





#### THE MOTOROLA GATEWAY BOARD

(MCF5202 Microprocessor To MC68EC000 Bus Interface Card)

Jeff Miller

October 15, 1997

#### 1.0 Introduction

The integrated Gateway circuit board will bridge an existing MC68EC000 system to the new ColdFire® MCF5202 VL-RISC microprocessor, to evaluate the possibility of moving toward a higher performance architecture. It can be used to evaluate system enhancements such as on-chip instruction and/or data cache and bursting to external memory. It can also be used to port software code to the ColdFire architecture directly in a customer's system as opposed to the traditional method of porting code to an evaluation platform. This paper describes the use and operation of the Gateway board as well as technical information that can be used as a reference design.

#### 2.0 Gateway Board Overview

#### 2.1 Software Considerations

The principal use of this board is to help port system software code from the M68000 architecture to the Cold-Fire architecture. Users will have to recompile the system software to target the MCF5202 instead of targeting the M68000. Even though the system will see a hardware interface that looks like a MC68EC000, the software must consist of ColdFire instructions for the MCF5202 to work properly. Refer to Section 8, "Porting from M68K Architecture," of the MCF5202 User's Manual for an overview of the issues encountered when upgrading from the M68000 to the ColdFire microprocessor. In addition, you'll have to keep three key things in mind while porting system software code from the MC68EC000 system to the MCF5202 system

- 1. mapping 32-bit MCF5202 addresses to 24-bit 68EC000 addresses
- 2. cache coherency
- 3. RMW cycles



#### 2.1.1 Mapping 32-bit MCF5202 addresses to 24-bit 68EC000 addresses

The Gateway board transfers only the lower 24-bits of the address from the MCF5202 to the MC68EC000. This should make no difference in porting the system software (because a 24-bit addressing scheme can still be used, with the upper 8-bits as a "don't-care") except when the on-chip cache is to be used. The MCF5202 allows specific regions of address space to be assigned access control attributes via the Access Control Registers (ACR0 and ACR1). Also, within the MCF5202's Cache Control Register (CACR), the default cache mode can be set up for regions that are not mapped by the ACRs. Refer to the "Cache" section of the ColdFire MCF5202 User's Manual for more details. The MCF5202 ACRs use address bits 31-24 to determine the region of space to which the corresponding access control attributes are assigned. Because the original M68000 system used only addresses 23-0, this at first glance may seem to cause a problem when considering caching certain areas of memory that are smaller than 16Mbytes. However, virtual-to-physical memory mapping can be used to map unique regions in the 24-bit address space to unique 16Mbyte regions in the 32-bit address space, such that certain areas of the physical memory map can take advantage of the MCF5202 caching schemes. One example of implementing this would be to simply concatenate A[31:24] = \$01 in front of the first 24-bit address region, and control the caching scheme for this region using ACRO. Then concatenate A[31:24] = \$02 in front of the second 24-bit address region, which will have a separate caching scheme, and control the caching scheme for this region with ACR1. Finally, concatenate A[31:24] = \$03 in front of the third 24-bit address region, which could have yet another caching scheme, and control the caching scheme for this region using the default cache mode in the CACR register. This example memory map translation is shown in Table 1.

68000 MEMORY MAP 5202 MEMORY MAP CONTENTS **CACHE CONTROL** A[23:0] A[31:0] \$000000 \$01000000 Instructions ACR0 \$1FFFFF **\$011FFFF** \$200000 \$02200000 Data ACR1 \$3FFFFF \$023FFFFF \$400000 \$03400000 I/O CACR \$FFFFF \$03FFFFF

**Table 1: Example Memory Map Translation** 

For this example, ACR0 can be set up such that everything within the region \$01xxxxxx, which includes \$01000000 - \$011FFFFF containing instructions, can have a specific cache attribute such as copyback. ACR1 can be set up such that everything within the region \$02xxxxxx, which includes \$02200000 - \$023FFFFF containing data, can have another specific cache attribute such as writethrough. The CACR can be set up such that everything not mapped by the ACRs, which includes \$03400000 - \$03FFFFFF containing I/O, can have a third cache attribute such as cache inhibit. Now, when the software code is compiled, the new MCF5202 memory map that is specific to the customer's system must be used when assigning the corresponding instruction, data, and I/O sections.

#### **2.1.2** Cache Coherency

If the MCF5202 has its cache on and in copyback mode, and if there is another bus master in the system that can arbitrate the system bus away from the MCF5202 and modify a shared piece of memory, users should be careful about maintaining cache coherency. Cache coherency is the term used to describe the act of keeping the on-chip cache consistent (or coherent) with external memory, if other masters will be using the same memory. Refer to the "Cache Coherency" section of the *ColdFire MCF5202 User's Manual*. If cache coherency is required, then the simplest way to resolve this problem is to control the shared memory region with one of the ACRs and set this ACR's



cache mode to cache-inhibit. This will require the microprocessor to go to external memory to get accurate data as opposed to having a cache hit within internal memory which could possibly contain stale data.

#### 2.1.3 RMW cycles

If the TAS instruction is used in the original M68000 code for implementing the locked or read-modify-write transfer sequence in hardware, then new code will have to be written that essentially implements the same locked transfer in software. This can be done by raising the interrupt mask to 7 and then executing the read, modify, and write instructions, and then lowering the mask back down to the appropriate level. This will ensure that the sequence of instructions between the raising and lowering of the mask will execute uninterrupted, except for a level 7 interrupt which is nonmaskable.

#### 2.2 Hardware Considerations

The target system must have a female 68-pin PLCC socket such that it could hold a 68EC000 PLCC FN package not a 68EC000 QFP FU package. The Gateway board has a male connector arranged in a PLCC FN fashion that will sit in this socket. The Gateway board can operate in 8- or 16-bit data mode. The board can handle interrupt acknowledge cycles for external vector number acquisition or the AVEC\* signal can be used to allow internal vector generation. One difference between the MCF5202 and the 68EC000 is that DA\*[1:0] is always asserted whether AVEC\* is asserted or not. Also, the interrupt level being acknowledged is driven onto A/D[4:2] by the MCF5202, which has to be routed onto address lines A[3:1] for the 68EC000. See Figure 3 for more details. The board also has control logic to handle bus arbitration for alternate bus masters. If the HALT signal is asserted, the processor will stop bus activity at the completion of the current bus cycle and will place all control signals in the inactive state and place all three-state lines in the high-impedance state.

#### 3.0 Performance

The Gateway board performance will be first discussed generally and then specifically with an industry-standard benchmark. For each bus cycle, there is one extra clock required from the beginning of the ColdFire MCF5202 microprocessor bus cycle to the beginning of the 68EC000 bus cycle. This is due to the multiplexed ATM signal on the ColdFire which is required to create the FC signals on the 68EC000 bus. Also, there are some bus clocks inherent to the ColdFire cycle that occur after the 68EC000 bus cycle is done. This is zero to two extra clocks, depending on the size of the access and whether the access is a read or a write. Therefore, because the fastest possible bus transaction for the 68EC000 is 4 bus clocks, the fastest Gateway board bus transaction can be as few as 5 bus clocks for the first bus access of a longword write, or as many as 7 bus clocks if doing, for example, a single byte read. Table 2 and Table 3,compare all possible combinations of accesses between the MCF5202 and the MC68EC000.



**Table 2: Bus Clock Timing Comparison (16-bit mode)** 

MCF5202 DATA ACCESS	READ/	GATEWAY BOARD	EQUIVALENT MC68EC000 BUS
	WRITE	BUS CLOCKS	CLOCKS TO GET SAME DATA
Byte, Word	Read	7	4
Long		6+7=13	4+4=8
Byte, Word	Write	7	4
Long		5+7=12	4+4=8
Line Fill (4 Longs)	Read	6+6+6+6+6+6+7=49	4+4+4+4+4+4+4=32
Line Fill (4 Longs)	Write	5+5+5+5+5+5+7=42	4+4+4+4+4+4+4=32

**Table 3: Bus Clock Timing Comparison (8-bit mode)** 

MCF5202 DATA ACCESS	READ/ WRITE	GATEWAY BOARD BUS CLOCKS	EQUIVALENT MC68EC000 BUS CLOCKS TO GET SAME DATA
Byte		7	4
Word	Read	6+7=13	4+4=8
Long		6+6+6+7=25	4+4+4+16
Byte		7	4
Word	Write	5+7=12	4+4=8
Long		5+5+5+7=22	4+4+4=16
Line Fill (4 Longs)	Read	6+6+6+6+6+6+6+6+ 6+6+6+6+6+6+7=97	4+4+4+4+4+4+4+ 4+4+4+4+4+4+4+64
Line Fill (4 Longs	Write	5+5+5+5+5+5+5+ 5+5+5+5+5+5+7=82	4+4+4+4+4+4+4+ 4+4+4+4+4+4+4+64

The industry standard Dhrystone 2.1 benchmark was run on the Motorola Gateway board, as well as some other systems, and the results are shown in Table 4. If you notice in Table 4, the Gateway board requires about a 7.5MHz increase in frequency (12.5MHz to 20MHz) to get about the same MIPS performance of the 68EC000 evaluation board. This is attributable to the handshaking required between the MCF5202 and the 68EC000. Notice, however, if the internal cache of the MCF5202 is used, the MIPS performance of the system is increased dramatically—more than 8 times better than with cache off. In addition, if system bus interface changes are made to take advantage of the MCF5202 bus interface, such as widening the data bus and allowing bursting (which will be discussed later), even greater system performance will result.



**Table 4: Dhrystone 2.1 Benchmark Performance** 

SYSTEM	DATA WIDTH	FREQUENCY	DRAM ACCESSES (TO GET 16 BYTES)	CACHE MODE	MIPS (@ GIVEN FREQUENCY)
MC68EC000 Board	16 bit	12.5 MHz	8-8-8-8-8-8-8	N/A	1.01
Gateway Board	8 bit	20 MHz	R: 10-10-10-11-10-10-10-11- 10-10-10-11-10-10-10-11 W: 9- 9- 9-11- 9- 9- 9-11- 9- 9- 9-11- 9- 9- 9-11	Off	0.56
Gateway Board	16 bit	20 MHz	R: 10-11-10-11-10-11-10-11 W: 9-11- 9-11- 9-11- 9-11	Off	1.07
Gateway Board	8 bit	20 MHz	R: 10-10-10-10-10-10-10-10- 10-10-10-10-10-10-10-11 W: 9- 9- 9- 9- 9- 9- 9- 9- 9- 9- 9- 9- 9- 9- 9-11	Copy-Back	5.95
Gateway Board	16 bit	20 MHz	R: 10-10-10-10-10-10-11 W: 9- 9- 9- 9- 9- 9- 9-11	Copy-Back	9.12
MCF5202 Board	32 bit	20 MHz	8-4-4-4	Copy-Back	12.6

#### 4.0 Potential Performance and System Improvements

To fully take advantage of the MCF5202 performance in a target system, the 68EC000 bus could be changed to interface better to the MCF5202 bus. First, the maximum frequency of operation for the Gateway board's MCF5202 is 33MHz, which can be a substantial improvement over the 12.5MHz, 16.7MHz, or even the 20MHz version of the 68EC000. So, if the 68EC000 system was designed to operate at higher frequencies, this would be an easy way to increase overall system performance. Second, the 16-bit 68EC000 data bus could be widened to 32-bits so that the MCF5202 can get a longword in one bus transaction instead of the two bus transactions that are required now through the Gateway board. Three, when the MCF5202 does a burst access (gives one address, expects 4 longwords of data), if the 68EC000 system could be changed to provide the secondary 3 longwords faster than the full bus transaction required by the current 68EC000 system, the overall MCF5202 performance can be improved dramatically. For example, if the data bus was widened to 32-bits and page mode DRAM was used in the system, the MCF5202 could potentially do a cache line fill (4 longwords) in 7 bus clocks (4-1-1-1) instead of 49 bus clocks (6-6-6-6-6-6-6-7).

The MCF5202 was chosen for the Gateway board because of its on-chip 2KB unified cache that allows customers to experiment among various on-chip memory configurations. For example, the 2KB unified cache can be configured to be 2KB of I-cache only, 2KB of D-cache only, 1KB of I-cache and 1KB of D-cache, or as a normal 2KB unified cache with a dynamic mixture of both instructions and data. Other ColdFire microprocessors can be selected according to specific system requirements. For example, the MCF5204, which would not require latches and buffers because it has a demultiplexed address and data bus (just like the 68EC000) has a little less on-chip memory (512 byte I-cache and 512 byte SRAM) compared to the MCF5202. Therefore, using the MCF5204 would most likely give a little less performance, but would save overall system cost.

#### **5.0 Debug Support**

There is a ColdFire BDM connector (labeled J2) on the Gateway board that is a 26-pin Berg Connector arranged in two rows of thirteen pins each. This connector is commonly used by software debugger vendors to allow such features as real-time trace, real-time debug, and background debug.



#### **6.0 Bus Operation**

The Gateway board supports a synchronous interface between the MCF5202 bus and the MC68EC000 bus. The waveforms in this document are meant to provide a functional description of the bus cycles required for data transfer operations. The examples below show a longword read and write to a 16-bit wide data bus of the MC68EC000 as well as an Interrupt Acknowledge Cycle. Note that at all times the MCF5202 will not burst (TBI\*=0) and that the address phase lasts for only one clock (AA\*=0).

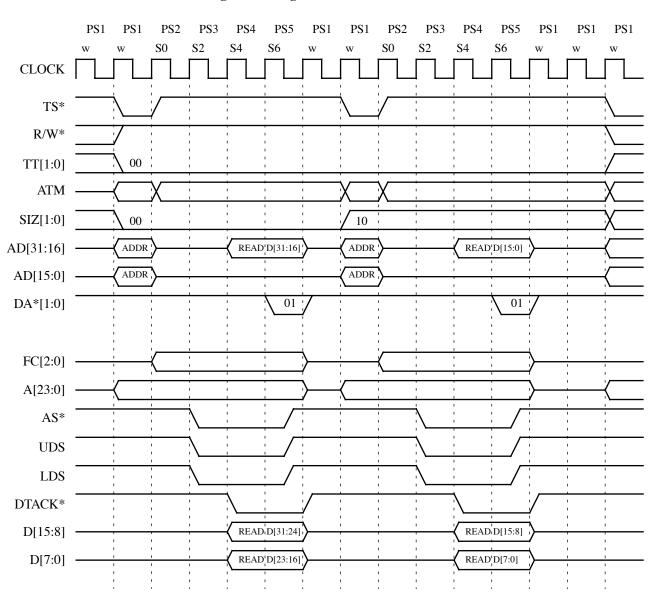


Figure 1: Longword Read To A 16-Bit Port



Figure 2: Longword Write To A 16-Bit Port

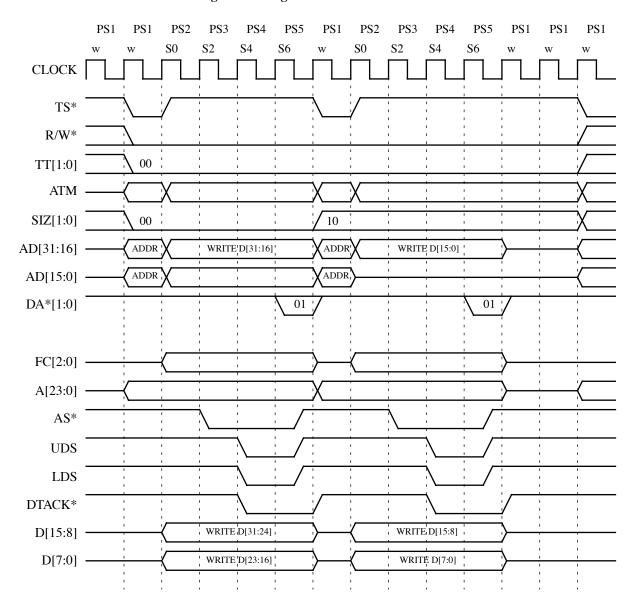
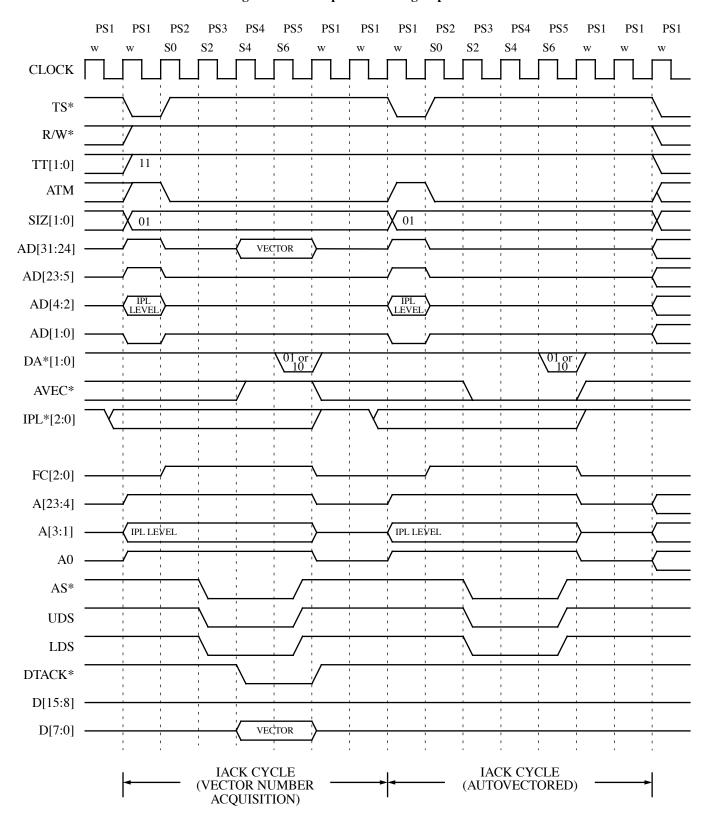




Figure 3: Interrupt-Acknowledge Operation

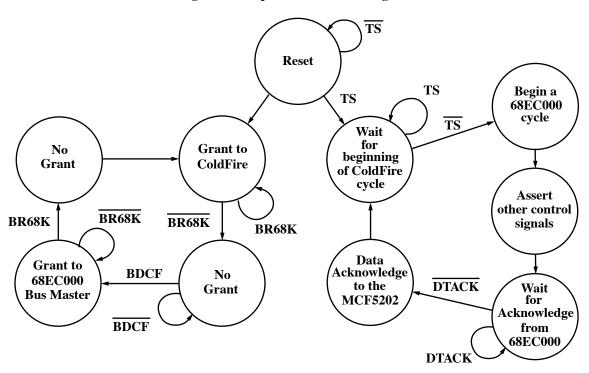


# NP

## Freescale Semiconductor, Inc.

#### 7.0 PLD State Diagram

Figure 4: Simplified PLD State Diagram



#### 8.0 PLD ABEL Code

```
MODULE gateway
TITLE
         'The controlling signals between a 5202 and a 68EC000'
gateway device 'ispLSI';
pLSI property 'PART ispLSI1016-80LT44';
pLSI property 'IGNORE FIXED PIN OFF';
pLSI property 'PULLUP ON';
pLSI property 'Y1_AS_RESET ON';
pLSI property 'LOCK AVEC
                                 1';
pLSI property 'LOCK HALT
                                 2';
pLSI property 'LOCK PCLK
                                 5';
"pLSI property 'LOCK SDI
                                 8';
pLSI property 'LOCK TT1
                                 9';
pLSI property 'LOCK TT0
                                10';
pLSI property 'LOCK ATM
                                11';
pLSI property 'LOCK BR68K
                                15';
"pLSI property 'LOCK SDO
                                18';
pLSI property 'LOCK SIZ1
                                19';
pLSI property 'LOCK BDCF
                                21';
"pLSI property 'LOCK SCLK
                                27';
"pLSI property 'LOCK RSTI
                                 29';
```



```
"pLSI property 'LOCK ISPMODE
                               30';
pLSI property 'LOCK AD0
                               38';
pLSI property 'LOCK MODE
                               40';
pLSI property 'LOCK SIZO
                               41';
pLSI property 'LOCK RnW
                              42';
pLSI property 'LOCK DTACK
                              43';
pLSI property 'LOCK TS
                               44';
pLSI property 'LOCK AENORM
                              3';
pLSI property 'LOCK AEIACK
                              4';
pLSI property 'LOCK FC2
                               12';
pLSI property 'LOCK FC1
                              13';
pLSI property 'LOCK FC0
                              14';
pLSI property 'LOCK BG68K
                             16';
pLSI property 'LOCK BGCF
                              20';
pLSI property 'LOCK LDAT
                              22';
pLSI property 'LOCK OEBA8
                              23';
pLSI property 'LOCK OEAB8
                              24';
pLSI property 'LOCK OEBA16
                              25';
pLSI property 'LOCK OEAB16
                              26';
pLSI property 'LOCK UDS
                               31';
pLSI property 'LOCK LDS
                               32';
pLSI property 'LOCK ADLT
                               33';
                               34';
pLSI property 'LOCK AS
pLSI property 'LOCK DA1
                              35';
pLSI property 'LOCK DA0
                               36';
pLSI property 'LOCK AEUP
                               37';
DECLARATIONS
"Inputs - All Positive Logic
                                   "nAVEC
!AVEC
         pin 1 istype 'input';
 !HALT
          pin 2 istype 'input';
                                   "nHALT
 PCLK
         pin 5 istype 'input';
                                   "CLK from motherboard to uP
         pin 8 istype 'input';
                                   SDI - only used for in-circuit programming of PLD
 TT1
         pin 9 istype 'input';
TTO pin 10 istype 'input';
ATM pin 11 ister
!BR68K pin 15 istype 'input';
                                   "nBR68K
" pin 18 istype 'input';
                                   SDO - only used for in-circuit programming of PLD
         pin 19 istype 'input';
 SIZ1
 !BDCF pin 21 istype 'input';
                                   "nBDCF
          pin 27 istype 'input';
                                   SCLK - only used for in-circuit programming of PLD
  !RSTI
          pin
                 istype 'input';
                                   "pin 29 - nRSTI - RESET pin
          pin 30 istype 'input';
                                   ISPMODE - only used for in-circuit programming of PLD
 AD0
          pin 38 istype 'input';
                                   "AD0 (unlatched)
                                   "Dedicated IN3 - 0=8-bit, 1=16-bit
 MODE
          pin 40 istype 'input';
 SIZ0
          pin 41 istype 'input';
 RnW
          pin 42 istype 'input';
  !DTACK pin 43 istype 'input';
                                   "nDTACK
  !TS
          pin 44 istype 'input';
                                   "nTS
"Outputs - All Positive Logic
  !AENORM pin 3 istype 'output';
                                   "Addr Enable for NORM Op - AENORM=0=HIZ, AENORM=1=output
  !AEIACK pin 4 istype 'output';
                                   "Addr Enable for IACK Op - AEIACK=0=HIZ, AEIACK=1=output
 FC2
          pin 12 istype 'output';
  FC1
          pin 13 istype 'output';
         pin 14 istype 'output';
```



```
!BG68K pin 16 istype 'output';
                                  "nBG68K
                                  "nBGCF
  !BGCF pin 20 istype 'output';
 LDAT pin 22 istype 'output'; "(!nLE16 8) - 0=transparent latches, L-2-H=latches data
 !OEBA8 pin 23 istype 'output'; "nOEBA8 =0=HIZ, 1=output from B (TDAT) to A (AD) enabled
  !OEAB8 pin 24 istype 'output'; "nOEAB8 =0=HIZ, 1=output from A (AD) to B (TDAT) enabled
  !OEBA16 pin 25 istype 'output';
                                  "nOEBA16=0=HIZ, 1=output from B (TDAT) to A (AD) enabled
                                  "nOEAB16=0=HIZ, 1=output from A (AD) to B (TDAT) enabled
 !OEAB16 pin 26 istype 'output';
          pin 31 istype 'output';
                                  "nUDS
 !UDS
  !LDS
          pin 32 istype 'output';
                                  "nLDS
 ! ADT<sub>'</sub>T
          pin 33 istype 'output';
                                  "Addr Latch - 0=transparent latches, L-2-H=latches data
 !AS
          pin 34 istype 'output';
 !DA1
          pin 35 istype 'output';
                                  "nDA1
         pin 36 istype 'output'; "nDA0
 !DA0
 !AEUP pin 37 istype 'output'; "Addr Enable for A[23:8] - AEUP=0=HIZ, AEUP=1=output
"Internal Nodes
      PQ0,PQ1,PQ2 node istype 'reg, buffer';
                    node istype 'req,buffer';
      ATMA
                    node istype 'reg,buffer';
      NO1
                    node istype 'reg';
      NO2
                    node istype 'reg';
      NCLK
                    node;
                    node istype 'reg,buffer';
      BQ0,BQ1
"Constants
      c,k,x,z = .C.,.K.,.X.,.Z.; "this is used for test vectors
"State Value Constants
      psreg = [PQ2, PQ1, PQ0];
                                 "Positive Clk State Register
             = [0,0,0];
                                  "!PQ2&!PQ1&!PQ0
      PS1
             = [0,0,1];
                                  "!PO2&!PO1&PO0
      PS2
             = [0,1,1];
                                  "!PQ2&PQ1&PQ0
           = [0,1,0];
      PS3
                                  "!PQ2&PQ1&!PQ0
      PS4
             = [1,1,0];
                                  "PQ2&PQ1&!PQ0
      PS5
           = [1,0,0];
                                  "PQ2&!PQ1&!PQ0
      PSTATE0
                   = !PO2&!PO1&!PO0;
      PSTATE1
                   = !PQ2&!PQ1&PQ0;
      PSTATE2
                   = !PQ2&PQ1&PQ0;
      PSTATE3
                   = !PQ2&PQ1&!PQ0;
      PSTATE4
                    = PQ2&PQ1&!PQ0;
      PSTATE5
                   = PQ2&!PQ1&!PQ0;
      bsreg = [BQ1,BQ0];
                                 "BusArb State Register
      BS0 = [0,0];
      BS1 = [0,1];
      BS2 = [1,1];
             = [1,0];
Equations
"Initializations
      psreg.clk = PCLK;
      psreq.ar = RSTI;
      A0.clk = TS;
                          "ADO is latched when TS is asserted
```



```
ATMA.clk = TS;
                            "ATM is latched when TS is asserted
       NQ1.ar = RSTI;
       NQ2.ar = RSTI;
       NO1.clk = PCLK;
                             "Clock NegClk machine 1 with pos clk
       NQ2.clk= !PCLK;
                             "Clock NegClk machine 2 with the inverted pos clk
      bsreg.clk = PCLK;
       bsreq.ar = RSTI;
"Output enables
       As.oe = !BG68K;
                            "enable when the 68K is not granted the bus
       UDS.oe = !BG68K;
                            "enable when the 68K is not granted the bus
                            "enable when the 68K is not granted the bus
      LDS.oe = !BG68K;
       FC0.oe = !BG68K;
                            "enable when the 68K is not granted the bus
       FC1.oe = !BG68K;
                            "enable when the 68K is not granted the bus
       FC2.oe = !BG68K;
                            "enable when the 68K is not granted the bus
"Sequential Logic
      A0 := AD0;
      ATMA := ATM;
      NQ1 := !NQ1;
      NQ2 := NQ1;
"Combinational Logic - (See NOTE 2)
   NCLK = NQ1 ! $ NQ2;
                        "XNOR the outputs of the two NegClk state machines to produce NCLK
          " AS is asserted for PS3, PS4, and the posclk of PS5
   AS = PSTATE3 # PSTATE4 # PSTATE5&!NCLK;
          " OEBA16 = (CF is master & not halted & during AS)&(16-bit read & !IACK)
   OEBA16 = (!BG68K & !HALT & AS) & ( RnW&MODE & !(TT1 & TT0) );
          " OEBA8 = (CF is master & not halted & during AS)&(8-bit read # IACK)
   OEBA8 = (!BG68K & !HALT & AS) & ( RnW&!MODE # TT1&TT0 );
          " OEAB16 = (CF is master & not halted & during AS)&(16-bit write)
   OEAB16 = (!BG68K & !HALT & AS) & ( !RnW&MODE );
          " OEAB8 = (CF is master & not halted & during AS)&(8-bit write
   OEAB8 = (!BG68K & !HALT & AS) & ( !RnW&!MODE );
          " UDS = (Read&PS3 # PS4 # PCLK&PS5) & (16-bit) & !( Odd & Byte )
   UDS = (RnW&PSTATE3 # PSTATE4 # PSTATE5&!NCLK) & MODE & !( A0 & !SIZ1&SIZ0 );
          " LDS = (Read&PS3 # PS4 # PCLK&PS5) & !(16-bit & Even & Byte & !IACK)
   LDS = (RnW&PSTATE3 # PSTATE4 # PSTATE5&!NCLK) & !( MODE & !A0 & !SIZ1&SIZ0 & !(TT1&TT0) );
   DA1 = PSTATE5 & MODE;
                                           "PS5 & 16-bit
   DA0 = PSTATE5 & !MODE;
                                           "PS5 & !16-bit
   LDAT = PSTATE5 & NCLK;
                                           "PS5 & NCLK
          "ADLT = (TS&PS1 # PS2 # PS3 # PS4 # PCLK&PS5)
   ADLT = (TS&PSTATE1 # PSTATE2 # PSTATE3 # PSTATE4 # PSTATE5&!NCLK);
   AENORM = (!BG68K \& !HALT) \& !(TT1&TT0);
                                              "(CF is master & not halted) & !(IACK-Access)
   AEIACK = (!BG68K \& !HALT) & (TT1&TT0);
                                               "(CF is master & not halted) & (IACK-Access)
                                               "(CF is master & not halted)
   AEUP = (!BG68K & !HALT);
```



```
"Function Codes for EC000 - (See NOTE 1)
           " FC2 = ( (ATM & Normal-Access) # (IACK-Access) )
   FC2 = (ATM # (TT1&TT0));
           " FC1 = ( (ATMA & Normal-Access) # (IACK-Access) )
   FC1 = (ATMA # (TT1&TT0));
           " FC0 = ( (!ATMA & Normal-Access) # (IACK-Access) )
   FC0 = (!ATMA # (TT1&TT0));
STATE_DIAGRAM psreg;
                                    "RESET and waiting for TS to de-assert
STATE PS0:
      IF TS THEN PS0;
                                    "Wait for TS to de-assert
      ELSE PS1;
STATE PS1:
                                    "Waiting for TS to assert, Beginning of ColdFire cycle
       IF !TS THEN PS1;
                                    "Waiting for TS to assert
       ELSE PS2;
STATE PS2:
                                    "Beginning of 68K cycle, assert FC's and Address
      IF HALT THEN PS2;
                                    "If HALT is asserted then stay in state 2
      ELSE PS3;
                                    "else goto state 3
STATE PS3:
                                    "Assert other control signals
       GOTO PS4;
                                    "Unconditionally goto state 4
STATE PS4:
                                    "Waiting for DTACK from 68K
       IF (TT1 & TT0 & AVEC) THEN
                                    "if TT[1:0]=11 (IACK and AVEC) then
              PS5;
                                    "goto state 5 (just DA the cycle)
       ELSE
         IF (DTACK) THEN
                                    "else if (Normal or IACK without AVEC), look for DTACK
                                    "goto state 5
          ELSE
              PS4;
                                    "else stay in state 4
STATE PS5:
                                    "Data acknowledge to ColdFire
       GOTO PS1;
                                    "Unconditionally goto state 1
STATE DIAGRAM bsreq;
STATE BS0:
                                    "Give the bus to CF, and wait for Request
       BGCF=1;
                                    "Assert Grant to CF
       BG68K=0;
                                   "Do not assert Grant to 68K
       IF BR68K THEN
                                    "If there is a Bus Request,
                                    "goto state 1
              BS1;
       ELSE
                                    "else if no request,
              BS0;
                                    "stay in state 0
```



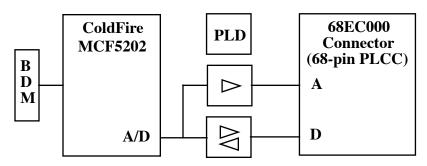
```
STATE BS1:
                                     "Got a Request, wait for CF to quit driving the bus
       BGCF=0; BG68K=0;
                                     "Do not assert either Grant
       IF !BDCF THEN
                                    "If CF is not driving the bus,
              BS2;
                                     "then goto state 2
       ELSE
                                     "else if CF is driving the bus,
              BS1:
                                     "stay in state 1
STATE BS2:
                                     "Done driving the bus, give the bus to 68K, wait for
Request to go away
       BGCF=0;
                                     "Do not assert Grant to CF
                                     "Assert Grant to 68K
       BG68K=1;
       IF BR68K THEN
                                     "If 68K is still requesting the bus,
                                     "then stay in state 2
              BS2;
       ELSE
                                     "else if no longer requesting the bus,
              BS3;
                                     "goto state 3
STATE BS3:
                                     "Request went away, delay one clock, then bus back to CF
       BGCF=0; BG68K=0;
                                     "Do not assert either Grant
       GOTO BS0;
                                     "goto state 0
"NOTE 1:
       ATMa
              ATMd
                     TT1
                             TT0
                                    FC2
                                           FC1
                                                   FC0
                                                          Notes
                      0
                             0
                                     0
                                            0
                                                           Normal User Data
       1
                      0
                             0
                                    0
                                            1
                                                           Normal User Instruction
                                                           Normal Supervisor Data
              1
                                    1
                                            0
                                                   1
       1
                      0
                             0
                                    1
                                                   0
                                                           Normal Supervisor Instruction
              1
                                            1
       Х
              X
                      0
                             1
                                    0
                                            0
                                                   0
                                                          Reserved
       Х
              Х
                      1
                             0
                                    ?
                                            ?
                                                   ?
                                                           Emulator Access
       Х
              Х
                      1
                             1
                                    1
                                            1
                                                   1
                                                           CPU Space or IACK
"NOTE 2:
"RnW
      MODE A0 (!SIZ1& AENORM
                                  AEIACK
                                           UDS LDS
                                                       OExxxx
                                                                Notes
                  SIZ0)
" 1
       1
                             1
                                                       OEBA16
                                                                Read, 16-bit, even, byte, Normal
" 1
       1
                    1
                             0
                                      1
                                             1
                                                       OEBA8
                                                                Read, 16-bit, even, byte, IACK
" 1
       1
            0
                    0
                             х
                                      x
                                             1
                                                       OEBA16
                                                                Read, 16-bit, even, !byte
" 1
       1
            1
                   1
                                             0
                                                       OEBA16
                                                                Read, 16-bit, odd, byte
                             х
                                      х
" 1
       1
            1
                   0
                             х
                                      х
                                             1
                                                1
                                                       OEBA16
                                                                Read, 16-bit, odd, !byte (N/A)
" 1
                                                1
       0
            х
                   х
                             х
                                      х
                                             0
                                                       OEBA8
                                                                Read, 8-bit
" 0
       1
            0
                   1
                             х
                                      х
                                             1
                                                0
                                                       OEAB16
                                                                Write, 16-bit, even, byte
" 0
            0
                   0
       1
                             х
                                      х
                                             1 1
                                                       OEAB16
                                                                Write, 16-bit, even, !byte
" 0
       1
            1
                   1
                             х
                                      x
                                             0 1
                                                       OEAB16 Write, 16-bit, odd, byte
                                             1 1
" 0
       1
            1
                   0
                             х
                                                       OEAB16
                                                                Write, 16-bit, odd, !byte (N/A)
                                      х
" 0
                             Х
                                                       OEAB8
                                                                Write,8-bit
```

END



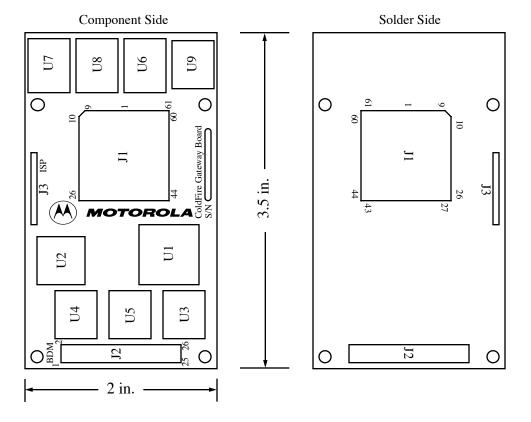
## 9.0 Block Diagram

Figure 5: Gateway Board Block Diagram



## 10.0 Gateway Board Physical Layout

Figure 6: Physical Layout (Actual Size)





## 11.0 Gateway Board Bill Of Material

**Table 5: Bill Of Material** 

ITEM	QTY	MANUFACTURER	PART NO.	REF. DES.	DESCRIPTION
1	1	Motorola	XCF5202PU33A	U1	IC, MCF5202, 33 MHz, 100pin, TQFP
2	1	Lattice	ISPLSI1016-90LT44	U2	IC, PLD, 44 pins, TQFP
3	4	Motorola	MC74F573DW	U3-U5, U9	IC, 74F573, 20 pins, SOL20
4	3	Motorola	MC74F543DW	U6-U8	IC, 74F543, 24 pins, SOL24
5	4	Venkel	CR1206-8W-103JT	R1-R4	Res, 10K, 5%, 1/8W, 1206
6	4	Venkel	CR1206-8W-472JT	R5-R8	Res, 4.7K, 5%, 1/8W, 1206
7	4	Samtec	TMS-117-55-G-S	J1	Conn, HDR, 17 pins, 50Mil ctr, single row, 1X17
8	1	AMP	1-103783-3	J2	Conn, HDR, 26 pins, 100Mil ctr, dual row, 2X13
9	1	AMP	1-87499-3	J3	Conn, HDR, 8 pins, 100Mil ctr, single row, 1X8
10	1	Samwa Venkel	CS3216X7R103K500R C1206X7R500-103KNE	C1	Cap, 0.01UF, 10%, 50V, 1206
11	1	Panasonic	S1012-36-ND	C2	Cap, 33UF, 10%, 16V, 1206, TANT
12	22	Samwa Venkel	CS3216X7R104K500R C1206X7R500-104KNE	C3-C24	Cap,0.1UF, 10%, 50V, 1206
13	1	Samwa Venkel	CS3216COG100K500R C1206C0G500-100JNE	C25	Cap, 10PF, 10%, 50V, 1206

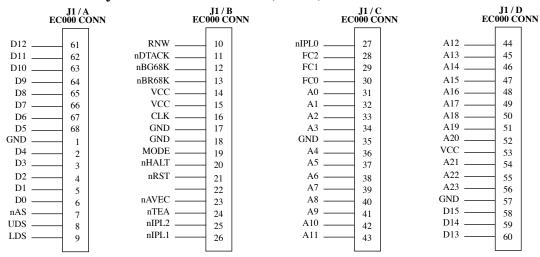


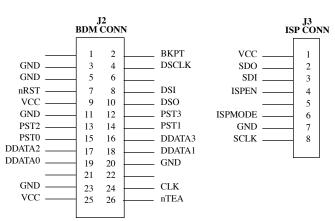
#### 12.0 ColdFire Gateway Board Schematics (1 of 2)

	U1	V	U9	
	MCF5202		74F573	0.01UF
CLK87	CLK	R/W 39 PNW	$ \begin{array}{c c} AEUPPER & 1 & OE \\ ADLT & 11 & C \end{array} $	ISPEN —   — GND
nRST 67	RST	R/W 39 RNW TT0 35 TT0		10 PF
mx51	KOI	TT1 36 TT1	AD16 2 D0 Q0 19 A16 AD17 3 D1 O1 18 A17	VCC —   — GND
74	TCK	SIZ0 41 SIZ0 SIZ1 42 SIZ1	AD17 3 D1 Q1 18 A17 AD18 4 D2 Q2 17 A18	33 UF
BKPT	TMS/BKPT	$ \begin{array}{c cccc} SIZ1 & 42 & SIZ1 \\ ATM & 63 & ATM \end{array} $	AD19 $\frac{5}{15}$ D3 Q3 $\frac{16}{15}$ A19	VCC — GND
DSI <u>72</u> DSO <u>71</u>	TDI/DSI TDO/DSO		AD20 6 D4 Q4 15 A20 AD21 7 D5 Q5 14 A21	0.1 UF
DSCLK 70	TRST/DSCLK	TS 40 nTS	AD22 8 D6 Q6 13 A22	VCC — GND
DDATA0 81 82 82	DDATA0 DDATA1	AA = 48 = GND	AD23 9 D7 Q7 12 A23	0.1 UF
DDATA2 83	DDATA2	$ \begin{array}{c c} DTIP & 47 \\ DA0 & 43 \\ \end{array} $ nDA0		VCC — GND
DDATA3 84	DDATA3	DA1 44 nDA1	U3	0.1 UF
PST0 <u>56</u> PST1 <u>57</u>	PST0 PST1	TEA 55 nTEA GND	74F573	VCC —   — GND
PST2 58	PST2		$ \begin{array}{c c} AEUPPER & \underline{1} & OE \\ ADLT & \underline{11} & C \end{array} $	0.1 UF
PST359	PST3	BR 50 BD 49 nBDCF	2 2 19 10	VCC   GND
GND85	JCE	BD 49 nBDCF BG 51 nBGCF	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	0.1 UF
VCC	MTMOD0	(4 777.0	AD10 4 D2 Q2 17 A10	0.1 UF
$\begin{array}{c} \text{GND} & \phantom{00000000000000000000000000000000000$	MTMOD1 MTMOD2	IPL0 64 nIPL0 IPL1 65 nIPL1	AD11 5 D3 Q3 16 A11 AD12 6 D4 Q4 15 A12	VCC   GND
VCC	HIZ	IPL2 66 nIPL2	AD12 7 D5 Q5 14 A13	0.1 UF 5
AD089	A/D0	AVEC 62 nAVEC	$AD14 = \frac{8}{9} D6 Q6 = \frac{13}{12} A14$	
AD190	A/D1	6 000	AD15 9 D7 Q7 12 A15	0.1 UF Q
AD2 91 AD3 92	A/D2 A/D3	$ \begin{array}{c c} GND & \underline{} & GND \\ GND & \underline{} & GND \end{array} $		VCC     GND   S
AD3 AD495	A/D4	GND $\frac{18}{24}$ GND	U5 74F573	0.1 UF SE
AD5 <u>96</u> AD6 97	A/D5 A/D6	$ \begin{array}{c c} GND & \underline{24} & GND \\ GND & \underline{31} & GND \end{array} $	,	VCC   GND   GND   GND
$\begin{array}{c} AD6 & \underline{97} \\ AD7 & \underline{98} \end{array}$	A/D0 A/D7	$\frac{1}{1}$ GND $\frac{1}{1}$ GND	$ \begin{array}{c c} AEIACK & \underline{1} & OE \\ ADLT & \underline{11} & C \end{array} $	VCC   GND
AD8 1	A/D8	$ \begin{array}{c c} \text{GND} & \underline{45} & \text{GND} \\ \hline 52} & \text{GND} \end{array} $	AD5 2 D0 Q0 19 A0	0.1 UF
AD9 2 AD10 3	A/D9 A/D10	$ \begin{array}{c c} GND & \phantom{00000000000000000000000000000000000$	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	VCC     GND
AD114	A/D11	GND $\frac{68}{70}$ GND	$AD3 = \frac{4}{5}$ $D2 = Q2 = \frac{17}{16}$ $A2 = \frac{17}{16}$ $A3 = \frac{17}{16}$	0.1 UF
AD125 AD138	A/D12 A/D13	$ \begin{array}{c c} \text{GND} & -79 & \text{GND} \\ \hline 86 & \text{GND} \end{array} $	AD4 5 D3 Q3 16 A3 AD5 6 D4 Q4 15 A4	VCC   GND
AD14 9	A/D14	$ \begin{array}{c c} GND & 93 & GND \\ \hline GND & 99 & GND \end{array} $	AD5 $\frac{7}{9}$ D5 Q5 $\frac{14}{12}$ A5	0.1 UF
AD15 10 AD16 11	A/D15 A/D16	GND 99 GND	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	VCC — GND
AD1611 AD1714	A/D17	VCC 7 VCC	1.E. 7.	0.1 UF
AD18 15	A/D18 A/D19	VCC 13 VCC	114	0.1 UF
AD1916 AD2017	A/D19 A/D20	$\begin{array}{c cccc} VCC & \underline{19} & VCC \\ VCC & \underline{25} & VCC \end{array}$	U4 74F573	VCC   GND
AD2120	A/D21	VCC 32 VCC	AENORM 1 OE	
AD22 21 AD23 22	A/D22 A/D23	$\begin{array}{c cccc} VCC & 38 & VCC \\ VCC & 46 & VCC \\ \end{array}$	ADLT 11 C	0.1 UF
AD24 23	A/D24	VCC $53$ $VCC$	AD0 2 D0 Q0 19 A0	VCC — GND
AD25 <u>26</u> AD26 <u>27</u>	A/D25	$VCC = \frac{60}{60} VCC$	AD1 3 D1 Q1 18 A1	0.1 UF
AD27 28	A/D26 A/D27	$\begin{array}{c cccc} VCC & 69 & VCC \\ \hline VCC & 80 & VCC \\ \end{array}$	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	VCC   GND S
AD28	A/D28	VCC 88 VCC	AD4 6 D4 Q4 15 A4	$\begin{array}{c} 0.1 \text{ UF} \\ VCC \longrightarrow \begin{array}{c} -1 \text{ GND} \end{array} \begin{array}{c} -2  \\ -1  \end{array}$
AD29 30 AD30 33	A/D29 A/D30	$\begin{array}{c cccc} VCC & 94 & VCC \\ VCC & 100 & VCC \end{array}$	AD5 7 D5 Q5 14 A5 A6 D6 Q6 13 A6	0.1 UF 2
AD31 34	A/D31	VCC 100 VCC	$ \begin{array}{c cccc} AD6 & & \circ & D6 & Q6 & 13 & A6 \\ AD7 & & 9 & D7 & Q7 & 12 & A7 \end{array} $	VCC → ⊢ GND
				0.1 UF S
4	7K	4.777	4.7K	VCC     GND S
	₩— vcc	4.7K nBGCF ——\\\\\\	4 4 4 4	0.1 UF
10	K	4.7K	4.7K	VCC — GND M
nTEA\(\square\)	W∕ vcc K	nIPL0 ——VWV—— 10K	- VCC nIPL2 —— VCC 10K	0.1 UF VCC— — GND
	₩— vcc		- VCC ISPEN —— VCC	



#### **ColdFire Gateway Board Schematics (2 of 2)**





		74F	543		
nOEAB16 GND nLE16_8	13 11 14	OEAB EAB LEAB	OEBA EBA LEBA	2 23 1	nOEBA10 GND nLE16_8
AD16 AD17 AD18 AD19 AD20 AD21 AD22 AD23	3 4 5 6 7 8 9	A0 A1 A2 A3 A4 A5 A6 A7	B0 B1 B2 B3 B4 B5 B6	22 21 20 19 18 17 16 15	D0 D1 D2 D3 D4 D5 D6
AD23		A	В/		DΙ

**U6** 

U2 ISPLSI-1016						
nAVEC 1 nHALT 2 AENORM 3 AEIACK 4 CLK 5 VCC 6 ISPEN 7 SDI 8 TT1 9 TT0 10 ATM 11 FC2 12 FC1 13 FC0 14 nBR68K 15 nBG68K 16 GND 17 SDO 18 SIZ1 19 nBGCF 20 nBDCF 21	I/O 28 I/O 29 I/O 30 I/O 31 Y0 VCC ISPEN/NC SDI/IN 0 I/O 1 I/O 2 I/O 3 I/O 4 I/O 5 I/O 6 I/O 7 GND SDO/IN 1 I/O 8 I/O 9 I/O 10	I/O 27 I/O 26 I/O 25 I/O 24 IN 3 GND I/O 23 I/O 22 I/O 21 I/O 20 I/O 19 I/O 18 I/O 17 I/O 16 IN 2/MODE Y1/RESET VCC Y2/SCLK I/O 15 I/O 14 I/O 13	44 nTS 43 nDTACK 42 RNW 41 SIZ0 40 MODE 39 GND 38 AD0 37 AEUPPER nDA0 35 nDA1 34 nAS 33 ADLT 32 LDS 31 UDS 30 ISPMODE 29 nRST 28 VCC 27 SCLK 26 nOEAB16 25 nOEBA16 24 nOEAB8			
nLE16_822	I/O 11	I/O 12	23 nOEBA8			

OEAB	OEBA	2	nOEBA16				
EAB	EBA	23	GND				
LEAB	LEBA	1	nLE16_8				
A0	B0	22	D8				
A1	B1	21	D9				
A2	B2	20	D10				
A3	B3	19	D11				
A4	B4	18	D12				
A5	B5	17	D13				
A6	B6	16	D14				
A7	B7	15	D15				
U8							
	EAB LEAB A0 A1 A2 A3 A4 A5 A6 A7	EAB EBA LEBA LEBA LEBA LEBA LEBA LEBA LE	A0 B0 22 A1 B1 21 A2 B2 20 A3 B3 19 A4 B4 18 A5 B5 17 A6 B6 A7 B7				

74F543

74F543							
nOEAB8	OEAB EAB LEAB	OEBA EBA LEBA	2 23 1	nOEBA8 GND nLE16_8			
AD243	A0	В0	22	D0			
AD254	A1	B1	21	D1			
AD265	A2	B2	20	D2			
AD276	A3	В3	19	D3			
AD287	A4	B4	18	D4			
AD298	A5	B5	17	D5			
AD309	A6	В6	16	D6			
AD3110	A7	В7	15	D7			



Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.