System Bus Specification

General

The system bus is an N-port time-division multiplexed transmission switch. A module installed in any of the N ports (backplane positions/slots) can send (receive) transmissions to (from) any of the N ports, including itself. Each transmission consists of a SOURCE port address, a DESTINATION port address, a transmission TYPE and 8 bytes of "DATA". The transmission TYPE determines what the DATA field contains. Error detection is provided for the SOURCE slot address, DESTINATION slot address and transmission TYPE fields and optionally provided for the DATA field. The maximum value of N is 16 (ports). The clock rate is 20 MHz. The instantaneous peak data transfer rate is (8 Bytes * 20 MHz =) 160 MB/sec.

Modules installed in bus ports interact with each other by exchanging transmissions over the bus. There are two types of transmissions, COMMANDS and RESPONSES. A module on the bus begins an interaction with another module by sending a COMMAND. The source of the COMMAND is the MASTER for that interaction; the destination of the COMMAND is the SLAVE. The SLAVE sends a RESPONSE back to the MASTER if required to complete the COMMAND.

Bus Arbitration and Flow Control

The bus arbiter controls access to the bus. To transmit on the bus, a module asserts ARB_REQUEST, ARB_DEST[3:0] (the DESTINATION port address) and optionally, a request modifier (ARB_RESP and/or ARB_MODIFY) to the arbiter. If the request is to send a COMMAND, the arbiter checks that the destination port has a COMMAND input buffer available. A port with a COMMAND input buffer available is said to be READY. If the request is to send a RESPONSE, the module also asserts ARB_RESP to the arbiter. In this case the destination port is required to have to have enough RESPONSE input buffers available for the size of the RESPONSE it requested. Ports wanting to send COMMANDS to destinations that are READY and ports wanting to send RESPONSES arbitrate for time slots on the bus. Arbitration occurs for each available time slot. The arbiter asserts ARB_GRANT to each port that wins an arbitration.

Arbitration priority is a function of the port number of the requesting port. Ports are numbered (N-1) to 0. The ports are divided into 2 groups. Ports (N-1) thru (N-n) form the first group. Ports (N-n-1) thru 0 form the second group. The priority of ports in the first group is the same as their port number. Port (N-1) has the highest priority, and port (N-n) has the lowest. The priority of ports in the second group is lower than that of the ports in the first group. And unlike the first group, the

priority of the ports in the second group is periodically changed so that each port in the group has the same average priority. Half the time, the priority of the ports in the second group is the same as their port number (Port (N-n-1) has the highest priority, and port 0 has the lowest). The rest of the time, the priority of the ports is inverse to their port number (port 0 has the highest priority, and port (N-n-1) has the lowest).

The initial implementations of the sbus and arbiter have the same number of ports in each group (n = N/2).

Each module indicates to the arbiter when it has COMMAND input buffers available. Each COMMAND input buffer holds one transmission. If the module has at least one COMMAND buffer available, it asserts ARB_READY1 to the arbiter. If the module has at least 3 COMMAND buffers available, it may also assert ARB_READY3 to the arbiter. When the arbiter grants the transmission of a COMMAND, it asserts ARB_RDY_DEC to the destination port to inform the destination of the grant. This allows the destination port to update its values of ARB_READY1 and ARB_READY3 while the transmission is occurring.

The maximum rate at which a module can receive COMMANDS is determined by whether ARB_READY3 is asserted. Assertion of ARB_READY3 allows a COMMAND to arrive every bus clock tick. If ARB_READY3 is not asserted, COMMANDS can arrive no more frequently than every other bus clock tick. RESPONSES can arrive every bus clock tick.

Interlocked Sequences

The bus arbiter also supports interlocked sequences of operations. These sequences are required to to support the TAS, CAS and CAS2 instructions of the Motorola 68020 and are a generalization of the READ/MODIFY/WRITE operation. Interlocked sequences are atomic to each other and are composed of any number of READ and WRITE commands. The signal ARB_LOCK is asserted by the arbiter when an interlocked sequence is in progress.

To request an interlocked sequence, a module asserts ARB_REQUEST, ARB_DEST[3:0] (of the first interlocked COMMAND) and ARB_MODIFY to the arbiter. The interlocked sequence begins when ARB_LOCK is not asserted and a port asserting ARB_REQUEST and ARB_MODIFY to the arbiter is granted the bus. The interlocked sequence lasts as long as ARB_MODIFY is asserted by the locking module and ends when the locking module deasserts ARB_MODIFY.

To minimize performance loss, an interlocked sequence does not lock out commands from ports not asserting MODIFY to the arbiter.

Bus Transmissions

The arbiter grants access to the bus by asserting ARB_GRANT to each module that wins an arbitration. ARB_GRANT is asserted to a module for one bus clock cycle. A module receiving ARB_GRANT may transmit on the bus during the clock cycle immediately following the clock cycle in which ARB_GRANT was received.

When a module is GRANTED the bus to send a RESPONSE, the module may use

the bus for up to four consecutive clock cycles by asserting ARB_BURST to the arbiter during all but the last clock cycle. The assertion of ARB_BURST during a clock cycle prevents the arbiter from issuing ARB_GRANT to any module for use of the bus during the next clock cycle. Only RESPONSES can be sent in BURST mode. BURST mode RESPONSES can be to one or more DESTINATION ports.

When a module transmits on the bus, it asserts the signal BUS_ACTIVE to indicate the presence of a transmission.

Each module monitors all bus transmissions. Each transmission is first checked for destination field errors. A module recognizes a transmission as addressed to it if the destination field contains no detected errors and matches the module's port (slot) number. The transmission is then checked for source and type field errors. If the TYPE of the transmission is not CONTROL_WRITE and BUS_DATA_PAR_VLD was asserted, the DATA field is checked for parity errors. If the transmission is a RESPONSE, it must be expected and the SOURCE field must be that of the expected source.

Each transmission received without detected error is indicated by asserting BUS_ACK on the bus during the second clock cycle after the transmission. Transmissions received with one or more detected errors or with a TYPE field value that the module can not process are indicated by asserting BUS_NACK on the bus during the second clock cycle after the transmission.

Errors in the DATA field of transmissions of TYPE CONTROL_WRITE or in transmissions with BUS_DATA_PAR_VLD deasserted are ignored. These transmissions are ACKed or NACKed based on the detection of errors in the source and type fields.

Only the destination port asserts BUS_ACK or BUS_NACK for a transmission. Transmissions with destination field errors or destination fields not matching the port (slot) number of any installed module are neither ACKed nor NACKed.

When a system bus transmission fails, retry is permitted, but not required. If retry is attempted, the module that issued the COMMAND resulting in the failed transmission restarts the transaction by reissuing the COMMAND.

Transmission Format

Each transmission has a DESTINATION field, a SOURCE field, a TYPE field and a DATA field. The DESTINATION field contains the destination port number and, for error detection, the compliment of the DESTINATION port number . The SOURCE field contains the source port number and its compliment. The TYPE field indicates the type of COMMAND or RESPONSE and the format of the DATA field. The TYPE field has a parity bit for single-bit error detection. A parity bit is also defined for each byte of the DATA field for single-bit error detection, but implememtation is optional. The source of a bus transmission indicates whether data parity bits have been sent by asserting BUS_DATAPAR_VLD during the transmission.

The DATA field is organized as 8 bytes of 8 bits each. The bytes are numbered 0 through 7 with byte 0 the most significant and of lowest address. Bits within a byte are numbered 7 through 0 with bit 7 the most significant. Each bit in the DATA field has a name of the form BUS_DATA[B,b] where B is

the byte number and b is the bit number within the byte. The optional parity bit for byte B is BUS_DATA_PAR[B]. Data field parity is even. The bit BUS_DATAPAR_VLD indicates whether data field parity is implemented.

System addressing is by 4 bits of physical port number and 32 bits of offset. The address of an operand is the address of its first (most significant and lowest address) byte.

The bus COMMANDS are:

READ [size = 1, 2, 3, 4, 8, 16 or 32 bytes]

The operand of a READ command for 4 bytes or less must not cross a long word (4 byte) boundary. The operand of a READ command for 8, 16 or 32 bytes must be aligned on an 8, 16 or 32 byte boundary, respectively. READs of 8, 16 and 32 bytes are not supported by all module types. The offset address of the operand is sent in bytes 4 through 7 of the DATA field. Bytes 0 through 3 of the DATA field are undefined.

The operand is returned in the DATA field of one or more RESPONSE transmissions. The operand is aligned in the DATA field for an 8 byte wide port. Bytes in the DATA field that are not part of the requested operand are undefined.

READ operands longer than 8 bytes require multiple 8 byte RESPONSE transmissions. Bytes are returned in order of increasing byte address with the bytes of lowest address returned first. Multiple RESPONSE transmissions may be sent one at a time or in one or more bursts.

WRITE [size = 1, 2, 3 or 4 bytes]

The operand of a WRITE command must not cross a long word boundary. The offset address of the operand is sent in bytes 4 through 7 of the DATA field.

The WRITE DATA is sent in bytes 0 through 3 of the data field and must be aligned for a 4 byte wide port.

CONTROL WRITE [signal = 0, 1, 2 or 3] [value = 0 or 1]

The CONTROL_WRITE command allows one of several control signals in a module to be asserted or deasserted even in the presence of errors in the DATA field. The specified signal is set to the specified value. The DATA field is undefined.

The bus RESPONSES are:

RESPONSE [response type = DATA, ERROR DATA 0, ERROR DATA 1 or ERROR]

DATA is the normal response to a READ command. A READ command for more than 8 bytes requires more than one RESPONSE transmission.

ERROR DATA is the response to a READ command encountering a detected but uncorrectable data error. It contains the requested data as read or after correction has been attempted.

ERROR DATA 0 indicates that the error was detected by the device responding to the READ.

ERROR DATA 1 indicates that the error was a transmission error in the DATA field detected by the IO MODULE.

ERROR is the response to a READ command that is somehow recognized as having failed to read anything. The DATA field of the response is undefined.

The format of the transmission TYPE field is:

BUS_TYPE [5:0] = [3 bit type field],[3 bit modifier field]

BUS_TYPE [5:3]	type
0	RESPONSE
1	--
2	--
3	READ
4	--
5	WRITE
6	CONTROL WRITE
7	--

BUS_TYPE [2:0]	size	response type
0	4 bytes	ERROR
1	1 byte	--
2	2 bytes	ERROR DATA 0
3	3 bytes	ERROR DATA 1
4	8 bytes	DATA
5	16 bytes	--
6	32 bytes	--
7	--	--

BUS_TYPE [2:1]	control signal
0	--
1	module enable
2	module interface enable
3	--

SBUS Configuration Conventions

The assignment of modules to SBUS ports (slots) is determined by a module's maximum transfer rate requirements. The higher the maximum transfer rate, the higher the assigned SBUS priority.

In general, ports N thru (N-n) are for I/O Modules (IOM) and Memory Modules (MM) with I/O Modules having the higher priority. A maximum of 4 I/O Modules can be installed in an SBUS. Ports (N-n-1) thru 0 are for Processor Modules (PM) and Service Processor Modules (SPM).

Transmission Conventions

With the exception of CONTROL WRITE, the target of a READ or WRITE command is specified by its system address. Within a module, the defined offset address space is 256 megabytes or 4 gigabytes.

The control and status registers of a module are located at the top of the offset address space. For ease of use and testing, any bit that can be written in a control register can be read at the same byte and bit address and with the same sense. By definition, status registers are read-only.

If relevant, the size of a module's offset address space is controlled by a bit in one of the module's control registers. On RESET, the size of the offset address space is 256 megabytes. This requirement applies primarily to large memory modules (> 256 MB), I/O modules and I/O adapters.

The ID of each module is contained in a long word status register at offset address 0xfffffffc.

Intelligent modules also have high level message buffers and interrupt request buffers at the top of their offset address space.

The Service Processor module has interrupt acknowledge buffers at the top of the offset address space.

System wide address conventions:

```
0xffff ff04-07  Interrupt level 0 acknowledge (service processor)
0xffff ff0c-0f  Interrupt level 1 acknowledge (service processor)
0xffff ff14-17  Interrupt level 2 acknowledge (service processor)
0xffff ff1c-1f  Interrupt level 3 acknowledge (service processor)
0xffff ff24-27  Interrupt level 4 acknowledge (service processor)
0xffff ff2c-2f  Interrupt level 5 acknowledge (service processor)
0xffff ff34-37  Interrupt level 6 acknowledge (service processor)
0xffff ff3c-3f  Interrupt level 7 acknowledge (service processor)

0xffff ff44-47  Interrupt level 8 acknowledge (service processor)
0xffff ff4c-4f  Interrupt level 9 acknowledge (service processor)
0xffff ff54-57  Interrupt level A acknowledge (service processor)
0xffff ff5c-5f  Interrupt level B acknowledge (service processor)
0xffff ff64-67  Interrupt level C acknowledge (service processor)
0xffff ff6c-6f  Interrupt level D acknowledge (service processor)
0xffff ff74-77  Interrupt level E acknowledge (service processor)
0xffff ff7c-7f  Interrupt level F acknowledge (service processor)

0xffff ff80-9f  Higher level message buffers
```

```
                        (processor, service processor)

        0xffff ff80-83  Command buffer
                        (processor, service processor)

        0xffff ffa0-a3  Interrupt request (service processor)

        0xffff ffb0-b3  Interrupt vector (memory)

        0xffff ffc0-ff  Control and Status registers (All)
        0xffff ffe4-e7  Memory check bits and control (memory)
        0xffff ffec-ef  Memory error information (memory)
        0xffff fff4-f7  Memory error address (memory)
        0xffff fffc-fe  Memory status register (memory)
        0xffff fffc     68020 Interrupt request level set (Computational)
        0xffff ffff     Module ID (All)
```

Electrical Interface

The bus signals are:

BUS_DEST[3:0]	SBus Destination Slot Address [3:0]
BUS_DEST[3:0]*	SBus Destination Slot Address [3:0]*
BUS_SRC[3:0]	SBus Source Slot Address [3:0]
BUS_SRC[3:0]*	SBus Source Slot Address [3:0]*
BUS_TYPE[5:0]	SBus Type [5:0]
BUS_TYPE_PARITY	SBus Type Field Parity
BUS_DATA[07:00]	SBus Data Byte 0 [7:0]
BUS_DATA[17:11]	SBus Data Byte 1 [7:0]
BUS_DATA[27:20]	SBus Data Byte 2 [7:0]
BUS_DATA[37:30]	SBus Data Byte 3 [7:0]
BUS_DATA[47:40]	SBus Data Byte 4 [7:0]
BUS_DATA[57:50]	SBus Data Byte 5 [7:0]
BUS_DATA[67:60]	SBus Data Byte 6 [7:0]
BUS_DATA[77:70]	SBus Data Byte 7 [7:0]
BUS_DATA_PAR[0:7]	SBus Data Parity [0:7]
BUS_DATAPAR_VLD	SBus Data Parity Valid
BUS_ACTIVE	SBus Active
BUS_ACK	SBus Acknowledge

BUS_NACK	SBus Negative Acknowledge

The signals from the arbiter to each port are:

ARB_CLOCK*	The SBUS clock. SBUS clock cycles begin and end on the falling edge of ARB_CLOCK*.
ARB_RDY_DEC*	When asserted, ARB_RDY_DEC indicates that the arbiter has granted the transmission of a COMMAND to this port and new values of ARB_READY1 and/or ARB_READY3 must be computed during the current clock cycle.
ARB_GRANT*	When asserted, ARB_GRANT indicates that the port may transmit on the bus during the next clock cycle and must deassert ARB_REQUEST at the beginning of the next clock cycle unless another transmission request is ready.
ARB_LOCK*	When asserted, ARB_LOCK indicates that an interlocked sequence of operations is in progress. The signal is provided for test purposes only and is not required for normal operation of the bus.
ARB_GRANTERR*	When asserted, GRANTERR indicates that two or more GRANT's were issued for one bus cycle. GRANTERR is asserted during the third cycle after the cycle in which the multiple GRANT fault occurred. The signal is for fault isolation only.

The signals from each port to the arbiter are:

ARB_READY1*	When asserted, ARB_READY1 indicates to the arbiter that the port has at least 1 free COMMAND input buffer.
ARB_READY3*	When asserted, ARB_READY3 indicates to the arbiter that the port has at least 3 free COMMAND input buffers.
ARB_REQUEST*	When asserted, ARB_REQUEST indicates that the port wants transmit a COMMAND or RESPONSE on the bus. ARB_REQUEST must be deasserted in the cycle following the reception of ARB_GRANT unless another ARB_GRANT is desired. In the latter case, ARB_RESP or the next value of ARB_DEST[3:0] must be asserted during the cycle following the reception of ARB_GRANT.
ARB_DEST[3:0]	The destination port address for a COMMAND. ARB_DEST[3:0] must be valid for all cycles

that ARB_REQUEST is asserted and ARB_RESP not asserted. ARB_DEST[3:0] need not be valid when ARB_RESP is asserted or ARB_REQUEST not asserted.

ARB_RESP* — When asserted in conjunction with ARB_REQUEST, ARB_RESP indicates that the port wants to transmit a RESPONSE.

ARB_MODIFY* — When asserted in conjunction with ARB_REQUEST, ARB_MODIFY indicates that the port wants to transmit a READ or WRITE COMMAND that is part of an interlocked sequence of operations. Once a port is granted the bus to begin an interlocked sequence, no other port can begin an interlocked sequence until the locking port deasserts ARB_MODIFY.

ARB_BURST* — When asserted, ARB_BURST prevents any ARB_GRANT from being issued for the next clock cycle. A module may assert ARB_BURST to send several RESPONSE transmissions during consecutive SBUS clock cycles. The module must receive an ABR_GRANT for the SBUS cycle in which it first asserts ARB_BURST and may assert the signal for a maximum of three (3) consecutive SBUS clock cycles. This permits a maximum burst length of 4 clock cycles.

Timing:

The timing of signals passing between a bus module and the arbiter is specified at the backplane connector which connects the pc board of the module to the arbiter. The timing of all signals is relative to the falling (LOW going) edge of ARB_CLOCK*

Signal	Setup Time	Hold Time
ARB_READY1*	15.0 ns	5.0 ns
ARB_READY3*	15.0 ns	5.0 ns
ARB_REQUEST*	15.0 ns	5.0 ns
ARB_RESP*	15.0 ns	5.0 ns
ARB_MODIFY*	15.0 ns	5.0 ns
ARB_DEST[3:0]	15.0 ns	5.0 ns
ARB_BURST*	34.0 ns	5.0 ns
ARB_GRANT*	7.0 ns	6.0 ns

ARB_RDY_DEC*	35.0 ns	4.0 ns

The timing of BUS signals passing between a bus module and the backplane is specified at the backplane connector. The timing of all signals is relative to the falling (LOW going) edge of ARB_CLOCK*

Signal	Setup Time	Hold Time
All P1 signals		
incoming	2.0 ns	7.0 ns
outgoing	24.0 ns	7.0 ns

Drivers:

With the exception of BUS_ACTIVE, BUS_IOX_ACTIVE, BUS_ACK and BUS_NACK, all P1 bus signals are driven with 74F241's with half the output enables driven from the Q side of a 74F74 or 74F109 flipflop and the other half of the output enables driven from the Q* side of the same flipflop.

BUS_ACTIVE, BUS_IOX_ACTIVE, BUS_ACK and BUS_NACK are driven with discrete open emitter 100 mA drivers.

Receivers:

ARB_CLOCK* is received with an AS1804 and loaded with six AS1804 or equivilant inputs. The fixed number of input loads standardizes the capacitive load on ARB_CLOCK* so as to minimize the module to module received clock skew.

All P1 bus signals are received with 74F374 or 74F534 D-type registers clocked by ARB_CLOCK.

System Bus - P1

Pin	Row A	Row B	Row C
1	COMMON	BUS_DATA_PAR[0]	BUS_DATA[07]
2	COMMON	BUS_DATA[06]	BUS_DATA[05]
3	COMMON	BUS_DATA[04]	BUS_DATA[03]
4	COMMON	BUS_DATA[02]	BUS_DATA[01]
5	COMMON	BUS_DATA[00]	BUS_DATA_PAR[1]
6	COMMON	BUS_DATA[17]	BUS_DATA[16]
7	COMMON	BUS_DATA[15]	BUS_DATA[14]
8	COMMON	BUS_DATA[13]	BUS_DATA[12]
9	COMMON	BUS_DATA[11]	BUS_DATA[10]
10	COMMON	BUS_DATA_PAR[2]	BUS_DATA[27]

11	COMMON	BUS_DATA[26]	BUS_DATA[25]
12	COMMON	BUS_DATA[24]	BUS_DATA[23]
13	COMMON	BUS_DATA[22]	BUS_DATA[21]
14	COMMON	BUS_DATA[20]	BUS_DATA_PAR[3]
15	COMMON	BUS_DATA[37]	BUS_DATA[36]
16	COMMON	BUS_DATA[35]	BUS_DATA[34]
17	COMMON	BUS_DATA[33]	BUS_DATA[32]
18	COMMON	BUS_DATA[31]	BUS_DATA[30]
19	COMMON	BUS_DATA_PAR[4]	BUS_DATA[47]
20	COMMON	BUS_DATA[46]	BUS_DATA[45]
21	COMMON	BUS_DATA[44]	BUS_DATA[43]
22	COMMON	BUS_DATA[42]	BUS_DATA[41]
23	COMMON	BUS_DATA[40]	BUS_DATA_PAR[5]
24	COMMON	BUS_DATA[57]	BUS_DATA[56]
25	COMMON	BUS_DATA[55]	BUS_DATA[54]
26	COMMON	BUS_DATA[53]	BUS_DATA[52]
27	COMMON	BUS_DATA[51]	BUS_DATA[50]
28	COMMON	BUS_DATA_PAR[6]	BUS_DATA[67]
29	COMMON	BUS_DATA[66]	BUS_DATA[65]
30	COMMON	BUS_DATA[64]	BUS_DATA[63]
31	COMMON	BUS_DATA[62]	BUS_DATA[61]
32	COMMON	BUS_DATA[60]	BUS_DATA_PAR[7]
33	COMMON	BUS_DATA[77]	BUS_DATA[76]
34	COMMON	BUS_DATA[75]	BUS_DATA[74]
35	COMMON	BUS_DATA[73]	BUS_DATA[72]
36	COMMON	BUS_DATA[71]	BUS_DATA[70]
37	COMMON	BUS_SRC[3]	BUS_SRC[3]*
38	COMMON	BUS_SRC[2]	BUS_SRC[2]*
39	COMMON	BUS_SRC[1]	BUS_SRC[1]*
40	COMMON	BUS_SRC[0]	BUS_SRC[0]*
41	COMMON	BUS_DEST[3]	BUS_DEST[3]*
42	COMMON	BUS_DEST[2]	BUS_DEST[2]*
43	COMMON	BUS_DEST[1]	BUS_DEST[1]*
44	COMMON	BUS_DEST[0]	BUS_DEST[0]*
45	COMMON	BUS_TYPE_PARITY	BUS_TYPE[5]
46	COMMON	BUS_TYPE[4]	BUS_TYPE[3]
47	COMMON	BUS_TYPE[2]	BUS_TYPE[1]
48	COMMON	BUS_TYPE[0]	BUS_DATAPAR_VLD
49	COMMON	BUS_IOX_ACTIVE	BUS_ACTIVE
50	COMMON	BUS_ACK	BUS_NACK

NOTES:

BUS_DATA[], BUS_DATA_PAR[], BUS_DATAPAR_VLD, BUS_SRC[], BUS_SRC[]*, BUS_DEST[], BUS_DEST[]*, BUS_TYPE[], BUS_TYPE_PARITY and spare are terminated at each with 150 Ohms +/- 2%, 100 mW to +5V and 100 Ohms +/-2%, 100 mW to COMMON (equivilent to 60.0 Ohms in series with 2.00 v, nominal).

BUS_ACTIVE, BUS_IOX_ACTIVE, BUS_ACK and BUS_NACK are terminated at each end with 820 Ohms +/- 5%, 100 mW to +5V and 43 Ohms +/- 5%, 100 mW to COMMON (equivilent to 40.9 Ohms in series with 0.25 v, nominal).

System Bus - P2

Pin	Row A	Row B	Row C
2		+5V	
4	+12V	n/c	+12V
5	ARB_CLOCK*	COMMON	BUS_C5
6	ARB_RDY_DEC*	COMMON	BUS_SLOT[3]
7	ARB_DEST[3]	COMMON	BUS_SLOT[2]
8	ARB_DEST[2]	COMMON	BUS_SLOT[1]
9	ARB_DEST[1]	COMMON	BUS_SLOT[0]
10	ARB_DEST[0]	COMMON	BUS_C10
11	ARB_GRANT*	COMMON	BUS_C11
12	ARB_BURST*	COMMON	BUS_C12
13	ARB_REQUEST*	COMMON	BUS_C13
14	ARB_RESP*	COMMON	BUS_C14
15	ARB_MODIFY*	COMMON	BUS_C15
16	ARB_LOCK*	COMMON	COMMON
17	ARB_READY1*	COMMON	BUS_C17
18	ARB_READY3*	COMMON	COMMON
19	ARB_GRANTERR*	COMMON	BUS_RESET*
20	BUS_A20	COMMON	COMMON
21	BUS_A21	COMMON	BUS_C21
22	BUS_A22	COMMON	BUS_C22
23	BUS_A23	COMMON	BUS_C23
24	BUS_A24	COMMON	BUS_C24
25	+12VAUX	+12VAUX	+12VAUX
26	BUS_A26	COMMON	BUS_C26
27	-12VAUX	-12VAUX	-12VAUX
28	BUS_A28	COMMON	BUS_C28
29	-12V	n/c	-12V
31		+5VAUX	

NOTES:

BUS_RESET*, BUS_AXX and BUS_CXX are terminated at each end with 150 Ohms +/- 5% to +5V and 220 Ohms +/- 5% to COMMON (equivilent to 89.2 Ohms in series with 2.97 V, nominal).

BUS_SLOT[3:0] are not bused. BUS_SLOT[3;0] for each slot are selectively connected to COMMON to encode the slot number in positive logic. The low value is provided by the connection to COMMON on the backplane; the high value is provided by a pullup resistor per signal on each module.