PTB, and PTC in Figure 3. The total of the 49, plus 4, plus 3 results in 56 p-terms. In this diagram, we dispense with the X and Disc notation and simply park little Disc symbols at intersections that could be programmable.

The abbreviated PLA p-term outputs are presented to an OR array which attaches to a single input OR gate shown at the bottom. Again, little disks are shown to indicate connectability to the OR term consituting individual inputs to the OR gate. Within the FB, there are 15 more copies of the logic on the right hand side of Figure 5, starting at the EX-OR gate. That way, the same set of 56 product terms connects through the OR and Mux structure to the other 15 macrocell EX-OR gates. Note that the Sum of Products site is the output of the OR, and is formally the output of the PLA. The EX-OR is added in for flexibility and polarity control.

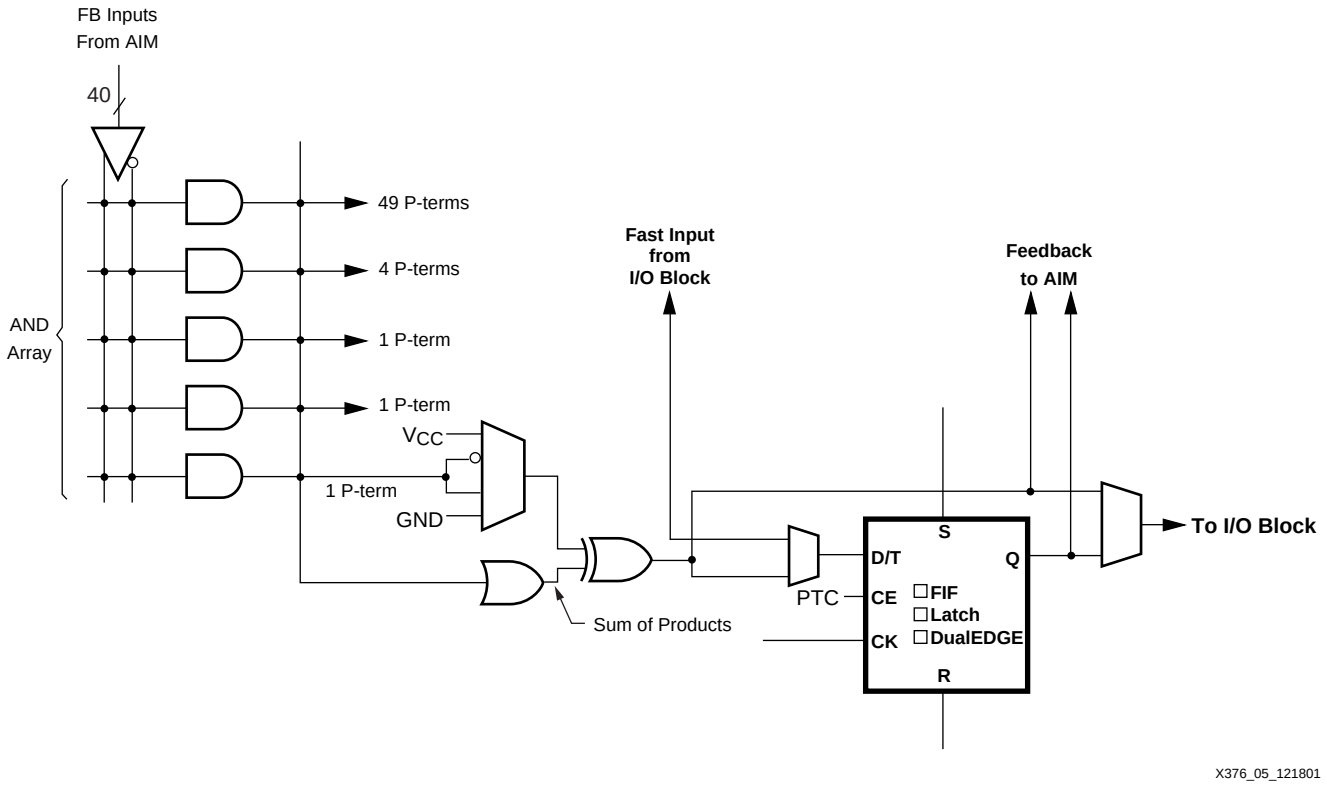

Figure 5: **Partially Stripped Macrocell Diagram**

Zooming further in, Figure 6 shows just the PLA feeding into the EX-OR circuit. One more cutaway lets us zoom into Figure 7, where focus is given on the Mux/OR/EX-OR structure. Figure 7 retains one p-term shown entering the OR, and another p-term shown entering the Mux. It should be noted that both p-terms enter the OR array, but here we focus on the logic structure produced by two p-terms working in the the Mux/OR/EX-OR structure.

The Mux driving one EX-OR leg offers a choice of V_{CC} (logic 1), GND (logic 0) and the directly attached p-term or its complement. First, consider the V_{CC} and GND options. A constant of one on one EX-OR leg makes the EX-OR output be the complement of the signal driven on the other leg — an inverter. A constant of zero on one EX-OR leg makes the EX-OR output be simply the signal which is driven on the other leg. More simply, the mux selecting V_{CC} or GND dictates whether the EX-OR output becomes the other leg signal or its complement.

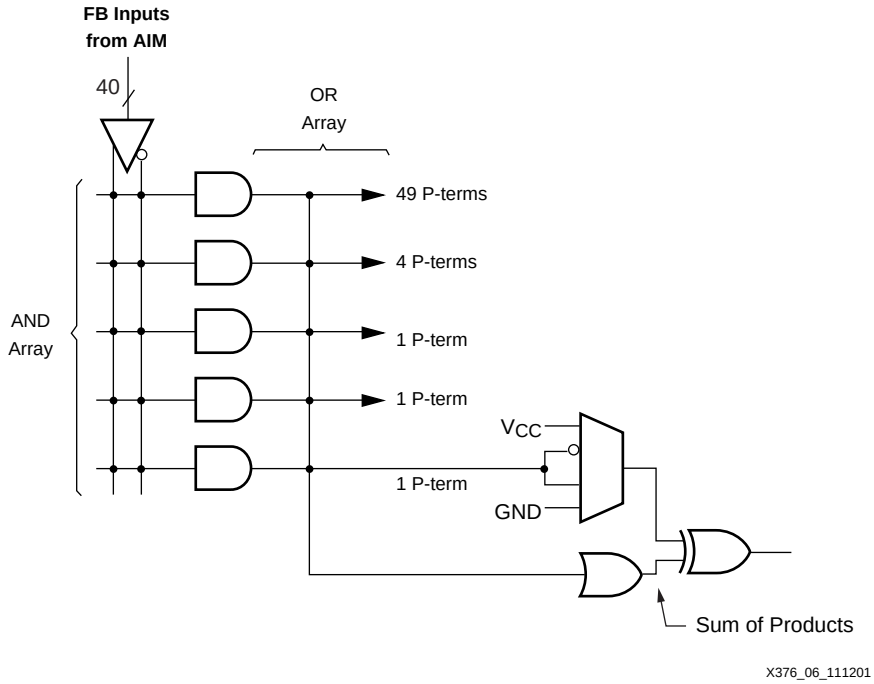

Figure 6: **CoolRunner-II PLA Structure**

Switching our attention from the V_{CC}/GND aspects of the mux to the bold path shown in Figure 7, we see a special high-speed option. A single AND gate (p-term) can directly pass through the mux going out the EX-OR, or have its complement (NAND) exit the EX-OR. This signal can then either attach to the flop on the other side or bypass it to the I/O pin. By avoiding the PLA OR term, extra speed (typically 0.3 ns) is gained, making this a valuable high-speed decoding path. Frequently, microprocessor address decoding needs maximum speed for a wide address decoder, and this path delivers it. The same fast AND/NAND path can be used to build faster shift registers, counters, and some simple state machines.

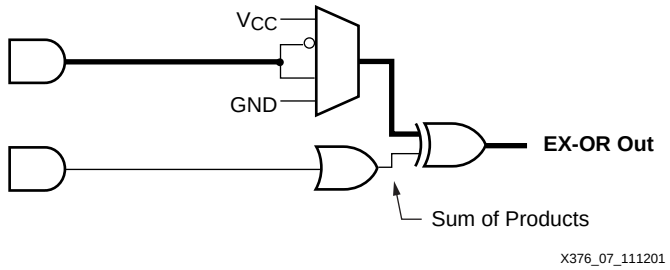

Figure 7: **CoolRunner-II Macrocell OR/EX-OR Section**

Using the Product Term Resources

A reduced version of Figure 3 is shown in Figure 8. This version exposes the ability of the macrocell to use one product term (PTA) to be used as a product term set signal and another product term (PTC) to be a product term clock. Note that the software would have to produce programming bits that would make the trapezoidal mux select that particular attachment. Product term resources are available to a specific macrocell, and not to any other macrocells. It should be noted that the PTA for macrocell number 1 is a different product term than PTA is for macrocell 2, 3, or 16. The PTA for macrocell 16 would be another one of the 49 p-terms shown in Figure 8.

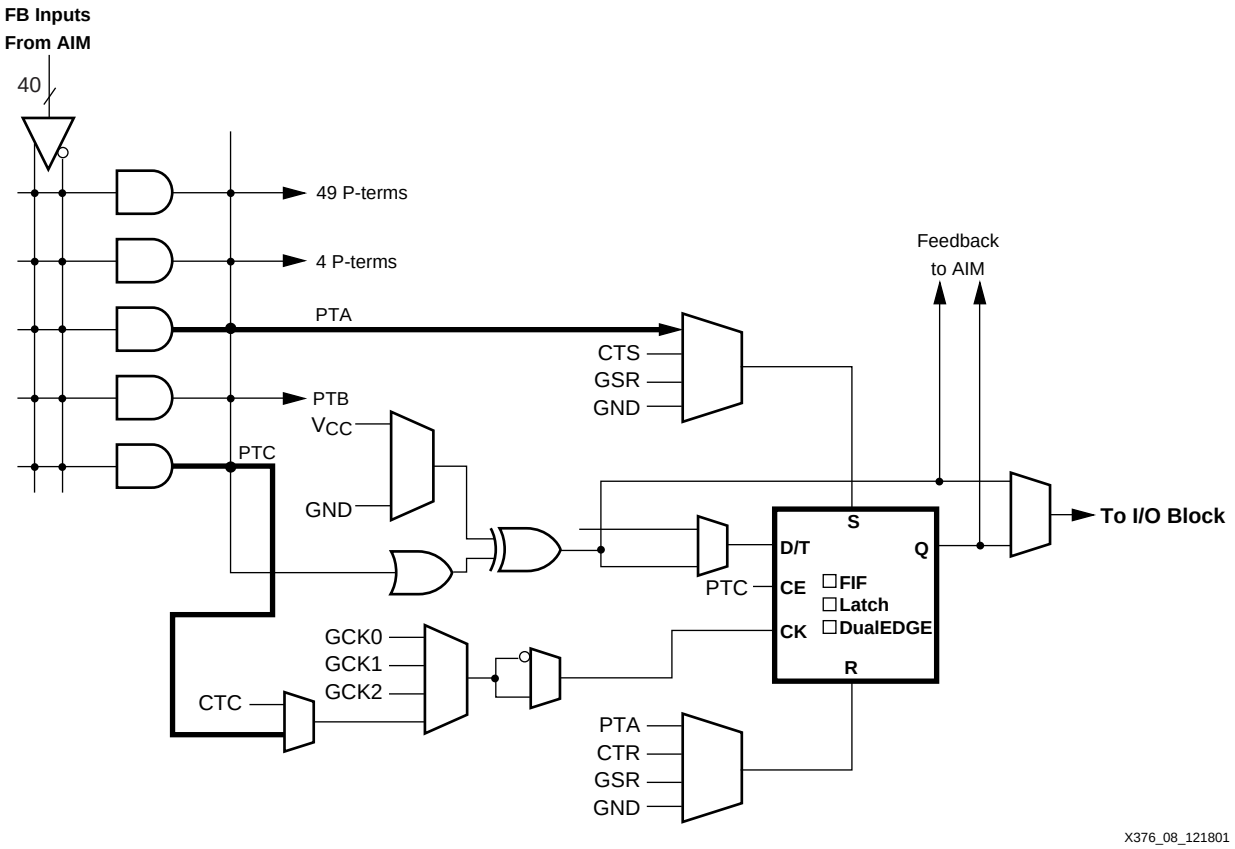

Figure 8: **Macrocell Simplified to Clarify Product Term Set and Product Term Clock**

Using the Control Term Resources

Sometimes, several macrocells within the same FB would need the exact same AND term driving their Clock, Reset, Set, or PTOE site. In that case, the Xilinx fitter software would use a control term instead of a product term. Control terms are the same as product terms except they connect to the same mux sites on every macrocell within the same FB. Because this occurs frequently in designs, this is a very useful resource to conserve the contents of the 56 p-term AND array. Clearly, it would be preferrable to use a single control term to satisfy even two macrocells with the same need, than to expend two p-term resources at individual macrocells. Figure 9 shows how control term set and control term clocks are used in a macrocell and also made available to other macrocells.

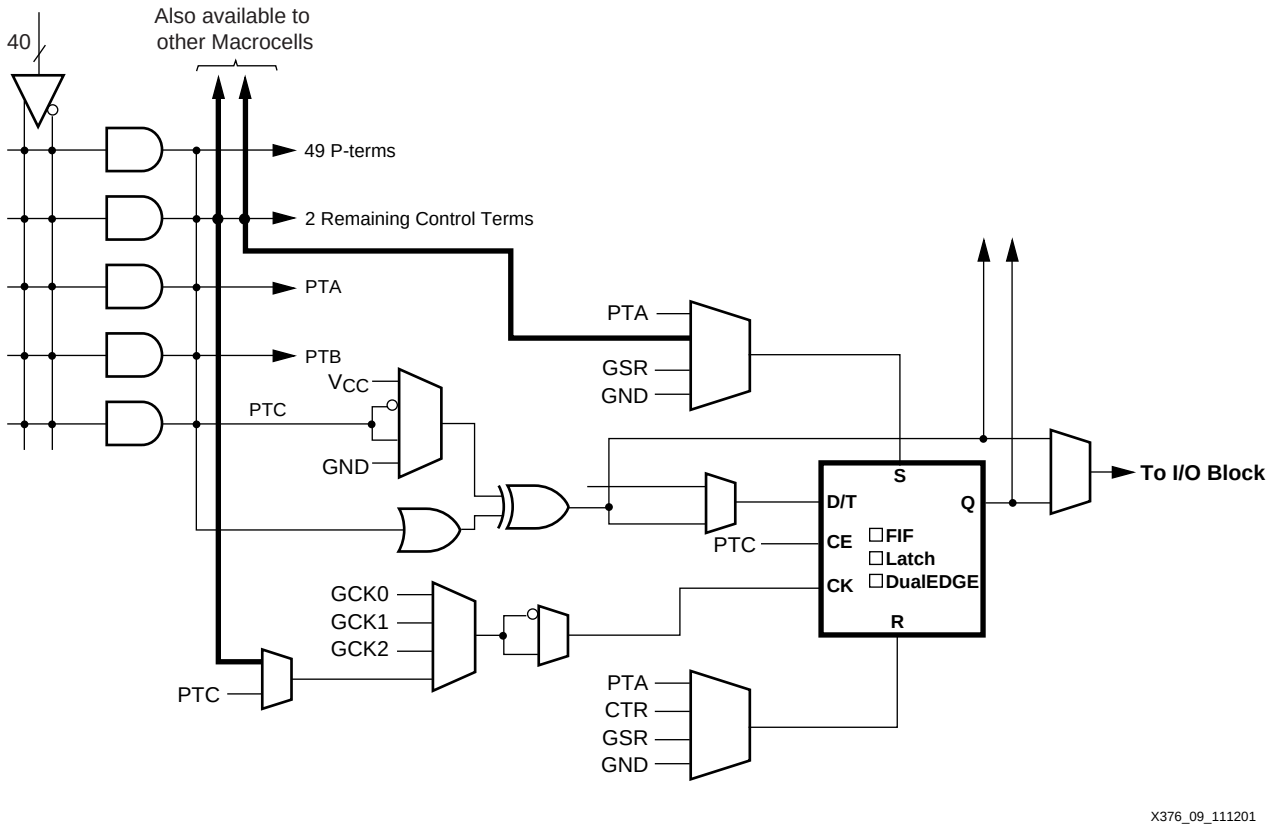

Figure 9: **Macrocell Simplified to Clarify Control Term Set and Control Term Clock**

Figure 10 shows how a control term (CTC) has influence scope over all macrocells within the same FB. If the same clock were to be used over several FBs, it would be appropriate for the software to assign a global resource (clock, s/r, output enable) rather than expend product terms. In some cases, it might take less power to use a control or product term resource than a global resource due to the capacitance of the global net. A speed/power trade-off would dictate the choice.

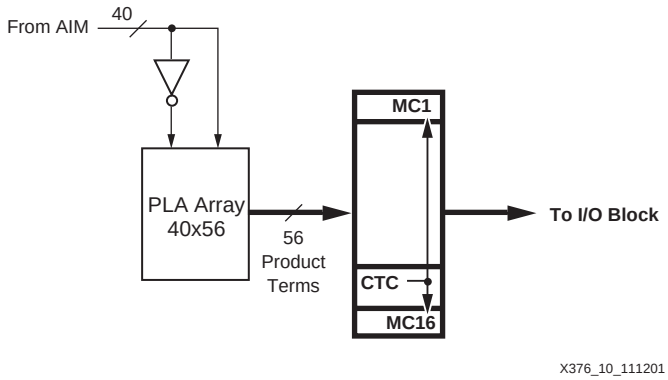

Figure 10: **Higher Level Function Block Showing Control Term C Scope**

Combining Product Term and Control Term Resources

Figure 11 shows how it is possible to mix and match product term and control term resources within a single macrocell. In this example, we see a product term set and a control term clock. The control term clock is also available to other macrocells within the FB. Typically, the software would make a choice like this if a specific macrocell needed a given set condition — unique to it — while its clocking need is identical to at least one other macrocell within the FB. Trading off which resources are declared to be product term generated and which are global or control term generated is an optimization generally solved by the Xilinx fitter software.

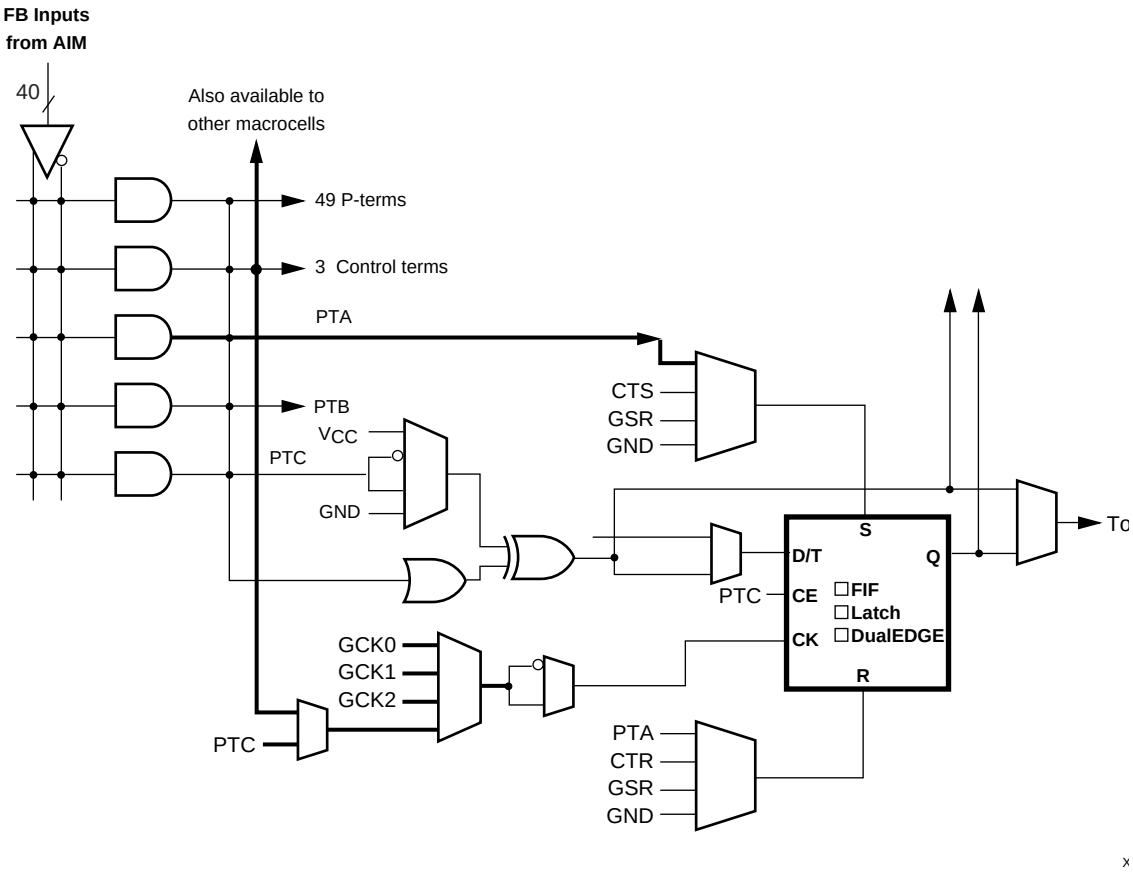

Figure 11: **Macrocell Simplified to Clarify Product Term and Control Term Usage**

Conclusions

High speed and low power present an interesting design trade-off. Typically, high-speed CPLDs have required high current levels to deliver fast switching times. CoolRunner-II delivers the logic required for high-speed general purpose CPLD applications while taking the consumed power to the absolute minimum.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
01/03/02	1.0	Initial Xilinx release.