PLUS405-55 – the ideal high speed interface AN034

INTRODUCTION

Philips Semiconductors PLUS405–55 is ideal for high performance microprocessor interfacing applications. Being a programmable integrated circuit, it adapts to nearly any bus or microprocessor protocol. The PLUS405–55 can make state machines, counters, and shifters running at speeds of 55 megahertz. The architecture of the PLUS405–55 combines a powerful programmable logic array with 16 JK flip-flops to form a programmable part superior to any comparable PAL part.

WHAT IS INTERFACING?

Interfacing is the translation of digital signals from one target device to another (see Figure 1). Each target device has their own unique signal behavior, and may not be directly connected to each other. Correct connection occurs with the use of a flexible interface. For instance, today's chip set integrated circuits connect a microprocessor to its memory and I/O devices. The signals presented by the microprocessor are not the specific /CAS and /RAS needed by the memory. Nor are the microprocessor signals the correct chip enables to attach to the UAR/T or graphics controller. Forming the translated signals which are appropriate for the memories, UAR/T's and other controllers is the job of the interface chips. Unfortunately, there are not off-the-shelf interface parts for all applications. That is where programmable parts excel.

With this information, we can ask: what are the qualities of an ideal interface part?

An ideal interface part must have sufficient logic inside to make correctly any logic translation needed. Because almost every interface is between two target devices which use separate clocks, conventional PAL parts are not adequate. This includes the popular 22V10 and most other registered PAL parts. To reconcile that an interface may exist between two target parts using different clocks, the interface must be able to synchronize signals from either or both target devices. So, an ideal interface must be clockable from at least two different clocks.

An ideal interface part must have enough flip-flops inside to capture data or control information from the target parts, and to re-synchronize control signals. This suggests it needs at least 16 flip-flops because the interface might receive 8 from either target device.

To support the use of the 16 flip-flops as handshaking flip-flops, a typical number of logic AND gates is about 4 gates per flip-flop. This number of gates can be less if the type of flip-flop is a JK rather than a D flip-flop.

An ideal interface must be as fast – if not faster – than the fastest target devices it must interface. This requirement is sometimes misunderstood. Most microprocessor parts have a clock input which is the fastest signal present at the outside world. Very few signals coming from the microprocessor are as fast or faster than the clock because they are usually made from flip-flop circuits inside the microprocessor.

In the past, many designers believed that interface PAL parts had to be twice as fast as the system clock. This was because either the system clock was operating at twice the crystal frequency or the PAL part had to compensate for the fact that some events occur on rising clock edges and other events happen on falling clock edges. This misconception should be reexamined. What is needed is an interface part which can respond to both edges of the basic clock rate, as the interface dictates.

Finally, the ideal interface must be electrically compatible with both devices that must communicate. This is almost always either CMOS or TTL and today's technologies usually support a standard TTL interface.

THE PLUS405-55 . . . THE IDEAL INTERFACE!

To meet the needs of the ideal interface, Philips Semiconductors designed the PLUS405–55 (see Figure 2). The 405 is fast enough to respond to microprocessor clocks in excess of 50MHz, contains 16 JK flip-flops and a programmable logic array to control the flip-flops. It accepts two clock sources and permits the internal flip-flops to be grouped in one of two standard ways. The PLUS405–55 handles simple data synchronization or complex bus handshaking between two target devices, within a single part. It is widely supported by Philips Semiconductors SNAP as well as third-party design tools.

Let's look inside the PLUS405 to see how it works. First, there are 15 input pins, each supplying a signal or its complement, to the main AND array of the 405. There is an additional input pin which can optionally bring in a clock. There are 8 output pins which are directly tied to specific flip-flop outputs (labeled F0 to F7). This is the fastest flip-flop configuration possible. It should be noted that the clock to Q time delay (t_{CKQ}), measured from the 405's pins, is 8 nanoseconds (max). This means that signals can get into and out of the PLUS405–55 very fast.

The choice of flip-flops was the JK flip-flop. For building counters, JKs require only one gate per bit of additional logic. For building shift registers inside, no gates are required except that all connections use gates, so it uses a small number to connect the shift register.

For making state machines, the number of gates per flip-flop is up to the application. There is no design restriction with programmable AND gates, which can be assigned as needed to any OR gates. This is superior to the ordinary PAL approach or even the 22V10 approach where each OR gate permanently connects to specific AND gates. AND gate outputs are shared as needed and there is no limit on how many OR gates may be driven from a single AND term. JK flip-flops are superior to D flip-flops for all state machine applications, because they do not need as much external logic to control their behavior.

At this point, we see that the PLUS405–55 combines two superior elements. Namely, JK flip-flops for making state machines and a programmable logic array (PLA) for forming logic expressions. However, there are still more features which the PLUS405–55 includes. There is a special pin for tri-state control of the 8 output pins or alternately it can initialization the 405. When used to initialize, the pin can applies any value to the asynchronous set or preset of each flip-flop. Initialization is to any state chosen.

The internal flip-flop connection in the 405 is also critical. Flip-flops are first grouped into two categories – 8 internal flip-flops and 8 output flip-flops. The internal flip-flops do not directly access the output pins and the output flip-flops do not feedback. To form state machines inside the 405, the internal flip-flops are best used. This doesn't mean that the output flip-flops can't be used, but rather that their use with feedback requires external connection. This is seldom necessary.

Another important flip-flop grouping is inside the 405. Four output flip-flops (F0–F3) link with their clock inputs to four internal flip-flops. The other four output flip-flops similarly link with four other internal flip-flops. This permits two separate state machines to be built. One state machine uses 4 internal flip-flops and four output flip-flops on one clock and four other internal flip-flops linked to their four output flip-flops on another clock. Alternately, the can all be linked together to a

common clock, or by an external inverter, one state machine can operate on the clock while the other one operates on the clock's complement.

Almost hidden from view in Figure 2 are two additional OR gates which do not drive any flip-flops. These two OR gates are inverted and fedback to the input of the AND logic array. The two inverted OR gates (i.e., NOR gates) are called complement arrays. They are used to save AND gates when state transitions are complicated. Luckily, the design software – SNAP and SLICE – automatically use these gates to save the designer from having to use them. The complement arrays also permit automatic homing to known states, if a power transient accidentally puts a state machine into an undefined state.

PERFORMANCE

As mentioned previously, the clock to Q time delay is 8 nanoseconds maximum. That is only one part of the performance equation. The flip-flop speed is:

$$f_{MAX} = \frac{1}{t_{SETUP} + t_{CKQ}}$$

When a flip-flop is put into a circuit with logic driving it, the logic adds delay which slows the circuit down. For a flip-flop with extra logic, that logic delay is included in the performance equation as follows:

$$f_{MAX} = \frac{1}{t_{SETUP} + t_{DELAY} + t_{CKQ}}$$

Passing signals through the PLA section of the PLUS405 (the complement array) will add additional time delay as follows:

t_{SETUP} = 10 nsec

t_{CKQ} = 8 nsec

t_{DELAY} = 8 nsec

Without the complement array, the f_{MAX} is 55.6MHz found as the inverse of 18 nanoseconds. This is because the specification includes a single signal pass through the PLA as part of the flip-flop setup time. Viewed from the outside, this makes sense because there is no way to get a signal to the flip-flop without entering the PLA. With the complement array, f_{MAX} is the inverse of 26 nanoseconds, found by adding the t_{DELAY} term to the f_{MAX} expression. This sets f_{MAX} with the complement array at 38.5MHz. An additional pass through the complement array is never needed, so the PLUS405–55 will never be slower than 38MHz. The additional complement array is included in case two distinct state machines are built, where each needs one.

The PLUS405–55 includes a PLA which has 64 AND gates in it. This permits an average of 4 AND gates per flip-flop, but this many are seldom needed, because the JK flip-flops are so efficient. Because the AND gate outputs are shared as needed, redundant terms are never used. There are additional buffers to do the asynchronous flip-flop control, and each JK flip-flop includes its own OR gates.

DESIGN SUPPORT

Designers need tools which can capture the design, compile it to a fusemap and download commercially available programmers. Philips Semiconductors supports the PLUS405–55 with SNAP design software (Figure 3). This software runs on personal computers and permits designs to be formulated with Boolean logic equations, state equations or schematics. SNAP, the full-featured product, includes advanced simulation capability found only in Field Programmable Gate Array (FPGA) or ASIC design software.

SNAP includes a simulator with back annotation of time delays to accurately model the PLUS405-55 as well as Philips Semiconductors full PLD product line. Additional support for the PLUS405–55 can be found in third-party design tools.

PLUS405-55 – the ideal high speed interface AN034

AN EXAMPLE OF INTERFACING WITH THE PLUS405-55

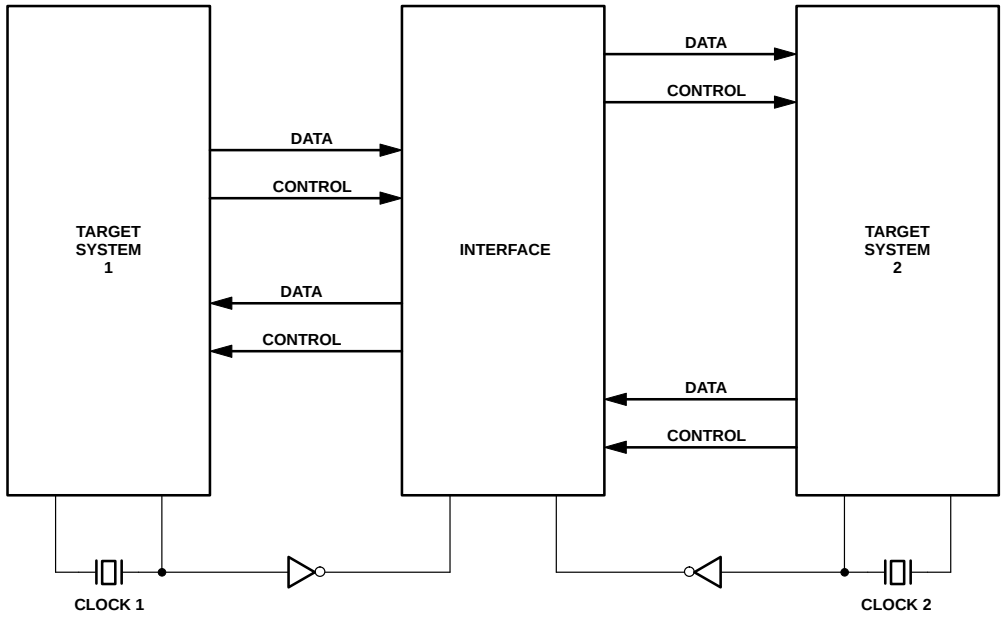

Figure 1. Digital Interfacing

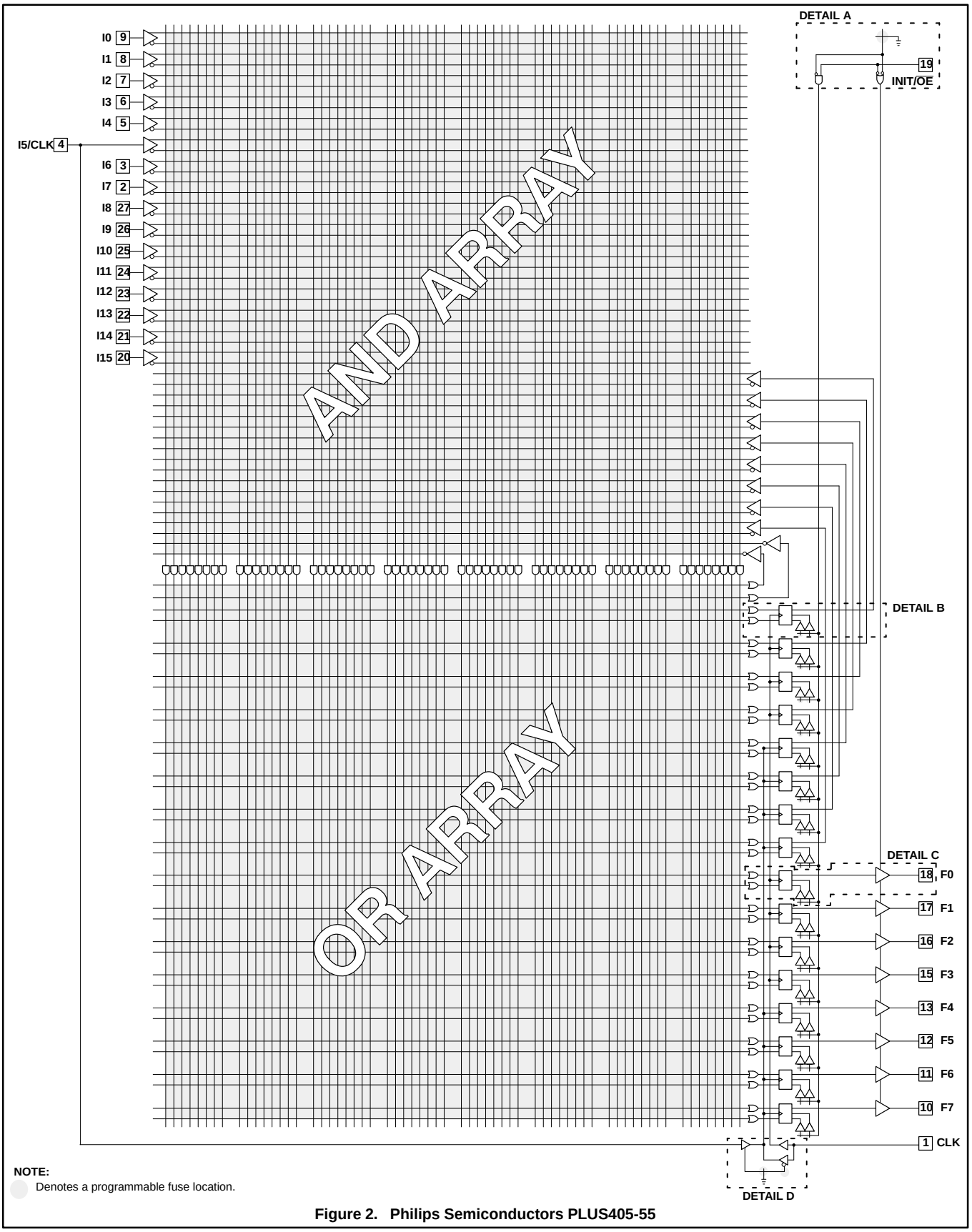

Figure 2. Philips Semiconductors PLUS405-55

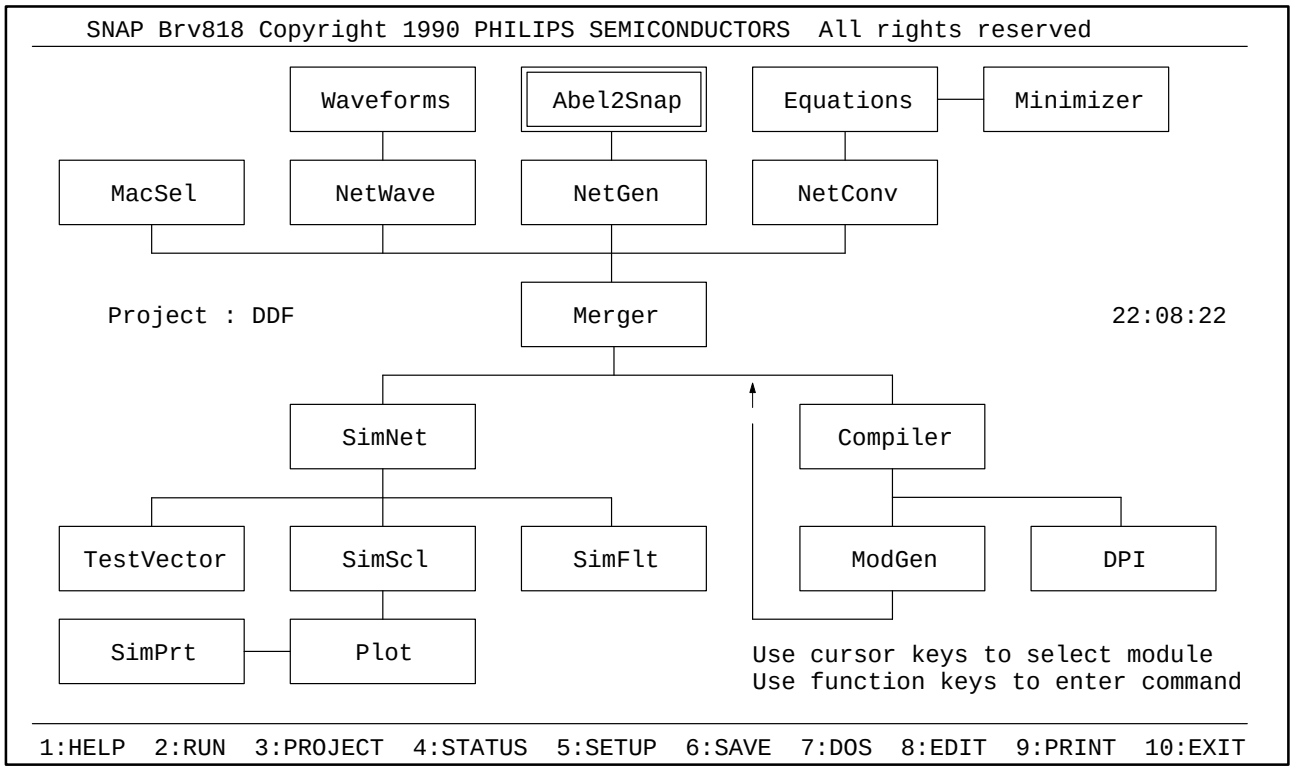

Figure 3. SNAP Shell