3.0 DARF Description

3.1 Introduction

This section describes the Data Address Register File (DARF). A general architectural description of the DARF is provided, followed by detailed descriptions of the signal pins and major DARF functional modules. Descriptions include:

- Signals generated by each module
- Requirements of each module
- Interactions between modules
- Relevant module performance

Traditional VME card design forces the CPU to enter wait states whenever it requires the VMEbus, and forces the VMEbus to wait for access to the local bus, wasting VMEbus and local bus bandwidth. The DARF provides FIFOs that decouple the local bus and VMEbus. The FIFOs immediately accept writes from one bus, release that bus for other use, and then finish the write to the destination bus. The high performance decoupled mode generates standard D32 VMEbus cycles in the DARF32 and the DARF64, while the DARF64 provides BLT and MBLT cycle support in addition to the standard D32 VMEbus cycles. A message passing FIFO linked to the Location Monitor provides Interprocessor communication.

The major differences between the DARF32 and the DARF64 are summarized in Table 3.1. To upgrade from a DARF32 to a DARF64, $\overline{\text{VLWORD}}$ must be grouped with other address lines so that it is output enabled with VADDROUT.

3.1.1 Table Shading Convention

The convention used in the following sections to highlight differences is as follows:

TEXT	TEXT	TEXT
This highlights the control bit(s) that are being defined	This represents new control bit(s) in the DARF 64 only	This represents DARF 32 and 64

Table 3.1 : Comparison of DARF32 and DARF64 Characteristics

DARF32	DARF64
25-30 Megabyte/sec VMEbus transfer rate	60-70 Megabyte/sec VMEbus transfer rate
Master D8, D16, D32	Master D8, D16, D32, D64
Slave D08(EO), D16, D32	Slave D08(EO), D16, D32, D64
Master A16/A24/A32	Master A16/A24/A32/A64
Slave A24/A32	Slave A24/A32/A64
7 deep transmit/receive FIFO's	15 deep transmit/receive FIFO's
DMA write capability	DMA read *and* write capability
	BLT/MBLT Slave
	BLT/MBLT Master - DMA initiated
	Local bus burst mode for BLT/MBLT cycles
	68040 local slave interface
	Byte swapping disable
	Unaligned cycle support
	RETRY functionality

The DARF provides the address and data path to link the local bus to the VMEbus, plus the logic required to perform data transfers as a master or slave on both VMEbus and local bus. It is capable of two operational modes: decoupled and coupled, also known as atomic mode. The decoupled mode offers much higher data transfer rates than coupled mode, while the coupled mode offers support for software development, diagnostics, on-card EPROM programming and other applications.

The base addresses of the DARF VMEbus A32 and A24 slave images are programmable, as are the sizes of each image. The DARF64 provides an A64 slave image in addition to the A32 and A24 slave images. Access protection is provided to write protect, or to read and write protect, a programmable amount of the DARF VMEbus slave image. Accesses to a card's own slave image are reflected directly back to memory, without use of the VMEbus.

To access the local bus from the VMEbus, the DARF uses a memory map that divides the 4 gigabyte 32-bit local address space into 32 pages of 128 Mbytes each. Each page can be mapped to the VMEbus or to an external device, such as a VSB interface. The DARF operates as an address decoder when accesses are mapped to an external device, and a select pin, $\overline{\text{VSBSEL}}$, is asserted for the duration of the cycle. The DARF64 provides an A64 capability which overlays the A32 address space.

A dual bus DMA controller in the DARF32 transfers longwords from local bus address space (usually memory) to any address on the VMEbus. It does not implement reads from the VMEbus to local RAM.

The DARF64 also provides a dual bus DMA controller, but can transfer data to or from the VMEbus. This DMA controller performs D64, D32 and D16 data transfers to and from the VMEbus.

The Location Monitor queues data on the VMEbus when the topmost longword of the DARF slave image is written into the message queue. The DARF generates an interrupt while there are entries in the FIFO.

3.2 List of Features

3.2.1 DARF32

- Complete VMEbus address and data interface, except buffers
- Decoupling of CPU and VMEbus
- Sustained 25-30 Megabyte/second VMEbus transfer rate
- Selectable coupled or decoupled mode
- Programmable A32 and A24 slave image bases and sizes
- Programmable access protection
- Integral DMA for local RAM to VMEbus transfers
- Location monitor with 31-longword deep data FIFO
- Low power CMOS implementation
- 224 Pin Grid Array package
- 240 Plastic Quad Flat Pack
- Available in MIL-STD 883 Class B version

3.2.2 DARF64

- 64-bit Data/Address Register File (DARF64)
- Pin compatible with DARF32
- IEEE-1014/VMEbus Rev D compatible
- Full master/slave A64/D64, A32, A24/D32, D16, D08(EO) interface
- Master A16/D16, D08(EO) capabilities
- A32/A24 BLT
- A64/A32/A24 MBLT
- High performance bus bandwidth:
  30 MB/s - Standard cycles
  35 MB/s - BLT Mode
  70 MB/s - MBLT Mode
- 15 deep transmit and receive decoupling FIFOs
- Direct connection with 68020/030
- Support for 68040 and RISC processors
- Location monitor with 31 deep message FIFO for inter-process communication
- Local bus burst mode

- Programmable
  Coupled and decoupled modes
  A64 master and slave base address
  A32 slave image base address and size
  A24 slave image base address and size
  Slave image access protection
  VMEbus/VSB routing
- Integral A64, A32, A24/D64, D32, D16 DMAC
  DMAC initiated read or write cycles
  Discrete, block, multiplexed block transfers
  Up to 4 megabyte block length
  Automatic address phase insertion
- BI-mode : bus isolation mode
- Low power CMOS implementation
- Card and system testability support
- 224 pin PGA package
- 240 pin Plastic Quad Flat Pack
- Commercial, military temperature and MIL-STD-883 processed versions

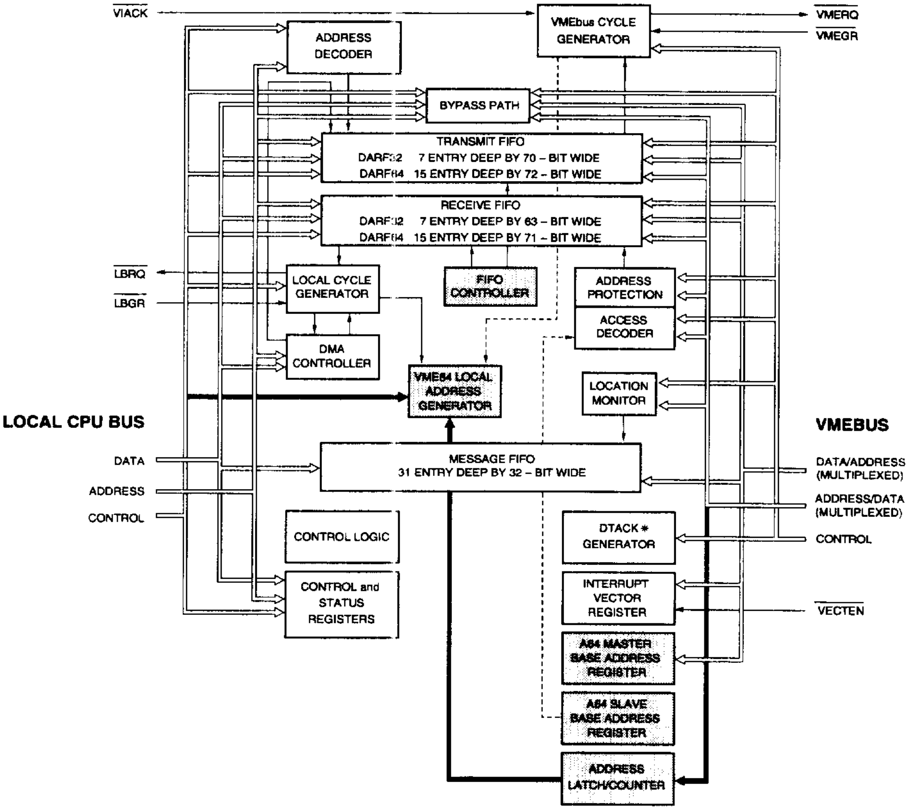

VIACK
VMEbus CYCLE GENERATOR
VMERQ
VMEGR
ADDRESS DECODER
BYPASS PATH
TRANSMIT FIFO
DARF32 7 ENTRY DEEP BY 70 - BIT WIDE
DARF64 15 ENTRY DEEP BY 72 - BIT WIDE
RECEIVE FIFO
DARF32 7 ENTRY DEEP BY 63 - BIT WIDE
DARF64 15 ENTRY DEEP BY 71 - BIT WIDE
LBRQ
LBGR
LOCAL CYCLE GENERATOR
FIFO CONTROLLER
ADDRESS PROTECTION
ACCESS DECODER
DMA CONTROLLER
VME64 LOCAL ADDRESS GENERATOR
LOCATION MONITOR
LOCAL CPU BUS
VMEBUS
DATA
ADDRESS
CONTROL
MESSAGE FIFO
31 ENTRY DEEP BY 32 - BIT WIDE
DATA/ADDRESS (MULTIPLEXED)
ADDRESS/DATA (MULTIPLEXED)
CONTROL
CONTROL LOGIC
DTACK* GENERATOR
CONTROL and STATUS REGISTERS
INTERRUPT VECTOR REGISTER
VECTEN
A64 MASTER BASE ADDRESS REGISTER
A64 SLAVE BASE ADDRESS REGISTER
ADDRESS LATCH/COUNTER

Figure 3.1 : DARF Block Diagram

3.3 Architecture

3.3.1 Decoupled CPU Concept

Decoupling two resources removes restraints that each resource places on the other. In the case of the DARF, the CPU is decoupled from the VMEbus during write cycles and the VMEbus is decoupled from the destination memory during write cycles. Read cycles are not decoupled using the DARF32. A decoupled read functionality is provided by the DARF64 when operating as the bus master using DMA .

Combining decoupling principles with quick VMEbus master and slave interfaces results in more useable bandwidth on the VMEbus. The DARF immediately accepts and acknowledges the write cycles received, which allows the VMEbus cycle to terminate and free the bus for use by another card. The DARF queues the write cycle in its receive FIFO and performs the cycle on the local bus of the destination card as soon as that bus is available.

Decoupling increases local bus bandwidth, thereby increasing CPU performance. Similarly, write cycles to the VMEbus are completed immediately, without the usual arbitration overhead for the VMEbus on the source module, and then for the local bus on the destination module.

Decoupling, as used in the DARF, hides the latencies associated with bus ownership delays, unless the system is highly loaded and the FIFOs fill up. When the FIFOs are full, and a CPU attempts to do a VMEbus write cycle, then local bus wait states occur until at least one FIFO entry becomes available. When the receive FIFO is filled, the VMEbus waits until at least one entry is available.

Data coherency is maintained, since all write cycles that leave a given card are queued

The interrupt vector source register of the VMEbus Interrupter is located in the DARF, and works in conjunction with the interrupt generator in the ACC (see paragraph 2.10). The ACC can be programmed to assert one of the VMEbus interrupts and signal the DARF to provide the interrupt vector and DTACK✳ signal when required.

The DARF32 status and control register block contains sixteen 32-bit wide registers, while the DARF64 block contains twenty registers. These are used to program the DMA, status indicators, mode selection, slave image configuration and card diagnostics.

The DARF64 supports BLT cycles and MBLT cycles. Also, a local burst mode is provided, that can be configured for 4 to 32 longword bursts on the local bus.

When BLT or MBLT cycles are enabled, they may be mapped into burst cycles on the local bus. BLT and MBLT VMEin write accesses to the RXFIFO may also be enabled as bursts on the local bus. If bursts are not enabled on the local bus, MBLT cycles are mapped into two indivisible longword cycles.

The DARF is designed to work with on-card 68020 or 68030 processors. Further, any other local bus master or a CPU may be expected to retry cycles and to perform bus sizing when requested by the DARF (if non-aligned or over-size transfers are attempted). Over-size transfers are 24 or 32 bit accesses to the 16 bit data space of the VMEbus.

Dynamic bus sizing of the DARF32 forces cycles to be subdivided into smaller bus cycles to accommodate the space available. For example, a D32 can be divided into two D16 transfers to access a D16 space. Access to a D16 space by a local bus master will result in the DARF signalling the local bus master as a D16 slave. Similarly, unaligned transfers can be divided into multiple cycles.

The DARF64 can disable the subdividing feature, using the BUSSIZ bit in the MODE register. When dynamic bus sizing is disabled, the DARF64 attempts unaligned transfers as a 32 bit cycle, and does not subdivide them into multiple cycles. Similarly, a D16 transfer is attempted as a D32 transfer

External buffers and transceivers are required to connect the DARF to the VMEbus.

3.3.3 DARF Functional Blocks Overview

The major functional blocks of the DARF shown in Figure 3.1, are listed below with a brief description of secondary modes and related issues.

- Memory Map

  Memory map implemented by the DARF after receiving a $\overline{\text{VMEOUT}}$ signal

  Determines whether the DARF uses the VMEbus or reflects the cycle to VSB or local memory

  Addressing mode selected, whether 64, 32, 24, or 16-bit, if the VMEbus is being used

- Master VMEbus, VSB Interfaces

  Requesting and using the VMEbus

  Coupled and decoupled mode differences

  Reflecting cycles to the VSB

- Slave VMEbus Interface

  Slave response of the DARF to the VMEbus, including programmed base address, access protection, access by the local CPU to its own slave image and operations in coupled and decoupled modes
- Local Bus Interface

  Local bus mastership — obtaining local bus, using and releasing it
  Operation of the DMA controller
  DARF64 Local Bus Burst Mode
  68040 Mode
- Location Monitor and FIFO

  Operation of the Location Monitor, FIFOs, and BI-mode protocol
- VMEbus Interrupter

  Operation of the VMEbus interrupter and its companion section in the ACC
- Control and Status Registers

  Accessing control and status registers
- BI-mode Operation

  Entering and exiting BI-mode
  DARF response while in BI-mode
- Test and Diagnostic Modes

  Operating modes and registers provided for error recovery and diagnostics

3.4 Memory Map

The DARF memory map determines whether an off-card cycle is performed on the VMEbus or on an external device (such as the VSB), as well as the addressing mode and data width to be used. The programmed slave image may also redirect the cycle to local RAM without using the VMEbus.

The DARF does not directly decode off-card bus requests, rather, a select input named $\overline{\text{VMEOUT}}$ is asserted that enables the DARF to decode a bus request. The $\overline{\text{VMEOUT}}$ signal comes from the address decoder associated with the local CPU. It is asserted throughout the address range(s) where the CPU wants access to VMEbus or VSB. The DARF memory map is shown in Figure 3.2. The memory map for a sample single board computer (SBC) with AVICS and VSB is shown in Figure 3.3

The DARF supports a 32-bit local address, giving a 4 gigabyte range. This is divided into 32 pages of 128 megabytes. Each page can be selected to be a VMEbus region or a VSB region. If a VMEbus region, the DARF performs the cycle on the VMEbus or to its own VMEbus image. If a VSB region, the DARF redirects the cycle to a VSB interface or other resource by asserting its $\overline{\text{VSBSEL}}$ pin.

Bus mapping is controlled by the BUSSEL register, shown in Table 3.2 VMEbus/VSB Select register.

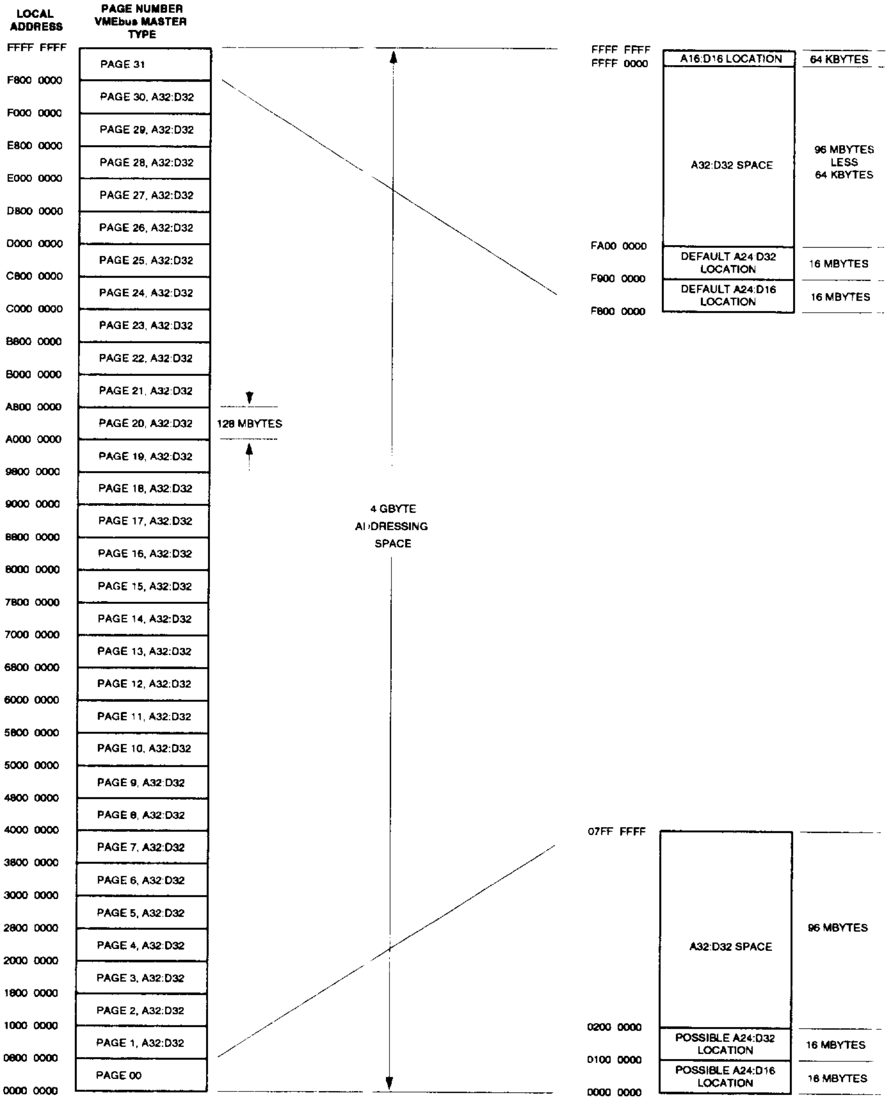


Figure 3.2 : DARF Memory Map

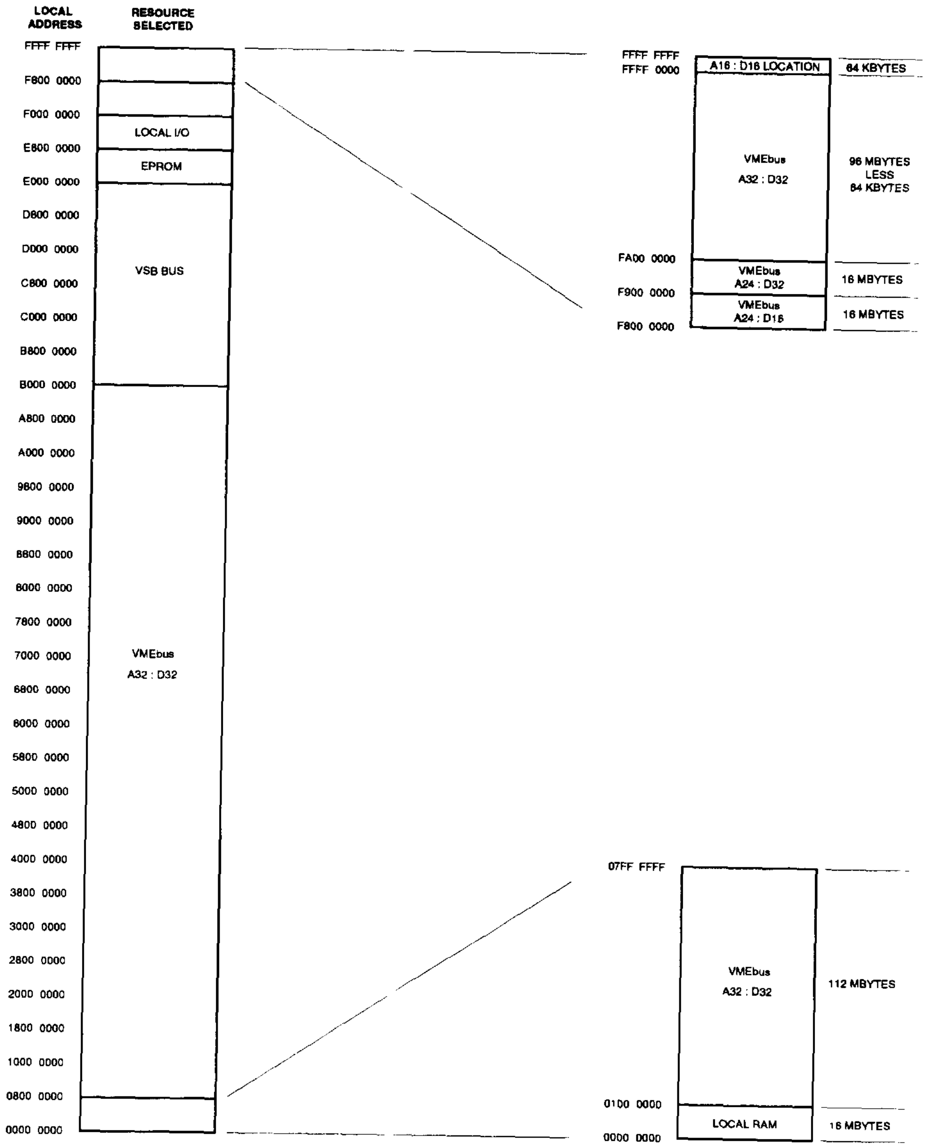


Figure 3.3 : Memory Map for Sample SBC with AVICS and VSB

Table 3.2 : VMEbus/VSB Select Register

Bits	Function
31-24	VSBEN (VMEbus/VSB Select bits) Byte 3
23-16	VSBEN (VMEbus/VSB Select bits) Byte 2
15-06	VSBEN (VMEbus/VSB Select bits) Byte 1
07-00	VSBEN (VMEbus/VSB Select bits) Byte 0

Bits	State	Function
VSBENx		One bit for each 32 x 128 Mbyte page
	0	VMEbus selected (default)
	1	VSB selected

Each bit in the register corresponds to a page in the address space (actual addresses are listed in the VMEbus/VSB Bus Select Register description in Appendix C). The register defaults to zero after reset, which maps each page onto the VMEbus. Setting a bit to one (1) maps the corresponding page onto the VSB.

When the CPU accesses the VMEbus, the address used by the CPU determines the address and data modes used by the DARF to perform the cycle. Addressing modes include 32, 24 or 16-bit addresses and data modes include 32 or 16-bit data paths. the DARF64 also supports 64 bit addressing and 64 bit data during DMA operations

The A16:D16 region is located at the top of the memory map, in the last 64 KB. This space can be disabled using bit A16DI in the DARF Mode Control Register (Table 3.3) in which case the A16:D16 address range becomes A32:D32.

Table 3.3 : DARF Mode Control Register

Bits	Function							
31-24	DTCSIZ	RESERVED	RMCRETRY	DMABEN	RMCPIN	BUSSIZ	SWAP	040
23-16	DPRIV	NOREL	FILL	DMAA64	MBLT	BLT	DMAD16	DMARD
15-08	FIFOBEN	BLEN *(2 bits)*		RXATOM	TXATOM	A24SLVEN	DMA24	BERRCHK
07-00	TASCON	A24P0	LPBK	DISRX	A24DI	A16DI	PROT	VINEN

Two A24 master modes are provided, A24:D32 and A24:D16. By default, these are both in Page 31 (at the top of memory), A24:D16 in the first 16 MB and A24:D32 in the second 16 MB.

This pair of A24 spaces can be relocated to page 0 (at the bottom of memory) using bit A24P0 in the Mode Control Register, or turned off , so that the address range they previously occupied becomes A32:D32. Both locations are shown in the DARF Bus Memory Map (Figure 3.2). Setting A24DI in the Mode register disables A24 addressing.

Accesses to other addresses cause the DARF to default to A32:D32 mode when using the VMEbus.

Access to A64 space requires the DMAA64 and MBLT bits in the Mode register to be set and the Master A64 Base Address Register to be programmed. When DMA operation is initiated, A63-32 will be driven onto VMEbus data lines D31-0 during the address phase of the MBLT cycle.

These mapping options affect outgoing VMEbus accesses only and do not affect accesses to the card dual-port memory from the VMEbus. Address and data modes used during card accesses are controlled by the programmed base addresses, size and mode that are used by the VMEbus master during the cycle. Addressing modes during DMA operations are programmed in the Mode register. Additional information on this subject is provided in paragraph 3.7.2 DMA Controller.

Accesses by the local CPU to its own slave image are not affected by VMEbus or VSB mapping, nor is the address or data mode affected. The local memory is selected within the programmed slave image, and acknowledge signals are provided by the memory control logic.

3.5 Master VSB and VMEbus Interfaces

3.5.1 Master VSB Interface

Master VSB accesses are not performed by the DARF — if an access is made within a page mapped to the VSB by the BUSSEL Register, the DARF asserts the $\overline{\text{VSBSEL}}$ pin until the end of the CPU cycle. A VSB interface elsewhere on the card is responsible for performing the cycle and for generating $\overline{\text{KDSACKn}}$ or $\overline{\text{KBERR}}$ to the CPU.

An external address decoder can select the VSB instead of the $\overline{\text{VSBSEL}}$ output, , or in combination with it. In address ranges where such external logic decodes the VSB, $\overline{\text{VMEOUT}}$ must not be asserted to the DARF — as the cycle may be performed on both buses (if the BUSSEL Register is incorrectly programmed) or timing to the VSB interface may be incorrect.

Because the DARF acts only as an address decoder during Master VSB accesses, any device can be connected to the $\overline{\text{VSBSEL}}$ pin. For example, $\overline{\text{VSBSEL}}$ could enable a block of local memory).

BI-mode operation with respect to the VSB is dependent on card design, though it is recommended that BI-mode operation isolate the card from both the VMEbus and the VSB. Where there is no such logic to isolate the card from the VSB, the DARF can decode accesses to the VSB and assert $\overline{\text{VSBSEL}}$ while in BI-mode.

3.5.2 Master VMEbus Interface

The DARF can request the VMEbus for any of the following conditions:

- Transmit FIFO contains write cycles
- CPU begins a read cycle to the VMEbus
- DARF is in coupled mode and the CPU begins a write cycle to the VMEbus
- DARF was signalled with $\overline{\text{VIACKRQ}}$ to begin a VMEbus interrupt acknowledge cycle
- DARF64 initiates a DMA read operation

A VMEbus Requester module in the ACC communicates with the DARF through the $\overline{\text{VMERQ}}$ signal and the $\overline{\text{VMEGR}}$ signal. The DARF asserts $\overline{\text{VMERQ}}$ to request the bus ; when it is obtained, $\overline{\text{VMEGR}}$ is asserted. When the DARF requests the bus, and it is not already owned by the ACC (Release-On-Request mode), the ACC may have to arbitrate for it.

The ACC to DARF protocol allows the DARF to give up the bus by negating $\overline{\text{VMERQ}}$. The ACC can also request that the DARF release the bus, by negating $\overline{\text{VMEGR}}$. The state machines in each device remain in step by not starting a new request/grant interaction until the previous one has ended. The ACC does not assert $\overline{\text{VMEGR}}$ until the DARF negates its previous $\overline{\text{VMERQ}}$ and the DARF does not assert $\overline{\text{VMERQ}}$ until $\overline{\text{VMEGR}}$ is negated.

The DARF negates $\overline{\text{VMERQ}}$ early in its last cycle on the VMEbus, which allows the ACC to release BBSY✳ early. This allows arbitration for the VMEbus to overlap with the data transfer cycle.

The ACC can be programmed to limit the time that the DARF uses the bus, ranging from immediately off, to 2, 4 or 8 µs, or to unlimited use. When the limit is reached, the ACC negates $\overline{\text{VMEGR}}$, forcing the DARF to complete its current cycle (or one more, if the DARF had already selected the next cycle from the transmit FIFO). After the DARF has negated $\overline{\text{VMERQ}}$, the ACC releases ownership of the VMEbus.

The ACC can also request the DARF to give up the bus for other reasons, including:

- Pending VMEbus ownership requests (Release-On-Request mode)
- Receipt of a BCLR✳ from the system arbiter if Bus Clear recognition is enabled (Bit BCEN in ACC Register VREQ is set to one (1))
- $\overline{\text{L7IMEM}}$ to the ACC is asserted
- Entry into BI-mode

If the DARF is waiting for VMEbus ownership to perform a coupled operation when some other VMEbus card begins a coupled operation to the local memory through the DARF, then the DARF asserts $\overline{\text{LBRQ}}$ to the ACC and requests that the local CPU retry its cycle by asserting $\overline{\text{KBERR}}$ and $\overline{\text{KHALT}}$. Examples of these coupled operations are:

- Write while in the coupled mode
- Read or RMW
- Interrupt acknowledge

This forces the CPU off the bus and allows the incoming cycle to be completed first, resolving the deadlock. The 68020 does not retry read-modify-write cycles, so the DARF asserts only $\overline{\text{KBERR}}$ in this case and the exception handler initiates a new read-modify-write cycle. The 68030 does retry read-modify-write cycles. Bit RMCRETRY in the DARF64 MODE register can be set so that the DARF64 will terminate deadlocked RMW cycles by asserting both $\overline{\text{KBERR}}$ and $\overline{\text{KHALT}}$.

While performing bus transactions from the FIFO, the DARF uses address pipelining on the VMEbus. Addresses for the next cycle are placed on the bus after the slave asserts DTACK*. The DARF32 generates and accepts only standard cycles. The DARF64 generates and accepts standard cycles. as well as BLT and MBLT block transfers.

Both BI-mode and local reset cause the DARF to immediately end any VMEbus cycle in progress without waiting for an acknowledge , or ensuring standard bus timing, after those signals are asserted. When BI-mode is entered, any cycles remaining in the transmit FIFO are lost as the CPU cannot unload them from the FIFO, nor are they completed when the DARF leaves BI-mode.

3.5.3 Address and Data Spaces

Separate VMEbus address and data spaces are provided by the DARF to ensure compatibility with other cards used in a system. The local CPU determines the mode to be used by accessing the VMEbus within a specific address range; the DARF then generates the appropriate address modifier codes from the address and CPU function codes. DARF cycles conform to VMEbus specifications shown in Table 3.5.

Two 24 bit address (A24) areas are provided, one for 32-bit data and one for 16-bit data. These are shown on the DARF memory map (Figure 3.2). Within these areas, the DARF sets address bits 31 – 24 to one (1) and uses address modifier codes to indicate standard addressing. This pair of A24 spaces can be relocated lower in the local memory space if so desired.

The sixteen bit (A16) address space is at the top of the local address space. If that space is enabled, the DARF performs the cycle on the VMEbus after setting address bits 31 through 16 to one (1) and using address modifier codes to indicate short addressing.

Data sizes are formed with the cooperation-operation of the local 68020/030. If an aligned access is made in a D32 or D16 space, the transfer is accepted by the DARF and the appropriate combination of LWORD, DS1, DS0 and A01 are used on the VMEbus for the number of bytes (1 through 4) and offset (0 through 3) within the longword addressed.

If the CPU generates a non-aligned transfer, or if an access is made to a D16 defined area on the VMEbus, the DARF responds to the CPU as a D16 slave. This causes the CPU to break the cycle into as many cycles as needed to transfer the data, if the transfer cannot be performed in the first two aligned bytes.

The DARF64 can disable dynamic bus sizing with the BUSSIZ bit in the MODE register. Dynamic bus sizing is discussed in the Motorola MC68020 User's Manual (Section 5.1.1) and MC68030 User's Manual (Section 7.2.1).

Address modifier codes generated by the DARF, and VMEbus address modifier codes, are given in Table 3.4. The address space is determined by the address range which is used to access the VMEbus.

In decoupled mode, the DMA within the DARF32 supports D32 VMEbus and D32 local bus cycles only. The DARF64 supports D16 cycles and D64 MBLT cycles, in addition to the D32 cycles.

Table 3.4 : Master and Slave Access Address Modifier Codes

CPU Function Code		Address	VME Address Modifier	
Type	**Code**	**Space**	**Code**	**Type**
User Data	1	A16	29	Short non-privileged data access*
User Program	2	A16	2A	Short non-privileged program access
Supervisor Data	5	A16	2D	Short supervisory data access*
User Data	1	A24	39	Standard non-privileged data access
User Program	2	A24	3A	Standard non-privileged program access
Block Transfer	3	**A24**	**38**	**Standard non-privileged 64-bit block transfer**
Block Transfer	3	**A24**	**3B**	**Standard non-privileged block transfer**
Block Transfer	3	**A24**	**3C**	**Standard supervisory 64-bit block transfer**
Block Transfer	3	**A24**	**3F**	**Standard supervisory block transfer**
Supervisor Data	5	A24	3D	Standard supervisory data access
Supervisor Program	6	A24	3E	Standard supervisory program access
User Data	1	A32	09	Extended non-privileged data access
User Program	2	A32	0A	Extended non-privileged program access
Block Transfer	3	**A32**	**08**	**Extended non-privileged 64-bit block transfer**
Block Transfer	3	**A32**	**0B**	**Extended non-privileged block transfer**
Block Transfer	3	**A32**	**0C**	**Extended supervisory 64-bit block transfer**
Block Transfer	3	**A32**	**0F**	**Extended supervisory block transfer**
Supervisor Data	5	A32	0D	Extended supervisory data access
Supervisor Program	6	A32	0E	Extended supervisory program access
Block Transfer	3	**A64**	**00**	**64-bit block transfer**

Note:
* VME master only

Table 3.5 : Types Of DARF Cycles

Characteristic	VMEbus Specification	Notes
VMEbus master	A32, A24/D32, D16, D08 (EO); A16/D16, D08 (EO)	By CPU. D32 transfers cannot be done in the A16 space.
	UAT; RMW	Unaligned cycles and read-modify-write cycles can be generated.
	A32, A24/D32,	By DMAC.
	A32, A24/D16	By DMAC.
	BLT A32, A24/D64, D32, D16; BLT A64/D64	
	no ADO	Address-only cycles are not generated.
VMEbus slave	A32, A24/D32, D16, D08(EO)	DARF64 is not an A16 slave.
	BLT A32, A24/D64, D32, D16, D08(EO)	
	BLT A64/D64	
	RMW	Address-only cycles ignored. Unaligned cycles and read-modify-write cycles accepted.
	ADO, UAT	
Location monitor	A32, A24/D32, D16(E);	No A64 or A16 location monitor; word transfers on odd-word boundaries and byte transfers not detected; read cycles are ignored.
Interrupt handler	IH(1-7),D08(O)	In conjunction with the ACC
Interrupt generator	I(1-7),D08(O),ROAK	In conjunction with the ACC

Note:

Shaded specifications apply to the DARF64 only.

3.5.4 Coupled Mode

The transmit and receive channels of the DARF can be programmed to operate in coupled or decoupled mode. These channels, and the channels on the slave being accessed, may be in either mode , possibly causing differences in system performance.

The DARF Mode Control Register (Table 3.6) controls the channel modes. Bit TXATOM controls the transmit mode, and bit RXATOM controls the receive mode. These bits default to the low (0) state (decoupled mode). Bit TXATOM may be changed at any time.

Table 3.6 : DARF Mode Control Register

Bits	Function							
31-24	DTCSIZ	RESERVED	RMCRETRY	DMABEN	RMCPIN	BUSSIZ	SWAP	040
23-16	DPRIV	NOREL	FILL	DMAA64	MBLT	BLT	DMAD16	DMARD
15-08	FIFOBEN	BLEN *(2 bits)*		RXATOM	TXATOM	A24SLVEN	DMA24	BERRCHK
07-00	TASCON	A24P0	LPBK	DISRX	A24DI	A16DI	PROT	VINEN

In coupled master mode accesses, the CPU transfer is not queued by the transmit FIFO but rather the CPU is directly linked to the VMEbus after ownership is obtained. The CPU enters wait states until a DTACK* or BERR* is generated by the VMEbus slave. The DARF copies the acknowledge signal through to the CPU, which then ends the cycle.

If a BERR* is received as a result of any type of cycle, the bit VBERR flag in the Control and Status Register (Table 3.7) is set, and a $\overline{\text{VMEINT}}$ interrupt to the CPU is generated. Although the transmit FIFO accepts entries, the DARF does not perform any VMEbus master cycles until the VBERR flag is cleared.

Table 3.7 : Control and Status Register

Bits	Function							
31-24	NOT USED				DEVICE *(4 bits)*			
23-16	NOT USED							CERR
15-08	TXSHFT	RETRY	RMCRETRY	A64BARDY	BARDY	RXSHFT	RXRST	TXRST
07-00	RXHD	TXHD	DLBER	LMHD	LBERR	VBERR	DONE	DMAGO

CPU read cycles and interrupt acknowledge cycles are always coupled, as are reads and writes forming a read-modify-write operation. The cycles of an RMW are identified by the assertion of the CPU $\overline{\text{KRMC}}$ signal.

If the DARF transmit channel is in decoupled mode, a pending coupled cycle from the local CPU waits until all write cycles queued up in the FIFO are completed. Data coherency is maintained by preventing read cycles from skipping ahead of uncompleted write cycles.

3.5.5 Decoupled Mode

CPU write cycles to the VMEbus are completed with 0 wait states when the DARF transmit channel is in decoupled mode. The transmit FIFO queues the 32-bit address, 32-bit data and all control signals needed to perform the cycle on the VMEbus when the cycle arrives at the front of the FIFO. The DARF then asserts the $\overline{\text{KDSACK0}}$ and $\overline{\text{KDSACK1}}$ signals to terminate the local bus cycle. DMA transfers to the VMEbus are processed the same way as CPU write cycles — as the DMA enters the transfer into the next available slot in the FIFO when it reads from local memory. The DARF64 DMA can also transmit read cycles from the VMEbus to the Local Bus. As the DMA reads from the VMEbus, address, data and control information is queued within the receive FIFO. Write operations to local memory are completed when access to the local CPU bus is obtained.

The DARF32 FIFO is 7 stages deep and requests VMEbus ownership whenever there are entries in the FIFO. The DARF64 FIFO is 15 stages deep and may request the VMEbus whenever there are entries in the FIFO or alternatively, can be programmed to wait until the FIFO is full before requesting the VMEbus. The option is selected by the FILL bit in the Mode Control Register.

The DARF begins the cycles after it has obtained the VMEbus, operating with near minimum VMEbus specified timing to start and end each cycle. Address pipelining is used, where the address for the next cycle is placed on the bus as soon as an acknowledge for the current cycle is received. The DARF signals the ACC to release BBSY* early when performing the last cycle by negating VMERQ .

When a VMEbus Bus Error (VBERRI) is received by the DARF, it stops performing VMEbus cycles, sets VBERR (Bit 2) in the Control and Status register (Table 3.7), asserts VMEINT and negates VMERQ .

The transmit FIFO remains frozen until bit VBERR is cleared by the CPU, and then resumes sending cycles to the VMEbus. To assure data coherency, read and IACK cycles to the VMEbus are blocked until all FIFO entries are processed. The TXCTL, TXADDR and TXDATA registers in the DARF reflect the values that are present at the output stage of the FIFO when the error occurred. They may be read by the CPU to obtain the address, data and control codes in the failed write cycle. This information can be examined by software to determine the fault, or used to rerun the cycle. The failed cycle is not re-run by the DARF when the VBERR bit is cleared, as it is no longer in the FIFO, and must be regenerated.

Note that apart from the address there is no information to differentiates CPU cycles from DMA cycles.

If the FIFO is full when the CPU writes to the VMEbus, the DARF does not respond until room is available, causing the CPU to enter wait states. If an ACC is providing the local bus timeout, and the CPU waits for more than 512 µs for access to the VMEbus, then the ACC asserts a KBERR to the CPU. This condition is indicated by setting bit LTO flag in the ACC Status Register 0 (Table 3.8).

Table 3.8 : ACC Status Register 0

Bits	Function							
31-24	NOT USED							
23-16	NOT USED							
15-08	NOT USED							
07-00	CLRDOG	CLRTIK	LTO	VI1	SYFIP	ACFIP	MEMIP	NMIP

3.5.6 Read-Modify-Write Cycles

The DARF supports read-modify-write (RMW) cycles by treating them as a locked group of coupled transfers. A 68020 or 68030 identifies the group of reads and writes forming the RMW cycle by keeping its KRMC pin asserted throughout the group of cycles. There are two classes of read-modify-write cycles:

- The first class operates on one address only; the 68020/030 TAS and CAS instructions, if the operand can be fetched in a single read
- The second class operates on multiple addresses; for example, CAS on a mis-aligned operand or the CAS2 instruction.

In a cycle to the VMEbus, the DARF keeps its $\overline{\text{VMERQ}}$ pin asserted while the CPU $\overline{\text{KRMC}}$ pin is asserted, and does not respond to any request to release the bus. The arbiter must keep BBSY✱ asserted in response to the assertion of $\overline{\text{VMERQ}}$, thereby preventing other VMEbus masters from using the VMEbus during the operation. Assertion of BBSY✱ does not assure that access to the slave memory is also locked, as a CPU on the slave card can gain access to its own memory between reads and writes. Consequently, there are two other methods to lock access to the slave memory.

- Setting bit TASCON in the DARF Mode Control Register (see Table 3.9) keeps the VMEbus address strobe low while the CPU $\overline{\text{KRMC}}$ pin is low. This locks access to the slave card memory, and may be used with any single address RMW.
- If used with a multiple address RMW, the cycle fails because subsequent reads and writes in the RMW are performed on the first address.

Table 3.9 : DARF Mode Control Register

Bits	Function							
31-24	DTCSIZ	RESERVED	RMCRETRY	DMABEN	RMCPIN	BUSSIZ	SWAP	040
23-16	DPRIV	NOREL	FILL	DMAA64	MBLT	BLT	DMAD16	DMARD
15-08	FIFOBEN	BLEN *(2 bits)*		RXATOM	TXATOM	A24SLVEN	DMA24	BERRCHK
07-00	TASCON	A24P0	LPBK	DISRX	A24DI	A16DI	PROT	VINEN

To lock multiple address RMWs, the DARF asserts its $\overline{\text{VRMC}}$ pin ($\overline{\text{VRMC}}$ is usually connected through a jumper block to the VMEbus P2-B3 Reserved pin). A compatible slave card, including any card using AVICS, detects this condition and keeps ownership of the slave card local bus until $\overline{\text{VRMC}}$ is negated. TASCON does not affect assertion of $\overline{\text{VRMC}}$ and must not be set if a multiple address RMW is about to be executed. P2 pin B3, "Reserved" is defined in the Rev. D VMEbus Specification as a RETRY✱ pin. The DARF64 provides for optional definition of the $\overline{\text{VRMC}}$ pin as a RMW lock signal, as described above, or as RETRY✱. The option is selected by setting the parameter RMCPIN in the Mode Control Register.

To resolve a lock-up cycle when the CPU is attempting RMW to the VMEbus and the VMEbus is waiting to do a coupled access to local memory, the DARF must force the CPU off the bus and complete the VME-in cycle first. The 68020 can re-try normal reads and writes, but not RMWs. Therefore, if an RMW lock-up occurs, the DARF ends the cycle by asserting $\overline{\mathrm{KBERR}}$ to the CPU. Refer to the Motorola MC68020 User's Manual, Section 6.4, or MC68030 User's Manual, Section 8.2, *Bus Fault Recovery*, for details on re-running RMW cycles. For non MC680x0, the DARF64 RMCRETRY bit in the MODE register can be set to terminate RMW lockups with $\overline{\mathrm{KBERR}}$ and $\overline{\mathrm{KHALT}}$.

When BERR✳ and RETRY✳ are asserted and $\overline{\mathrm{VRMC}}$ is configured as the RETRY✳ line, $\overline{\mathrm{KBERR}}$ and $\overline{\mathrm{VMEINT}}$ are asserted on the local side, and the RETRY bit in the DCSR register will be set. When assertion of $\overline{\mathrm{KBERR}}$ and $\overline{\mathrm{VMEINT}}$ have been identified, the DCSR register should be checked for assertion of RETRY✳. If it is asserted, the local CPU should re-attempt its last VME transfer again

3.5.7 Interrupt Acknowledge Cycles

The DARF begins an interrupt acknowledge cycle on the VMEbus when its $\overline{\mathrm{VIACKRQ}}$ pin is asserted. If the DARF does not already have the bus, it asserts $\overline{\mathrm{VMERQ}}$ to request the bus and begins the cycle once $\overline{\mathrm{VMEGR}}$ is received.

The interrupt acknowledge cycle is coupled and is processed the same way as a read cycle. This means that it must wait for pending write cycles in the FIFO before it is performed. The $\overline{\mathrm{VMEOUT}}$ signal must not be asserted during an IACK cycle.

After the DARF owns the bus, it behaves like an 8-bit interrupt handler, asserting $\overline{\mathrm{VDS0}}$ and $\overline{\mathrm{VIACK}}$, and keeping $\overline{\mathrm{VDS1}}$ and $\overline{\mathrm{VLWORD}}$ high. After the interrupter has provided its vector on VMEbus D07 – D00 and asserted $\overline{\mathrm{VDTACK0}}$, the DARF transfers the vector to local data bus bits D31 – D24 and asserts $\overline{\mathrm{KDSACK0}}$ to the CPU.

The interrupt level is indicated on VMEbus lines A03 – A01. The other address bits are driven to the same level provided by the CPU, "high" for a 68020/030 during an IACK cycle.

If $\overline{\mathrm{VIACKRQ}}$ is asserted to the DARF while BIMODE is also asserted, the DARF remains isolated from the VMEbus and does not respond to the request.

3.5.8 VMEbus Mastership

The DARF determines a requirement to initiate a VMEbus cycle due to any of the following conditions:

- A write cycle waiting in the transmit FIFO
- A pending coupled cycle
- An interrupt acknowledge cycle
- A DARF64 DMA read operation

When one of these conditions occurs, the DARF asserts $\overline{\text{VMERQ}}$ and waits for $\overline{\text{VMEGR}}$. The $\overline{\text{VMEGR}}$ signal indicates that the DARF can begin the address cycle of a VMEbus transfer. If the previous data cycle is not complete, then the address cycle is pipelined. The DARF waits for the previous DTACK* and BERR* to be negated before beginning its data transfer cycle.

The DARF keeps $\overline{\text{VMERQ}}$ asserted until its last VMEbus cycle, when there are no more cycles pending in the FIFO or coupled cycles pending from the CPU. The DARF releases $\overline{\text{VMERQ}}$ early in this cycle, allowing BBSY* to be released early. The DARF address strobe is the last master signal to be negated on the VMEbus.

The DARF can be requested to give up the VMEbus by negating $\overline{\text{VMEGR}}$ at any time. The DARF performs one or two more cycles, and then negates $\overline{\text{VMERQ}}$ during its last cycle.

Asserting BIMODE or reset immediately ends the current cycle, without waiting for an acknowledgment. Any cycles remaining in the transmit FIFO are lost. While the BIMODE pin is asserted, the DARF ignores all attempted cycles to the VMEbus in both coupled and decoupled modes, and waits for the local bus timeout circuit to end the cycle with $\overline{\text{KBERR}}$.

If the bit VBERR flag in the Control and Status Register is set, the DARF continues queuing decoupled writes, but does not request the bus. Unless the VBERR flag is cleared, the FIFO eventually fills up and the local bus times out on the next VMEbus access.

The local CPU is forced to retry its cycle if it is attempting a coupled access to the VMEbus, or a write to a full DARF transmit FIFO, while the VMEbus is trying to access the local memory. Because the DARF has already requested the VMEbus through the ACC, and the local CPU is not assured to retry the cycle immediately, the DARF negates its $\overline{\text{VMERQ}}$ signal immediately after $\overline{\text{VMEGR}}$ is asserted, to dispose of the VMEbus grant as soon as it is received.

3.6 Slave VMEbus Interface

The DARF does not provide a slave image on the VMEbus until its base address and image size have been programmed. Access protection for the DARF32 must be initialized as there are no defaults defined. The access protection default for the DARF64 is to provide no protection. Separate images, different in size, can be set up in the A32 and A24 address spaces. Neither the DARF32 nor the DARF64 can provide an A16 slave image. An A64 slave image is provided in the DARF64 for A64 block transfers.

Slave access types include reads, writes, read-modify-writes and interrupt acknowledge cycles. The Location Monitor is located within the slave image, but its particular address responds differently and is described under *Location Monitor and FIFO.*

The local CPU can access its own slave images at their programmed address bases, within the size configured. During this cycle, the DARF acts only as an address decoder and asserts $\overline{\text{RAMSEL}}$. It does not implement access protection, nor use the VMEbus and FIFOs, unless it is set in loopback mode. The slave images are accessible by both the VMEbus and the local CPU, so that reads or writes by a process access the same memory location, independent of the card on which the process is running.

3.6.1 Slave Image Configuration

The A32 and A24 slave images are defined by the Slave Base Address (VMEBAR) Register, shown in Table 3.10. The A32 image base can be programmed to any 128 MB boundary and to any size from 4 KB to 128 MB in binary increments. The A24 image base can be programmed to any half megabyte boundary or multiple of its programmed size, whichever is larger, within the 16 MB A24 addressing space. The A24 image size can be programmed to 512 KB, 1, 2 or 4 MB. Each image supports D32 through D08(EO) accesses. The DARF64 supports block mode accesses and D64 accesses in multiplexed block mode in addition to the D32 to D08(EO) accesses.

Table 3.10 : VMEbus Slave Base Address Register

Bits	Function
31-24	NOT USED
23-16	NOT USED A24SIZ (2 bits) A24BA (5 bits)
15-08	NOT USED A32SIZ -
07-00	>A32SIZ A32 BA (5 bits)

Address bits A31 to A24 are driven low on the local bus during all A24 accesses to the DARF. All address bits which are not required to uniquely address the slave image due to the programmed slave image size or base address selection are also driven low.

In the DARF64, an A64 slave image can be created at any 4 gigabyte boundary. It is 128 megabytes in size, imaged throughout a 4 gigabyte address range, and supports only D64 multiplexed block transfers.

Although the slave image bases and sizes can be programmed to virtually any value, only certain combinations are valid. The restriction described below ensures that the same bytes of memory are selected when the local memory is addressed directly by the CPU, by the CPU through one of its slave images, or by the VMEbus through the DARF

If the restrictions described below are violated, different areas of local memory may be accessed through these three paths.

The slave image base size is restricted to a multiple of the amount of memory installed on the CPU card. For example, if the card has 4 MB of memory, the possible A24 slave image base sizes are: 00 0000, 40 0000, 80 0000 and C0 0000, regardless of the A24 size programmed. This restriction is not a problem for A32 images, as they are already multiples of 128 MB.

3.6.2 AM Codes

The slave AM codes to which the DARF responds on the VMEbus are given in Table 3.4, section 3.5.3.

3.6.3 Access Protection

All or part of the A24 and A32 slave image can be protected from writes, or reads and writes. If access to a protected area is attempted from the VMEbus, the DARF responds with a VBERRO, which generates a BERR✳ on the VMEbus. Protection is not implemented for memory access by the local CPU to its own slave images.

If write protection has been programmed, and a read-modify-write operation is attempted on the protected memory, the RMW read cycles complete successfully, but the write cycles are terminated with BERR✳ and not performed by the DARF.

Bits 00 to 03 of the Access Protect Boundary Register (APBR), shown in Table 3.11, select protection for the lower part of the slave image. Settings for these bits are:

0 – none protected

1 – lower 64 KB protected

•
•
•

F – lower 128 MB protected

If the entire slave image is protected, the Location Monitor is not protected (from the VMEbus only). The APBR Register in the DARF32 is undefined after reset and must be programmed before the slave image is used. The APBR register in the DARF64 is reset to 0 and need not be programmed if no access protection is required.

Table 3.11 : Access Protect Boundary Register

Bits	Function				
31-24	NOT USED				
23-16	NOT USED				
15-08	NOT USED				
07-00	NOT USED	APB03	APB02	APBO1	APB00

The type of protection is determined by bit PROT in the Mode Control Register shown in Table 3.12. Setting PROT to 0 (default value), selects write protection. Setting PROT to 1 blocks both read and write accesses.

Table 3.12 : DARF Mode Control Register

Bits	Function							
31-24	DTCSIZ	RESERVED	RMCRETRY	DMABEN	RMCPIN	BUSSIZ	SWAP	040
23-16	DPRIV	NOREL	FILL	DMAA64	MBLT	BLT	DMAD16	DMARD
15-08	FIFOBEN	BLEN *(2 bits)*		RXATOM	TXATOM	A24SLVEN	DMA24	BERRCHK
07-00	TASCON	A24P0	LPBK	DISRX	A24DI	A16DI	PROT	VINEN

3.6.4 Bus Error Checking

Local bus cycle failures can be signalled by asserting KBERR only to end the cycle, or by asserting KBERR and KDSACKn . The DARF sets the LBERR flag in the Control and Status Register (see Table 3.13) and asserts VBERRO (BERR✳) instead of VDTACKO (DTACK✳ on the VMEbus) if the cycle was coupled.

If the cycle was a decoupled write, the VMEbus cycle has already received DTACK✳. The DARF interrupts the CPU if LBERR is set.

Table 3.13 : Control and Status Register

Bits	Function							
31-24	NOT USED				DEVICE *(4 bits)*			
23-16	NOT USED							CERR
15-08	TXSHFT	RETRY	RMCERR	A64BARDY	BARDY	RXSHFT	RXRST	TXRST
07-00	RXHD	TXHD	DLBER	LMHD	LBERR	VBERR	DONE	DMAGO

The DARF can be programmed to check for a late bus error by setting bit BERRCHK in the Mode Control Register (Table 3.14). If this bit is set, asserting KBERR during the two clock periods following the end of the DARF cycle indicates that the cycle just ended has failed, and bit LBERR in the Control and Status Register is set.

If the DARF is executing a coupled cycle and BERRCHK is set, the DARF ends the local cycle and then waits two clocks to check for late bus errors before asserting either VDTACKO (DTACK✳) or VBERRO (BERR✳) to end the local cycle.

Table 3.14 : DARF Mode Control Register

Bits	Function							
31-24	DTCSIZ	RESERVED	RMCRETRY	DMABEN	RMCPIN	BUSSIZ	SWAP	040
23-16	DPRIV	NOREL	FILL	DMAA64	MBLT	BLT	DMAD16	DMARD
15-08	FIFOBEN	BLEN *(2 bits)*		RXATOM	TXATOM	A24SLVEN	DMA24	BERRCHK
07-00	TASCON	A24P0	LPBK	DISRX	A24DI	A16DI	PROT	VINEN

3.6.5 Coupled Mode

The DARF performs VMEbus coupled slave cycles using the following procedures:

- Requesting the local bus
- Starting the cycle with standard 68020 timing
- Waiting for KDSACKn or KBERR on the local bus

The local cycle ends and the bus is released after the DARF receives either KDSACKn or KBERR . VDTACKO or VBERRO are maintained on the VMEbus until the cycle there also ends. The DARF provides slave only image address decoding, access protection and early local bus release in coupled mode. If the area addressed is protected, the DARF asserts VBERRO (BERR✳ on the VMEbus) without making a request for the local bus.

The DARF samples KDSACKn and KBERR on the falling edge of the local clock. On the first edge that KDSACKn or KBERR is asserted, the cycle ends with standard 68020 timing. If bit BERRCHK in the Mode Control Register is not asserted, and KDSACKn or KBERR , (or both) were asserted locally, the DARF generates VDTACKO or VBERRO (DTACK✳ or BERR✳ on the VMEbus) on the rising edge of S5. If KDSACKn or KBERR are not asserted locally, the cycle times out, and the Syscon must terminate the failed cycle with a BERR✳. The ACC must be programmed to assert BERR✳ signals to handle this condition.

The DARF accepts any KDSACKn to successfully end a local cycle, regardless of the size indicated by the VMEbus or local bus data strobes. The DARF has no dynamic bus sizing capabilities as a master and assumes that all requested data was provided.

To alternate the receive mode between coupled and decoupled, first disable the incoming cycles by writing zero to bit VINEN in the DARF Mode Control Register (Table 3.15), then wait until the DARF acknowledges the new mode by setting VINEN to zero. RXATOM can then be changed and VINEN set high again. Any slave VMEbus accesses to the DARF during this time are ignored and the VMEbus master performing such cycles may receive a timeout BERR✳.

Table 3.15 : DARF Mode Control Register

Bits	Function							
31-24	DTCSIZ	RESERVED	RMCRETRY	DMABEN	RMCPIN	BUSSIZ	SWAP	040
23-16	DPRIV	NOREL	FILL	DMAA64	MBLT	BLT	DMAD16	DMARD
15-08	FIFOBEN	BLEN *(2 bits)*		RXATOM	TXATOM	A24SLVEN	DMA24	BERRCHK
07-00	TASCON	A24P0	LPBK	DISRX	A24DI	A16DI	PROT	VINEN

3.6.6 Decoupled Mode

In decoupled mode, writes from the VMEbus are immediately loaded into the receive FIFO. They are acknowledged with $\overline{\text{VDTACKO}}$ (DTACK* on the VMEbus), after the slave address decoder and protection logic validate the access. If the access is protected, the DARF generates $\overline{\text{VBERRO}}$ (BERR* on the VMEbus).

If the receive FIFO is full, the DARF does not accept writes until the FIFO contents are unloaded onto the local bus, making space available. If the receive FIFO cannot perform its writes on the local bus, the VMEbus transfer ends with a timeout BERR*.

Read accesses are always coupled; if the DARF is in decoupled mode, the access is detected by the address decoder and protection logic and waits until any writes queued up in the FIFO are completed. The read is then performed as described under Coupled Mode. Because as many as seven writes may be queued ahead of the read when the DARF32 is used, and 15 writes when the DARF64 is used, it is recommended that the VMEbus data transfer timeout period be longer than 4 µs.

If the DARF receives a local bus error while performing a queued write, bit LBERR is set in the Control and Status Register and the $\overline{\text{VMEINT}}$ interrupt pin is asserted. The DARF does not stop transferring cycles in or out, and does not save the failed cycle for diagnostic analysis.

3.6.7 Read-Modify-Write Cycles

The standard VMEbus implements a locked read-modify-write cycle by maintaining AS* low between the read and write cycles that form the RMW, instead of returning it high. Continuing to assert AS* locks the access path all the way to the slave memory, which can be dual-ported to some other device.

The standard RMW can operate on only one address, as a result of keeping address strobe low. The DARF accepts the standard RMW cycle, and implements a locked multiple address RMW by accepting a $\overline{\text{VRMC}}$ signal through an unused VMEbus pin (P2 pin B3, "Reserved"). This is normally derived from the RMC pin on the 68020/030 and is asserted throughout read-modify-write cycles. P2 pin B3, "Reserved" is defined in the Rev. D VMEbus Specification as a RETRY pin. The DARF64 provides for optional definition of the $\overline{\text{VRMC}}$ pin as a RMW lock signal, as described above, or as RETRY. The option is selected by setting the bit RMCPIN in the Mode Control Register.

If either $\overline{\text{VAS}}$ (AS*) or $\overline{\text{VRMC}}$ remain low after a local cycle ends, the DARF keeps ownership of the slave card local bus

3.7 Local Bus Interface

3.7.1 Local Bus Mastership

The DARF requests the local bus whenever:

- There are cycles waiting in the receive FIFO
- A coupled VME-in cycle is pending
- The DMA needs to read local memory

To request the bus, the DARF asserts $\overline{\mathrm{LBRQ}}$ to the local bus arbiter, usually the ACC. The arbiter then requests the bus from the local CPU. The arbiter must assert $\overline{\mathrm{KBGACK}}$ to the CPU, to take ownership of the bus on behalf of the DARF. When the grant is received from the CPU, the arbiter asserts $\overline{\mathrm{LBGR}}$ to the DARF after the previous master has released its address strobe and $\overline{\mathrm{KBGACK}}$ signals. The DARF can then begin using the bus. The DARF keeps $\overline{\mathrm{LBRQ}}$ asserted and uses the bus until it has no cycles pending.

While the DARF is using the local bus, its signal timing and use of $\overline{\mathrm{KDSACKn}}$ or $\overline{\mathrm{KBERR}}$ to end a cycle are approximately the same as for a 68020 (refer to AC Timing Characteristics in Appendix D for details). Neither the DARF32 nor the DARF64 supports automatic bus sizing on the local bus. The DARF expects the local bus slave to support the signalled type of cycle, and accepts any $\overline{\mathrm{KDSACKn}}$ to end the cycle. Further, the byte lane swapping that is necessary to match the VMEbus byte lanes with a local 68020, or 68030, is performed. The DARF can also be programmed to check for late bus errors after the cycle ends. If the DARF is in decoupled mode with late bus error checking enabled and other cycles are waiting, it starts the next cycle immediately and overlaps checking for late bus errors with the first two clocks of the new cycle.

Early in its last cycle, the DARF releases $\overline{\mathrm{LBRQ}}$. The arbiter in the ACC keeps $\overline{\mathrm{KBGACK}}$ to the CPU asserted until both the DARF signals $\overline{\mathrm{LBRQ}}$ and $\overline{\mathrm{KAS}}$ are negated. The arbiter then releases the bus to the CPU.

The DARF can be requested to give up the local bus if a higher priority device needs it. To do this, the arbiter deasserts $\overline{\mathrm{LBGR}}$ and waits for the DARF to negate $\overline{\mathrm{LBRQ}}$. Within one or two data transfer cycles, the DARF releases the bus as described above, even though there may still be pending cycles. The DARF can assert its bus request again anytime after $\overline{\mathrm{LBGR}}$ has been negated.

If the DARF is performing a read-modify-write operation, indicated by VMEbus AS* remaining low or $\overline{\mathrm{VRMC}}$ being asserted, the DARF does not negate $\overline{\mathrm{LBRQ}}$ until the cycles in the RMW operation are completed.

If the DARF64 has local bus mastership, it waits for any queued write cycles to complete before performing slave MBLT read cycles. If the DARF does not have local bus mastership, it waits for a bus grant, and then translates the MBLT cycle into two indivisible standard 32 bit read cycles. Bus ownership is released after every two transfers, or when the Receive FIFO is empty.

3.7.2 DMA Controller

The DMA controller in the DARF32 can transfer longwords from local memory to any A32:D32 or A24:D32 address on the VMEbus. The decoupled mode is used to minimize local and VMEbus ownership. Transfers are always done in longwords (D32 on the VMEbus), in blocks up to 4K longwords (16 KB).

The DMA controller in the DARF64 can, in addition, transfer double longwords from local memory to any A32:D64 or A64:D64 address on the VMEbus. Each D64 transfer is converted to or from two indivisible longword cycles on the local bus. In A64 mode, the high 32 address bits are provided by the Master A64 Base Address Register (MA64BAR) and the lower 32 bits are provided by the DMA VMEbus Address Register (DMAVAR). Data transfers can be configured to occur in discrete, block, or multiplexed block modes. Discrete and block modes can be further configured as D32 or D16 widths. These modes are selected in the Mode Control Register. The DARF64 DMAC may transfer up to 4 megabytes of data, inserting address phases when necessary on the local bus and VMEbus.

A single read on the local bus is used for each longword, since the DMA loads the VMEbus address directly into the transmit FIFO during the read cycle. The DMA uses the DARF32 transmit FIFOs and decoupling logic and thus does not support reads from the VMEbus to local RAM. The DARF64 uses both transmit FIFO and receive FIFO in decoupled mode to support reads from the VMEbus in addition to writes to the VMEbus. DMA transfer direction is controlled by the DMARD bit in the Mode Control Register.

All DMA operations use decoupled mode. Bits RXATOM and TXATOM in the Mode Control Register must be cleared before DMA transfers can be performed.

Five registers are involved in controlling the DMA, one of which is the DARF Mode Control Register (Table 3.16). Bit DMA24 in this register determines whether DMA transfers occur in the A32 or A24 address space on the VMEbus.

In A24 mode, the DARF32 uses AM code 39 (standard non-privileged data access) and sets A31 – A24 to ones when using the bus, regardless of how those bits are programmed in the DMA Destination Register. In A32 mode, the DARF32 uses AM code 09 (extended non-privileged data access) and uses the destination register as programmed. The DMA24 bit overrides the addressing mode that is implied by the VMEbus address being transferred (A24:D16, etc.). Bit A24P0 in the DARF Mode Control Register has no effect on the DMA transfers. The DARF64 provides a programming bit in the Mode Control Register that enables selection of the privilege type - supervisory or non-privileged - for all DMA transfers.

Table 3.16 : DARF32 Mode Control Register

Bits	Function							
31-24	DTCSIZ	RESERVED	RMCRETRY	DMABEN	RMCPIN	BUSSIZ	SWAP	040
23-16	DPRIV	NOREL	FILL	DMAA64	MBLT	BLT	DMAD16	DMARD
15-08	FIFOBEN	BLEN *(2 bits)*		RXATOM	TXATOM	A24SLVEN	DMA24	BERRCHK
07-00	TASCON	A24P0	LPBK	DISRX	A24DI	A16DI	PROT	VINEN

The DARF32 DMA Local Address register (Table 3.17), provides the local memory address and always points to the next longword to be read. Although the register can be programmed to transfer from any local address in the 0 to 128M range, there are restrictions similar to those on the valid slave image base addresses described earlier. The DMA must transfer data from a block entirely within the slave image.

Advanced features are provided in the DARF64 to tune DMAC and VME performance.

- The transmit FIFO can be configured to begin transfers as soon as data is received, or configured to wait until the FIFO is full. This option is programmed by the FILL bit in the Mode Control register.
- The bit NOREL (No Release) in the Mode Control register causes the DARF64 to keep ownership of the VMEbus until either the ACC requests the DARF64 to release the bus, or until the bit is turned off. The VMEbus ownership timer in the ACC can be used in conjunction with the NOREL bit to throttle DMAC use of the VMEbus.

Table 3.17 : DMA Local Address Register

Bits	Function		
31-24	NOT USED	DSA (DMA Local Address)	
23-16	DSA (DMA Local Address) BYTE 2		
15-08	DSA (DMA Local Address) BYTE 1		
07-00	DSA (DMA Local Address) BYTE 0	0	0

The DARF32 DMA VMEbus Address Register (Table 3.18), provides the VMEbus address for the transfer and can be programmed to any longword address in the A32 or A24 space. The DMA cannot transfer to the A16 space. The register points to the next VMEbus address, although some of the most recent cycles may still be queued up in the transmit FIFO waiting to be written to their destination.

Table 3.18 : DMA VMEbus Address Register

Bits	Function		
31-24	DSA (DMA VMEbus Address) BYTE 3		
23-16	DSA (DMA VMEbus Address) BYTE 2		
15-08	DSA (DMA VMEbus Address) BYTE 1		
07-00	DSA (DMA VMEbus Address) BYTE 0	0	0

The DARF64 has a DMA VMEbus Address Register (DMAVAR) and a DMA Local Address Register (DMALAR) in place of the source and destination registers in the DARF32. This provides a consistent register for programming VME and local addresses independent of transfer direction. DMAVAR must always be programmed with the VMEbus address, either as a source or destination. The DMALAR must always be programmed with the Local bus address, either as a source or destination.

The DMA Transfer Count Register in the DARF32 (Table 3.19), indicates the number of longwords to be transferred. The register can be programmed for up to 4096 longwords, or 16 KB where the value 0 indicates 4096. While DMA is running, the register shows the number of transfers remaining; the DMA stops when the register counts down to zero.

The DARF64 Transfer Count Register (DMATC) can be programmed to transfer 1,048,576 longwords or 4 Mbytes. This extended transfer count range is enabled only when DTCSIZ is set in the Mode Control Register. When the DTCSIZ bit is set to 0, the DARF64 DMATC registers functions in the same way as the register in the DARF32, using bits 0 to 11. When the DTCSIZ bit is set to 1, the DARF64 uses bits 0 to 19.

An additional register, DMA VMEbus Transfer Count (DMAVTC), is provided in the DARF64 to indicate the number of VMEbus transfers remaining during DMA BLT and MBLT transfers. The DMA Transfer Count Register (DMATC) indicates the number of local transfers remaining. These values may differ as cycles are queued in the transmit and receive FIFO's.

Before attempting to start a DMA transfer, the following bits must be set by programming the MA64BAR register and the VMEBAR register:

- MODE register bit A64BARDY and bit BARDY
- DCSR register bit A64BARDY and bit BARDY

DMA transfers using A64 must ensure that both the MA64BAR and the VMEBAR registers are programmed as indicated by the corrersponding bits A64BARDY and BARDY in the DCSR register before attempting to start a DMA transfer.

Table 3.19 : DMA Transfer Count Register

Bits	Function	
31-24	Unused	
23-16	Unused	**DMA VMEbus Transfer Count ->**
15-08	**->DMA VMEbus Transfer Count ->**	
07-00	->DMA VMEbus Transfer Count *(Tota. 12 bits for DARF32, 20 bits for DARF64)*	

Caution: Software must ensure the DMA is stopped before reading the address and transfer count registers, so that the values do not change while being read.

The Control and Status Register (Table 3.20) contains the DMA start and stop control bits DMAGO and DONE. These bits are managed by both the CPU and DMA.

- DMA is started by setting DMAGO to one (1). To set this bit:

  the DARF transmit FIFO must be in decoupled mode,
  the slave image must be programmed (indicated by BARDY high) and
  the VBERR and DLBER flags must be cleared.

- Normally the DMA completes the transfer and stops by itself. The DMAGO bit is cleared by the DMA, the DONE bit is set, the $\overline{\text{VMEINT}}$ pin is asserted, and $\overline{\text{VMEINT}}$ is negated when the bit DONE flag in the Control and Status Register is cleared.

The DMA can also be stopped when the CPU writes 0 to the DMAGO bit. The DMA then completes any cycles already pending (one or two). The DMA stops if:

- it receives a local bus error while reading local RAM,
- the DARF receives a VMEbus BERR✳ while writing to the destination, or
- the DARF receives a late bus error when BERRCHK is set.

When these errors occur, the DLBER flag or the VBERR flag, in the Control and Status Register, is set, the bit DMAGO is cleared by the DMA and $\overline{\text{VMEINT}}$ is asserted. The DONE bit is not set, as the programmed transfers are not finished by the DMA at this point . $\overline{\text{VMEINT}}$ is negated when the bit DLBER and bit VBERR flags are cleared.

Table 3.20 : Control and Status Register

Bits	Function							
31-24	NOT USED				DEVICE *(4 bits)*			
23-16	NOT USED							CERR
15-08	TXSHFT	RETRY	RMCERR		BARDY	RXSHFT	RXRST	TXRST
07-00	RXHD	TXHD	DLBER	LMHD	LBERR	VBERR	DONE	DMAGO

Configuration Errors. If a DMA transfer is attempted when the DMA is incorrectly programmed the CERR bit in the DCSR register is set, and signal $\overline{\text{VMEINT}}$ is asserted. The following configurations result in CERR being set:

MBLT set	and	BLT set
MBLT set	and	D16 set
DMAA64 set	and	MBLT cleared
DMAA64 set	and	DMA24 set
DMARD set	and	RXATOM set
DMARD set	and	LPBK set

3.7.3 Local Burst Mode

As a local bus master, the DARF64 provides a burst mode on the local bus that can be enabled and configured for bursts of 4, 8, 16, or 32 longwords. The DARF64 begins the cycle with normal address and data strobe assertion, and identifies it as a burst by using a function code of 3. Address is asserted at the start of the cycle only and held throughout the burst transfer. Data is transferred synchronous to KCLK using $\overline{\text{DSACK1}}$ to indicate a ready state for the target device.

All burst start addresses must be longword aligned (addresses 00H, 04H, 08H, 0CH) for BLTs and double-longword aligned (addresses 00H, 08H) for MBLTs. Burst cycles are automatically terminated by deassertion of $\overline{\text{KAS}}$ and $\overline{\text{KDS}}$ at every burst-length boundary and restarted with a new address phase. A burst length boundary is defined as the address given by the programmed burst length times the data width. For a programmed burst length of 8 during BLT transfers, burst length boundaries occur at 20H, 40H, etc.

As an example, if a burst of 8 longwords is programmed and 8 longwords are to be transferred starting at address 0CH, the DARF64 will terminate the burst and restart the cycle with a new address phase at address 20H. The cycle will appear as a burst of 5, followed by a burst of 3.

Burst mode can be enabled during DMA operations using BLT or MBLT VMEbus cycles or during BLT or MBLT VMEin write cycles. Two bits in the Mode control register enable burst mode transfers: DMABEN and FIFOBEN. BLEN in the Mode register must be programmed to set the required burst length. Burst programming options are summarized below.

DMABEN	FIFOBEN	
0	0	no burst mode operation enabled
0	1	VMEin burst writes enabled
1	0	DMA burst reads enabled
1	1	DMA read/write bursts enabled, VMEin burst writes enabled

3.7.3.1 Write Burst Operation

Burst writes are enabled by setting the FIFOBEN bit in the Mode register. This enables cycles stored within the receive FIFO (RXFIFO) to be completed as burst cycles on the local bus. Write bursts to the local bus are initiated when a minimum of 4 BLT or 2 MBLT cycles have been queued within the RXFIFO. If burst mode is enabled and insufficient entries exist in the RXFIFO, these entries will be completed as standard cycles until 4 burst entries have been stored. FIFO entries preceding BLT or MBLT cycles are always completed normally before queued BLT/MBLT cycles are completed as bursts and do not contribute to the 4 entry threshold. Once a burst has been initiated, all subsequent BLT/MBLT cycles that form part of the same block will also be completed as burst cycles as long as local bus ownership is maintained.

As an example, if a burst of 4 longwords is programmed and 4 longwords are to be transferred starting at address 08H, an address phase will be inserted after the first 2 transfers have completed. This will leave 2 entries which will also be completed as burst cycles after the address phase and if local bus ownership is maintained. If the local bus is re-arbitrated at the address phase, and if 3 or fewer cycles remain in the RXFIFO, these will be transferred as standard cycles when local bus ownership is regained.

$\overline{\text{KDSACK1}}$ is sampled on the second rising edge of KCLK after the cycle is initiated to indicate whether or not the target device is ready to accept data. If $\overline{\text{KDSACK1}}$ is asserted, data is written on the next rising clock edge. $\overline{\text{KDSACK1}}$ is sampled on the subsequent rising edge of KCLK after data is written and if asserted will cause an additional write cycle at the next rising edge of clock. If $\overline{\text{KDSACK1}}$ is not asserted at the rising edge of KCLK immediately following any data change, wait states are inserted and data is not transferred. $\overline{\text{KDSACK1}}$ will continue to be sampled at every rising clock edge until $\overline{\text{KDSACK1}}$ is asserted to indicate device ready and cause data to be written at the next rising clock edge. Completion of the burst cycle is indicated by deassertion of $\overline{\text{KAS}}$ and $\overline{\text{KDS}}$ on the second falling edge of KCLK after the last data is written. A data transfer rate of one transfer every two clock cycles may be sustained during the burst.

The DARF64 is not guaranteed to maintain local bus ownership during the burst cycle. The programmed burst length is a maximum only and does not guarantee the number of cycles to be transferred as burst cycles. The burst cycle will halt if data from the VMEbus is not available, or if the current BLT or MBLT cycle ends before the programmed number of burst transfers are completed. A new burst cycle will be started if a new BLT or MBLT cycle is started and adequate cycles are queued in the RXFIFO. If a higher priority local bus request is made, the DARF64 will release the local bus when the burst cycle reaches a burst length boundary. Any remaining BLT or MBLT FIFO entries will be completed as burst cycles when the DARF64 regains the local bus if sufficient entries are queued to initiate burst operation.

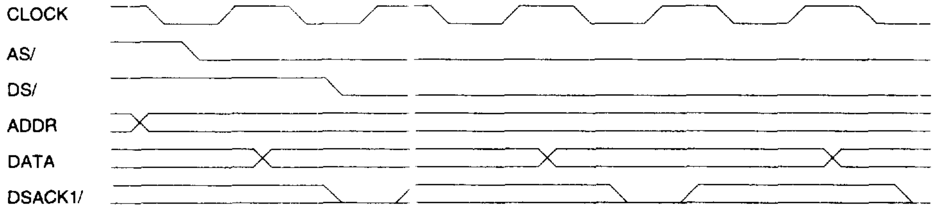


Figure 3.4 : DARF64 Write Burst

3.7.3.2 Read Burst Operation

Burst reads may be enabled during DMA operations to load the transmit FIFO (TXFIFO) for BLT and MBLT data transfers to the VMEbus. After the DMA has been initiated, the DARF64 will request the local bus and initiate the burst read by asserting $\overline{\text{KAS}}$ and $\overline{\text{KDS}}$ and using function code 3 to indicate a burst cycle on the local bus. Burst read cycles will halt if the programmed burst length is reached, if the DMA transfer count has been completed, if local bus ownership is given to a higher priority local bus master or if no room remains in the TXFIFO. If the TXFIFO fills, the DARF64 will release the local bus before any additional cycles cause the FIFO to overflow. During BLT transfers, the local bus will be released after the fifteenth entry to prevent overflow.

During MBLT cycles the local bus will be released after the fourteenth entry to prevent overflow by the 2 consecutive reads needed to complete an MBLT transfer.

$\overline{\text{KDSACK1}}$ is sampled on the second rising edge of KCLK after the cycle is initiated and on every rising edge of KCLK thereafter. On assertion of $\overline{\text{KDSACK1}}$, data is loaded into the DARF64 at the next falling clock edge. Data is loaded into the DARF64 at every falling clock edge during which $\overline{\text{KDSACK1}}$ is asserted or until the cycle is terminated. Completion of the burst cycle is indicated by deassertion of $\overline{\text{KAS}}$ and $\overline{\text{KDS}}$ *after* the last data transfer and at the same edge as a subsequent data transfer would start at. Data is transferred at the rate of 1 longword every clock during the burst.

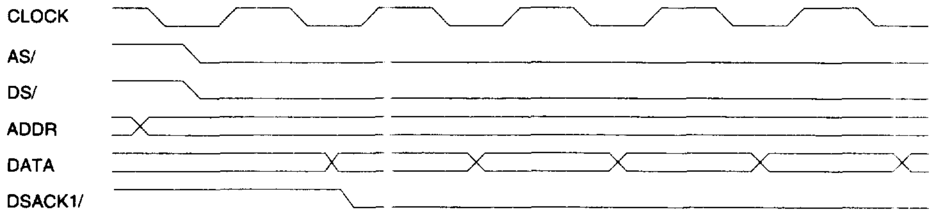


Figure 3.5 : DARF64 Read Burst

3.7.5 Other Processor Support

3.7.5.1 68040 Mode

The DARF64 is designed to work with 68040. In this mode, it accepts 68040 local bus timing and performs the data and control signal multiplexing required to interface the 68040 with the VMEbus. The 68030 timing still used when the DARF64 accesses local memory. When processors other than the 680x0 series are used, functions such as dynamic bus sizing, can be selectively disabled if appropriate.

The 68040 support provided by the DARF64 includes turning off dynamic bus sizing and a synchronous local bus interface that permits direct signal connection. In order to invoke this 68040 mode, $\overline{\text{KDS}}$ must be grounded on the local bus and is sampled after the reset sequence. However, when the DARF64 becomes local bus master, it generates 68030 timing. Data from the DARF64 is always valid for 0 wait state writes, enabling the $\overline{\text{KDS}}$ pin to be interpreted in this way. The 040 bit in the MODE register indicates that the DARF64 is in the 68040 mode

3.7.5.2 Swap

The SWAP bit in the DARF64 Mode register may be used to disable swapping of byte lane information between the local CPU bus and the VMEbus. This feature may be useful where non-Motorola architectures are used to avoid Motorola specific byte lane conventions. All byte lanes on the local bus are mapped directly into the same byte lanes on the VMEbus for all transfer sizes.

Note that use of this feature may produce illegal or unrecognized VMEbus cycles.

3.7.5.3 Bussiz

The BUSSIZ bit in the DARF64 Mode register is used for support of 68040 local bus designs. In general, if an access is made to the D16 space of the VMEbus, the DARF64 will respond to the local bus as a D16 slave causing the local CPU to break up the cycle as required. By disabling BUSSIZ, the DARF64 will always respond to the local bus as a D32 slave independent of the target data space on the VMEbus.

3.8 Location Monitor and FIFO

The Location Monitor is provided to assist in inter-processor or inter-process communication. It includes a 32-bit wide, 31-entry deep, message FIFO loaded from the data bus, so that either 16 or 32-bit messages or memory pointers can be sent between processes. If an even word is written, the upper 16 bits in the FIFO entry are set to one (1). While there are entries in the FIFO, the DARF keeps $\overline{\text{LMINT}}$ asserted to generate an interrupt to the local CPU.

The Location Monitor exists at the top longword, and the lower (even) word of the top longword, in each of the A32 and A24 slave images of the DARF. It is accessible from both the local bus and VMEbus. The Location Monitor replaces the longword of memory which otherwise exists at that address. As a result, writes to the Location Monitor do not enter into memory. A read access to the Location Monitor address asserts BERR* on the VMEbus.

When the CPU writes to its own Location Monitor through a slave image, the transfer does not involve use of the transmit FIFO or the VMEbus. The Location Monitor does not exist at the top of the basic image of local memory, which is selected by address decoding logic external to the DARF, nor does it respond to accesses to the upper word of the top longword or misaligned-aligned transfers.

The Location Monitor is an address detector which controls the message FIFO. The message FIFO is 32 bits wide by 31 longwords deep and queues the data which is present on the bus writing to it. If the FIFO is full when a Location Monitor write is made, no response is made to the cycle. If the local CPU reads an item from the other end of the FIFO while a write is waiting at the other end, then a position becomes available and the write is accepted. Otherwise, the bus timeout expires and ends the cycle with a Bus Error.

When both the CPU and VMEbus write to the Location Monitor at the same time, priorities are resolved and both writes are queued according to priorities, as long there is room for them.

The DARF asserts LMINT while there are entries in the queue and negates it when the queue becomes empty. The LMHD status bit in the Control and Status Register is high while the FIFO has data (see Table 3.21).

Table 3.21 : Control and Status Register

Bits	Function							
31-24	NOT USED				DEVICE *(4 bits)*			
23-16	NOT USED							CERR
15-08	TXSHFT	RETRY	RMCERR	A64BARDY	BARDY	RXSHFT	RXRST	TXRST
07-00	RXHD	TXHD	DLBER	LMHD	LBERR	VBERR	DONE	DMAGO

Messages in the FIFO are read from DARF LMFIFO Register. When the FIFO is empty, further reads return undefined data (with a KDSACKn). The DARF does not signal this as an error condition. LMINT is only asserted while there are entries in the FIFO, so software can poll the signal or use an interrupt service routine to support the Location Monitor.

A write to the Location Monitor is also used to exit from BI-mode. The DARF pulses the BIREL pin each time the Location Monitor is written. The ACC uses this event to clear the BIMODE signal if none of the conditions for initiating BI-mode are present.

3.9 VMEbus Interrupter

The DARF contains the interrupt vector provider section of a VMEbus Interrupter. The interrupt asserter, and interrupt acknowledge cycle detector (IACK daisy chain driver) are provided by external logic, such as the ACC.

The DARF provides an 8-bit interrupt vector on the VMEbus when VECTEN is asserted. This vector is programmed by software into the VMEbus Interrupter Vector Register (Table 3.22). Once the DARF has asserted VDTACKO (DTACK* on the VMEbus), VECTEN is negated. The DARF maintains the vector on the VMEbus until DS0* is negated.

The interrupt vector is contained in the lower 8 bits of the register and is not modified before being supplied to an acknowledge cycle. The upper 3 bytes of the register are not used — the register must be accessed as a longword port.

Eight bit interrupters are allowed to respond to any IACK cycle directed at them, whether the vector size requested by the data strobes is 8, 16 or 32 bits, as long as D31 – D08 are not driven low. The DARF therefore only uses the data strobes for timing and drives D31 – D08 high while the vector is on D07 – D00.

Table 3.22 : VMEbus Interrupter Vector Register

Bits	Function
31-24	NOT USED
23-16	NOT USED
15-08	NOT USED
07-00	IVECT (Interrupt Vector)

3.10 Control and Status Registers

There are sixteen control and status registers in the DARF32, accessible as aligned longwords only. The DARF does not respond to other sizes of accesses, relying on a local bus timeout to assert $\overline{\text{KBERR}}$ in those cases. Access to the registers is selected when $\overline{\text{DARFSEL}}$ is asserted, whether or not the card is in BI-mode. The DARF32 itself does not make the registers directly accessible to the VMEbus. External logic can be added to assert $\overline{\text{DARFSEL}}$ for the DARF64 for accessing its registers from the VMEbus.

There are twenty registers provided in the DARF64. The additional registers are provided for:

- Master A64 base addresses
- Slave A64 base addresses
- VMEbus transfer count register
- Local address generator register which generates local addresses during BLT and MBLT transfers.

During valid register accesses, the DARF asserts $\overline{\text{KDSACK1}}$ and $\overline{\text{KDSACK0}}$ for a 1 wait state cycle and rescinds them at the end of the cycle. The start of the cycle is detected on the falling edge of S2, where $\overline{\text{KAS}}$ and $\overline{\text{DARFSEL}}$ are evaluated. $\overline{\text{KDSACKn}}$ are then asserted on the next rising edge of the clock.

Register contents not initialized by reset are undefined. Some like the Access Protect Boundary Register in the DARF32, must be programmed before associated functions can be used. This register in the DARF64 is reset to all zeroes to provide a default setting for access protection.

The following registers are intended for use during test or diagnostic mode only, and are not assured to be stable while being read at other times.

RXDATA	Receive FIFO Data
RXADDR	Receive FIFO Address
RXCTL	Receive FIFO Control

The following registers are provided for error recovery after bit VBERR in the Control and Status Register has been set. They may change while being read at any other time.

TXDATA	Transmit FIFO Data Output Latch
TXADDR	Transmit FIFO Address Output Latch
TXCTL	Transmit FIFO AM Code and Control Output Latch

The DMA Source and VMEbus Address Registers and the Transfer Count Register may change while they are being read, unless the DMA is stopped before reading. There is usually no need to read these while DMA is working.

The Location Monitor FIFO may output a data transient on the local bus (data changes in the middle of a cycle) if a read is done while it is empty; a message may arrive from the VMEbus while the CPU is reading the Location Monitor. Do not read the FIFO unless its interrupt, or the LMHD bit in the Control and Status Register, is active.

Some of the Control and Status Register values may change while being read. Each of these is described in Table 3.23 If a decision requires a stable value for these bits, read the register until two successive reads provide the same bit value or change the mode of the DARF so the bit in question cannot change.

Table 3.23 : Control and Status Register Value Changes

Bit	Conditions causing change while being read
RXHD	0 to 1 due to a received VMEbus cycle
TXHD	1 to 0 due to last outgoing cycle to VMEbus
LMHD	0 to 1 if a location monitor write is received
VBERR	0 to 1 due to VMEbus BERR* on outgoing cycle
DONE	0 to 1 due to DMA completed, except during DMA writes
DMAGO	1 to 0 due to DMA stopping

3.11 DARF Programming Considerations

3.11.1 DARF Initialization

Until the local CPU needs the VMEbus or its slave image of memory, the DARF need not be programmed. Before the DARF performs a VMEbus cycle for the CPU, its BI-mode signal (BIMODE) must be deasserted. This is accomplished on cards using the ACC and DARF by creating a VMEbus slave image by programming the VME Base Address Register, then writing to the Location Monitor that exists at the top of that slave image. The DARF asserts $\overline{\text{BIREL}}$ to the ACC, which then negates BIMODE if all other BI-mode initiator signals are negated.

For the VMEbus to access the dual ported memory on the card, the VME Base Address and Access Protect Boundary Registers must be programmed to create the slave image and initialize the access protection. The slave image must also be enabled by programming the VINEN bit in the Mode Control register and the DARF must be out of BI-mode. Access protection does not need to be programmed in the DARF64.

The DARF defaults to decoupled mode. If the DARF receives a VMEbus BERR* ($\overline{\text{VBERRI}}$), software on the card must clear the VBERR flag in the Control and Status Register before further VMEbus master accesses are possible.

3.11.2 System Configuration

If the DARF forces the CPU to retry a VME-out cycle, the DARF must have already requested the VMEbus from the ACC. The DARF waits for the VMEbus grant, then immediately negates its request without using it so that spurious bus requests do not disrupt VMEbus operation. The DARF is then ready for a new VME-out cycle. The DARF state machines do not support new VMEbus requests from the CPU during this sequence.

However, if the ACC Requester is programmed in *Fair* and ROR mode, and some other card is using the bus on the same bus request level, the CPU and DARF may toggle endlessly, requesting the bus grant and discarding it. To avoid this configuration, use different request levels, but not the *Fair* and ROR modes at the same time.

3.12 BI-mode Operation

While the DARF is in BI-mode, its VMEbus transceivers are pointed inwards and it does not assert any signals on the VMEbus. Even though its slave images may have been programmed, it does not respond to any accesses — and the VMEbus times out if such an access occurs. During writes to the Location Monitor, the DARF asserts $\overline{\text{BIREL}}$ and responds to the VMEbus with a $\overline{\text{VDTACKO}}$ (DTACK*).

If the local CPU attempts to access the VMEbus while the DARF is in BI-mode, and timeout is enabled in the ACC, the DARF ignores the cycle and the local bus times out.

If the DARF enters BI-mode with cycles still in its two FIFOs, the receive FIFO finishes all writes to memory. However, all cycles in the transmit FIFO are lost.

If $\overline{\text{VECTEN}}$ is asserted to the DARF while BIMODE is asserted, the DARF does not drive its interrupt vector onto the VMEbus.

The internal registers of the DARF are fully accessible while in BI-mode. The local memory slave image, at the programmed base address and size, is also accessible to the local CPU. Such an access uses neither VMEbus nor FIFOs, unless the DARF is in loopback mode (described in *Test and Diagnostic Modes* following). Address decoding for the VSB also remains active, since those pages have been mapped over to that bus, though other parts of the circuit may externally block such access while in BI-mode.

3.13 Test and Diagnostic Modes

Several read back registers and FIFO control bits are provided for functionally testing the DARF FIFOs, for address decoding, and to provide recovery from decoupled write cycles terminating in a bus error.

The DARF also has a test mode for use during chip manufacture when the TESTMODE pin is tied high. This feature is not useful for card level diagnostics and the TESTMODE pin must always be grounded during normal operation.

3.13.1 Decoupled Write Diagnostics

The transmit FIFO data, address and control fault latches may be read from their respective registers, shown in Table 3.24 These latches are loaded with the write cycle the DARF was attempting on the VMEbus when it received a $\overline{\text{VBERRI}}$ (BERR* from the VMEbus). The registers do not show the cycle waiting to be performed next at the output stage of the transmit FIFO. The DARF64 can shift the TXFIFO forward using the TXSHFT bit in the DCSR register. Shifted TXFIFO contents may then be read through the TXCTL, TXADDR, and TXDATA registers.

Table 3.24 : Transmit FIFO Registers

Bits	Function							
31-24	NOT USED							
23-16	NOT USED							
15-08	NOT USED							
07-00	MBLT	BLT	SIZ1	SIZ0	SPC1	SPC0	SUPER	TYPE

a) **TXCTL** Register

Bits	Function
31-24	TXA (Address at Output Stage of Transmit FIFO) BYTE 3
23-16	TXA (Address at Output Stage of Transmit FIFO) BYTE 2
15-08	TXA (Address at Output Stage of Transmit FIFO) BYTE 1
07-00	TXA (Address at Output Stage of Transmit FIFO) BYTE 0

b) **TXADDR** Register

Bits	Function
31-24	TXD(Data at Output Stage of Transmit FIFO) - BYTE 3
23-16	TXD (Data at Output Stage of Transmit FIFO) BYTE 2
15-08	TXD (Data at Output Stage of Transmit FIFO) BYTE 1
07-00	TXD (Data at Output Stage of Transmit FIFO) BYTE 0

c) **TXDATA** Register

The failed cycle read back from these registers may be retried by the CPU by writing the same data to the address again. This write will occur after any other writes queued ahead are completed. The order of writes may be significant in some situations. The cycle is not automatically retried by the DARF when the VBERR flag is cleared - the DARF resumes with the next cycle in the FIFO.

3.13.2 Loopback Mode

The DARF can be put into a loopback mode. In this mode, any write cycle by the local CPU to its own slave image does not go directly to memory but goes out to the VMEbus, and then back in through the FIFOs. VMEbus address and data transceivers drive the cycle onto the VMEbus, but the DARF loads the cycle into its receive FIFO from its own output pins, so the loopback mode does not test the external VMEbus buffers. $\overline{\text{VDTACKO}}$ (DTACK✱ on VMEbus) is asserted and received through the usual VMEbus receiver by the DARF to end the cycle. Loopback is implemented in the same way for the DARF32 and the DARF64, although consideration should be given to the larger size Transmit and Receive FIFOs provided in the DARF64.

Read cycles are not looped back, because the DARF needs local bus mastership to complete the read and the CPU is the current owner. These cycles are directed into memory using the $\overline{\text{RAMSEL}}$ pin.

The loopback control bit LPBK is in the Mode Control Register (Table 3.25). LPBK can be set or cleared at any time, as it only affects the path of CPU cycles to its own slave image as such an access will not be in progress during programming. Bit LPBK can be set only if the DARF is in decoupled mode. The mode does not function while the card is in BI-mode, since the VMEbus is required for the looped-back transfer. The DARF does not accept the transfer at all in this case.

Table 3.25 : DARF Mode Control Register

Bits	Function							
31-24	DTCSIZ	RESERVED	RMCRETRY	DMABEN	RMCPIN	BUSSIZ	SWAP	040
23-16	DPRIV	NOREL	FILL	DMAA64	MBLT	BLT	DMAD16	DMARD
15-08	FIFOBEN	BLEN *(2 bits)*		RXATOM	TXATOM	A24SLVEN	DMA24	BERRCHK
07-00	TASCON	A24P0	LPBK	DISRX	A24DI	A16DI	PROT	VINEN

While bit LPBK is set, data coherency is no longer assured. For example, the CPU can dispatch a decoupled write to its own memory through the DARF FIFOs, and then immediately begin a read to the direct image of memory. The read would likely start before the write cycle is propagated to the VMEbus and back again, causing stale data to be read.

While the DARF is in loopback mode, the base address and size of the slave image, the access protection and the integrity of the FIFOs can all be tested. Address and data faults either on the VMEbus or in the VMEbus transceivers cannot be detected, since address and data signals are received by the DARF from its own output pins. Access protection can be verified as follows:

1. Write to the slave image with no protection
2. Allow time for the write to occur, and then verify that the write succeeded
3. Enable protection
4. Check that Status and Control Register bit VBERR is set when the write is repeated, or when another write is attempted

The base and size of the slave image can be tested in two ways. The local slave image detector can be verified while still in BI-mode. Without using loopback, reads or writes to memory are completed just as to the direct image. The second slave image detector exists on the VMEbus side and loopback mode or interactive testing using another VMEbus master can be used to verify that circuit.

Integrity of the FIFOs can be checked by disabling the receive FIFO and shifting and unloading each cycle from the FIFO. This is done by reading the address, data and control sections through the DARF registers. This can be done only when no other card is accessing the DARF. The receive FIFO disable bit, (DISRX, Bit 4), is shown in the Mode Control Register (Table 3.26). Setting DISRX high stops the FIFO from requesting the local bus and executing its own write cycles. The receive FIFO shift bit, (RXSHFT, Bit 10), is in the Control and Status Register (Table 3.27). Writing one to RXSHFT shifts the receive FIFO one step — the DARF clears RXSHFT after completing the shift.

Table 3.26 : DARF Mode Control Register

Bits	Function							
31-24	DTCSIZ	RESERVED	RMCRETRY	DMABEN	RMCPIN	BUSSIZ	SWAP	040
23-16	DPRIV	NOREL	FILL	DMAA64	MBLT	BLT	DMAD16	DMARD
15-08	FIFOBEN	BLEN *(2 bits)*		RXATOM	TXATOM	A24SLVEN	DMA24	BERRCHK
07-00	TASCON	A24P0	LPBK	DISRX	A24DI	A16DI	PROT	VINEN

Table 3.27 : Control and Status Register

Bits	Function							
31-24	NOT USED				DEVICE *(4 bits)*			
23-16	NOT USED				CERR			
15-08	TXSHFT	RETRY	RMCERR	A64BARDY	BARDY	RXSHFT	RXRST	TXRST
07-00	RXHD	TXHD	DLBER	LMHD	LBERR	VBERR	DONE	DMAGO

The three receive FIFO sections are read through separate registers (Table 3.28). The address read from the register is the value latched from the VMEbus, prior to any address bit zeroing that is done because of the addressing mode (A32 or A24).

Table 3.28 : Receive FIFO Registers

Bits	Function							
31-24	NOT USED							
23-16	NOT USED							
15-08	NOT USED							
07-00	MBLT	BLT	SIZ1	SIZ0	SPC1	SPC0	SUPER	TYPE

a) **RXCTL Register**

Bits	Function
31-24	RADDR (Address at Output Stage of Receive FIFO) BYTE 3
23-16	RADDR (Address at Output Stage of Receive FIFO) BYTE 2
15-08	RADDR (Address at Output Stage of Receive FIFO) BYTE 1
07-00	RADDR (Address at Output Stage of Receive FIFO) BYTE 0

b) **RXADDR Register**

Bits	Function
31-24	RDATA (Address at Output Stage of Receive FIFO) BYTE 3
23-16	RDATA (Address at Output Stage of Receive FIFO) BYTE 2
15-08	RDATA (Address at Output Stage of Receive FIFO) BYTE 1
07-00	RDATA (Address at Output Stage of Receive FIFO) BYTE 0

c) **RXDATA Register**

3.14 Description of DARF Signals

A detailed description of the signals connected to the DARF is given below. The signals are organized by functional group: VMEbus Signals , CPU Signals (direct connect) and Other Local Signals.

3.14.1 VMEbus Signals

The DARF is connected to the VMEbus through external buffers, with VMEbus address, data and strobe flow direction controlled by the VADDROUT, VDATAOUT and VSTRBOUT signals respectively.

VAS	**ACTIVE LOW TRISTATE BIDIRECT**
VMEbus Address Strobe — falling edge indicates valid addresses are on the bus. By continuing to assert VAS , ownership of the bus is maintained after BBSY* is negated.	
VDS 1 – 0	**ACTIVE LOW TRISTATE BIDIRECT**
VMEbus Data Strobes — During write cycles, the falling edge indicates valid data on the bus. — During read cycles, assertion indicates a request to provide valid data, until negated.	
VWR	**ACTIVE LOW TRISTATE BIDIRECT**
VMEbus Write signal — timing is defined relative to the data strobes and indicates the direction of data transfer.	
VRMC	**ACTIVE LOW TRISTATE BIDIRECT**
VMEbus Read-Modify-Write signal — locks access to the slave card memory for the duration of a multiple-address read-modify-write operation VRMC use is optional as the IEEE 1014 Rev C specification does not define this signal. This pin may be programmed as the Rev D, VMEbus RETRY pin. When the DARF64 is a master, it monitors the pin, in conjunction with BERR, and provides the RETRY pin status in the Control and Status register.	

Caution: Connecting this signal to the P2 - Reserved "B3" VMEbus pin may cause problems if non-IEEE 1014 Rev D boards installed in the system are also using this pin.

VMEbus Signals (Continued)

$\overline{\text{VLWORD}}$	**ACTIVE LOW TRISTATE BIDIRECT**
VMEbus Longword Data Transfer Size indicator — used in conjunction with the two data strobes ($\overline{\text{VDS 1 – 0}}$) and VADDR 01 to indicate the number of bytes (1 – 4) in the current transfer. In the DARF64, during MBLT transfers, $\overline{\text{VLWORD}}$, serves as data bit D32.	
$\overline{\text{VIACK}}$	**TRISTATE BIDIRECT**
VMEbus Interrupt Acknowledge signal IACK✳ — asserted to indicate the bus transfer is acknowledging an interrupt. The interrupt level is indicated by VMEbus address bits 3 through 1 (VADDR 03 – VADDR 01).	
VAM 5 – 0	**TRISTATE BIDIRECTS**
VMEbus Address Modifier Codes — indicate the address space being accessed (A16, A24, A32, A64), the privilege level (user, supervisor), the cycle type (standard, BLT, MBLT) and the data type (program, data).	
VADDR 31 – 01	**TRISTATE BIDIRECTS**
VMEbus Address Lines 31 through 01. Address bit 0 is indicated indirectly by data strobes $\overline{\text{VDS1}}$ and $\overline{\text{VDS0}}$. In DARF64, during MBLT transfers, VADDR31-01 serve as data bits D33 to D63.	
VDATA 31 – 00	**TRISTATE BIDIRECTS**
VMEbus Data Lines 31 through 00.	
$\overline{\text{VDTACKI}}$	**ACTIVE LOW INPUT**
VMEbus DTACK✳ input — used by the DARF while it is the VMEbus master (and while trying to become master). Indicates that the VMEbus data transfer was successfully completed. This is a direct connect signal for the DARF64.	
$\overline{\text{VDTACKO}}$	**ACTIVE LOW OUTPUT**
VMEbus DTACK✳ output — generated by the DARF when it is the VMEbus slave to indicate that the data transfer was successfully completed.	
$\overline{\text{VBERRI}}$	**ACTIVE LOW INPUT**
VMEbus BERR✳ input — a direct connect signal monitored by the DARF while it is the VMEbus master (and while trying to become master). Indicates that the VMEbus data transfer was not successfully completed.	
$\overline{\text{VBERRO}}$	**ACTIVE LOW OUTPUT**
VMEbus BERR✳ output — generated by the DARF when it is the VMEbus slave. The DARF asserts $\overline{\text{VBERRO}}$ to enforce access protection or to indicate that the data transfer was not successfully completed.	

3.14.2 CPU Signals

Most of the data transfer signals on the local bus are driven by the current bus master; either the CPU (68020 or 68030) or the DARF.

<table>
<tr><td><b>$\overline{\text{KAS}}$</b></td><td><b>ACTIVE LOW TRISTATE BIDIRECT</b></td></tr>
<tr><td colspan="2">Address Strobe on the local CPU bus
as an input, it is used in conjunction with the CPU clock (KCLK) to locate the start of bus cycles
as an output, it indicates that valid addresses, sizes, function codes and read/write state information are on the bus.</td></tr>
<tr><td><b>$\overline{\text{KDS}}$</b></td><td><b>ACTIVE LOW TRISTATE OUTPUT(DARF32)</b>
<b>ACTIVE LOW TRISTATE BIDIRECT(DARF64)</b></td></tr>
<tr><td colspan="2">Data Strobe on the local CPU bus
as an output during read cycles, it indicates that the addressed slave should drive the data bus
as an output during write cycles, it indicates that valid data is on the bus.
When the DARF64 is held low during reset, it can be placed in the 040 mode.</td></tr>
<tr><td><b>$\overline{\text{KWR}}$</b></td><td><b>ACTIVE LOW TRISTATE BIDIRECT</b></td></tr>
<tr><td colspan="2">This signal indicates the direction of data transfer — a high level indicates a read cycle, a low level indicates a write cycle. Note that this is defined relative to the current local bus master.</td></tr>
<tr><td><b>$\overline{\text{KRMC}}$</b></td><td><b>ACTIVE LOW TRISTATE BIDIRECT</b></td></tr>
<tr><td colspan="2">This signal indicates that the current cycle is part of a read-modify-write cycle that must remain indivisible. $\overline{\text{KRMC}}$ remains asserted throughout the read and write cycles forming the RMW operation.</td></tr>
<tr><td><b>KFC 2, 0</b></td><td><b>TRISTATE BIDIRECTS</b></td></tr>
<tr><td colspan="2">Local Bus Function Code bits which indicate the type of transfer — data or code, supervisor or user. Reserved User Function Code 3 is used to indicate DARF64 local bus burst transfers.</td></tr>
<tr><td><b>KFC 1</b></td><td><b>TRISTATE OUTPUT</b></td></tr>
<tr><td colspan="2">Local Bus Function Code bits: indicates the type of transfer: data or code and supervisor or user. Bit 1 is not required by the DARF for VMEbus transfers. Reserved User Function Code 3 is used to indicate DARF64 local bus burst transfers.</td></tr>
</table>

CPU Signals (Continued)

KSIZE 1 – 0	**TRISTATE BIDIRECTS**

Local Bus Data Transfer Size — indicates the number of bytes (including current cycle) to be transferred to complete the operation.

KSIZE1	KSIZE0	Size of Transfer
0	0	32 – bit
0	1	8 – bit
1	0	16 – bit
1	1	24 – bit

KADDR 31 – 0	**TRISTATE BIDIRECT**

Local bus address bits 31 through 00.

KDATA 31 – 0	**TRISTATE BIDIRECT WITH PULL-UP**

Local bus data bits 31 through 00.

$\overline{\text{KDSACK 1 – 0}}$	**ACTIVE LOW TRISTATE BIDIRECTS**

Local bus data transfer and size acknowledge. The slave for the current cycle asserts one or both of these signals to acknowledge the cycle and indicates the size of transfer accepted. The 68020 generates a new cycle with the excess bytes if a slave acknowledges an amount less than that originally requested by the CPU.

$\overline{\text{KDSACK1}}$	$\overline{\text{KDSACK0}}$	Number of Bytes Acknowledged
0	0	4 bytes
0	1	2 bytes
1	0	1 byte
1	1	Wait states inserted

$\overline{\text{KBERR}}$	**OPEN DRAIN BIDIRECT**

Local Bus Error signal — indicates a failed transfer attempt. The DARF also asserts $\overline{\text{KBERR}}$ in conjunction with $\overline{\text{KHALT}}$ to force the local CPU to retry a cycle — when the DARF needs the local bus before responding to an access by the CPU (deadlock resoluιion). When $\overline{\text{KBERR}}$ is released it is momentarily pulled high.

$\overline{\text{KHALT}}$	**ACTIVE LOW TRISTATE BIDIRECT**

Asserted by the DARF in conjunction with $\overline{\text{KBERR}}$ to force the CPU to retry a cycle, as described under the $\overline{\text{KBERR}}$ description above.

KCLK	**INPUT**

Local CPU Clock signal — must be connected directly to the CPU clock input pin to maintain timing relationships between KCLK, $\overline{\text{KAS}}$ and other local bus signals. The DARF32 KCLK input has a TTL switching threshold, while the DARF64 has a CMOS switching threshold.

CPU Signals (Continued)

$\overline{\text{RESET}}$	**ACTIVE LOW INPUT**
Local Reset signal — ends any cycles being performed by the DARF and clears various internal registers.	

3.14.3 Other Local Signals

$\overline{\text{DARFCS}}$	**ACTIVE LOW INPUT**
DARF Chip Select — asserted during a local bus cycle to access the DARF internal registers.	
$\overline{\text{RAMSEL}}$	**ACTIVE LOW OUTPUT**
Memory Select signal — asserted by the DARF to select local memory during accesses from the VMEbus, when the DARF DMA is reading memory or when the local CPU is accessing the VMEbus slave address range of the card.	
$\overline{\text{VSBSEL}}$	**ACTIVE LOW OUTPUT**
VSB Select — asserted by the DARF when a local CPU access to the VMEbus has been mapped onto the VSB (by DARF internal registers). The DARF does not perform a VMEbus cycle, but expects a VSB interface to complete the transfer.	
$\overline{\text{LBRQ}}$	**ACTIVE LOW OUTPUT**
CPU Bus Request — asserted by the DARF when it requires the local bus. $\overline{\text{LBRQ}}$ is normally connected to the ACC.	
$\overline{\text{LBGR}}$	**ACTIVE LOW INPUT**
CPU Bus Grant input to the DARF — indicates the local bus is available for use. When negated, this signal requests that the DARF give up the bus. $\overline{\text{LBGR}}$ is normally connected to the ACC	
$\overline{\text{LMINT}}$	**ACTIVE LOW OUTPUT**
Indicates that there are entries in the Location Monitor FIFO — and can be used as an interrupt to the local CPU. $\overline{\text{LMINT}}$ is negated when the FIFO becomes empty. This signal is usually connected to the ACC as one of the auto-vectored interrupt inputs.	
$\overline{\text{VMEINT}}$	**ACTIVE LOW OUTPUT**
Indicates an event related to VMEbus use has occurred — such as DMA finished or bus error ($\overline{\text{KBERR}}$ or BERR*) received. The signal is negated when appropriate flags in the DARF status register are cleared. $\overline{\text{VMEINT}}$ can also be used as an interrupt to the local CPU. This signal is usually connected to the ACC as one of the auto-vectored interrupt inputs.	

Other Local Signals (Continued)

$\overline{\text{VMEOUT}}$ — **ACTIVE LOW INPUT**

Generated by the local address decoder to indicate that the DARF should attempt to initiate a VMEbus cycle. The address maps within the DARF may indicate that this should be a local cycle or sub-system bus cycle and assert $\overline{\text{RAMSEL}}$ or $\overline{\text{VSBSEL}}$ respectively. Otherwise, a VMEbus cycle is initiated.

$\overline{\text{VIACKRQ}}$ — **ACTIVE LOW INPUT**

VMEbus Interrupt Acknowledge — indicates that the local CPU is acknowledging a VMEbus interrupt. The DARF treats a VMEbus interrupt acknowledge in the same manner as a VMEbus read cycle, except that it also asserts $\overline{\text{VIACK}}$. The $\overline{\text{VIACKRQ}}$ signal is normally provided by the ACC.

$\overline{\text{VECTEN}}$ — **ACTIVE LOW INPUT**

Vector Enable — asserted to the DARF during a VMEbus interrupt acknowledge cycle. This causes the DARF to drive its programmed interrupt acknowledge vector onto the VMEbus and then assert $\overline{\text{VDTACKO}}$. The $\overline{\text{VECTEN}}$ signal is usually connected to the ACC.

$\overline{\text{VMERQ}}$ — **ACTIVE LOW OUTPUT**

VMEbus Request — used to request ownership of the VMEbus. It is negated when the DARF no longer needs the bus. The $\overline{\text{VMERQ}}$ signal is normally connected to the ACC.

$\overline{\text{VMEGR}}$ — **ACTIVE LOW INPUT**

VMEbus Grant — indicates that ownership of the VMEbus has been obtained, although there may still be a data transfer in progress. The signal may be negated to request the DARF to give up the VMEbus. The $\overline{\text{VMEGR}}$ signal is normally connected to the ACC.

VSTRBOUT — **ACTIVE HIGH OUTPUT**

VMEbus Address and Data Strobe Transceiver direction control — the DARF points the control strobe transceivers outward when it is the bus master. In all other cases, the transceivers are pointed inward, presenting a high impedance to the VMEbus.

VADDROUT — **ACTIVE HIGH OUTPUT**

VMEbus Address Transceiver direction control — the DARF controls the direction of the address and AM code transceivers as required for master, slave and bus isolation modes.

VDATAOUT — **ACTIVE HIGH OUTPUT**

VMEbus Data Transceiver direction control — the DARF controls the direction of the data transceivers as required for read, write and bus isolation modes.

$\overline{\text{VASDLY}}$ — **ACTIVE LOW INPUT**

Delayed VMEbus Address Strobe input — used for slave address decode timing. The recommended delay is 20 ns; the delay line is connected between $\overline{\text{VAS}}$ and $\overline{\text{VASDLY}}$

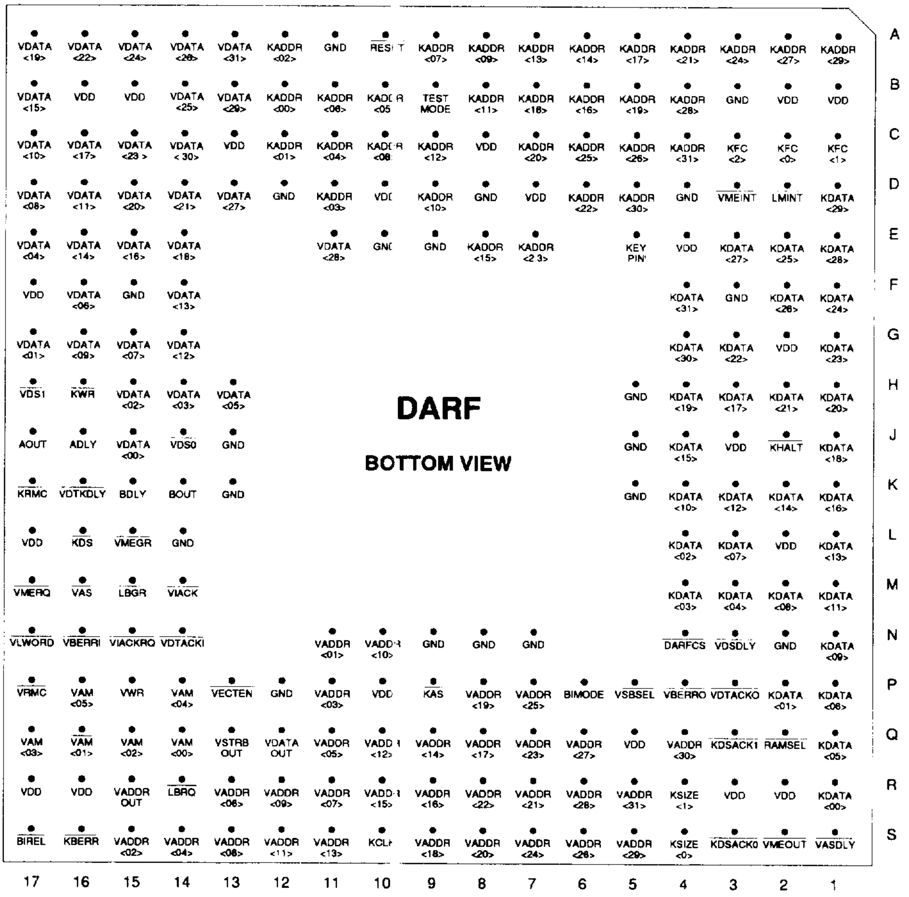
Other Local Signals (Continued)

<table>
<tr><td><b>$\overline{\text{VDSDLY}}$</b></td><td align="right"><b>ACTIVE LOW INPUT</b></td></tr>
<tr><td colspan="2">Delayed first VMEbus data strobe input becoming active — used for minimum DS* assertion time and minimum DS* negation time between back to back cycles. The recommended delay is 40 ns (delay line connected between $\overline{\text{VDSDLY}}$ and the logical OR of $\overline{\text{VDS0}}$ and $\overline{\text{VDS1}}$, such that when either data strobe is true the OR function generates a low).</td></tr>
<tr><td><b>$\overline{\text{VDTKDLY}}$</b></td><td align="right"><b>ACTIVE LOW INPUT</b></td></tr>
<tr><td colspan="2">Delayed VMEbus DTACK* input — used for internal event sequencing. The recommended delay is 30 ns; the delay line is connected between $\overline{\text{VDTACKI}}$ and $\overline{\text{VDTKDLY}}$.</td></tr>
<tr><td><b>$\overline{\text{AOUT}}$</b></td><td align="right"><b>ACTIVE LOW OUTPUT</b></td></tr>
<tr><td colspan="2">Output to an external 40 ns delay line.</td></tr>
<tr><td><b>ADLY</b></td><td align="right"><b>ACTIVE LOW INPUT</b></td></tr>
<tr><td colspan="2">Input from the delay line connected to $\overline{\text{AOUT}}$ — used for timing $\overline{\text{VAS}}$ and other internal operations.</td></tr>
<tr><td><b>$\overline{\text{BOUT}}$</b></td><td align="right"><b>ACTIVE LOW OUTPUT</b></td></tr>
<tr><td colspan="2">Output to an external 40 ns delay line.</td></tr>
<tr><td><b>$\overline{\text{BDLY}}$</b></td><td align="right"><b>ACTIVE LOW INPUT</b></td></tr>
<tr><td colspan="2">Input from the delay line connected to $\overline{\text{BOUT}}$ — used for timing $\overline{\text{VDS 1 – 0}}$ and other internal operations.</td></tr>
<tr><td><b>BIMODE</b></td><td align="right"><b>ACTIVE HIGH INPUT</b></td></tr>
<tr><td colspan="2">BI-mode input
— used to isolate the DARF from the VMEbus.
— normally connected to the ACC
The DARF tristates its VMEbus buffers, points its VMEbus transceivers inward and does not initiate or respond to any VMEbus cycles until BIMODE is negated.</td></tr>
<tr><td><b>$\overline{\text{BIREL}}$</b></td><td align="right"><b>ACTIVE LOW OUTPUT</b></td></tr>
<tr><td colspan="2">Output from the DARF used to clear the BI-mode signal (BIMODE). $\overline{\text{BIREL}}$ is pulsed every time a write is made to the DARF Location Monitor. A write to the Location Monitor is the protocol for exiting BI-mode. The $\overline{\text{BIREL}}$ signal is normally connected to the ACC.</td></tr>
<tr><td><b>TESTMODE</b></td><td align="right"><b>ACTIVE HIGH INPUT</b></td></tr>
<tr><td colspan="2">Input signal used to test the DARF during chip fabrication.</td></tr>
</table>

⚠ *Caution: This input must be connected to ground for proper operation.*

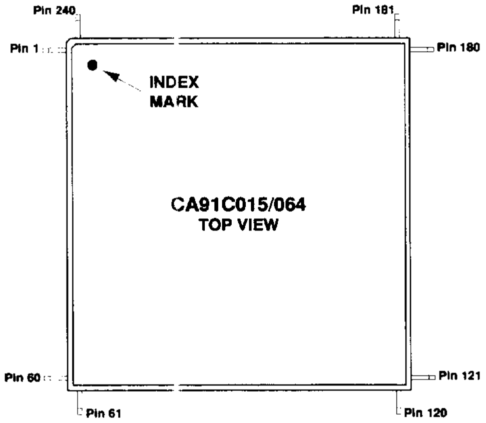
3.15 Pin Configuration

The 224 pin PGA configuration of the DARF is shown in 5, and the 240 pin PQFP configuration in Figure 3.4b.



17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
VDATA <19>	VDATA <22>	VDATA <24>	VDATA <28>	VDATA <31>	KADDR <02>	GND	RESET	KADDR <07>	KADDR <09>	KADDR <13>	KADDR <14>	KADDR <17>	KADDR <21>	KADDR <24>	KADDR <27>	KADDR <29>	A
VDATA <15>	VDD	VDD	VDATA <25>	VDATA <29>	KADDR <00>	KADDR <06>	KADDR <05	TEST MODE	KADDR <11>	KADDR <18>	KADDR <16>	KADDR <19>	KADDR <28>	GND	VDD	VDD	B
VDATA <10>	VDATA <17>	VDATA <23 >	VDATA < 30>	VDD	KADDR <01>	KADDR <04>	KADDR <08	KADDR <12>	VDD	KADDR <20>	KADDR <25>	KADDR <26>	KADDR <31>	KFC <2>	KFC <0>	KFC <1>	C
VDATA <08>	VDATA <11>	VDATA <20>	VDATA <21>	VDATA <27>	GND	KADDR <03>	VDD	KADDR <10>	GND	VDD	KADDR <22>	KADDR <30>	GND	VMEINT	LMINT	KDATA <29>	D
VDATA <04>	VDATA <14>	VDATA <16>	VDATA <18>			VDATA <28>	GND	GND	KADDR <15>	KADDR <2 3>		KEY PIN	VDD	KDATA <27>	KDATA <25>	KDATA <28>	E
VDD	VDATA <06>	GND	VDATA <13>										KDATA <31>	GND	KDATA <26>	KDATA <24>	F
VDATA <01>	VDATA <09>	VDATA <07>	VDATA <12>										KDATA <30>	KDATA <22>	VDD	KDATA <23>	G
VDS1	KWR	VDATA <02>	VDATA <03>	VDATA <05>								GND	KDATA <19>	KDATA <17>	KDATA <21>	KDATA <20>	H
AOUT	ADLY	VDATA <00>	VDS0	GND								GND	KDATA <15>	VDD	KHALT	KDATA <18>	J
KRMC	VDTKDLY	BDLY	BOUT	GND								GND	KDATA <10>	KDATA <12>	KDATA <14>	KDATA <16>	K
VDD	KDS	VMEGR	GND										KDATA <02>	KDATA <07>	VDD	KDATA <13>	L
VMERQ	VAS	LBGR	VIACK										KDATA <03>	KDATA <04>	KDATA <08>	KDATA <11>	M
VLWORD	VBERRI	VIACKRQ	VDTACKI			VADDR <01>	VADDR <10>	GND	GND	GND			DARFCS	VDSDLY	GND	KDATA <09>	N
VRMC	VAM <05>	VWR	VAM <04>	VECTEN	GND	VADDR <03>	VDD	KAS	VADDR <19>	VADDR <25>	BIMODE	VSBSEL	VBERRO	VDTACKO	KDATA <01>	KDATA <08>	P
VAM <03>	VAM <01>	VAM <02>	VAM <00>	VSTRB OUT	VDATA OUT	VADDR <05>	VADDR <12>	VADDR <14>	VADDR <17>	VADDR <23>	VADDR <27>	VDD	VADDR <30>	KDSACK1	RAMSEL	KDATA <05>	Q
VDD	VDD	VADDR OUT	LBRQ	VADDR <06>	VADDR <09>	VADDR <07>	VADDR <15>	VADDR <16>	VADDR <22>	VADDR <21>	VADDR <28>	VADDR <31>	KSIZE <1>	VDD	VDD	KDATA <00>	R
BIREL	KBERR	VADDR <02>	VADDR <04>	VADDR <08>	VADDR <11>	VADDR <13>	KCLK	VADDR <18>	VADDR <20>	VADDR <24>	VADDR <28>	VADDR <29>	KSIZE <0>	KDSACK0	VMEOUT	VASDLY	S

Figure 3.6a : Pin Configuration for 224-PIN PGA Package



240 PIN PLASTIC QFP

1. VDD
2. KADDR31
3. KFC0
4. KFC2
5. KFC1
6. VSS
7. LMINT
8. VMEINT
9. KDATA31
10. KDATA30
11. KDATA29
12. VDD
13. KDATA28
14. KDATA27
15. KDATA26
16. KDATA25
17. KDATA24
18. VSS
19. VSS
20. KDATA23
21. KDATA22
22. KDATA21
23. VDD
24. KDATA20
25. KDATA19
26. KDATA18
27. KDATA17
28. KHALT
29. VSS
30. VSS
31. VSS
32. KDATA16
33. KDATA15
34. KDATA14
35. VDD
36. VDD
37. KDATA13
38. KDATA12
39. KDATA11
40. KDATA10
41. KDATA09
42. VSS
43. VSS
44. KDATA08
45. KDATA07
46. KDATA06
47. VDD
48. KDATA05
49. KDATA04
50. KDATA03
51. KDATA02
52. KDATA01
53. VSS
54. KDATA00
55. VDSDLY
56. RAMSEL
57. DARFCS
58. VASDLY
59. VDTACKO
60. VDD
61. VDD
62. VBERRO
63. VMEOUT
64. KDSACK1
65. KDSACK0
66. KSIZE1
67. KSIZE0
68. VSBSEL
69. BIMODE
70. VSS
71. VADDR31
72. VADDR30
73. VADDR29
74. VDD
75. VDD
76. VADDR28
77. VADDR27
78. VADDR26
79. VADDR25
80. VADDR24
81. VADDR23
82. VADDR22
83. VADDR21
84. VADDR20
85. VADDR19
86. VADDR18
87. VADDR17
88. VADDR16
89. VSS
90. VSS
91. VSS
92. KCLK
93. NC
94. KAS
95. VADDR15
96. VADDR14
97. VADDR13
98. VADDR12
99. VADDR11
100. VADDR10
101. VADDR09
102. VDD
103. VDD
104. VADDR08
105. VADDR07
106. VADDR06
107. VADDR05
108. VADDR04
109. VADDR03
110. VADDR02
111. VADDR01
112. VSS
113. VDATOUT
114. VADDOUT
115. VSTRBOUT
116. KBERR
117. LBRQ
118. BIREL
119. VECTEN
120. VDD
121. VDD
122. VAM 00
123. VAM 01
124. VAM 02
125. VAM 03
126. VAM 04
127. VAM 05
128. VWR
129. VRMC
130. VSS
131. VIACK
132. VDTACKI
133. VLWORD
134. VIACKRQ
135. VAS
136. VBERRI
137. VMERQ
138. LBGR
139. VDD
140. VDD
141. VMEGR
142. VDTKDLY
143. KDS
144. KRMC
145. BOUT
146. AOUT
147. BDLY
148. ADLY
149. NC
150. VSS
151. VSS
152. VSS
153. VDS1
154. VDS0
155. KWR
156. VDATA00
157. VDATA01
158. VDATA02
159. VDD
160. VDD
161. VDATA03
162. VDATA04
163. VDATA05
164. VDATA06
165. VDATA07
166. VDATA08
167. VDATA09
168. VDATA10
169. VSS
170. VDATA11
171. VDATA12
172. VDATA13
173. VDATA14
174. VDATA15
175. VDATA16
176. VDATA17
177. VDATA18
178. VDATA19
179. VDATA20
180. VDD
181. VDD
182. VDATA21
183. VDATA22
184. VDATA23
185. VDATA24
186. VDATA25
187. VDATA26
188. VDATA27
189. VSS
190. VDATA28
191. VDATA29
192. VDATA30
193. VDATA31
194. VDD
195. KADDR00
196. KADDR01
197. KADDR02
198. KADDR03
199. VSS
200. VSS
201. KADDR04
202. KADDR05
203. KADDR06
204. RESET
205. VDD
206. VDD
207. KADDR07
208. KADDR08
209. TESTMODE
210. VSS
211. VSS
212. VSS
213. KADDR09
214. KADDR10
215. KADDR11
216. KADDR12
217. KADDR13
218. VDD
219. VDD
220. KADDR14
221. KADDR15
222. KADDR16
223. VSS
224. KADDR17
225. KADDR18
226. KADDR19
227. KADDR20
228. KADDR21
229. VDD
230. KADDR22
231. KADDR23
232. KADDR24
233. KADDR25
234. VSS
235. KADDR26
236. KADDR27
237. KADDR28
238. KADDR29
239. KADDR30
240. NC

NC=NO CONNECTION

Figure 3.6b : Pin Configuration for 240-PIN PQFP Package

A list of the signals connected to the DARF is provided in Table 3.29 and Table 3.30. The signals are organized in the following functional groups:

- VMEbus and Local Bus signals
- VMEbus Address Input and Output Signal Bits
- VMEbus Data Input and Output Signal Bits
- Local Bus Address Input and Output Signal Bits
- Local Bus Data Input and Output Signal Bits
- Power and Ground Pins

Table 3.29 : VMEbus and Local Bus Signals

Signal	Pins		Signal	Pins		Signal	Pins	
	PGA	PQFP		PGA	PQFP		PGA	PQFP
$\overline{\text{ADLY}}$	J16	148	KSIZE1	R4	66	VDATAOUT	Q12	113
$\overline{\text{AOUT}}$	J17	146	$\overline{\text{LBGR}}$	M15	138	$\overline{\text{VDS0}}$	J14	154
$\overline{\text{BDLY}}$	K15	147	$\overline{\text{LBRQ}}$	R14	117	$\overline{\text{VDS1}}$	H17	153
BIMODE	P6	69	$\overline{\text{LMINT}}$	D2	7	$\overline{\text{VDSDLY}}$	N3	55
$\overline{\text{BIREL}}$	S17	118	N/C	E5		$\overline{\text{VDTACKI}}$	N14	132
$\overline{\text{BOUT}}$	K14	145	$\overline{\text{RAMSEL}}$	Q2	56	$\overline{\text{VDTACKO}}$	P3	59
$\overline{\text{DARFSEL}}$	N4	57	$\overline{\text{RESET}}$	A10	204	$\overline{\text{VDTKDLY}}$	K16	142
$\overline{\text{KAS}}$	P9	94	TESTMODE	B9	209	$\overline{\text{VECTEN}}$	P13	119
$\overline{\text{KBERR}}$	S16	116	VADDROUT	R15	114	$\overline{\text{VIACK}}$	M14	131
KCLK	S10	92	VAM 00	Q14	122	$\overline{\text{VIACKRQ}}$	N15	134
$\overline{\text{KDS}}$	L16	143	VAM 01	Q16	123	$\overline{\text{VLWORD}}$	N17	133
$\overline{\text{KDSACK0}}$	S3	65	VAM 02	Q15	124	$\overline{\text{VMEGR}}$	L15	141
$\overline{\text{KDSACK1}}$	Q3	64	VAM 03	Q17	125	$\overline{\text{VMEINT}}$	D3	8
KFC0	C2	3	VAM 04	P14	126	$\overline{\text{VMEOUT}}$	S2	63
KFC1	C1	5	VAM 05	P16	127	$\overline{\text{VMERQ}}$	M17	137
KFC2	C3	4	$\overline{\text{VAS}}$	M16	135	$\overline{\text{VRMC}}$	P17	129
$\overline{\text{KHALT}}$	J2	28	$\overline{\text{VASDLY}}$	S1	58	$\overline{\text{VSBSEL}}$	P5	68
$\overline{\text{KRMC}}$	K17	144	$\overline{\text{VBERRI}}$	N16	136	VSTRBOUT	Q13	115
KSIZE0	S4	67	$\overline{\text{VBERRO}}$	P4	62	$\overline{\text{VWR}}$	P15	128

Table 3.30a : VMEbus Address Input and Output Signal Bits

Signal	PGA	PQFP	Signal	PGA	PQFP	Signal	PGA	PQFP
	Pin	Pin		Pin	Pin		Pin	Pin
VADDR 01	N11	111	VADDR 12	Q10	98	VADDR 23	Q7	81
VADDR 02	S15	110	VADDR 13	S11	97	VADDR 24	S7	80
VADDR 03	P11	109	VADDR 14	Q9	96	VADDR 25	P7	79
VADDR 04	S14	108	VADDR 15	R10	95	VADDR 26	S6	78
VADDR 05	Q11	107	VADDR 16	R9	88	VADDR 27	Q6	77
VADDR 06	R13	106	VADDR 17	Q8	87	VADDR 28	R6	76
VADDR 07	R11	105	VADDR 18	S9	86	VADDR 29	S5	73
VADDR 08	S13	104	VADDR 19	P8	85	VADDR 30	Q4	72
VADDR 09	R12	101	VADDR 20	S8	84	VADDR 31	R5	71
VADDR 10	N10	100	VADDR 21	R7	83			
VADDR 11	S12	99	VADDR 22	R8	82			

Table 3.30b : VMEbus Data Input and Output Signal Bits

Signal	PGA	PQFP	Signal	PGA	PQFP	Signal	PGA	PQFP
	Pin	Pin		Pin	Pin		Pin	Pin
VDATA 00	J15	156	VDATA 11	D16	170	VDATA 22	A16	183
VDATA 01	G17	157	VDATA 12	G14	171	VDATA 23	C15	184
VDATA 02	H15	158	VDATA 13	F14	172	VDATA 24	A15	185
VDATA 03	H14	161	VDATA 14	E16	173	VDATA 25	B14	186
VDATA 04	E17	162	VDATA 15	B17	174	VDATA 26	A14	187
VDATA 05	H13	163	VDATA 16	E15	175	VDATA 27	D13	188
VDATA 06	F16	164	VDATA 17	C16	176	VDATA 28	E11	190
VDATA 07	G15	165	VDATA 18	E14	177	VDATA 29	B13	191
VDATA 08	D17	166	VDATA 19	A17	178	VDATA 30	C14	192
VDATA 09	G16	167	VDATA 20	D15	179	VDATA 31	A13	193
VDATA 10	C17	168	VDATA 21	D14	182			

Table 3.30c : Local Bus Address Input and Output Signal Bits

Signal	PGA	PQFP	Signal	PGA	PQFP	Signal	PGA	PQFP
	Pin	Pin		Pin	Pin		Pin	Pin
KADDR 00	B12	195	KADDR 11	B8	215	KADDR 22	D6	230
KADDR 01	C12	196	KADDR 12	C9	216	KADDR 23	E7	231
KADDR 02	A12	197	KADDR 13	A7	217	KADDR 24	A3	232
KADDR 03	D11	198	KADDR 14	A6	220	KADDR 25	C6	233
KADDR 04	C11	201	KADDR 15	E8	221	KADDR 26	C5	235
KADDR 05	B10	202	KADDR 16	B6	222	KADDR 27	A2	236
KADDR 06	B11	203	KADDR 17	A5	224	KADDR 28	B4	237
KADDR 07	A9	207	KADDR 18	B7	225	KADDR 29	A1	238
KADDR 08	C10	208	KADDR 19	B5	226	KADDR 30	D5	239
KADDR 09	A8	213	KADDR 20	C7	227	KADDR 31	C4	2
KADDR 10	D9	214	KADDR 21	A4	228			

Table 3.30d : Local Bus Data Input and Output Signal Bits

Signal	PGA	PQFP	Signal	PGA	PQFP	Signal	PGA	PQFP
	Pin	Pin		Pin	Pin		Pin	Pin
KDATA 00	R1	54	KDATA 11	M1	39	KDATA 22	G3	21
KDATA 01	P2	52	KDATA 12	K3	38	KDATA 23	G1	20
KDATA 02	L4	51	KDATA 13	L1	37	KDATA 24	F1	17
KDATA 03	M4	50	KDATA 14	K2	34	KDATA 25	E2	16
KDATA 04	M3	49	KDATA 15	J4	33	KDATA 26	F2	15
KDATA 05	Q1	48	KDATA 16	K1	32	KDATA 27	E3	14
KDATA 06	P1	46	KDATA 17	H3	27	KDATA 28	E1	13
KDATA 07	L3	45	KDATA 18	J1	26	KDATA 29	D1	11
KDATA 08	M2	44	KDATA 19	H4	25	KDATA 30	G4	10
KDATA 09	N1	41	KDATA 20	H1	24	KDATA 31	F4	9
KDATA 10	K4	40	KDATA 21	H2	22			

Table 3.31 : Pin assignments for Power

V_{SS} Pins						V_{DD} Pins				
PGA			PQFP			PGA			PQFP	
A11	J5		6	112		B1	G2		1	121
B3	J13		18	130		B2	J3		12	139
D4	K5		19	150		B15	L2		23	140
D8	K13		29	151		B16	L17		35	159
D12	L14		30	169		C8	P10		36	160
E9	N2		31	189		C13	Q5		47	180
E10	N8		42	199		D7	R2		60	181
F3	N9		43	200		D10	R3		61	194
F15	P12		53	210		E4	R16		74	205
H5			70	211		F17	R17		75	206
			89	212					102	218
			90	223					103	219
			91	234					120	229

Notes:

1. There is no direct relationship between PGA pin power assignments and PQFP power assignments.
2. PQFP pins 93, 149, and 240 have no connections.