APPLICATION NOTE

AN216
Arbitration in shared resource systems

1988 Jul 18

Philips
Semiconductors

PHILIPS

Arbitration in shared resource systems AN216

INTRODUCTION

The need for more powerful and faster systems gave birth to multiprocessing and multi-tasking systems. But to achieve this, cost and reliability were not to be sacrificed. To reduce cost it is vital to share resources, but to do so requires reliable means of arbitration. In a multiprocessing system, a single bus may be shared between various processors or intelligent peripherals. The resources shared by processors (Figure 1) are generally termed as global resources and those shared between the local processor and the peripherals (Figure 2) are typically known as local resources. Whether local or global, there always exists a protocol that will connect and disconnect various devices to and from the shared resources. Various bus architectures in existence today have different ways of doing this.

No matter what the protocol of a specific bus, there is always a method which dictates how arbitration shall be performed between two or more devices. Some systems employ synchronous arbitration and some use an asynchronous approach. The third option is not to use arbitration at all, but instead to employ time-multiplexing. This is used mainly in data communications by dividing the common media into various time slots. Each processor (station) is assigned a predetermined time for using the media. If the station does not need to use the media during its assigned time-slot, it may pass control to the next station. This obviously results in an inefficient use of the bus bandwidth.

Synchronous and asynchronous arbitration have their advantages and disadvantages, and are both used in system designs. Some applications may even use a combination of the two. Generally, synchronous arbitration is used in systems where the designer can take the time to synchronize signals with the master clock. In synchronous arbitration the request is sampled on a clock edge, and therefore if it is asserted close to, but after the sampling clock edge, it will not be recognized until after a whole clock cycle. Today's applications, where speeds are being pushed to their limits, may not find that an optimal solution. Therefore, more and more designers tend towards asynchronous arbitration because it is much faster on the average. Since applications vary drastically from one to another, some may be better served by first-come-first-serve arbitration, some with fixed priority and some with dynamic priority.

In a first-come-first-served scheme, as the name implies, the request to be asserted first is selected first. All other requests made after the first are queued in their respective order of assertion. After the current request is serviced, the request asserted second will be selected, and so on. If the request just serviced is asserted again, before all other active requests are serviced, it will be placed at the end of the queue. In a fixed priority method all inputs have a hard-wired priority and cannot be changed. In a dynamic priority assignment, the user can change the priority depending upon the system needs. For example, processors performing vital tasks may be placed at a higher priority as compared to processors doing background tasks.

Arbitration, whether synchronous or asynchronous, always brings up the question of "metastability". A hard fact that relates itself all the way back to the beginning of the history of electronics. In its simplest definition, it is the state of a flip-flop that is neither a logic "1" nor a logic "0", and is a result of violations of its setup and hold times. This condition must be allowed and dealt with in arbitration and synchronization designs.

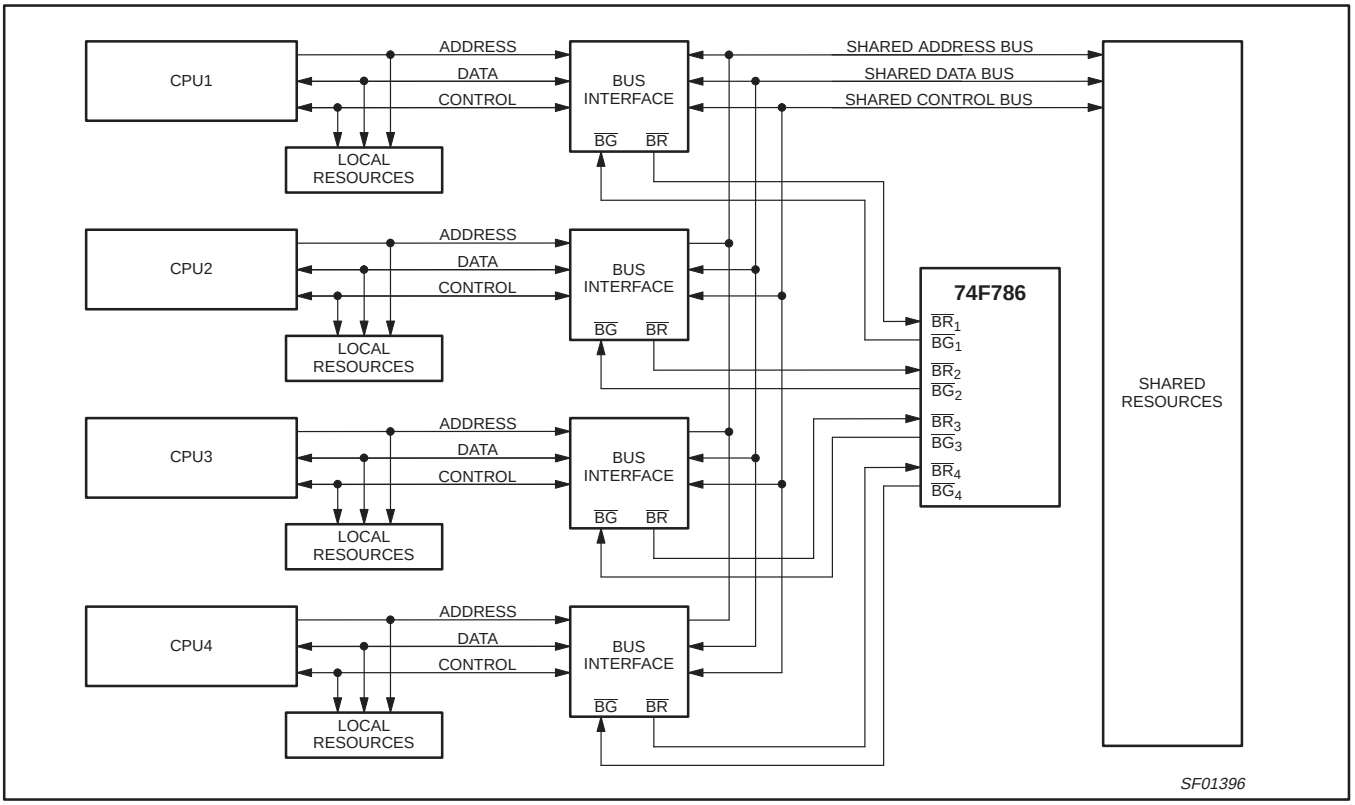


Figure 1. Sharing Global Resources

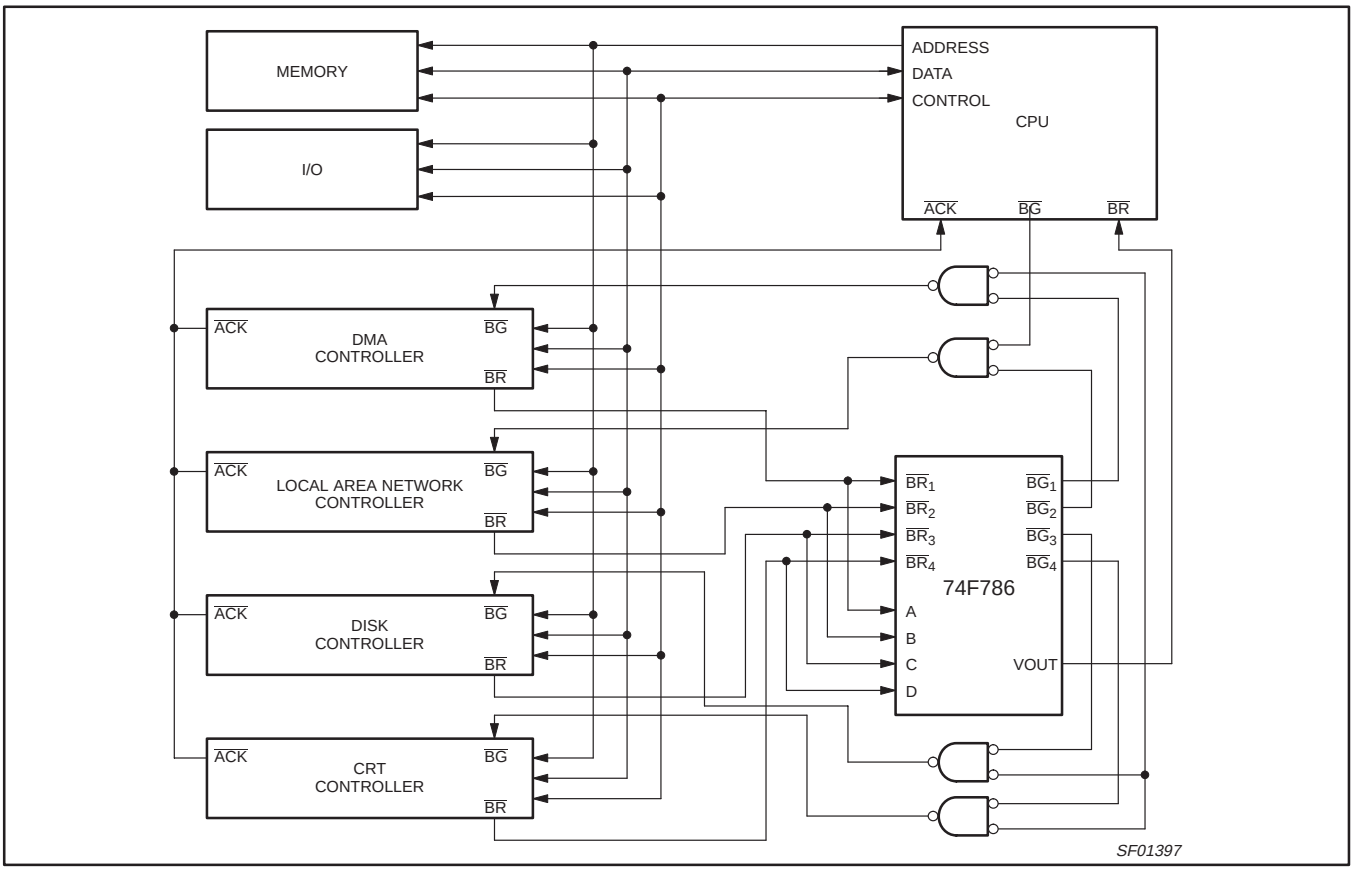


Figure 2. Sharing Local Resources

METASTABILITY

Various publications have discussed this subject and given recommendations for reducing but not completely eliminating this potential problem. Briefly the suggestions consist of using very fast flip-flops (with very small setup and hold times), using multiple flip-flops and delay lines and designing of metastable-hardened flip-flops. Please note that a metastable-hardened flip-flop does not necessarily mean that it will never enter a metastable state, but rather it is a flip-flop that is highly optimized to be used in applications where the system designer cannot guarantee the minimum setup and hold times specified by the manufacturer. Since, as of today, the design of a metastable free flip-flop is not practically possible, the next best thing that could be done is design of a flip-flop with significantly reduced setup and hold times and reduced propagation delays. This will ensure reduced probability of being in a metastable state. Since we still will have some probability of not meeting the minimum setup and hold times and potentially being in a metastable state, another requirement to be imposed on this flip-flop would be to hold its previous state and not to propagate this invalid state to its outputs until it has decided to settle in a “0” or a “1” state. By doing so, it could be guaranteed that the outputs of a flip-flop will never be in an undetermined state even though the flip-flop may internally be in a metastable state. The penalty that the user would expect to pay in such a design will be a propagation delay that can extend beyond the maximum specified in the data sheet.

74F786 — 4-INPUT ASYNCHRONOUS ARBITER

The key consideration when arbitrating for shared resources is that access may not be granted to more than one device at a given time. If this could be guaranteed, it would improve reliability. This Application note describes a product from Philips Semiconductors, which guarantees against simultaneous grants and does so at very high speeds. The Philips Semiconductors 74F786 (Figure 3) is a general purpose asynchronous bus arbiter designed to address the needs for real-time applications, where arbitration is desired between multiple devices sharing common resources. The design goal was to provide for a device, the outputs of which could be guaranteed against logic hazards (glitches), metastability and that no more than one output could be active at a given time. The arbiter has four Bus Request ($\overline{BR}_n$) inputs, which allow arbitration between two to four asynchronous inputs. The priority is determined on a first-come-first-served basis. Corresponding to each input is a separate Bus Grant ($\overline{BG}_n$) output, which indicates which one of the request inputs is served by the arbiter at a given time. All these outputs are enabled by a common enable ($\overline{EN}$) input. Also included on-chip, is a general purpose four-input AND gate that may be used to generate a bus request signal (Figure 2) or as an independent AND gate.

Since the Bus Request inputs have no inherent priority, the arbiter assigns priority to the incoming requests as they are received. Therefore, the first request asserted will have the highest priority.

When a Bus Request is received, its corresponding Bus Grant becomes active, provided $\overline{EN}$ is LOW, and no other Bus Grant is active. Typically, a Bus Grant is selected in 6.6ns from the time of assertion of a request input. If additional Bus Requests are made after the first request goes LOW, they are queued in their respective order, When the first request is removed, the arbiter services the request with the next highest priority, based upon a first-come-first-served algorithm.

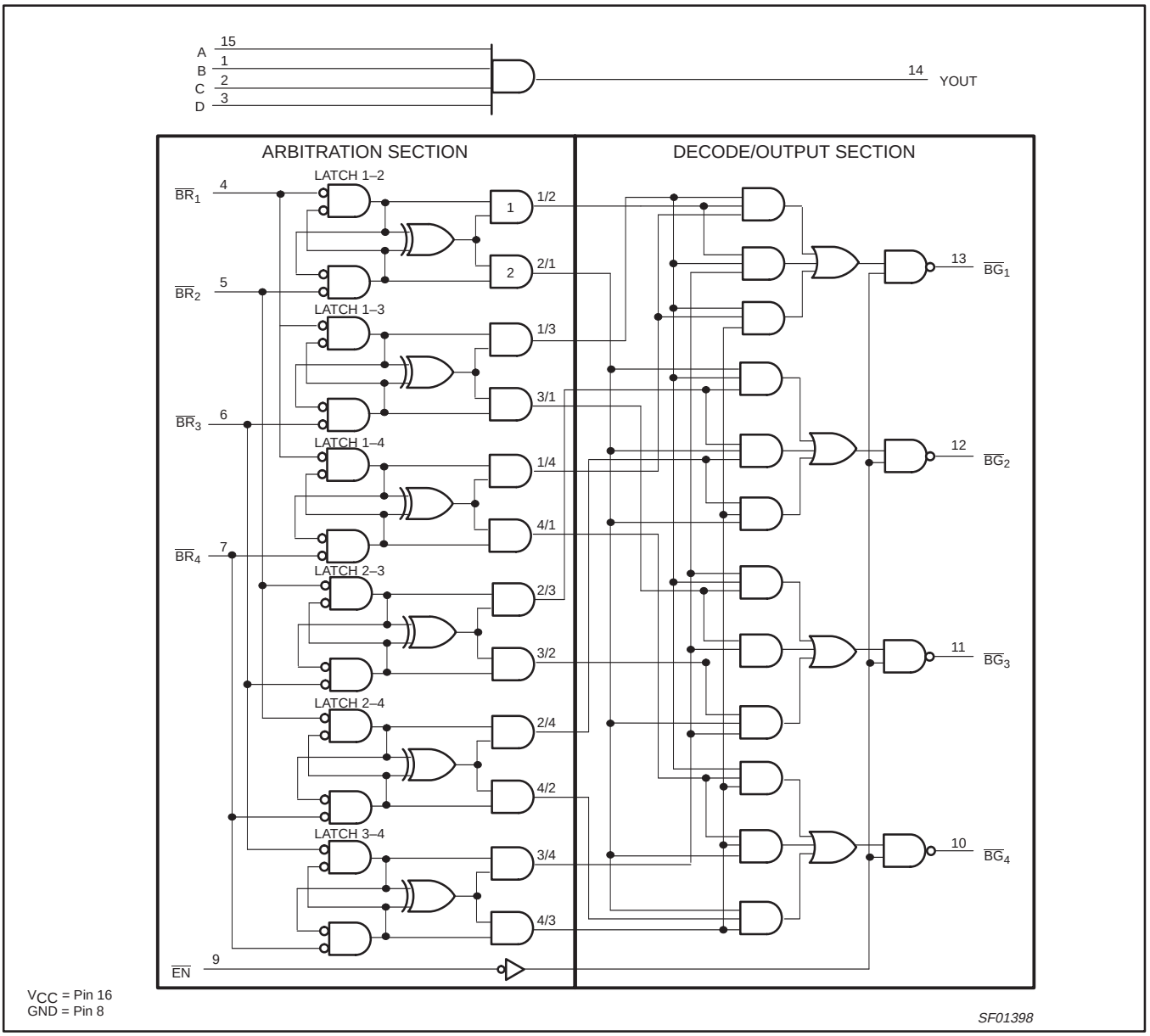
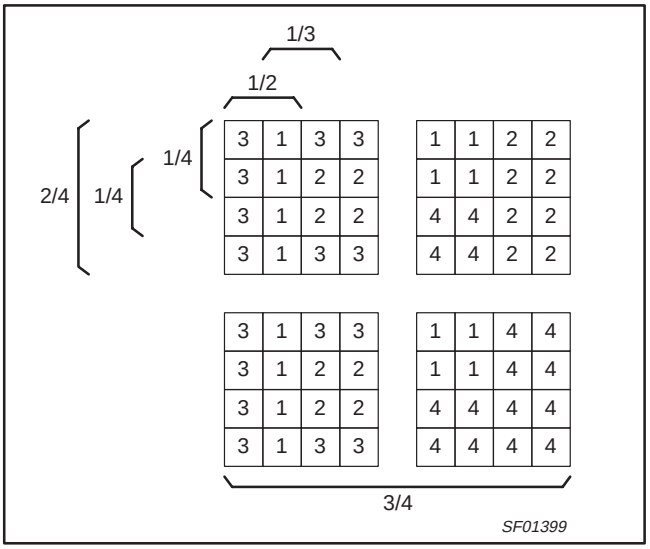


Figure 3. 74F786 Logic Diagram

Metastable-Free Outputs

The 74F786 logic diagram (Figure 3) consists of two sections: the arbitration section and the decoding/output section. Within the arbitration section lie six independent 2-input arbiters, each of which arbitrates between the two Bus Request ($\overline{BR}_n$) inputs connected to that specific arbiter. Each 2-input arbiter is comprised of two cross-coupled NOR gates, an Ex-OR gate and two AND gates. The cross-coupled NOR gates are designed so that they are securely latched when a Schottky diode voltage difference appears between the outputs of these NOR gates. The Ex-OR gate is designed so that its output will remain LOW until there is at least 1Vbe difference between its inputs. This creates a noise-margin of 1Vbe (base to emitter voltage)-1Vsky (Schottky voltage) = 0.3 Volts and assures that the output of the Ex-OR will not go HIGH until after the two NOR gates have resolved any contention problems. This guarantees that neither of the outputs of a 2-input arbiter can be in a metastable state, and also that both outputs cannot be HIGH simultaneously. As is clear from Figure 3, the first 2-input arbiter is responsible for deciding between the $\overline{BR}_1$, and $\overline{BR}_2$ inputs. Since both AND gate outputs cannot be HIGH at the same time, the other three possible configurations are: First, AND gate1 is HIGH indicating that $\overline{BR}_1$ arrived at the latch before $\overline{BR}_2$ (designated 1/2); second, AND gate2 is HIGH indicating BR_2 arrived before BR_1 (designated 2/1); and third, both AND gates are LOW, indicating that neither $\overline{BR}_1$ nor $\overline{BR}_2$ has been latched.

Table 1. 74F786 Karnaugh Mappings



Glitch-Free Outputs

The decode section of the 74F786 is responsible for insuring that the outputs do not glitch or produce a logic hazard. While there are three possible Karnaugh mappings, to produce an optimum decode section with a minimum number of transistors and balanced propagation times, the mapping in Table 1 was chosen. solving Table 1 for $\overline{BG}_1$–$\overline{BG}_4$ yields the following equations:

$$\overline{BG}_1 = 1/2 \cdot 1/3 \cdot 1/4 + 1/2 \cdot 1/3 \cdot 3/4 + 1/2 \cdot 1/4 \cdot 4/3$$

$$\overline{BG}_2 = 2/1 \cdot 2/3 \cdot 2/4 + 2/1 \cdot 2/3 \cdot 3/4 + 2/1 \cdot 2/4 \cdot 4/3$$

$$\overline{BG}_3 = 3/1 \cdot 3/2 \cdot 3/4 + 1/2 \cdot 3/1 \cdot 3/4 + 2/1 \cdot 3/2 \cdot 3/4$$

$$\overline{BG}_4 = 4/1 \cdot 4/2 \cdot 4/3 + 1/2 \cdot 4/1 \cdot 4/3 + 2/1 \cdot 4/2 \cdot 4/3$$

To see if a glitch can occur, let's take the worst possible case, that is, let $\overline{BR}_1$ beat $\overline{BR}_2$, 2 beat 3, 3 beat 4, and 4 beat 1 (a possible situation when all inputs are asserted simultaneously). Also, let's have the outputs of the arbitration section switch sequentially. Initially, all the variables in the equations are false (remember, the outputs of the arbitration section have three possible states). First, when 1/2 goes true, 2/1 must remain false. This eliminates several terms from playing a role in deciding which output becomes active. In fact, $\overline{BG}_2$ has been removed from the list and is no longer a contender. At this point, while all the outputs are HIGH (inactive), we have decided that $\overline{BG}_2$ will remain inactive. This leaves us with the following equations:

$$\overline{BG}_1 = 1/2 \cdot 1/3 \cdot 1/4 + 1/2 \cdot 1/3 \cdot 3/4 + 1/2 \cdot 1/4 \cdot 4/3$$

$$\overline{BG}_3 = 3/1 \cdot 3/2 \cdot 3/4 + 1/2 \cdot 3/1 \cdot 3/4 + \sout{2/1 \cdot 3/2 \cdot 3/4}$$

$$\overline{BG}_4 = 4/1 \cdot 4/2 \cdot 4/3 + 1/2 \cdot 4/1 \cdot 4/3 + \sout{2/1 \cdot 4/2 \cdot 4/3}$$

Similarly, when 2/3 goes true, 3/2 must remain false, which further eliminates a term from this set of 3 equations:

$$\overline{BG}_1 = 1/2 \cdot 1/3 \cdot 1/4 + 1/2 \cdot 1/3 \cdot 3/4 + 1/2 \cdot 1/4 \cdot 4/3$$

$$\overline{BG}_3 = \sout{3/1 \cdot 3/2 \cdot 3/4} + 1/2 \cdot 3/1 \cdot 3/4 + \sout{2/1 \cdot 3/2 \cdot 3/4}$$

$$\overline{BG}_4 = 4/1 \cdot 4/2 \cdot 4/3 + 1/2 \cdot 4/1 \cdot 4/3 + \sout{2/1 \cdot 4/2 \cdot 4/3}$$

Now when 3/4 goes true, 4/3 must remain false. This eliminates $\overline{BG}_4$ from the contending list and the contest now is between $\overline{BG}_1$ and $\overline{BG}_3$ as indicated from the following equations:

$$\overline{BG}_1 = 1/2 \cdot 1/3 \cdot 1/4 + 1/2 \cdot 1/3 \cdot 3/4 + \sout{1/2 \cdot 1/4 \cdot 4/3}$$

$$\overline{BG}_3 = \sout{3/1 \cdot 3/2 \cdot 3/4} + 1/2 \cdot 3/1 \cdot 3/4 + \sout{2/1 \cdot 3/2 \cdot 3/4}$$

When 4/1 goes true, 1/4 must remain false. Still no decision has been made, and is dependent on the two 2–4 and 1–3 latches not taken into account yet. In this case, the 2–4 latch status is a Don't Care, so the outcome of the 1–3 latch dictates the Bus Request granted:

$$\overline{BG}_1 = 1/2 \cdot 1/3 \cdot 3/4$$

$$\overline{BG}_3 = 1/2 \cdot 3/1 \cdot 3/4$$

If the 1–3 latch settles in the 1/3 state, $\overline{BR}_1$ gets the grant, and with 3/1 remaining false, $\overline{BG}_3$ will remain inactive. Similarly, if the 1–3 latch goes to the 3/1 state, $\overline{BR}_3$ gets the grant, and with 1/3 remaining false, $\overline{BG}_1$ will remain inactive.

Notice that the Bus Grant was given in this case without regard to the 2–4 latch. In fact, a quick review shows that neither the 2–3 latch nor the 1–4 latch played a role in making the decision. Each grant is dependent on the state of three latches. By the nature of the encoding logic, as the three activating latches are switched, three outputs are forced to remain in an inactive state. This insures a glitch-free output.

Let's assume that in the example above, the 1–3 latch goes to the 1/3 state, and hence, $\overline{BG}_1$ is asserted. At this time the other five latches in the circuit will be in 1/2, 4/1, 2/3, 2/4, and 3/4 states. If at this point $\overline{BR}_3$ is removed, then latch 3–4 changes from 3/4 to 4/3, and hence, $\overline{BR}_4$ steals the grant (with $1/2 \cdot 4/1 \cdot 4/3$). This concludes that if three or more requests are asserted precisely at the same time, and one of them is removed prior to being serviced, it may cause premature termination of the present grant and assertion of another grant. Therefore, when using three or more Bus Requests, it is not advised to remove a request before being serviced. On the other hand, arbitration between two requests does not have this restriction. The user, if necessary, may decide to remove an ungranted request at his discretion.

Extended Propagation Delays

Since the outputs of the six 2-input arbiters cannot display a metastable condition, the Bus Grant outputs cannot display a metastable condition because the decoding/output section does not have any storage element to go metastable. Even though the Bus Grant outputs cannot go metastable, the cross-coupled NOR gates can. To determine the metastability characteristics of these NOR gates, the 74F786 was evaluated by Mr. Thomas J. Chaney of Washington University in St. Louis, Missouri, who is considered to be a leading expert in this field. Of the 19 devices supplied to him, Table 2 gives the test results from the fastest, the slowest and a typical package. In order to determine the Mean Time Between Package Unresolved (MTBPU) with the relative arrival times of the two input signal transitions uniformly distributed, the following formula is used:

$$MTBPU = [\exp(t'/\tau)]/[T_0(\text{Input 1 rate})(\text{Input 2 rate})].$$

where:

t' = Time given to resolve contention between inputs after they are asserted,

and τ and T_0 are device parameters derived from tests and can most nearly be defined as:

τ = A function of the rate at which a latch in a metastable state resolves that condition

T_0 = A function of the measurement of the propensity of a latch to enter a metastable state. T_0 is also a very strong function of the normal propagation delay of the device.

Solving for t', the resolving time measured from the arrival of the first input, and setting up the equation so the value of T_0 in Table 2 (given in ns) can be substituted directly, is:

$$t' = (\tau)\text{In}[(T_0)(3E14)].$$

The implication of the above equation is that, even though typical propagation dely through the arbiter is about 6.6ns, contention between inputs may extend this time significantly and can be calculated from Table 2.

Table 2. 74F786 Test Results for All Latches for Three Packages

All tests with $V_{CC} = 5.0V_{DC}$ and at Room Temperature

PACKAGE	LATCH	OUTPUT MEASURED	τ (ns)	T_0 (ns)	h (ns)	t' FOR 1 FAILURE/CENTURY (inputs at 10E6hz)
FASTEST	1–2	13	0.38	175E2	6.6	16.6
	1–3	13	0.39	79E2	6.6	16.4
	1–4	13	0.39	69E2	6.6	16.4
	2–3	12	0.38	109E2	6.6	16.1
	2–4	12	0.39	68E2	6.6	16.5
	3–4	11	0.38	181E2	6.6	16.3
SLOWEST	1–2	13	0.44	34E2	6.6	18.1
	1–3	13	0.44	17E2	6.6	18.0
	1–4	13	0.43	26E2	6.6	17.8
	2–3	12	0.44	16E2	6.6	17.9
	2–4	12	0.46	8E2	6.6	18.5
	3–4	11	0.44	29E2	6.6	18.2
TYPICAL	1–2	13	0.41	56E2	6.6	17.3
	1–3	13	0.42	24E2	6.6	17.2
	1–4	13	0.43	17E2	6.6	17.5
	2–3	12	0.43	18E2	6.6	17.4
	2–4	12	0.39	72E2	6.6	16.6
	3–4	11	0.41	49E2	6.6	17.2

Where h = typical propagation delay through the device.

NOTES

Arbitration in shared resource systems AN216

Data sheet status

Data sheet status	Product status	Definition [1]
Objective specification	Development	This data sheet contains the design target or goal specifications for product development. Specification may change in any manner without notice.
Preliminary specification	Qualification	This data sheet contains preliminary data, and supplementary data will be published at a later date. Philips Semiconductors reserves the right to make chages at any time without notice in order to improve design and supply the best possible product.
Product specification	Production	This data sheet contains final specifications. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

[1] Please consult the most recently issued datasheet before initiating or completing a design.

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088–3409
Telephone 800-234-7381

Date of release: 04-98

Document order number: 9397 750-05224

Let's make things better.

Philips Semiconductors

PHILIPS