

A Complete Solution for Futurebus + Arbitration Schemes

National Semiconductor
Application Note 837
Shilpa Parikh
July 1992



A Complete Solution for Futurebus + Arbitration Schemes

IEEE 896.1 Futurebus+ Standard specifies two arbitration protocols: Distributed Arbitration and Central Arbitration. Each arbitration scheme has its merits and drawbacks. First a brief summary of both arbitration methodologies is presented. Then follows a discussion on National's present day solution and strategy for Futurebus+ arbitration schemes.

ARBITRATION SCHEMES

Distributed Arbitration

With the distributed arbitration scheme, all master capable modules participate in the arbitration process to determine the next bus master. During the control acquisition cycle, all competing modules assert their arbitration numbers onto the Futurebus+ arbitration bus lines. Contention logic is implemented to resolve the winner. To reduce bus latency for critical events, and maintain a fair and equitable access of the bus, a Round Robin protocol is implemented for competing modules with identical priority numbers. In a system with two priority levels, arbitration requires a single competition cycle (single pass) before a new bus master is selected. While systems with a greater number of priority levels (maximum 256 levels), arbitration requires two competition cycles (two pass) before a new bus master is selected. The competition number consists of: five geographical address bits, one round robin bit, either one priority bit (for 2 priority levels; one pass arbitration) or eight priority bits (for 256 priority levels; two pass arbitration), and one bit to indicate if the competition number is a single pass or two pass number. CN7, the most significant bit, has the highest priority down to CN0 which has the least priority. Both single pass and two pass arbitration numbers can be assigned in a system. The single pass numbers take higher precedence over two pass numbers. The competition occurs concurrently with

data transfers. So a master elect can take over immediately when the current master is ready to relinquish the bus. One competition cycle can typically range from 200 ns to 400 ns.

Central Arbitration

With the central arbitration scheme, one central arbiter will keep track of all pending and new requests for the Futurebus+. It will evaluate all signals and appropriately issue a bus grant signal to a module that will become the new bus master. The signals between the central arbiter and each module are: two request signals coming in from each module (rq0, rq1: rq1 is a higher priority signal than rq0), a grant signal going out to each module, a bussed preempt signal going out to all modules, and a bussed end of tenure signal coming into the central arbiter. A look up table with preprogrammed priorities for the various request signals may be incorporated within the central arbiter design along with a Round Robin protocol similar to distributed arbitration. 896.1 only specifies that rq1 is of greater priority than rq0. The rest is left for the system designer to implement based on the specific system needs. The central arbiter can typically issue a bus grant signal in 50 ns to 150 ns from bus request.

ARBITRATION TIMING

Distributed Arbitration

Distributed arbitration protocol is implemented on 14 backplane lines. The arbitration competition occurs on AB[7:0]*, and AB[P]*. Three signals: AP*, AQ*, and AR*, serve as synchronization signals. Only one of these lines will change per transition from one phase to the next phase of arbitration. The remaining two signals: AC0* and AC1* can be thought of as arbitration competition status indicators. The assertion of AC0* signal indicates that error has occurred. The assertion of AC1* indicates that transfer of tenure should not occur during the current arbitration pass.

Table I. Distributed Arbitration Number

Bit	CN7	CN6	CN5	CN4	CN3	CN2	CN1	CN0
Single Pass Arbitration Number								
Pass1	1	PR0	RR	GA4	GA3	GA2	GA1	GA0
Two Pass Arbitration Number								
Pass1	0	PR7	PR6	PR5	PR4	PR3	PR2	PR1
Pass2	1	PR0	RR	GA4	GA3	GA2	GA1	GA0

AN-837

One pass of the distributed arbitration competition cycle consists of five phases. For two pass cycles, the complete five phase cycle is repeated again. The following is a description of the events that occur during each arbitration phase: (see *Figure 1*).

Phase 0: No competition is in progress, or it is the midway point for a two pass arbitration cycle. Modules needing to use the bus or send a message assert ap^* . This is the first indication to other modules that an arbitration cycle is beginning.

Phase 1: All modules decide if they want to compete in this arbitration cycle or not. All modules desiring to compete, put their competition number onto the bus.

Phase 2: Competition takes place. Modules wait for the backplane competition bus lines to settle before evaluating their internal "win" signals. The arbitration settling time, t_a , is determined by each module based on the competition number. The winner of the competition asserts aq^* to transition to the next phase of the arbitration cycle.

Phase 3: All modules check if any errors occurred during arbitration. On error detection, $AC0^*$ and $AC1^*$ are both asserted. In phase 3, $AC1^*$ may be asserted for the following reasons: error occurrence, pre-emption (explained later), or pass1 of two pass arbitration. Modules transition to phase 4 upon the release of AS^* by the current master. This signals that a bus transaction has just ended and a transfer of tenure may occur.

Phase 4: If $AC0^*$ and/or $AC1^*$ are asserted, all modules immediately proceed to phase 5. If $AC0^*$ and $AC1^*$ are both released, all modules wait for the current master to release the bus within one microsecond. If the current master is ready to transfer tenure, it asserts ar^* . If the current master has additional transfers to perform, the master may cancel transfer of tenure by asserting $AC1^*$ and then asserting ar^* to transition onto the next phase.

Phase 5: If $AC0^*$ and $AC1^*$ are still both released, then a transfer of tenure takes place. If $AC0^*$ and/or $AC1^*$ are asserted, no transfer of tenure occurs and the arbitration cycle is ended by transitioning to Phase 0.

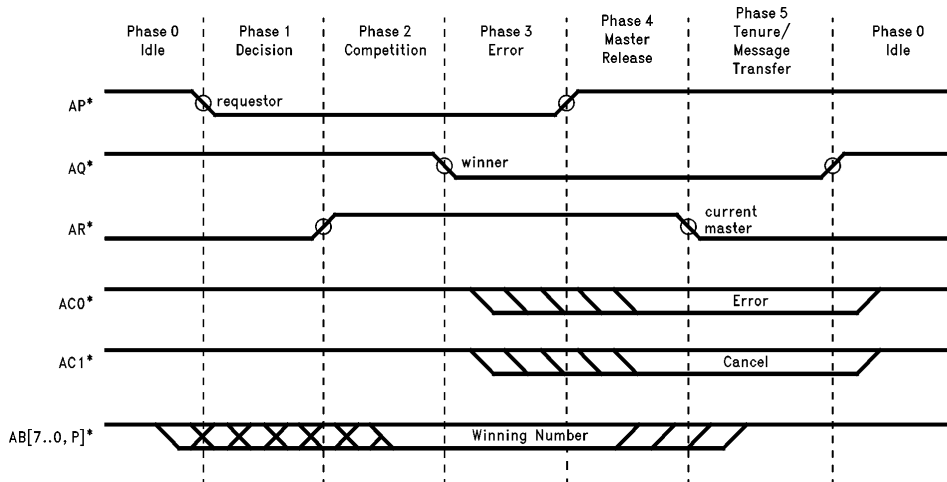


FIGURE 1. Distributed Arbitration Cycle

TL/F/11485-1

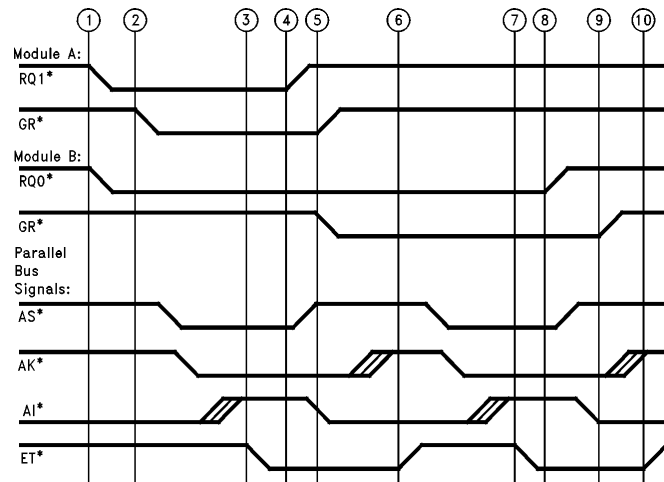
Central Arbitration

Unlike distributed arbitration, the central arbitration scheme is quite simple. As mentioned earlier, there are two request signals, RQ0* and RQ1*. RQ1* has higher priority than RQ0*. A module can gain tenure of the bus by using either (both) of the request signal(s). Various combinations can represent different priority levels. The request signal(s) remain asserted until the module becomes bus master, and is in its last transfer on the parallel bus. When both request signals are asserted, they are released at the same time.

The Central arbiter asserts grant in response to the request. Grant indicates to a module that it is the master elect. The module may then begin parallel bus transfers when it detects ET* released. The central arbiter maintains GR* asserted until the module releases the request.

The Central arbiter asserts PE* to indicate to the master that a preemptive condition exists. The arbiter releases PE* when it releases GR*.

The following example is taken from IEEE 896.1 specification.



TL/F/11485-2

FIGURE 2. Central Arbitration

Note 1: Modules A and B each assert their arbitration request. For this example module A's request is a higher priority than module B's request.

Note 2: The central arbiter asserts gr* to module A.

Note 3: The current master, module A, asserts et* after beginning a transaction.

Note 4: During its last transaction module A finishes its need for bus tenure and releases its request.

Note 5: Upon detecting that Module A has released its request the central arbiter releases the grant to Module A and asserts the grant to Module B. Module B at that point becomes the master elect.

Note 6: The master, Module A, after releasing the parallel bus signals releases et* to indicate to the master elect that it is the new master.

Note 7: After beginning its transaction the current master, Module B, asserts et*.

Note 8: During its last transaction Module B finishes its need for bus tenure and releases its request.

Note 9: Upon detecting that Module B has released its request the central arbiter releases the grant to Module B. Since no requests are pending the central arbiter issues no grant.

Note 10: When the current master, Module B, has released the parallel bus signals it releases et*.

Preemption

Both arbitration methods allow a higher priority module to preempt the current arbitration process when it needs access to the Futurebus+. For instance, in distributed arbitration, while the master elect is waiting for the current master to complete its tenure, the master elect can get preempted by a higher priority module (AC1* asserted). This will cancel transfer of tenure for the current arbitration cycle and initiate a subsequent cycle to reevaluate all requests for the bus. In the central arbitration case, the central arbiter will assert the preempt signal, PE*, when a preemptive condition is present. During preemption, if the current master removes its request for the bus, the central arbiter will immediately remove the grant signal and reevaluate all requests for the bus.

ARBITRATION MESSAGES

In addition to implementing preemption for higher priority modules to reduce bus latencies, both schemes support arbitration messages. Critical system messages or targeted messages can be quickly sent or received on the distributed arbitration bus lines: AB[0:7]. Any module can use the arbitration bus lines to send a message, independent of transactions that may be carried out on the parallel protocol bus. Up to 128 messages can be encoded on the arbitration bus lines. IEEE 896.1 only specifies the powerfail encoding. The remaining messages are to be encoded as per the specific needs of the system. There are only slight differences in format for sending a message using distributed arbitration or central arbitration.

In distributed arbitration, messages are sent during an arbitration competition cycle and compete same as normal competition cycles. During all competition cycles, all modules receive the winner's arbitration number. Thus during an arbitration competition cycle to send a message, the message is transferred but, no exchange of tenure occurs. Two arbitration competition cycles are needed to send the encoded message. To distinguish it as a message from an actual request for the bus, the arbitration competition number consists of all ones during the first arbitration competition cycle (pass1), and the message is encoded as the arbitration competition number during the second arbitration competition cycle (pass2).

In central arbitration, messages are sent using the distributed arbitration protocol in the same manner as described above. The arbitration message is encoded and sent using the AB[0:7] lines. Two message formats are specified in the central arbitration environment. The first message format is for system messages (General Messages) that are sent during the first competition cycle. Two arbitration cycles are not needed, like in distributed arbitration, since the arbitration bus, AB[0:7]* is used only to send/receive messages. (The central arbiter handles all requests for the Futurebus+.) The second message format is to change each module's assigned priority of the request signals (rq0*, rq1*) in the central arbiter (Central Messages). These messages are specifically targeted to the central arbiter in the system. In this case, during the first competition cycle (pass1), the new priority level of the signal is sent. During the second competition cycle (pass2), the module, and specific request signal that is to be updated is indicated.

Table II. Distributed Arbitration Number Message Format

Bit	CN7	CN6	CN5	CN4	CN3	CN2	CN1	CN0
Pass1	1	1	1	1	1	1	1	1
Pass2	1	AM6	AM5	AM4	AM3	AM2	AM1	AM0

Table III. Central Arbitration Number Message Formats

Bit	CN7	CN6	CN5	CN4	CN3	CN2	CN1	CN0
General Messages								
Pass1	1	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Central Messages								
Pass1	0	PR7	PR6	PR5	PR4	PR3	PR2	PR1
Pass2	1	PR0	RQ	GA4	GA3	GA2	GA1	GA0

PROFILES

In IEEE 896.2 various application specific profiles are defined which specify the particular arbitration scheme that is to be used. The following profiles are in 896.2: Profile A (General Purpose Multiprocessor Systems), Profile B (General Purpose I/O Systems), and Profile F (Very High Performance Multiprocessor). Currently, Profile M (Military) and Profile T (Telecommunications) are being defined in the

committee. In short, for Profiles B and F: central arbitration is the required method to achieve reduced arbitration latency and thus higher system performance in I/O and/or cache based environments. For Profile A: central arbitration is the preferred choice. Profiles M and T are leaning towards central arbitration with distributed arbitration required for back-up for fault tolerance reasons. This guarantees that there is no single point of failure and visibility of the winner for error recovery incidents.

Profile	Central Arbitration	Distributed Arbitration	Arbitrated Messages
Profile A	Optional	Optional	PowerFail arbitrated message required, all others are optional
Profile B	Required @ Power Up Defaults to Central	Optional	PowerFail message required, range: 0x60–0x7F for subrack power fail notification (and guaranteed to be generated by the system); generation and interpretation of other messages is optional, range: 0x00–0x5F.
Profile F	Required	Not Supported	PowerFail message is required, for monarch capable modules the backplane delay arbitrated message is required, all other messages are optional.
Profile M	In Committee	In Committee	In Committee
Profile T	In Committee	In Committee	In Committee

NATIONAL'S DISTRIBUTED ARBITRATION SOLUTION

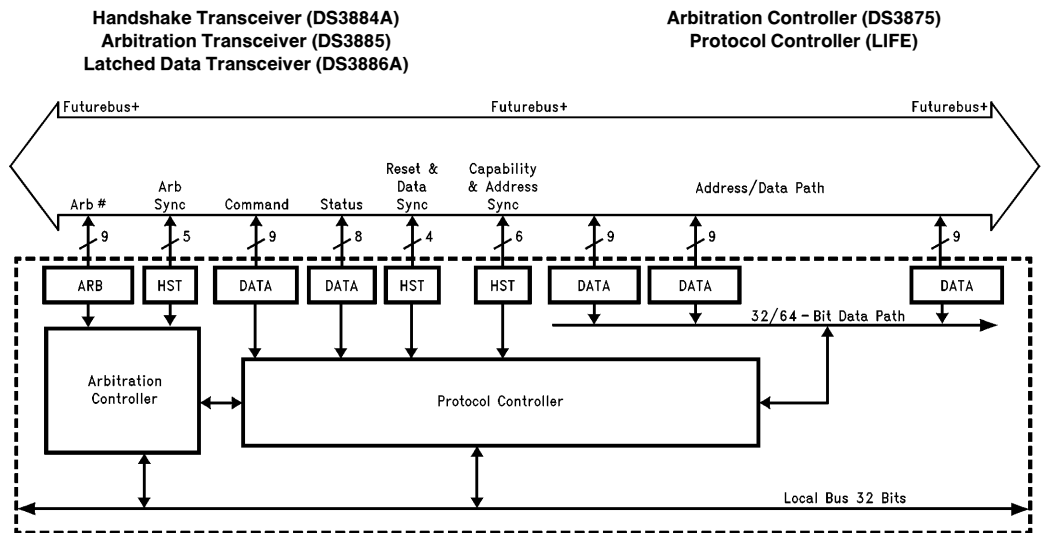


FIGURE 3. NSC's Futurebus+ Chipset Block Diagram (Distributed Arbitration)

TL/F/11485-3

National's DS3875 Futurebus+ Arbitration Controller, the DS3885 Futurebus+ Arbitration Transceiver, and DS3884A BTL Handshake Transceiver implement the complete requirements of the IEEE 896.1 distributed arbitration protocol. These devices were designed before the 896.1 specifications were finalized. Thus the Arbitration Controller has additional features than those required in the specifications. For instance, idle bus arbitration, Fast/Slow mode, and the Restricted mode are features that are still available, if desired. All features that are no longer in 896.1 are user programmable options. Thus, the DS3875 can be configured to operate in the 896.1 compliant mode. For instance, parking is register selectable, and either a one pass or two pass arbitration number may be programmed (Formally known as the Unrestricted Mode).

For performance, NSC designed the DS3885 Futurebus+ Arbitration Transceiver to incorporate the competition contention logic needed for the Arbitration Number signal lines: AB[0:7]*, AB[P]*. The DS3875 generates all the control signals needed for the DS3885 during the arbitration cycles. The DS3885 can be considered a slave to the DS3875 Arbitration Controller. Also in conjunction with these devices, the DS3884A BTL Handshake Transceiver that has selectable Wired-OR receiver glitch filtering and regular non filtered receiver outputs is used for the Arbitration Sequencing and Arbitration Condition signal lines: AP*, AQ*, AR*, AC0*, AC1*. These signals determine the current phase (progress) and the current status of the arbitration cycle. The DS3875 can be considered a slave to the LIFE. The LIFE generates the bus request signal to the DS3875 and the DS3875 responds with the bus grant signal. For additional information, please refer to the appropriate datasheets or contact the Interface and Peripherals Applications Group.

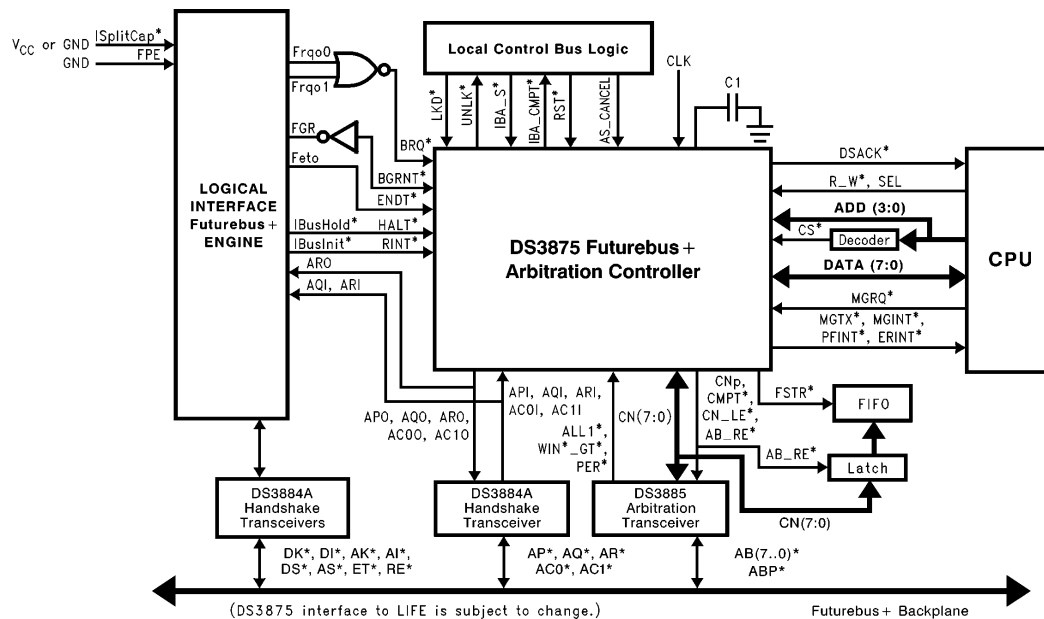


FIGURE 4. DS3875 Interface

TL/F/11485-4

A HIGHLIGHT OF THE DS3875 FEATURES FOLLOWS:

- Implements the complete requirements of the IEEE 896.1.
- Can be easily programmed with either a signal or two pass competition number. The arbitration number may be changed by loading a new number into the controller's internal registers. During a subsequent competition cycle, the new number will be in effect.
- The Parking feature is implemented for cases when the current bus master can get the bus when no other module desires to compete for the bus.
- Built in 1 μ s timer for use in the arbitration cycle.
- User programmable 16 arbitration competition settling time delays.
- Built in PLL for accurate delays. The PLL accepts clocks from 2 MHz to 40 MHz in steps of 1 MHz.
- Signal to unlock slave modules on transfer of tenure. Auto unlock through a dummy cycle if the current master locked resources.
- Programmable delay for releasing ar* after issuing compete. This is to ensure the assertion of the arbitration number during competition, before the release of ar*.
- Read/Write facility with data acknowledge for the host to load arbitration numbers, an arbitration message, and control registers.
- On-chip parity generator to indicate error occurrence and arbitration message received. Interrupts cleared on a register write. Error status is available in a separate status register.
- A special output pin to indicate that a PowerFail message was received.
- Hardwired register to hold the first word of the arbitration message.
- FIFO strobe provided to store more than one arbitration message externally to prevent overrun.

- Bus initialization, system reset and Live-insertion supported. (The logic to detect these conditions must be implemented externally).
- Testability in the form of reading from key registers which include the STATE, MCW, 1 μ s timer and programmable input clock divider.

NATIONAL'S CENTRAL ARBITRATION SOLUTION

National recommends a custom solution for implementing the system central arbiter. The central arbitration scheme lets the system designer determine the conditions for pre-emption, and **protocol** for round robin and/or priority based implementation for arbitration between the various modules. Recalling that 896.1 **only specifies** that rq1 is of greater priority than rq0, the arbiter can be designed and tailored for the exact needs of the specified system to achieve optimum performance. Also since the arbiter design is quite simple, quick request to grant times have been achieved. A flexible central arbiter controller design would be needed to address the majority of the possible implementations since the arbiter can be different for each system. This may result in a degradation in speed since the controller will no longer be optimized. In summary, a discrete central arbiter can be easily designed by the system designer to get optimum performance at low cost.

Currently National has an example of a Futurebus+ MAPL (Multiple Array Programmable Logic) based central arbiter design included in this book (AB-XXX). The implementation consists of two MAPL128, one PAL20L8 (a PAL16L8 can be substituted), and two 74AS00 (octal 2 input nand gates). For the signals to go through the backplane, two DS3886 9 Bit BTL Latched Data transceivers are also needed. The design is for 10 nodes (RQ0*, RQ1*, and GR* for each node). All the request RQ1* signals are assigned the same and higher priority level than the RQ0* signals. Within each of the two priority levels, the round robin protocol is implemented. The bussed PE* signal is generated whenever a higher priority request is present compared to the request with the current grant. With this design, request to grant is approximately 60 ns to 100 ns depending upon whether the particular grant signal uses one or two levels of logic.

NATIONAL'S CENTRAL ARBITRATION SOLUTION

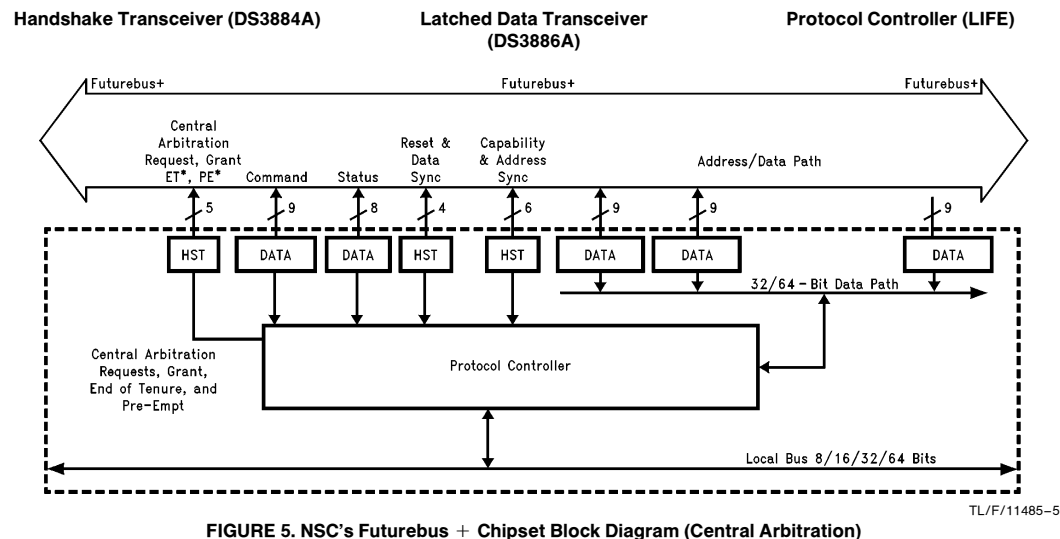


FIGURE 5. NSC's Futurebus+ Chipset Block Diagram (Central Arbitration)

This design can be expanded for systems with greater than ten modules (Futurebus+ allows up to 14 modules maximum). In this case, the logic in the MAPL devices will need to be reorganized since currently the device is used almost to maximum capacity.

Each board in a system communicates to the central arbiter through the central arbitration signals: RQ0*, RQ1*, GR*, PE* and ET*. These all interface to the LIFE through the DS3884A Handshake Transceiver. The LIFE contains **board level** central arbitration logic.

Looking at the basic central arbitration message sending/receiving requirements for the profiles and most Profile B designs: all modules should be able to receive the power fail message. For this purpose, a DS3884A BTL Handshake Transceiver and a PAL may be used to decode this message. Most designs that will be implementing this basic requirement will not need the DS3875 for decoding this particular arbitration message.

For systems that implement additional arbitration messages, the DS3875 can be used for central arbitrated message sending and receiving. At the time the DS3875 was designed, central arbitration was not even yet specified in the 896.1 specifications. Thus, additional logic will be needed to send and receive central arbitrated messages in the

appropriate central arbitration message formats. So, to get the desired central arbitration message passing formats to be correct, the DS3875 will need to treat the message like a request for the bus. (All real requests for the bus are directed to the central arbiter in the system). Thus, external logic is needed to generate the signals the DS3875 would normally generate in the message case. The DS3875 can send 64 (of the 128) possible system general messages successfully without any possibility of the message becoming altered by the round robin bit. In the arbitration number, one bit is allocated for keeping track of the round robin which automatically gets updated. Thus, not to have any effects from this bit, we need to code our messages so as not to overlap the messages which may appear to the arbiter in the same priority class. Again for the same reason, when changing the priorities of each module, each module should be assigned a different priority so that the round robin bit does not get altered. For example, of the 256 possible priorities that can be assigned, the DS3875 will allow 18 priorities to be allocated to each module without any overlap in a Futurebus+ 14 slot backplane environment. In summary, we can account for these slight variations in the message passing formats from distributed arbitration to central arbitration by actually using the DS3875 in the normal bus request format instead of the message request format.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.