1. CSS Bus

1.1 CSS Bus Overview

The system bus is an N-port time-division multiplexed transmission switch. 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 design value of N is 14 (ports); the design clock rate is 20 MHz.

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 interaction.

1.2 CSS Bus Operation

The CSS bus arbiter controls access to the bus. To transmit on the bus, a module issues a request, a request modifier and a destination port address 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 destination port is required to have to have enough RESPONSE buffer space 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 time slot.

Arbitration priority is determined by the port number of requesting port. Port N has the highest priority, and port 0 the lowest. Computational modules as a group are assigned the lowest priorities. A bus bandwidth spreading scheme insures that all computational modules get about the same amount of access to the bus.

A module on the bus is READY to receive a COMMAND when it has at least one COMMAND buffer free. Each module indicates to the arbiter how many COMMAND buffers it has free. The arbiter maintains a count of free COMMAND buffers for each module and decrements the count for the destination module as permission to send each COMMAND is GRANTED. Each module in turn signals the arbiter to increment its free buffer (READY) count whenever one of the module's COMMAND buffers becomes free.

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 LOCK is asserted by the arbiter when an interlocked sequence is in progress. A interlocked sequence begins when LOCK is not asserted and a port asserting BUS REQUEST and MODIFY to the arbiter is granted the bus. The sequence ends when the port executing the sequence deasserts MODIFY to the arbiter. To minimize performance loss, an interlocked sequence does not lock out commands from ports not asserting MODIFY to the arbiter.

The arbiter grants access to the bus by asserting GRANT to each module that wins an arbitration. GRANT is asserted to a module for one time slot (clock cycle). A module receiving a GRANT may transmit on the bus during the time slot immediately following the time slot in which that 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 time slots by asserting BURST to the arbiter during all but the last time slot. The assertion of BURST during a time slot prevents the arbiter from issuing a GRANT to any module for use of the bus during the next time slot. 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 and optionally, parity errors in the data field. 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 ACK on the bus during the second time slot after the transmission. Transmissions received with one or more detected errors are indicated by asserting NACK on the bus during the second time slot. The sole exception to this are transmissions of type CONTROL WRITE. These transmissions are ACKed or NACKed based on the detection of errors in the source and type fields. Any detected errors in the data filed are ignored.

Only the destination port asserts ACK or 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.

1.3 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 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 implementation is optional. The source of a bus transmission indicates whether data parity bits have been sent.

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 as follows.

1. 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, and must not cross a 4 kilobyte page boundary. 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.

   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.

2. 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.

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

   The CONTROL 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 as follows.

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

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

2. 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.

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

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

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

The format of the transmission TYPE field is as follows.

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

BUSTYPE [5:3] type

0	RESPONSE
1	Not Used (Reserved)
2	Not Used (Reserved)
3	READ
4	Not Used (Reserved)
5	WRITE
6	CONTROL WRITE
7	Not Used (Reserved)

BUSTYPE [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	--	--

BUSTYPE [2:1] control signal

0	Not Used (Reserved)
1	module enable
2	module interface enable
3	Not Used (Reserved)

1.4 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 range of defined offset addresses varies from 256 bytes to 4 gigabytes.

The offset address range 0xFFFF FF00 through 0xFFFF FFFF of all modules contains control and status registers. Some modules also have higher level message buffers in this address range. Some elements of this address range are found in more than one module and have the same address in each module in which they appear. For instance, the ID of the module installed in a bus port can be read at offset 0xFFFF FFFF.

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.

The number of offset address bits decoded depends on the module receiving a command. Memory modules and i/o modules which support 4 GB offset address spaces decode all offset address bits. Computational modules and Service modules which support 256 byte address spaces decode only the low order 8 offset address bits.

Use of the 0xFFFF FF00 through 0xFFFF FFFF is as follows.

0xFFFF FF00-1F	Higher level message buffers (Computational, Service & Test)
0xFFFF FF00-03	Command buffer
0xFFFF FF20-2F	Reserved
0xFFFF FF30-3F	Reserved
0xFFFF FF40-4F	Reserved
0xFFFF FF50-5F	Reserved
0xFFFF FF60-63	Interrupt request (Service & Test)
0xFFFF FF80-BF	Interrupt acknowledge (Service & Test)
0xFFFF FFC4-C7	Interrupt vector (All)
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 (Computational)
0xFFFF FFFF	Module ID (All)

1.5 Interface Specification

1.5.1 Bus Signals The bus signals are as follows.

1. BUS_DEST[3:0]
2. BUS_DEST[3:0]*
3. BUS_SRC[3:0]
4. BUS_SRC[3:0]*
5. BUS_TYPE[5:0]
6. BUS_TYPE_PARITY
7. BUS_DATA[0:77]
8. BUS_DATA_PAR[0:7]
9. BUS_DATAPAR_VLD
10. BUS_ACTIVE
11. BUS_ACK
12. BUS_NACK

The signals from the arbiter or backplane to each port are as follows.

1. ARB_CLOCK*

   Bus Clock.

2. ARB_DCLOCK*

   A delayed version of CLOCK*.

3. ARB_GRANT*

   When asserted, the port may transmit on the bus during the next time slot and must deassert ARB_REQUEST*.

4. ARB_LOCK*

   ARB_LOCK is asserted when an interlocked sequence of operations is in progress. The signal is provided for test purposes and is not required for normal operation of the bus.

5. ARB_GRANTERR*

   When asserted, ARB_GRANTERR* indicates that two or more GRANT's were issued for one bus cycle. ARB_GRANTERR* is asserted during the third cycle after the cycle in which the multiple GRANT fault occurred. The signal is for fault isolation only.

6. ARB_SLOT[3:0]

   Indicates the geographical bus slot number.

The signals from each port to the arbiter are:

1. ARB_RS_READY*

   When asserted, the READY counter is reset to ZERO (NOT READY).

2. ARB_INC_READY*

   When asserted, the READY counter is incremented by one on each falling edge of ARB_CLOCK*. The maximum READY count is 7. Excess assertion of ARB_INC_READY will not cause the count to exceed 7 or roll-over to 0.

3. ARB_CPU*

   When asserted, indicates that the port is occupied by a computational module.

4. ARB_DEST[3:0]

   Slot address of destination.

5. ARB_MODIFY*

   When asserted in conjunction with ARB_REQUEST*, 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*.

6. ARB_RESP*

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

7. ARB_REQUEST* When asserted, indicates that the port wants transmit a COMMAND or RESPONSE on the bus.

8. ARB_BURST*

   When asserted, prevents a GRANT from being issued for the next time slot.

1.5.2 Bus Timing The timing of signals passing between a bus module and the arbiter is specified at the point where the backplane connector and pc board of the module or arbiter join. 2 ns is allowed for a signal to pass through the two backplane connectors and the backplane.

Signals from a bus module to the arbiter must be valid 18 ns after the falling edge of ARB_CLOCK*.

Signals from the arbiter to a bus module must be valid 4 ns before the falling edge of ARB_CLOCK*.

Signals driven by a module onto the bus must be enabled and valid or disabled by 27 ns after the falling edge of ARB_CLOCK*.

1.5.3 Drivers With the exception of BUS_ACTIVE, BUS_ACK and BUS_NACK, all bus signals are driven with 74F244's.

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

1.5.4 Receivers All signals are received with 74F374's.

1.5.5 Bus Pinouts

1.5.5.1 Connector P1 Pinouts

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	RESERVED	BUS_ACTIVE
50	COMMON	BUS_ACK	BUS_NACK

1.5.5.2 P1 Signals - Bus Termination BUS_DATA[], BUS_DATA_PAR[], BUS_DATAPAR_VLD, BUS_SRC[], BUS_SRC[]*, BUS_DEST[], BUS_DEST[]*, BUS_TYPE[], BUS_TYPE_PARITY and RESERVED are terminated at each with 150 Ohms +/- 2%, 100 mW to +5V and 100 Ohms +/-2%, 100 mW to COMMON (equivalent to 60.0 Ohms in series with 2.00 V, nominal).

BUS_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 (equivalent to 40.9 Ohms in series with 0.25 V, nominal).

1.5.5.3 Connector P2 Pinouts

Pin	Row A	Row B	Row C
2		+5V	
4	+12V	N/C	+12V
5	ARB_CLOCK*	COMMON	ARB_DCLOCK*
6	ARB_CPU*	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_INCREADY*	COMMON	BUS_PWRFAILWRN*
18	ARB_RSREADY*	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	

1.5.5.4 P1 Signals - Bus Termination BUS_RESET*, BUS_PWRFAILWARN*, BUS_AXX and BUS_CXX are terminated at each end with 150 Ohms +/- 5% to +5V and 220 Ohms +/- 5% to COMMON (equivalent to 89.2 Ohms in series with 2.97 V, nominal).

BUS_SLOT[3:0] are not bussed. 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 pull-up resistor per signal on each module.

2. I/O Link Bus

2.1 I/O Link Bus Description

The link bus is the point to point communication link between between an I/O module and an I/O adapter. The bus is synchronous with a clock rate half that of the system bus. The design clock rate is 10 MHz. The "data" path is 32 bits wide. Maximum bandwidth is 20 MB per second when writing to the system bus and 35.6 MB per second when reading from the system bus.

The operation of the link bus is somewhat similar to that of the system bus. It supports the same commands, responses and data sizes. Each port can transmit to the other port or itself. Each transmission consists of a source port address, a destination port address, a transmission type and 4 bytes of "data". The transmission type determines what "data" contains.

Each port (I/O module, I/O adapter) tells the other how many buffers it has available for READ or MODIFY commands and how buffers it has available for WRITE commands. When a command is sent from one port to the other, the appropriate available buffer count maintained at the sending port for the receiving port is decremented. When an input buffer in one port becomes available, the other port is told to increment the appropriate free buffer count. These counts provide flow control on the bus. Separate counts for READ or MODIFY and WRITE commands permit optimizing the interleaving of commands for maximum data rate with a minimum number of buffers.

An arbiter located on the I/O module controls access to the link bus. The I/O module has the higher priority for use of the bus. The arbiter grants the I/O module a time slot on the bus when ever the module wants to send a response or send a command to an available command buffer. The arbiter grants the I/O adapter all time slots not granted to the I/O module.

Each transmission is checked for errors. The source field, destination field, type field and each byte of data is protected with a parity check bit. Transmissions that are received with correct parity and valid type field are indicated by asserting ACK on the bus during the second time slot after the transmission. Transmissions that are received with one or more detected errors are indicated by asserting NACK on the bus during the second time slot. Unreceived transmissions are indicated by the lack of either ACK or NACK being asserted. Only the destination port asserts ACK or NACK for a transmission.

When a link bus transmission fails, retry is permitted, but not required. If retry is attempted, the port that issued the COMMAND resulting in the failed transmission restarts the COMMAND.

2.2 IOL Commands

The I/O Link Commands are as follows.

1. READ [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 boundary. The operand of a READ command for more than 4 bytes must be quad word aligned and must not cross a 4 kilobyte page boundary.

2. WRITE [1, 2, 3 or 4 bytes]

The operand of a WRITE command must not cross a long word boundary. The WRITE data must be aligned for a 4 byte memory port.

3. MODIFY [1, 2, 3 or 4 bytes]

   The MODIFY command is the first portion of a read/modify/write operation. It reads the operand and locks the slot to other read/modify/write operations. The operation is completed with a regular WRITE command that writes the operand and unlocks the slot.

   The operand of a MODIFY command must not cross a long word boundary.

2.3 IOL Responses

The I/O Link Responses are as follows.

1. READ DATA

   The response to a READ command is the requested data aligned for an 8 byte wide memory. A request for more than 8 bytes requires more than one RESPONSE.

2. ERROR DATA

   The response to a READ command encountering an uncorrectable read error is the requested data as read or after correction has been attempted and aligned for a 8 byte wide memory.

. ERROR [size = 0]

   A READ command that is somehow recognized as having failed to read anything is acknowledged by returning the ERROR response. The data field of the response is undefined.

2.4 Transmission Format

The format of the TRANSMISSION TYPE is as follows.

[3 bit type field] [3 bit size field]

0	Illegal	0	0 bytes
1	READ	1	1 byte
2	WRITE	2	2 bytes
3	MODIFY	3	3 bytes
4	READ DATA	4	4 bytes
5	ERROR DATA	5	8 bytes
6	ERROR	6	16 bytes
7	Illegal	7	32 bytes

2.5 I/O Link Signals

The I/O Link signals are as follows.

1. Destination Slot Address [CSS bus slot 3:0][I/O bus slot 4:0]

2. Destination Parity

3. Source Slot Address [CSS bus slot 3:0][I/O bus slot 4:0]

4. Source Parity

5. Transmission Type [5:0]

6. Type Parity

7. Transmission Data [0:31]

8. Data Parity [0:3]

9. ACTIVE

10. ACK

11. NACK

12. CSS Bus Clock

13. I/O Link Bus Clock

14. Free Buffer Count Control [6 signals]

15. Reset

16. Module Alive

17. Adapter Alive

18. Adapter Burst

19. Adapter Grant

20. System Power Fail