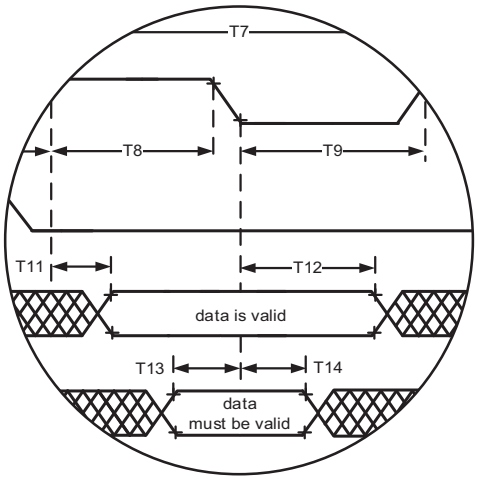
PERFORMANCE MOTION DEVICES

MOTION CONTROL AT ITS CORE



Magellan® Motion Control IC

MC58000 Electrical Specifications

Revision 2.7/ January 2024

Performance Motion Devices, Inc.

80 Central Street, Boxborough, MA 01719

www.pmdcorp.com

NOTICE

This document contains proprietary and confidential information of Performance Motion Devices, Inc., and is protected by federal copyright law. The contents of this document may not be disclosed to third parties, translated, copied, or duplicated in any form, in whole or in part, without the express written permission of PMD.

The information contained in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form, by any means, electronic or mechanical, for any purpose, without the express written permission of PMD.

Copyright 1998–2024 by Performance Motion Devices, Inc.

Juno, Atlas, Magellan, ION, Prodigy, Pro-Motion, C-Motion and VB-Motion are trademarks of Performance Motion Devices, Inc.

Warranty

Performance Motion Devices, Inc. warrants that its products shall substantially comply with the specifications applicable at the time of sale, provided that this warranty does not extend to any use of any Performance Motion Devices, Inc. product in an Unauthorized Application (as defined below). Except as specifically provided in this paragraph, each Performance Motion Devices, Inc. product is provided "as is" and without warranty of any type, including without limitation implied warranties of merchantability and fitness for any particular purpose.

Performance Motion Devices, Inc. reserves the right to modify its products, and to discontinue any product or service, without notice and advises customers to obtain the latest version of relevant information (including without limitation product specifications) before placing orders to verify the performance capabilities of the products being purchased. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement and limitation of liability.

Unauthorized Applications

Performance Motion Devices, Inc. products are not designed, approved or warranted for use in any application where failure of the Performance Motion Devices, Inc. product could result in death, personal injury or significant property or environmental damage (each, an "Unauthorized Application"). By way of example and not limitation, a life support system, an aircraft control system and a motor vehicle control system would all be considered "Unauthorized Applications" and use of a Performance Motion Devices, Inc. product in such a system would not be warranted or approved by Performance Motion Devices, Inc.

By using any Performance Motion Devices, Inc. product in connection with an Unauthorized Application, the customer agrees to defend, indemnify and hold harmless Performance Motion Devices, Inc., its officers, directors, employees and agents, from and against any and all claims, losses, liabilities, damages, costs and expenses, including without limitation reasonable attorneys' fees, (collectively, "Damages") arising out of or relating to such use, including without limitation any Damages arising out of the failure of the Performance Motion Devices, Inc. product to conform to specifications.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent procedural hazards.

Disclaimer

Performance Motion Devices, Inc. assumes no liability for applications assistance or customer product design. Performance Motion Devices, Inc. does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of Performance Motion Devices, Inc. covering or relating to any combination, machine, or process in which such products or services might be or are used. Performance Motion Devices, Inc.'s publication of information regarding any third party's products or services does not constitute Performance Motion Devices, Inc.'s approval, warranty or endorsement thereof.

Patents

Performance Motion Devices, Inc. may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Patents and/or pending patent applications of Performance Motion Devices, Inc. are listed at https://www.pmdcorp.com/company/patents.

Related Documents

Magellan® Motion Control IC User Guide

Complete description of the Magellan Motion Control IC features and functions with detailed theory of operation.

C-Motion Magellan Programming Reference

Descriptions of all Magellan Motion Control IC commands, with coding syntax and examples, listed alphabetically for quick reference.

DK58420 Developer Kit User Manual

How to install and configure the DK58420 and DK55420 developer kits.

Atlas Digital Amplifier User Manual

Description of the Atlas Digital Amplifier electrical and mechanical specifications along with a summary of its operational features.

Atlas Digital Amplifier Complete Technical Reference

Complete technical and mechanical description of the Atlas Digital Amplifier with detailed theory of operations.

Table of Contents

1. The MC50000 Family 9
1.1 Magellan Motion Control IC Family Overview 9
1.2 MC58420-Series ICs & Developer Kit 10

2. Functional Characteristics 13
2.1 Configurations, Parameters, and Performance 13
2.2 Physical Characteristics and Mounting Dimensions 15
2.3 Absolute Maximum Environmental and Electrical Ratings 17
2.4 MC58110 System Configuration — Single Chip, 1 Axis Control 17
2.5 MC58x20 System Configuration — Two Chip, 1 to 4 Axis Control 19

3. Electrical Characteristics 21
3.1 DC Characteristics for 58110, 58x20 CP 21
3.2 DC Characteristics for 58x20 IO 22
3.3 AC Characteristics 22

4. I/O Timing Diagrams 25
4.1 Clock 25
4.2 Quadrature Encoder Input 26
4.3 Reset 26
4.4 Host Interface, 8/16 Mode 27
4.5 Host Interface, 16/16 Mode 29
4.6 External Memory Timing 31
4.7 Peripheral Device Timing 33
4.8 SPI DAC Timing 34
4.9 SPI Atlas Timing 35

5. Pinouts and Pin Descriptions 37
5.1 Pinouts for the MC58110 37
5.2 MC58110 Pin Descriptions 38
5.3 Pinouts for the MC58x20 44
5.4 MC58x20 Pin Descriptions 48

6. Application Notes — MC58110 and MC58x20 59
6.1 General Design Notes 59
6.2 Peripheral Device Access 61
6.3 Power Supplies 64
6.4 Clock Generator, Grounding and Decoupling, and Device Reset 67
6.5 Serial Communication Interface (SCI) 72
6.6 CAN Communication Interface 77
6.7 External Memory 79
6.8 Asynchronous SRAM 79
6.9 Using the On-chip ADC 83
6.10 User I/O Space 89
6.11 Parallel Word Position Input 91
6.12 Parallel Communication Interface 93
6.13 Overcurrent and Emergency Braking Circuits for Motor Drivers 97
6.14 DC Brush Motor Control Using SPI Interfaced DACs 99
6.15 Brushless DC Motor Control Using High-Precision Parallel DACs 102

6.16 Single-Axis Magellan with Brushless DC Atlas 106
6.17 Multi-Axis Magellan with DC Brush & Step Motor Atlas 109
6.18 Pulse & Direction Mode Output Connected to Atlas 111
6.19 Using PWM for DC Brush, Brushless DC and Microstepping Motors 113
6.20 Using the Allegro A3977 to Drive Microstepping Motors 120

List of Figures

1-1 DK58420 Board 10
2-1 CP chip (all dimensions in millimeters) 15
2-2 IO Chip (all dimensions in millimeters) 16
2-3 MC58110 control and data paths 18
2-4 MC58x20 control and data paths 19
4-1 Clock timing 25
4-2 Quad encoder timing 26
4-3 Reset timing 26
4-4 Instruction write, 8/16 mode 27
4-5 Data write, 8/16 mode 27
4-6 Data read, 8/16 mode 28
4-7 Status read, 8/16 mode 28
4-8 Instruction write, 16/16 mode 29
4-9 Data write, 16/16 mode 29
4-10 Data read, 16/16 mode 30
4-11 Status read, 16/16 mode 30
4-12 External memory read 31
4-13 External memory write 32
4-14 Peripheral device read 33
4-15 Peripheral device write 34
4-16 SPI DAC timing 34
4-17 SPI Atlas timing 35
5-1 MC58110 pinouts 37
5-2 MC58420 pinouts 44
5-3 MC58320 pinouts 45
5-4 MC58220 pinouts 46
5-5 MC58120 pinouts 47
6-1 Basics, power supplies, 58000 66
6-2 Basics, clock and bypass caps, 58110 68
6-3 Basics, clock and bypass caps, 58420 69
6-4 Oscillator filter circuit 70
6-5 Basics, reset 71
6-6 Host communication, SCI 73
6-7 Host communication, RS232 and RS485/422 76
6-8 Host communication, CANbus 78
6-9 External memory, SRAM 512Kx16 80
6-10 External memory, Ready signal generator 82
6-11 External memory, DPRAM, host management 84
6-12 ADC, single-ended temperature sensor 86
6-13 ADC, differential interface 88
6-14 User I/O space, red LED 90
6-15 Parallel word feedback, AD2S1200, 4 axes 92
6-16 Host communication, parallel interface, 16/16 mode 94
6-17 Host communication, parallel interface, 8/16 mode 96
6-18 Motor driver protection circuitry 98
6-19 SPI DAC, DC brush, single axis 101
6-20 Parallel DAC, brushless DC, sheet 1 of 2 104
6-21 Parallel DAC, brushless DC, sheet 2 of 2 105
6-22 Brushless DC Atlas With Single-Axis Magellan 108

6-23 DC Brush & Step Motor Atlas With Multi-Axis Magellan 110
6-24 Step Motor Atlas Operating In Pulse & Direction Mode 112
6-25 PWM, brushless DC, L6234 114
6-26 PWM, DC brush, LMD18200 116
6-27 PWM, µStep, A3953S, fast decay 118
6-28 Encoded PWM signal spectra 119
6-29 Filter frequency response 119
6-30 Filter output to 150 Hz electrical cycle 120
6-31 Pulse and direction, A3977 122

1. The MC50000 Family

1

In This Chapter

- Magellan Motion Control IC Family Overview
- MC58420-Series ICs & Developer Kit

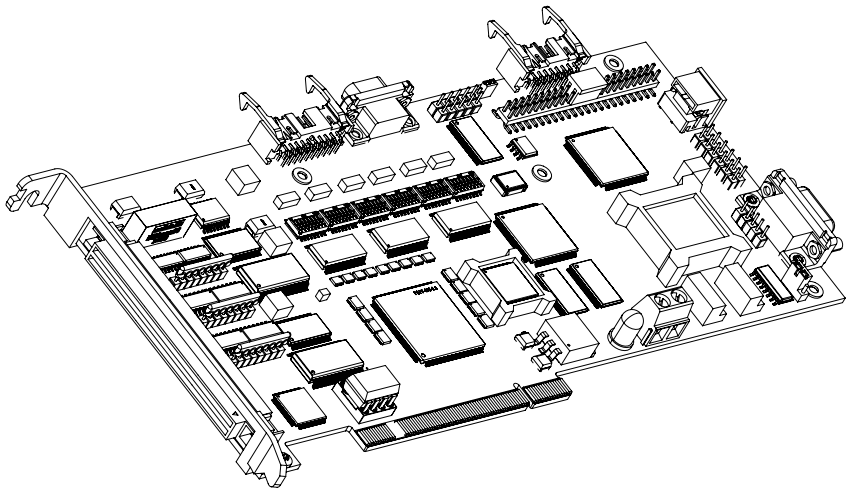
1.1 Magellan Motion Control IC Family Overview

	MC58000 Series (Except MC58113)	MC55000 Series	MC58113 Series
# of axes	1, 2, 3, 4	1, 2, 3, 4	1+ (primary & aux channel encoder input)
Motor types supported	DC Brush, Brushless DC, step motor	Step motor	DC Brush, Brushless DC, step motor
Output format	SPI Atlas, PWM, DAC, pulse & direction	Pulse & direction	SPI Atlas, PWM, DAC, pulse & direction
Parallel host communication	✓	✓	
Serial host communication	✓	✓	✓
CAN 2.0B host communication	✓	✓	✓
SPI host communication			✓
Incremental encoder input	✓	✓	✓
Parallel word device input	✓	✓	
Index & Home signals	✓	✓	✓
Position capture	✓	✓	✓
Directional limit switches	✓	✓	✓
PWM output	✓		✓
Parallel DAC output	✓		
SPI Atlas interface	✓		✓
SPI DAC output	✓		✓
Pulse & direction output	✓	✓	✓
Digital current control	✓ (with Atlas)		✓
Field oriented control	✓ (with Atlas)		✓
Under/overvoltage sense	✓ (with Atlas)		✓
I^2T Current foldback	✓ (with Atlas)		✓
DC Bus shunt resistor control			✓
Overtemperature sense	✓ (with Atlas)		✓
Short circuit sense	✓ (with Atlas)		✓
Trapezoidal profiling	✓	✓	✓
Velocity profiling	✓	✓	✓
S-curve profiling	✓	✓	✓
Electronic gearing	✓	✓	✓
On-the-fly changes	✓	✓	✓
PID position servo loop	✓		✓
Dual biquad filters	✓		✓
Dual encoder loop	✓ (multi-axis configurations only)		✓
Programmable derivative sampling time	✓		✓
Feedforward (accel & vel)	✓		✓

	MC58000 Series (Except MC58113)	MC55000 Series	MC58113 Series
Data trace/diagnostics	✓	✓	✓
Motion error detection	✓	✓ (with encoder)	✓
Axis settled indicator	✓	✓ (with encoder)	✓
Analog input	✓	✓	✓
Programmable bit output	✓	✓	✓
Software-invertible signals	✓	✓	✓
User-defined I/O	✓	✓	
Internal Trace Buffer			✓
External RAM support	✓	✓	
Multi-chip synchronization	✓		✓
Chipset configurations	MC58420 (4 axes, 2 ICs) MC58320 (3 axes, 2 ICs) MC58220 (2 axes, 2 ICs) MC58120 (1 axis, 2 ICs) MC58110 (1 axis, 1 IC)	MC55420 (4 axes, 2 ICs) MC55320 (3 axes, 2 ICs) MC55220 (2 axes, 2 ICs) MC55120 (1 axis, 2 ICs) MC55110 (1 axis, 1 IC)	MC51113 (1+ axis, 1 IC) MC53113 (1+ axis, 1 IC) MC54113 (1+ axis, 1 IC) MC58113 (1+ axis, 1 IC)
IC Package: CP chip	MC58x20: 144 pin TQFP MC58110: 144 pin TQFP	MC55x20: 144 pin TQFP MC55110: 144 pin TQFP	100 pin TQFP
IC Package: IO chip	MC58x20: 100 pin TQFP MC58110: NA	MC55x20: 100 pin TQFP MC55110: NA	N/A

1.2 MC58420-Series ICs & Developer Kit

Figure 1-1: DK58420 Board



There are five members of the MC58420-series ICs; MC58420, MC58320, MC58220, MC58120, and MC58110. A single developer kit supports all of these ICs:

Developer Kit p/n	Installed IC	Motors Supported
DK58420	MC58420	DC Brush, Brushless DC, step motor

The DK58420 developer kit comes installed with the four axis MC58420CP chip. For users interested in other MC58000 chips, for example MC58320, MC58220, MC58120, or MC58110 (three, two, one, and one axes respectively) there are two options. The first is to separately order the desired MC58000 family CP and take advantage of the fact that the CP chip is socketed by swapping the desired IC with the MC58420CP.

The second option is to leave the MC58420 installed and disable one or more higher order axes and reduce the sample time to match the default sample time of the desired IC. For example to make the installed four axis MC58420 mimic a two axis MC58220 the **SetOperatingMode** command would be used to disable axes #3 and #4, and the **SetSampleTime** command would be used to set the sample time to 153.6 μSec.

Note that the Magellan IO chip that accompanies the CP chip supports one, two, three, and four axes CP chips so there is no need to swap or change the functionality of the Magellan IO chip installed in the developer kit.

Note that throughout this manual the term MC58000 means the complete product family; MC58420, MC58320, MC58220, MC58120, and MC58110.

This page intentionally left blank.

2. Functional Characteristics

2

In This Chapter

- Configurations, Parameters, and Performance
- Physical Characteristics and Mounting Dimensions
- Absolute Maximum Environmental and Electrical Ratings
- MC58110 System Configuration — Single Chip, 1 Axis Control
- MC58x20 System Configuration — Two Chip, 1 To 4 Axis Control

2.1 Configurations, Parameters, and Performance

Configuration	4 axes (MC58420) 3 axes (MC58320) 2 axes (MC58220) 1 axis (MC58120 or MC58110)
Operating modes	**Servo** Closed loop (motor command is driven from output of servo filter) Open loop (motor command is driven from user-programmed register) **Microstepping** Open loop (motor command is driven from output of trajectory generator & microstep generator; encoder input used for stall detection) **Pulse & Direction** Open loop (pulse generator is driven by trajectory generator output, encoder input used for stall detection)
Communication modes	8/16 parallel 8-bit external parallel bus with 16-bit command word size 16/16 parallel16-bit external parallel bus with 16-bit command word size Point to point asynchronous serial Multi-drop asynchronous serial CAN bus 2.0B, protocol co-exists with CANOpen, 11-bit identifier.
Serial port baud rate range	1,200 baud to 460,800 baud
CAN port transmission rate range	10,000 baud to 1,000,000 baud
Profile modes	S-curve point-to-point Position, velocity, acceleration, deceleration, and jerk parameters Trapezoidal point-to-point Position, velocity, acceleration, and deceleration parameters Velocity-contouring Velocity, acceleration, and deceleration parameters Electronic Gear Encoder or trajectory position of one axis used to drive a second axis. Master and slave axes and gear ratio parameters.
Position range	-2,147,483,648 to +2,147,483,647 counts or steps
Velocity range	-32,768 to +32,767 counts or steps per cycle with a resolution of 1/65,536 counts or steps per cycle

Acceleration and deceleration ranges	0 to +32,767 counts or steps per cycle[2] with a resolution of 1/65,536 counts or steps per cycle[2]
Jerk range	0 to ½ counts or steps per cycle[3] with a resolution of 1/4,294,967,296 counts or steps per cycle[3]
Electronic gear ratio range	-32,768 to +32,767 with a resolution of 1/65,536 (negative and positive direction)
Filter modes	Scalable PID + Velocity feedforward + Acceleration feedforward + Bias. Also includes integration limit, settable derivative sampling time, output motor command limiting and two bi-quad filters. Dual encoder feedback mode where auxiliary encoder is used for backlash compensation
Filter parameter resolution	16 bits
Position error	32 bits
Position error tracking	Motion error windowAllows axis to be stopped upon exceeding programmable window Tracking windowAllows flag to be set if axis exceeds a programmable position window Axis settledAllows flag to be set if axis exceeds a programmable position window for a programmable amount of time after trajectory motion is complete
Motor output modes	PWM10-bit resolution at 20 kHz, or 8-bit resolution at 80 kHz Parallel DAC-compatible16 bits SPI DAC-compatible16 bits Pulse and directionMC58x20: 4.98 Mpulses/sec maximum MC58110: 97.6 kpulses/sec maximum SPI AtlasFour-signal SPI interface with 16-bit packet commands and SPI Atlas protocol..
Commutation rate	10 kHz
Microstepping waveform	Sinusoidal
Microsteps per full step	Programmable, 1 to 256
Maximum encoder rate	Incremental (up to 8 Mcounts/sec) Parallel-word (up to 160 Mcounts/sec)
Parallel encoder word size	16 bits
Parallel encoder read rate	20 kHz (reads all axes every 50 μsec)
Hall sensor inputs	3 Hall effect inputs per axis (TTL level signals)
Cycle timing range	51.2 microseconds to 1.048576 seconds
Minimum cycle time	51.2 microseconds
Multi-chip synchronization	<1 μsec difference between master and slave servo cycle
Limit switches	2 per axis: one for each direction of travel
Position-capture triggers	2 per axis: index and home signals
Other digital signals (per axis)	1 *AxisIn* signal per axis, 1 *AxisOut* signal per axis.
Software-invertible signals	*Encoder A, Encoder B, Index, Home, AxisIn, AxisOut, PositiveLimit, NegativeLimit, HallA, HallB, HallC* (all individually programmable per axis), *Pulse, Direction.*
Analog input	8 10-bit analog inputs
User defined discrete I/O	256 16-bit width user defined I/O
RAM/external memory support	65,536 blocks of 32,768 16-bit words per block. Total accessible memory is 2,147,483,648 16-bit words.
Trace modes	one-time, continuous
Maximum number of trace variables	4
Number of traceable variables	31

2.2 Physical Characteristics and Mounting Dimensions

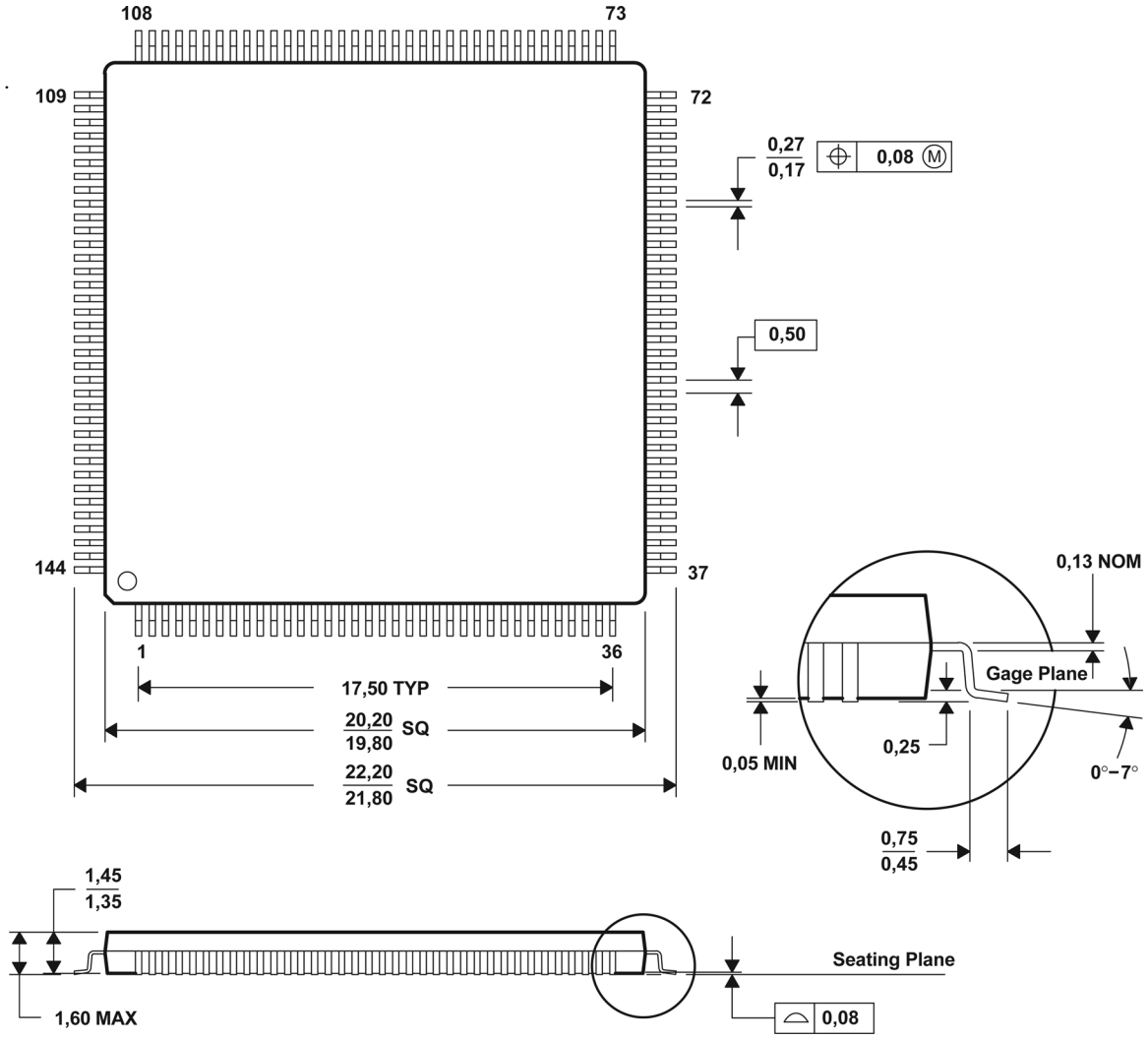
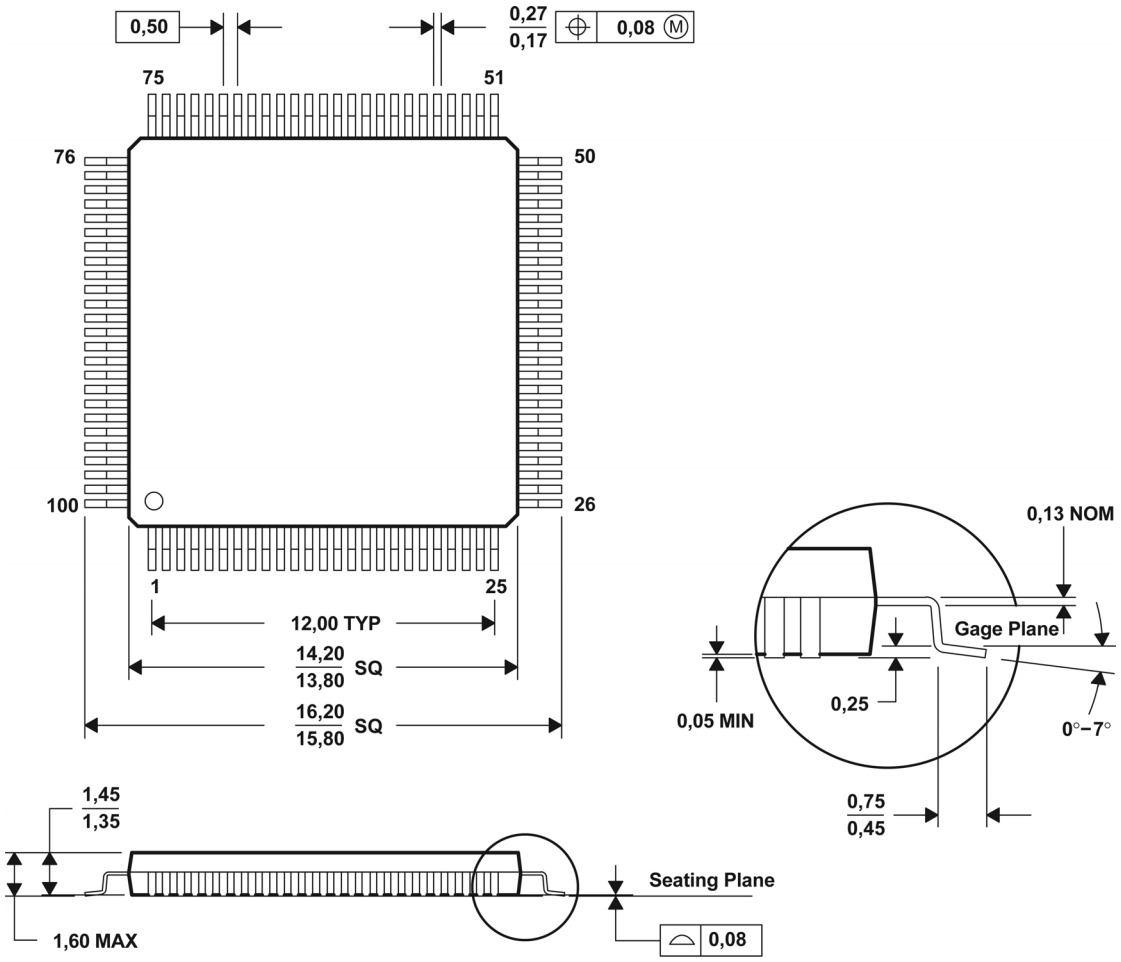


Figure 2-1: CP chip (all dimensions in millimeters)

Figure 2-2: IO Chip (all dimensions in millimeters)



2.3 Absolute Maximum Environmental and Electrical Ratings

2.3.1 CP 58110, 58x20

Supply Voltage (Vcc)	-0.3V to +4.6V
Vcc5 range	-0.3V to +5.5V
Input voltage (Vi)	-0.3V to +4.6V
Package thermal impedance (θ_{JA})	32°C/W
Junction temperature range (Tj)	-40°C to 150°C
Storage Temperature (Ts)	-65°C to 150°C
Nominal Clock Frequency (Fclk)	20.0 MHz

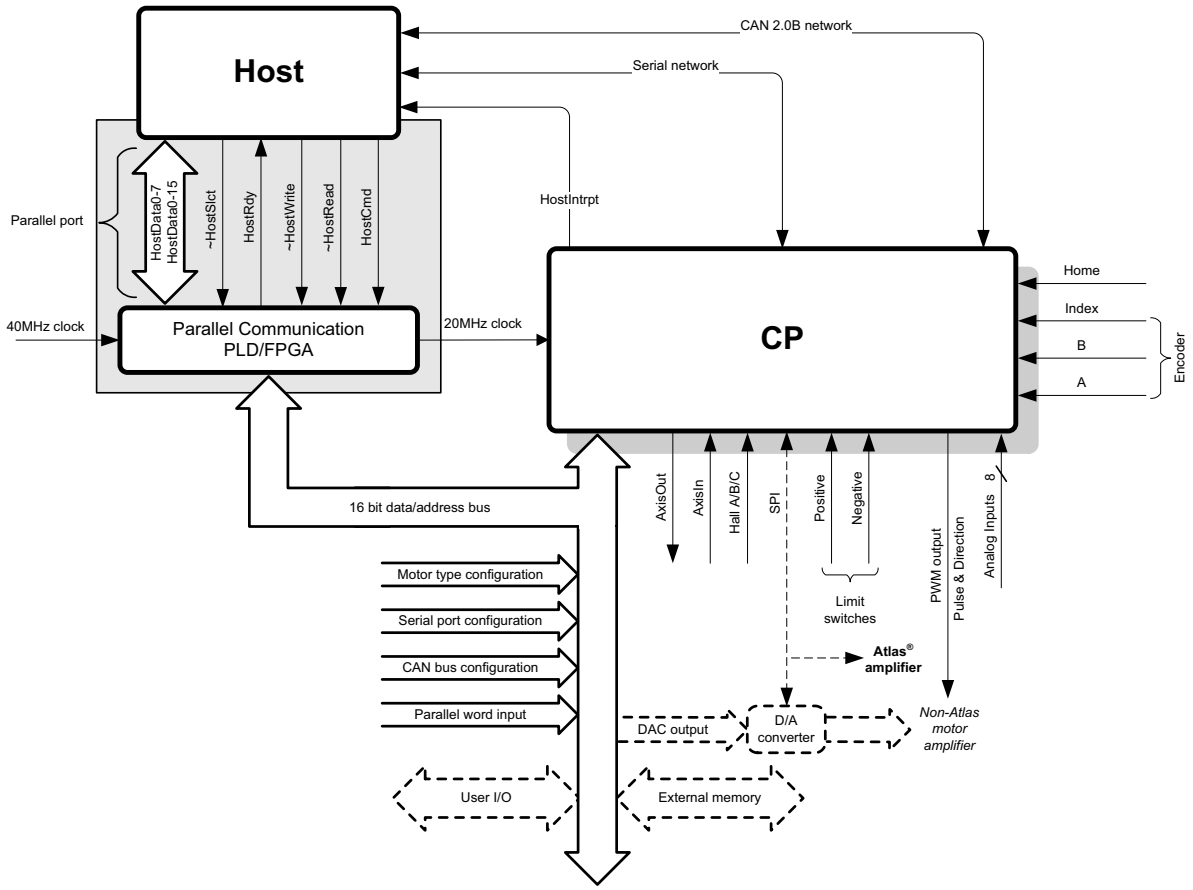
2.3.2 IO 58x20

Supply Voltage (Vcc)	-0.5V to +3.6V
Input voltage (Vi)	-0.5V to Vcc +0.5V
Package thermal impedance (θ_{JA})	39.7°C/W
Junction temperature range (Tj)	-40°C to 150°C
Storage Temperature (Ts)	-65°C to 150°C
Nominal Clock Frequency (Fclk)	40.0 MHz

2.4 MC58110 System Configuration — Single Chip, 1 Axis Control

The following figure shows the principal control and data paths in an MC58110 system.

Figure 2-3: MC58110 control and data paths



The CP chip is a self-contained motion control IC. In addition to handling all system functions, the CP chip contains the profile generator, which calculates position, velocity, acceleration, and values for a trajectory. When an axis is configured for servo motor control, a digital servo filter controls the motor output signal. When an axis is configured for microstepping motor control, a commutator controls the motor output signal. In either case, one of four types of output can be generated:

- a Pulse-Width Modulated (PWM) signal output
- a DAC-compatible value routed via the data bus to the appropriate D/A converter
- a DAC-compatible value routed via the SPI port to the appropriate D/A converter
- an SPI Atlas bus-compatible bi-directional amplifier interface

If an axis is configured for step motor control, the CP chip generates pulse and direction signals. Axis position information returns to the motion control IC in the form of encoder feedback using either the incremental encoder input signals, or via the bus as parallel word input.

The MC58110 can co-exist in a CANOpen network as a slave device. It is CAN 2.0B compliant.

The shaded area shows the PLD/FPGA that must be provided by the designer if parallel communication is required. For a description and an example of the necessary logic (in schematic format) contact PMD.

2.5 MC58x20 System Configuration — Two Chip, 1 to 4 Axis Control

The following figure shows the principal control and data paths in an MC58x20 system.

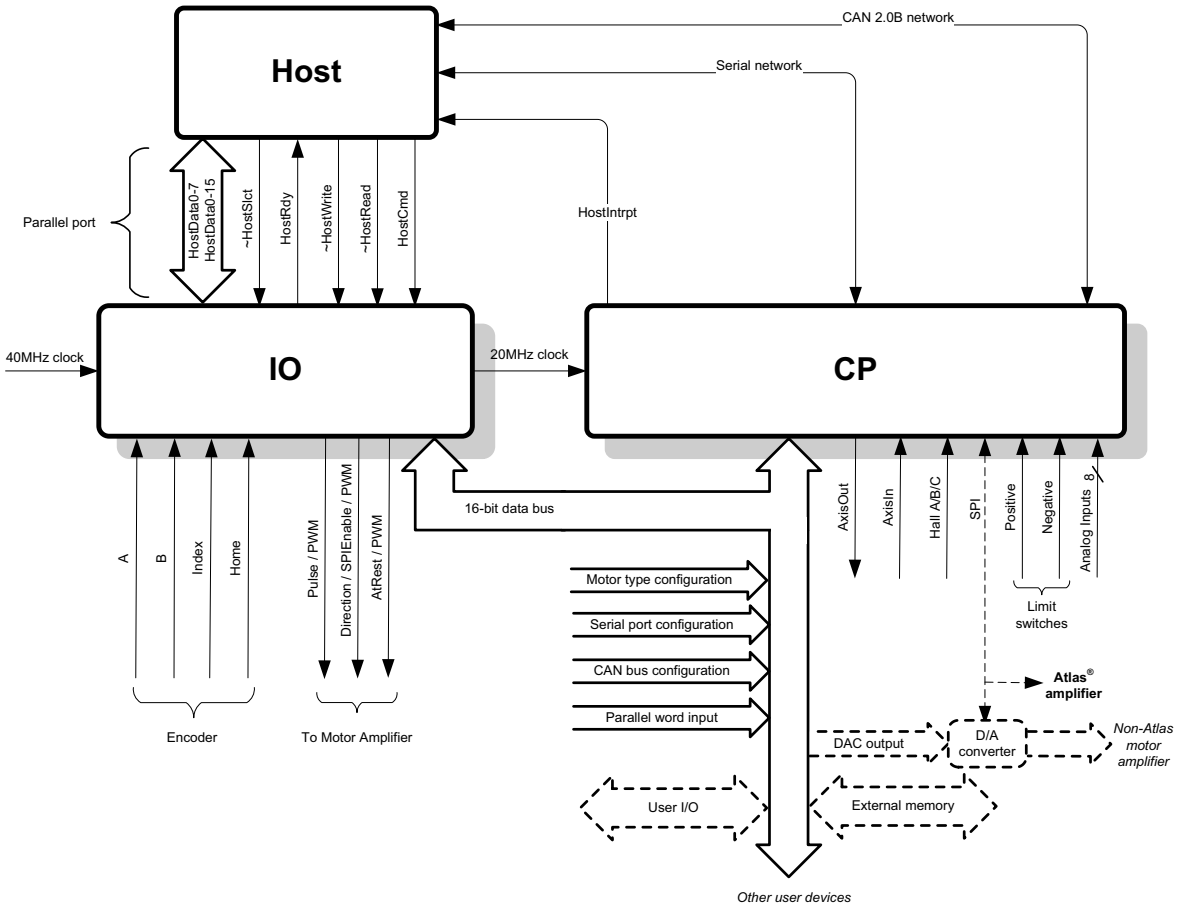


Figure 2-4: MC58x20 control and data paths

The IO chip contains the parallel host interface, the incremental encoder input along with motor output signals that are configured as PWM or pulse and direction signals according to the motor type selected for each axis.

The CP chip contains the profile generator, which calculates position, velocity, acceleration, and values for a trajectory. When an axis is configured for servo motor control, a digital servo filter controls the motor output signal. When an axis is configured for microstepping motor control, a commutator controls the motor output signal. In either case, one of four types of output can be generated:

- a Pulse-Width Modulated (PWM) signal output
- a DAC-compatible value routed via the data bus to the appropriate D/A converter
- a DAC-compatible value routed via the SPI port to the appropriate D/A converter
- an SPI Atlas bus-compatible bi-directional amplifier interface

When an axis is configured for step motor control, the IO chip generates the pulse and direction signals.

Axis position information returns to the motion control IC in the form of encoder feedback using either the incremental encoder input signals, or via the bus as parallel word input.

The MC58x20 can co-exist in a CANOpen network as a slave device. It is CAN 2.0B compliant.

This page intentionally left blank.

3. Electrical Characteristics

In This Chapter

- DC Characteristics for 58110, 58x20 CP
- DC Characteristics for 58x20 IO
- AC Characteristics

3.1 DC Characteristics for 58110, 58x20 CP

(V_{cc} and T_a per operating ratings, F_{clk} = 20.0 MHz)

Symbol	Parameter	Minimum	Maximum	Conditions
V_{cc}	Supply Voltage	3.0V	3.6V	
I_{dd}	Supply Current		120 mA	All I/O pins are floating
T_a	Operating free-air temperature	-40°C	85°C	
Input Voltages				
V_{ih}	Logic 1 input voltage	2.0V	V_{cc} + 0.3V	
V_{il}	Logic 0 input voltage	0	0.8V	
Output Voltages				
V_{oh}	Logic 1 Output Voltage	2.4V		Io = -2 mA
V_{ol}	Logic 0 Output Voltage		0.4V	Io = 2 mA
Other				
I_{out}	Tri-State output leakage current	-2 μA	2 μA	Vin = 0 or Vcc
I_{in}	Input current	-30 μA	30 μA	
C_{io}	Input/Output capacitance		2/3 pF	typical
Analog Input				
Z_{ai}	Analog input source impedance		1.4 kΩ	
I_a	Analog supply current		22 mA	
I_{refhi}	Vrefhi input current		1.5 mA	
C_{ai}	Analog input capacitance		30 pF	typical
E_{zo}	Zero-offset error		±2 LSB	typical
E_{dnl}	Differential nonlinearity error. Difference between the step width and the ideal value.		±2 LSB	
E_{inl}	Integral nonlinearity error. Maximum deviation from the best straight line through the ADC transfer characteristics, excluding the quantization error.		±2 LSB	

3.2 DC Characteristics for 58x20 IO

(Vcc and Ta per operating ratings, Fclk = 40.0 MHz)

Symbol	Parameter	Minimum	Maximum	Conditions
Vcc	Supply Voltage	3.0V	3.6V	
Idd	Supply Current		24 mA	All I/O pins are floating
Ta	Operating free-air temperature	-40°C	85°C	
Input Voltages				
Vih	Logic 1 input voltage	2.0V	Vcc	
Vil	Logic 0 input voltage	0	0.8V	
Output Voltages				
Voh	Logic 1 Output Voltage	2.4V		Io = -2 mA
Vol	Logic 0 Output Voltage		0.4V	Io = 6 mA
Other				
Iout	Tri-State output leakage current	-10 µA	10 µA	
Iin	Input current	-10 µA	10 µA	
Cio	Input/Output capacitance		7/7 pF	typical

3.3 AC Characteristics

See timing diagrams in Chapter 4, "I/O Timing Diagrams," for *Tn* numbers. The symbol "~" indicates active low signal.

Timing Interval	*Tn*	Minimum	Maximum
Clock			
IOClkIn Frequency (Fclk)[1]		8 MHz	40 MHz
IOClkIn pulse duration[3]	T1a	0.4 T2a	0.6 T2a
IOClkIn Period	T2a	25 nsec	125 nsec
CPClkIn Frequency (Fclk)[1]		4 MHz	20 MHz
CPClkIn pulse duration[3]	T1b	0.4 T2b	0.6 T2b
CPClkIn Period	T2b	50 nsec	250 nsec
CPClkIn rise/fall time	T58		5 nsec
Encoder			
Encoder Pulse Width	T3	200 nsec	
Dwell Time Per State	T4	100 nsec	
Index Setup and Hold (relative to Quad A and Quad B low)	T5	0 nsec	
Host IO			
~HostSlct Hold Time	T6	0 nsec	
~HostSlct Setup Time	T7	0 nsec	
HostCmd Setup Time	T8	0 nsec	
HostCmd Hold Time	T9	0 nsec	
Read Data Access Time	T10		25 nsec
Read Data Hold Time	T11		10 nsec
~HostRead High to HI-Z Time	T12		20 nsec
HostRdy Hold Time	T13	40 nsec	70 nsec
~HostWrite Pulse Width	T14	70 nsec	
Write Data Delay Time	T15		15 nsec
Write Data Hold Time	T16	0 nsec	
Read Recovery Time[2]	T17	60 nsec	

Timing Interval	*Tn*	Minimum	Maximum
Write Recovery Time[2]	T18	60 nsec	
~HostRead Pulse Width	T19	70 nsec	
External Memory Read			
ClockOut low to control valid	T20		4 nsec
ClockOut low to address valid	T21		8 nsec
Address valid to ~ReadEnable low	T22	5.5 nsec	
ClockOut high to ~ReadEnable low	T23		5 nsec
ClockOut low to ~ReadEnable high	T23a	-8 nsec	1 nsec
Data access time from Address valid	T24		40 nsec
Data access time from ~ReadEnable low	T25		31 nsec
Data setup time before ~ReadEnable high	T25a	8 nsec	
Data hold time after ~ReadEnable high	T26	0 nsec	
~ReadEnable high to Address invalid	T26a	0 nsec	
ClockOut low to control inactive	T27		5 nsec
Address hold time after ClockOut low	T28	2 nsec	
ClockOut low to Strobe low	T29		5 nsec
ClockOut low to Strobe high	T30		6 nsec
W/~R low to R/~W rising delay time	T31		5 nsec
External Memory Write			
ClockOut high to control valid	T32		4 nsec
ClockOut high to address valid	T33		10 nsec
Address valid to ~WriteEnable low	T34	3.5 nsec	
ClockOut low to ~WriteEnable low	T35		6 nsec
ClockOut low to ~WriteEnable high	T35a		6 nsec
Data setup time before ~WriteEnable high	T36	33 nsec	
Data bus driven from ClockOut low	T37	-3 nsec	
Data hold time after ~WriteEnable high	T38	2 nsec	
ClockOut high to control inactive	T39		5 nsec
Address hold time after ClockOut low	T40	-5 nsec	
ClockOut low to Strobe low	T41		6 nsec
ClockOut low to Strobe high	T42		6 nsec
R/~W low to W/~R rising delay time	T43		5 nsec
ClockOut high to control valid	T44		6 nsec
ClockOut high to R/~W high	T44a		6 nsec
Peripheral Device Read			
ClockOut high to ClockOut low[4]	T45	112.5 nsec	562.5 nsec
Data access time from Address valid	T46		65 nsec
Data access time from ~ReadEnable low	T47		56 nsec
Peripheral Device Write			
ClockOut low to ClockOut low[4]	T48	125 nsec	625 nsec
Data setup time before ~WriteEnable high	T49	58 nsec	
Device Reset[5]			
Reset low pulse width	T50	400 nsec	
Device Ready/ Outputs Initialized	T57		1.5 msec
SPI in DAC Mode			
SPIClock clock cycle time, typical	T70		250 nSec
Pulse duration, SPIClock high	T71	115 nSec	135 nSec
Pulse duration, SPIClock low	T72	115 nSec	135 nSec
~SPIEnablen active to first SPICLock high	T73	400 nSec	
SPIClock high to SPIXmt valid delay time	T74		21 nSec

Timing Interval	*Tn*	Minimum	Maximum
SPIXmt data valid time after SPIClock low	T75	40 nSec	
Last SPIClock low to ~SPIEnablen inactive	T76	40 nSec	
SPI In Atlas Mode			
SPIClock clock cycle time, typical	T78		250 nSec
Pulse duration, SPIClock high	T79	115 nSec	135 nSec
Pulse duration, SPIClock low	T80	115 nSec	135 nSec
~SPIEnablen active to first SPIClock high	T81	400 nSec	
SPIClock high to SPIXmt valid delay time	T82		21 nSec
SPIXmt data valid time after SPIClock low	T83	40 nSec	
SPIRcv setup time before SPIClock high	T84	26 nSec	
SPIRcv valid time after SPIClock low	T85	50 nSec	
Last SPIClock low to ~SPIEnablen inactive	T86	40 nSec	

1. Performance figures and timing information valid at Fclk = 40.0 MHz for the dual chip configuration and Fclk = 20.0 MHz for the single chip configurations only. For timing information and performance parameters at lower Fclk, see Section 6.1.4, "Using a Non-standard System Clock Frequency,"
2. For 8/16 interface modes only.
3. The clock low/high split has an allowable range of 40 - 60%.
4. The minimum and maximum values correspond to a 50 nsec and 250 nsec CPClkIn clock periods, or 25 nsec and 125 nsec IOClkIn clock periods, respectively.
5. For additional important information on *Reset* signal management and timing please see Section 6.4.6, "Reset Signal."

4. I/O Timing Diagrams

4

In This Chapter

- Clock
- Quadrature Encoder Input
- Reset
- Host Interface, 8/16 Mode
- Host Interface, 16/16 Mode
- External Memory Timing
- Peripheral Device Timing
- SPI DAC Timing
- SPI Atlas Timing

For the values of *Tn*, please refer to the table in Section 3.3, "AC Characteristics."

4.1 Clock

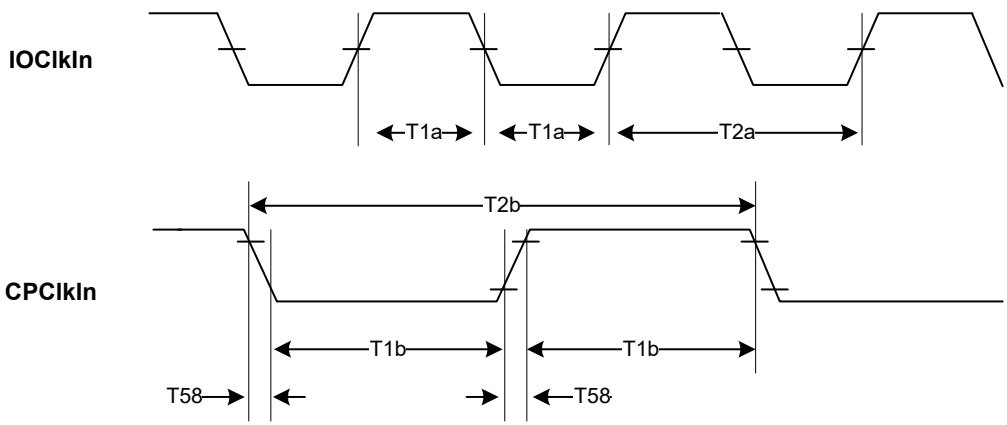
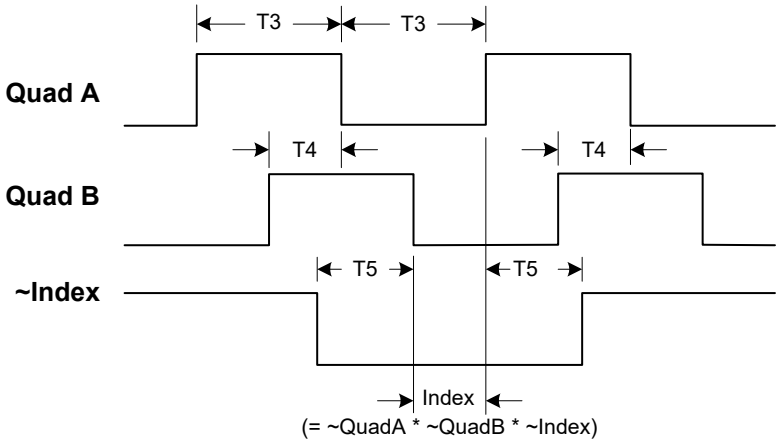


Figure 4-1: Clock timing

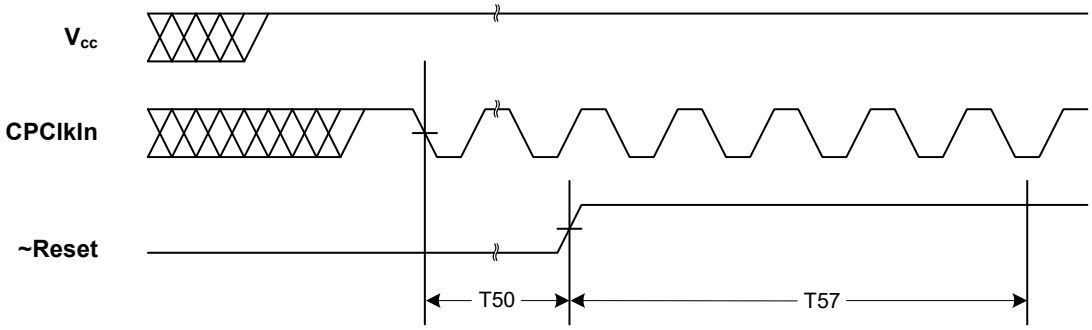
4.2 Quadrature Encoder Input

Figure 4-2: Quad encoder timing



4.3 Reset

Figure 4-3: Reset timing



NOTE: The device must be reset after power on.

NOTE: For additional important information on Reset signal management and timing please see Section 6.4.6, "Reset Signal."

4.4 Host Interface, 8/16 Mode

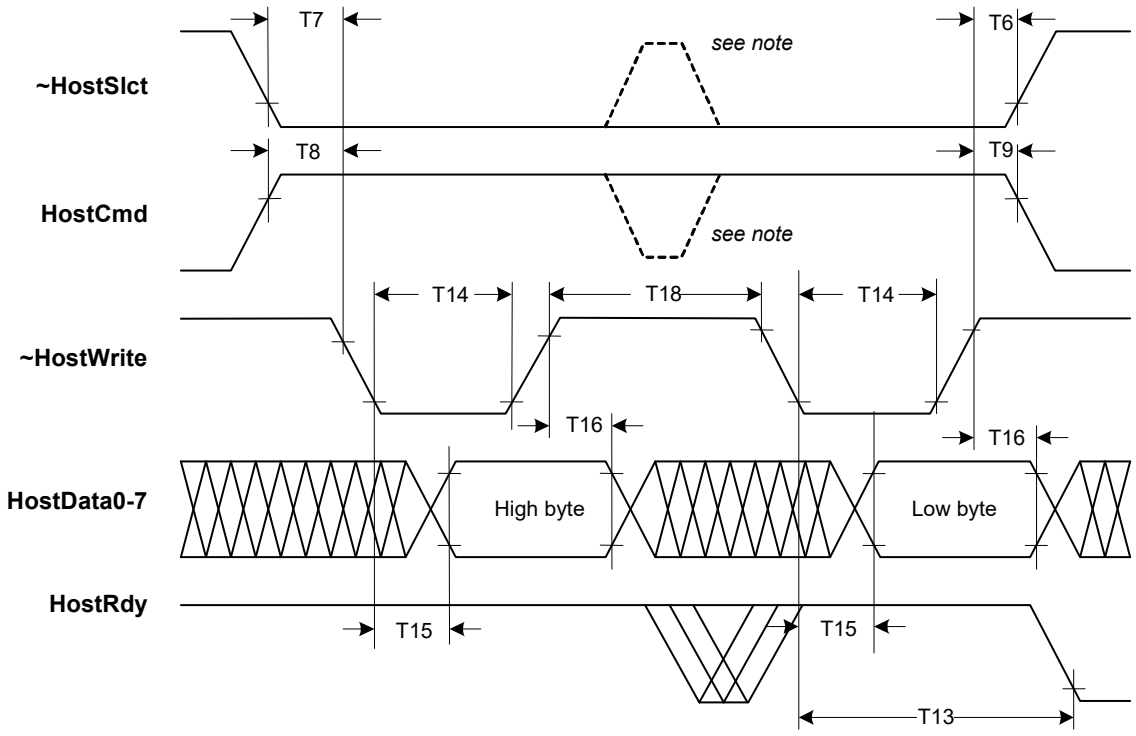


Figure 4-4: Instruction write, 8/16 mode

NOTE: If setup and hold times are met, ~HostSlct and HostComd may be de-asserted at this point.

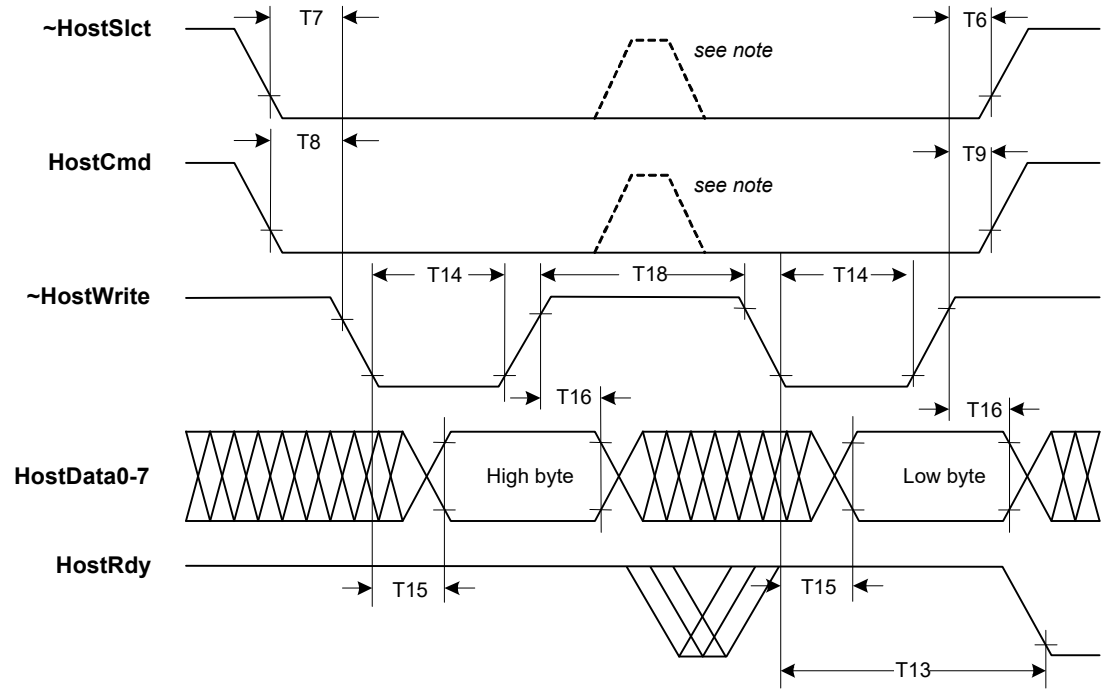
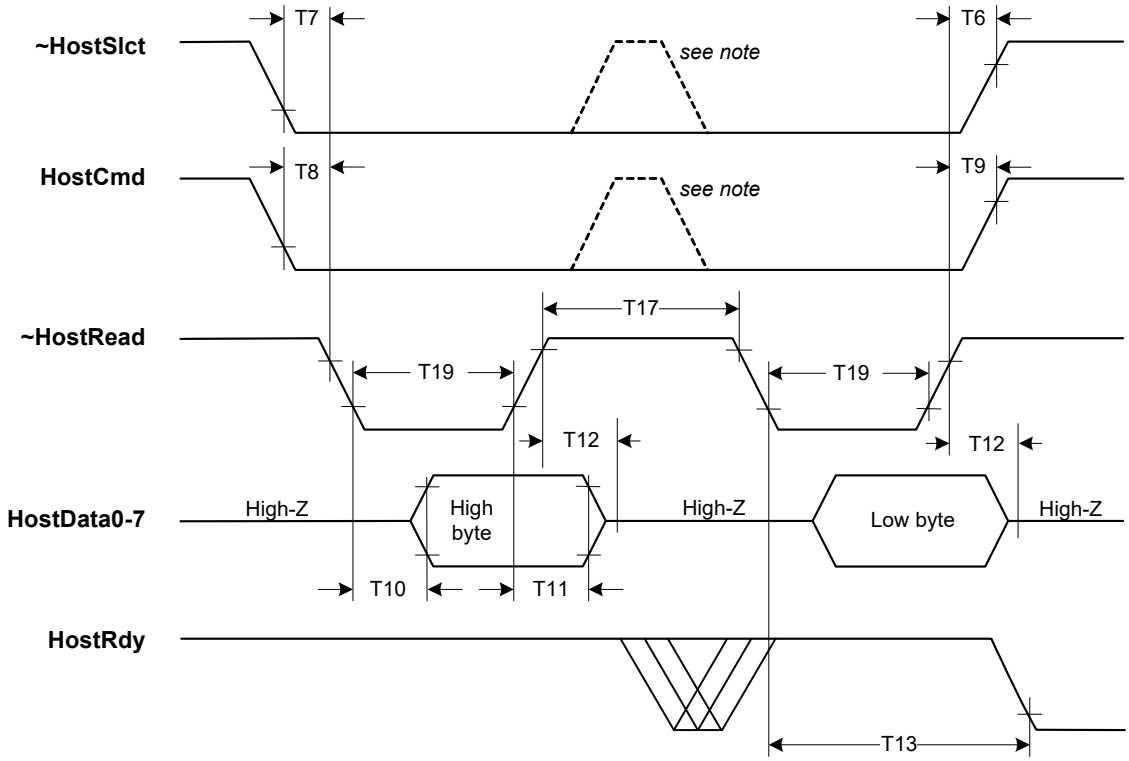


Figure 4-5: Data write, 8/16 mode

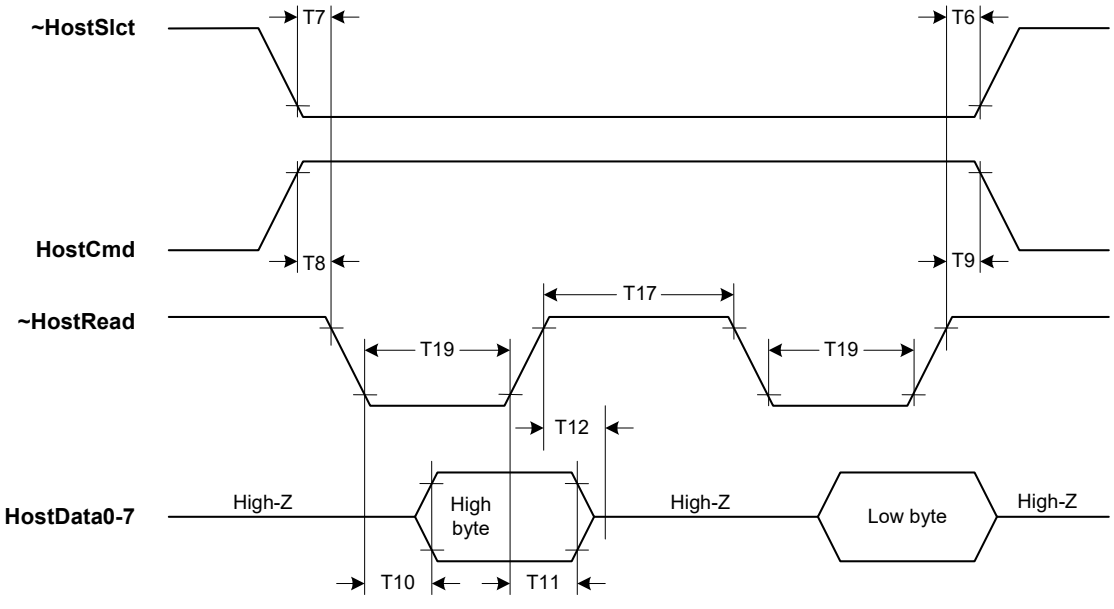
NOTE: If setup and hold times are met, ~HostSlct and HostComd may be de-asserted at this point.

Figure 4-6: Data read, 8/16 mode



NOTE: If setup and hold times are met, ~HostSlct and HostComd may be de-asserted at this point.

Figure 4-7: Status read, 8/16 mode



4.5 Host Interface, 16/16 Mode

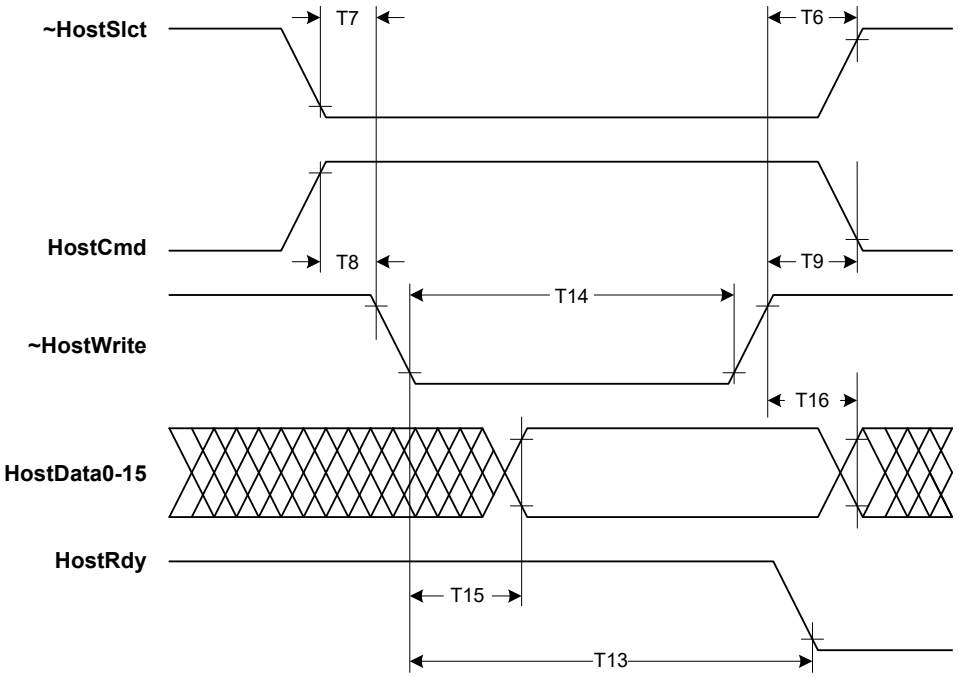


Figure 4-8: Instruction write, 16/16 mode

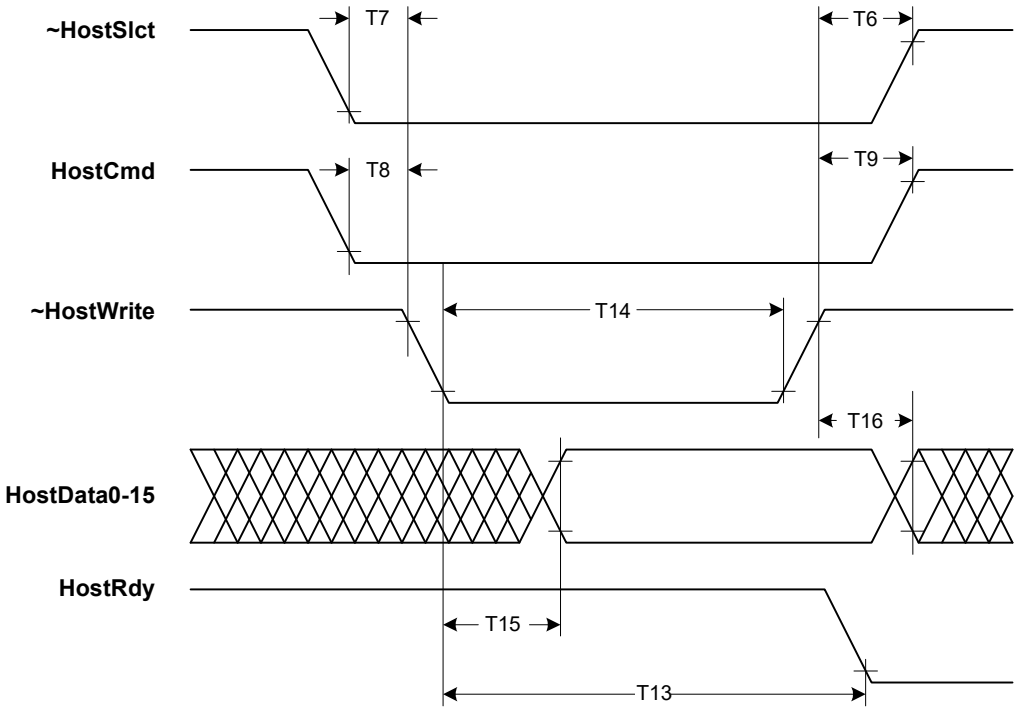


Figure 4-9: Data write, 16/16 mode

Figure 4-10: Data read, 16/16 mode

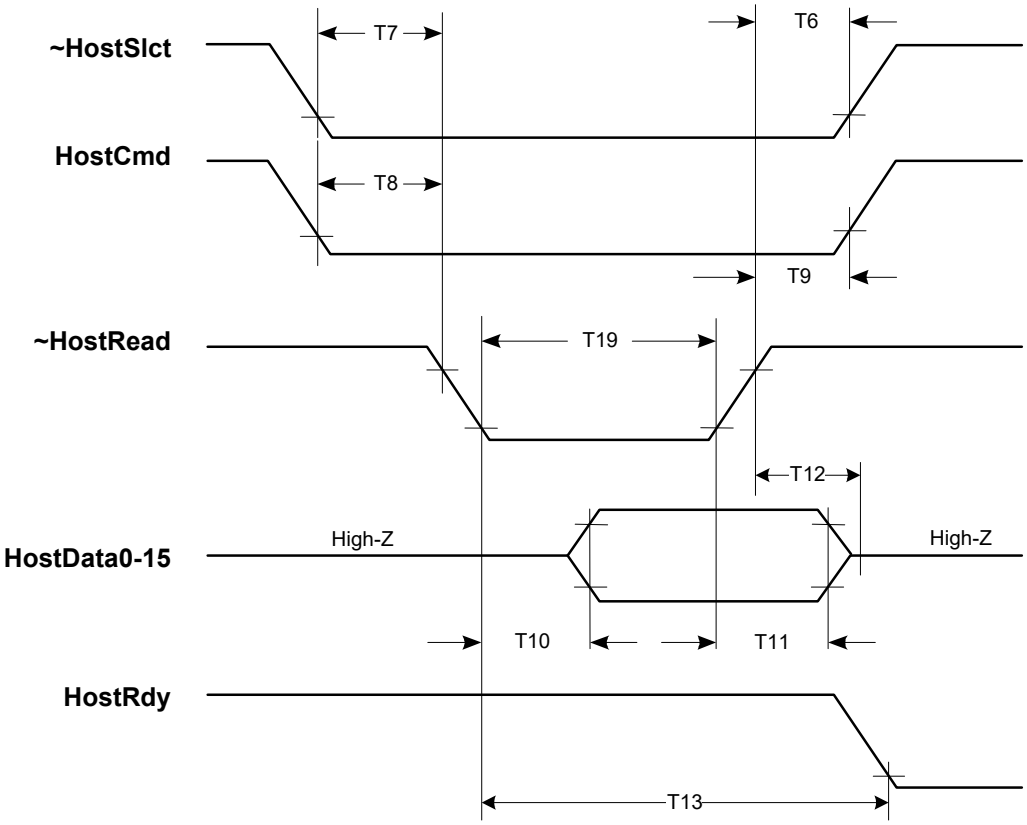
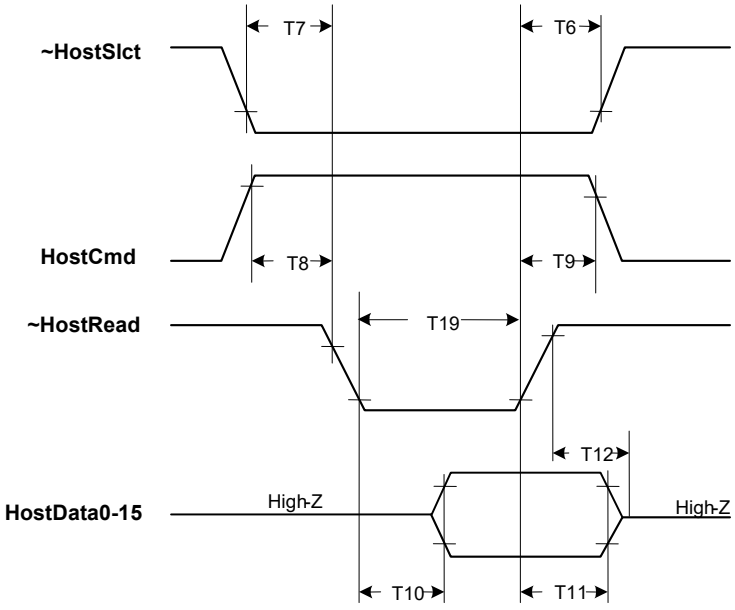


Figure 4-11: Status read, 16/16 mode



4.6 External Memory Timing

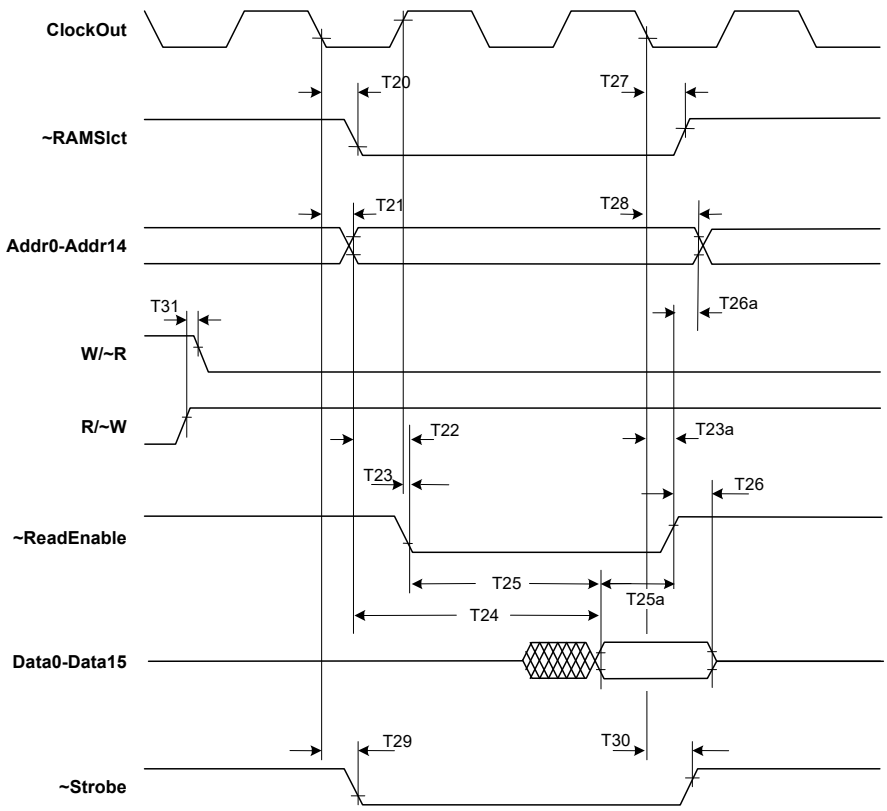
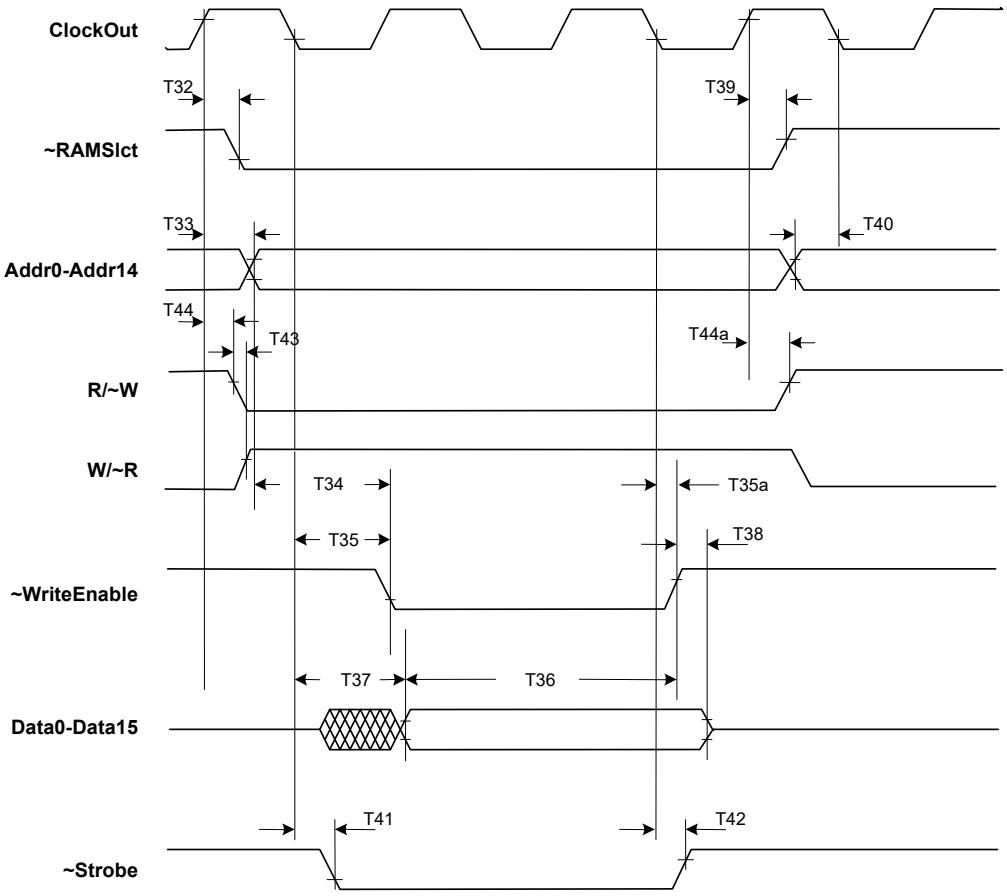


Figure 4-12: External memory read

Figure 4-13: External memory write



4.7 Peripheral Device Timing

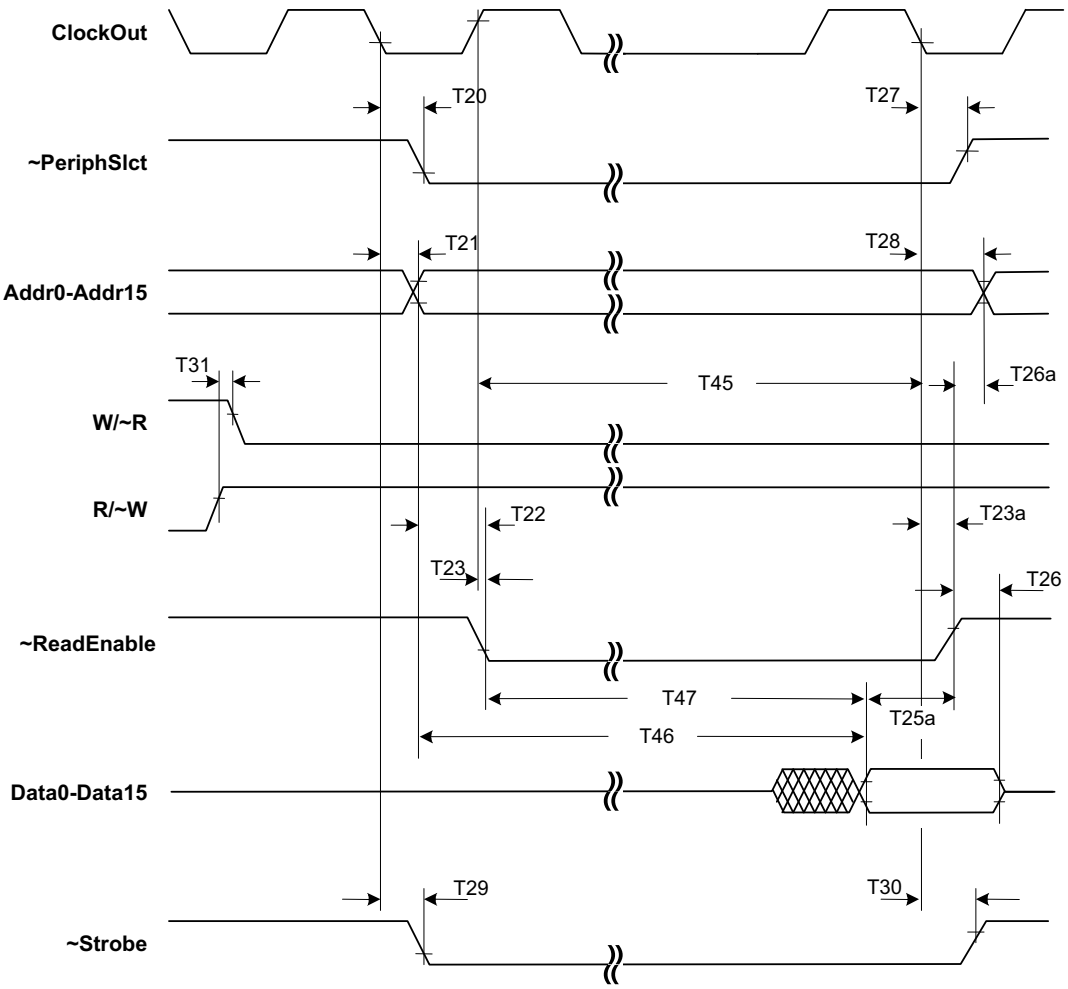
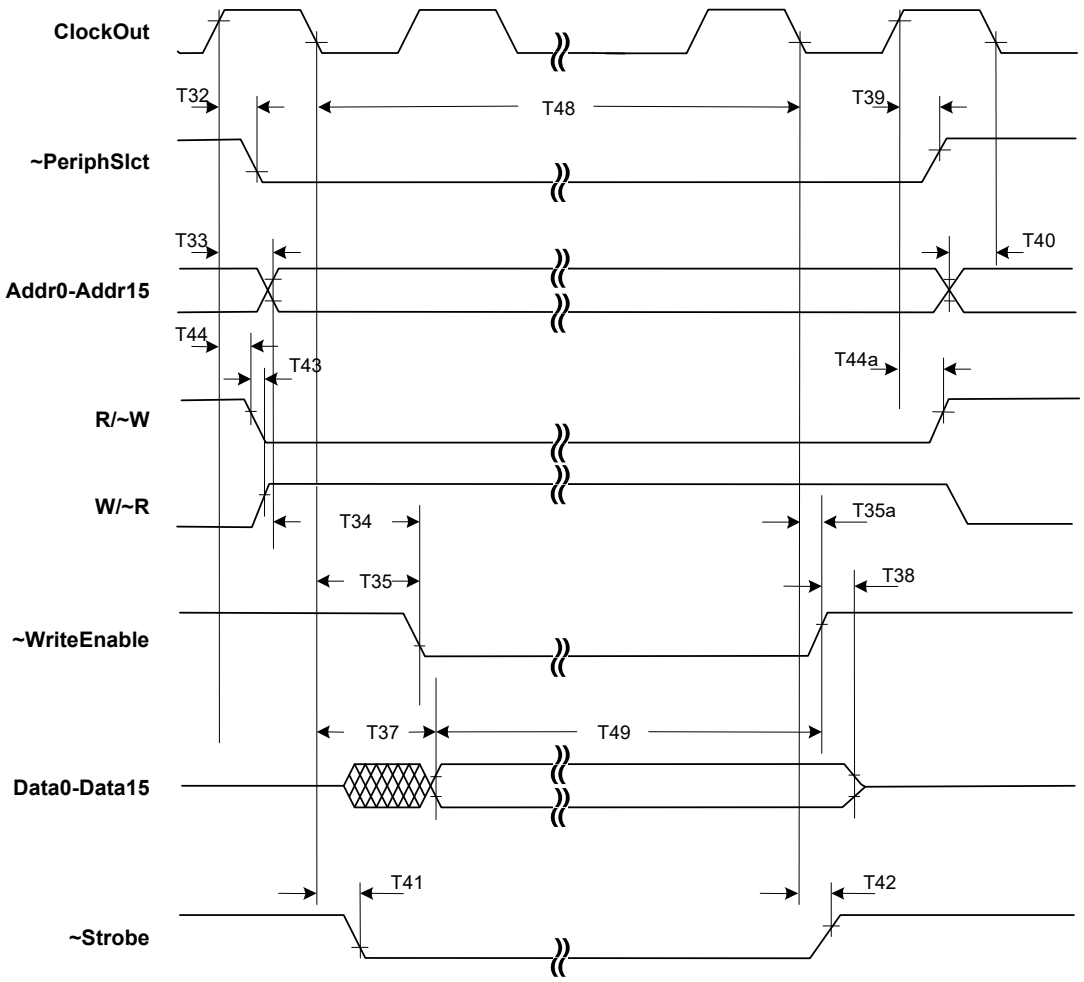


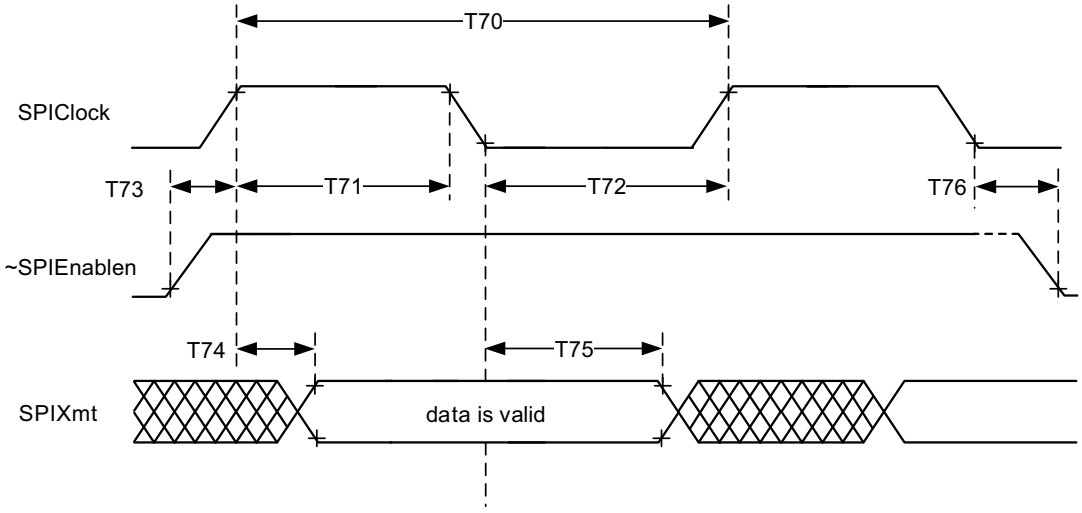
Figure 4-14: Peripheral device read

Figure 4-15: Peripheral device write



4.8 SPI DAC Timing

Figure 4-16: SPI DAC timing



4.9 SPI Atlas Timing

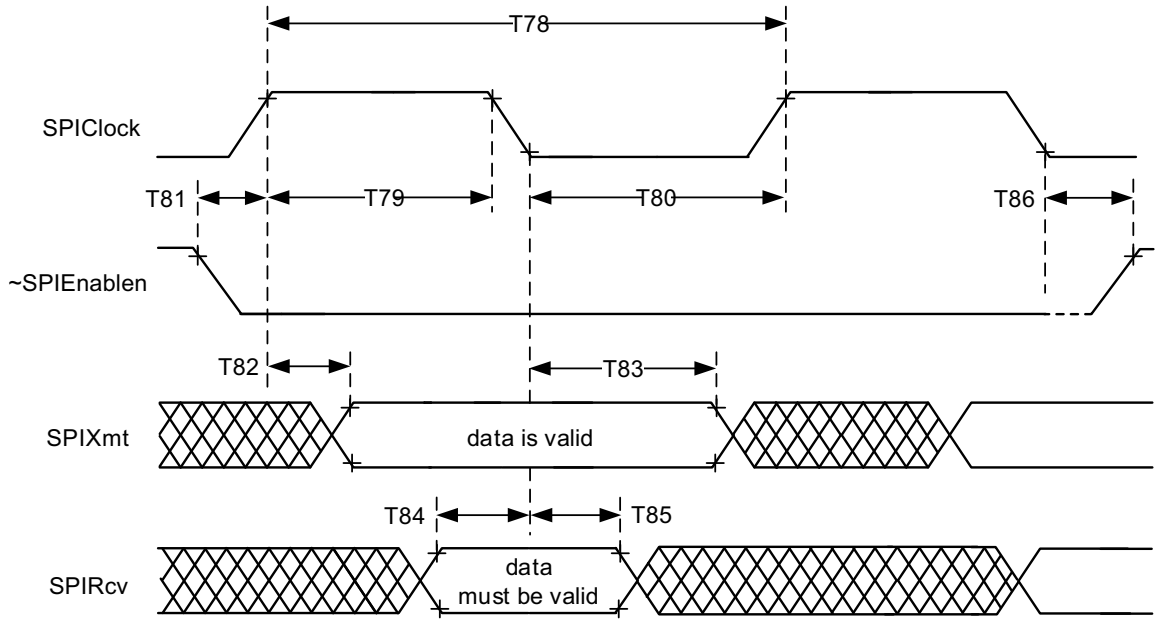


Figure 4-17: SPI Atlas timing

This page intentionally left blank.

5. Pinouts and Pin Descriptions

5

In This Chapter

- Pinouts for the MC58110
- MC58110 Pin Descriptions
- Pinouts for the MC58x20
- MC58x20 Pin Descriptions

5.1 Pinouts for the MC58110

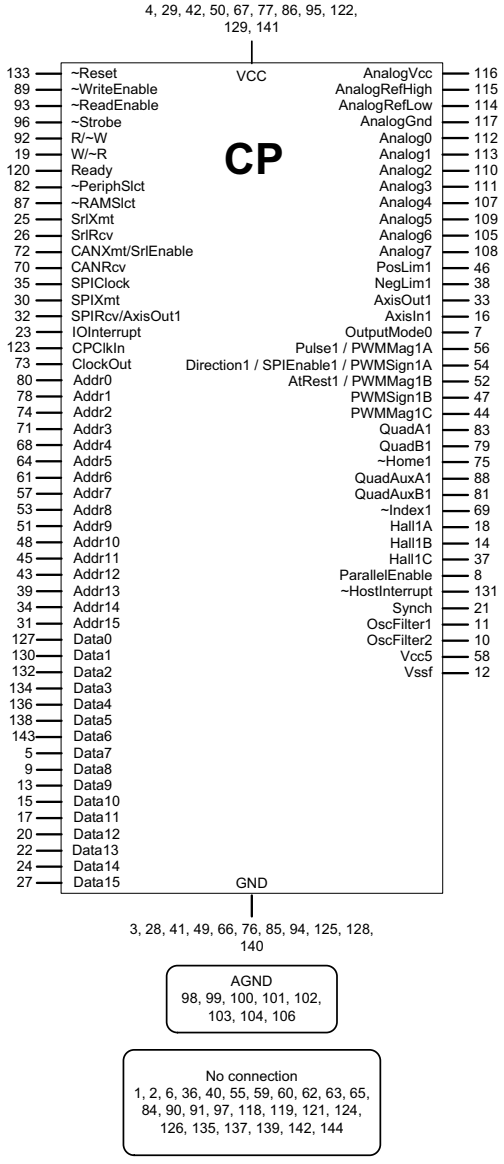


Figure 5-1: MC58110 pinouts

5.2 MC58110 Pin Descriptions

		58110 CP	
Pin Name and Number		**Direction**	**Description**
~Reset	133	input/output	This is the master reset signal. This pin must be brought *low* to reset the chipset to its initial condition. **NOTE**: A software reset will momentarily drive this pin *low*. **NOTE**: For additional important information on *Reset* signal management and timing please see Section 6.4.6, "Reset Signal."
~WriteEnable	89	output	This signal is the write-enable strobe. When *low*, this signal indicates that data is being written to the bus.
~ReadEnable	93	output	This signal is the read-enable strobe. When *low*, this signal indicates that data is being read from the bus.
~Strobe	96	output	This signal is *low* when the data and address are valid during CP communications.
R/~W	92	output	This signal is *high* when the CP chip is performing a read, and *low* when it is performing a write.
W/~R	19	output	This signal is the inverse of *R/~W*; it is *high* when *R/~W* is low, and vice versa. For some decode circuits and devices this is more convenient than *R/~W*.
Ready	120	input	*Ready* can be pulled low to add wait states for external accesses. *Ready* indicates that an external device is prepared for a bus transaction to be completed. If the device is not ready, it pulls the *Ready* pin *low*. The motion control IC then waits one cycle and checks *Ready* again. For a timing diagram and example schematic using the Ready signal refer to Section 6.8.1, "Slow Asynchronous SRAM." If used, this signal is typically asserted during RAM Read/Write or Peripheral Read/Write operations. However another use of this signal is to hold off on configuration reads done at startup. Keeping *Ready* asserted at the time of *Reset* release and beyond allows synchronization with FPGAs other external logic that may not yet be ready to receive configuration Peripheral reads. This signal may remain unconnected if it is not used.
~PeriphSlct	82	output	This signal is *low* when peripheral devices on the data bus are being addressed.
~RAMSlct	87	output	This signal is *low* when external memory is being accessed.
SrlXmt	25	output	This pin outputs serial data from the asynchronous serial port.
SrlRcv	26	input	This pin inputs serial data to the asynchronous serial port. If not used, this pin should be tied to Vcc.
CANXmt/ SrlEnable	72	output	When the CAN host interface is used, this pin transmits serial data to the CAN transceiver. When the multi-drop serial interface is used, this pin sets the serial port enable line and the *CANXmt* function is not available. *SrlEnable* is *high* during transmission for the multi-drop protocol and always *high* during point–point mode.
CANRcv	70	input	This pin receives serial data from the CAN transceiver. If not used, this pin should be tied to Vcc.
SPIClock	35	output	This pin is the clock signal used for strobing synchronous serial data on the SPI bus to DACs or to Atlas amplifiers. This signal is only active when SPI data is being transmitted.
SPIXmt	30	output	This pin transmits synchronous serial data on the SPI bus to DACs or to Atlas amplifiers.
SPIRcv/ AxisOut1	32	input or output	If OutputMode0 is low at powerup this pin serves as the SPIRcv input, used with the SPI Atlas amplifier interface. If OutputMode0 is left floating at powerup this pin provides the AxisOut1 output. AxisOut1 can be programmed to track the state of any bit in the status registers. If this pin is not used, it may remain unconnected.
IOInterrupt	23	input	This interrupt signal is used for IO to CP communication. This signal may remain unconnected if it is not used.
CPClkIn	123	input	This is the clock signal for the motion control IC. It is driven at a nominal 20 MHz.

58110 CP			
Pin Name and Number		**Direction**	**Description**
ClockOut	73	output	This signal is the reference output clock. Its frequency is twice the frequency of the input clock (which is normally 20 MHz), resulting in a nominal output frequency of 40 MHz. ClockOut will not be active when ~*Reset* is active.
Addr0 Addr1 Addr2 Addr3 Addr4 Addr5 Addr6 Addr7 Addr8 Addr9 Addr10 Addr11 Addr12 Addr13 Addr14 Addr15	80 78 74 71 68 64 61 57 53 51 48 45 43 39 34 31	output	Multi-purpose address lines. These pins comprise the CP chip's external address bus, which is used to select devices for communication over the data bus. Other address pins may be used for DAC output, parallel word input, external memory, or user-defined I/O operations. See Section 6.2, "Peripheral Device Access," for a complete memory map.
Data0 Data1 Data2 Data3 Data4 Data5 Data6 Data7 Data8 Data9 Data10 Data11 Data12 Data13 Data14 Data15	127 130 132 134 136 138 143 5 9 13 15 17 20 22 24 27	bi-directional	Multi-purpose data lines. These pins comprise the CP chip's external data bus, which is used for all communications with peripheral devices such as external memory or DACs. They may also be used for parallel-word input and for user-defined I/O operations.
AnalogVcc	116	input	Analog input V_{cc}. This pin should be connected to the analog input supply voltage, which must be in the range of 3.0V to 3.6V. If the analog input circuitry is not used, this pin should be tied to V_{cc}.
AnalogRefHigh	115	input	Analog high voltage reference for A/D input. The allowed range is *AnalogRefLow* to *AnalogVcc*. If the analog input circuitry is not used, this pin should be tied to V_{cc}.
AnalogRefLow	114	input	Analog low voltage reference for A/D input. The allowed range is *AnalogGND* to *AnalogRefHigh*. If the analog input circuitry is not used, this pin should be tied to *GND*.
AnalogGND	117	input	Analog input ground. This pin should be connected to the analog input power supply return. If the analog input circuitry is not used, this pin should be tied to *GND*.

58110 CP			
Pin Name and Number		**Direction**	**Description**
Analog0 Analog1 Analog2 Analog3 Analog4 Analog5 Analog6 Analog7	112 113 110 111 107 109 105 108	input	These signals provide general-purpose analog voltage levels which are sampled by an internal A/D converter. The A/D resolution is 10 bits. The allowed signal input range is *AnalogRefLow* to *AnalogRefHigh*. Any unused pins should be tied to *AnalogGND*. If the analog input circuitry is not used, these pins should be tied to *GND*.
PosLim1	46	input	This signal provides input from the positive-side (forward) travel limit switch. On power-up or after reset, this signal defaults to active *low* interpretation, but the interpretation can be set to active *high* interpretation using the **SetSignalSense** instruction. If this pin is not used, it may remain unconnected.
NegLim1	38	input	This signal provides input from the negative-side (reverse) travel limit switch. On power-up or after reset, this signal defaults to active *low* interpretation, but the interpretation can be set to active *high* interpretation using the **SetSignalSense** instruction. If this pin is not used, it may remain unconnected.
AxisOut1	33	output	For Magellan ICs revision 3.0 or higher this pin provides the AxisOut1 output. AxisOut1 can be programmed to track the state of any bit in the status registers.
AxisIn1	16	input	This pin is a general-purpose input which can also be used as a breakpoint input. If this pin is not used, it may remain unconnected.
OutputMode0	7	input	OutputMode0 should be tied low when SPI Atlas interfacing will be used, and with Magellan ICs revision 3.0 or higher. For Magellan ICs lower than revision 3.0 this pin should be left floating. The state of this pin is only checked during power up.

58110 CP			
Pin Name and Number		**Direction**	**Description**
PWMMag1A/ Pulse1	56	output	Depending upon the selected motor type and output mode, these signals have the following functions: *PWMMag* encodes the magnitude of the pulse width modulated output. *PWMSign* signals encode the sign of the pulse width modulated output. In 2- or 3-phase PWM 50/50 mode, *PWMMag1A/1B/1C* are the only signals, and encode both magnitude and direction in one signal. In single-phase PWM sign/magnitude mode, *PWMMag1A* and *PWMSign1A* are the PWM magnitude and direction signals respectively. In 2-phase PWM sign/magnitude mode, *PWMMag1A* and *PWMSign1A* are the PWM magnitude and direction signals for Phase A. *PWMMag1B* and *PWMMag1B* are the PWM magnitude and direction signals for Phase B. *SPIEnable* provides the enable signal when SPIDAC or SPI Atlas motor output mode is used. This signal is active high for SPI DAC, meaning this signal is *high* when the SPI DAC channel is being written to, and *low* at all other times. This signal is active low for SPI Atlas mode, meaning it is *low* when an SPI Atlas communication is occurring, and *high* at all other times. For SPI DAC output, SPI output can only be used when the axis being controlled is DC brushed, or when the amplifier expects a single-phase input and it performs brushless motor commutation. For SPI Atlas interfacing, DC Brush, Brushless DC, and step motors are supported. *Pulse* provides the pulse (step) signal to the motor. A step occurs when the signal transitions from a high state to a low state. This default behavior can be changed from a low to a high state transition using the command **SetSignalSense**. *Direction* indicates the direction of motion, and works in conjunction with the pulse signal. A high level on this signal indicates a positive direction move, and a low level indicates a negative direction move. *AtRest* indicates that the axis is at rest, and that the step motor can be switched to low power or standby. A *high* level on this signal indicates the axis is at rest. A *low* signal indicates the axis is in motion. Refer to the Motor Interfacing section of the *Magellan Motion Control IC User Guide* for more information on PWM encoding schemes.
PWMSign1A/ Direction1/ SPIEnable1	54		
AtRest1/ PWMMag1B	52		
PWMSign1B	47		
PWMMag1C	44		
QuadA1 QuadB1	83 79	input	These pins should be connected to the A and B quadrature signals from the incremental encoder. When the axis is moving in the positive (forward) direction, signal A leads signal B by 90°. **NOTE**: Some encoders require a pull-up resistor to 3.3V on each signal to establish a proper high signal. Check your encoder's electrical specification. If these pins are not used, they may remain unconnected.
~Home1	75	input	This pin provides the home signal; a general-purpose input to the position capture mechanism. A valid home signal is recognized by the motion control IC when *~Home* transitions from *high* to *low*. If this pin is not used, it may remain unconnected.
QuadAuxA1 QuadAuxB1 ~Index1	88 81 69	input	If index capture is required, the encoder A and B signals connected to *QuadA1* and *QuadB1* signals must also be connected to *QuadAuxA1* and *QuadAuxB1*. The index pin should be connected to the index signal from the incremental encoder. A valid index pulse is recognized by the motion control IC when it meets the criteria shown in Figure 4-2. If these pins are not used, they may remain unconnected. **WARNING! There is no internal gating of the index signal with the encoder A and B inputs. This must be performed externally if desired.**

58110 CP			
Pin Name and Number		**Direction**	**Description**
Hall1A Hall1B Hall1C	18 14 37	input	Hall sensor inputs. These signals encode six valid states as follows: A on, A and B on, B on, B and C on, C on, C and A on. A sensor is defined as being on when its signal is *high*. On power-up or after reset, these signals default to active *high* interpretation, but the interpretation can be set to active *low* interpretation using the **SetSignalSense** instruction. If these pins are not used, they may remain unconnected.
ParallelEnable	8	input	This signal enables/disables the parallel communication with the host. If this signal is tied *high*, the parallel interface is enabled. If this signal is tied *low,* the parallel interface is disabled. Contact PMD for more information on parallel communication. **WARNING! This signal should only be tied high if an external logic device which implements the parallel communication logic is included in the design.**
~HostInterrupt	131	output	When *low*, this signal causes an interrupt to be sent to the host processor.
Synch	21	input/output	This pin is the synchronization signal. In the disabled mode, the pin is configured as an input and is not used. In the master mode, the pin outputs a synchronization pulse that can be used by slave nodes or other devices to synchronize with the internal chip cycle of the master node. In the slave mode, the pin is configured as an input and should be connected to the Synch pin on the master node. This signal is falling edge triggered. A pulse on the pin synchronizes the internal chip cycle to the signal provided by the master node. If this pin is not used, it may remain unconnected.
OscFilter1 OscFilter2	11 10		These signals connect to the external oscillator filter circuitry. Section 6.4.5, "External Oscillator Filter," details the required filter circuitry.
V_{cc5}	58		This signal can be tied to a 5V supply if available. If 5V is not available this signal must be tied to GND. Being tied to GND will not adversely affect the device performance.
V_{ssf}	12		This signal must be tied to V_{cc}. It must also be tied to pin 28 via a bypass capacitor. A ceramic capacitor with a value between 0.1 µF and 0.01 µF should be used.
V_{cc}	4, 29, 42, 50, 67, 77, 86, 95, 122, 129, 141		CP digital supply voltage. All of these pins must be connected to the supply voltage. V_{cc} must be in the range of 3.0V to 3.6V.
GND	3, 28, 41, 49, 66, 76, 85, 94, 125, 128, 140		CP digital supply ground. All of these pins must be connected to the digital power supply return.
AGND	98, 99, 100, 101, 102, 103, 104, 106		These signals must be tied to *AnalogGND*. If the analog input circuitry is not used, these pins must be tied to *GND*.
No connection	1, 2, 6, 36, 40, 55, 59, 60, 62, 63, 65, 84, 90, 91, 97, 118, 119, 121, 124, 126, 135, 137, 139, 142, 144		These signals must remain unconnected.

5.2.1 MC58110 Pin Assignments for Multiple Motor Types

The MC58110 chip supports the output of PWM motor commands in sign/magnitude and 50/50 modes. It can also output pulse and direction signals. The CP chip assigns pin function according to the selected output mode.

If the output mode is set to **PWM sign/magnitude**, the following pinout should be used.

PWMMag1A PWMMag1B	56 52	output	These pins provide the Pulse Width Modulated signal to the motor. In PWM 50/50 mode, this is the only signal. In PWM sign-magnitude mode, this is the magnitude signal.
PWMSign1A PWMSign1B	54 47	output	In PWM sign-magnitude mode, these pins provide the sign (direction) of the PWM signal to the motor amplifier.

If the output mode is set to **PWM 50/50**, the following pinout should be used.

PWMMag1A PWMMag1B PWMMag1C	56 52 44	output	These pins provide the Pulse Width Modulated signals for each phase to the motor. If the number of phases is two, only phase A and B are valid. If the number of phases is three, phases A, B, and C are valid. The number of phases is set using the command **SetMotorType**. In PWM 50/50 mode, these are the only signals.

If the output mode is set to **Pulse and direction**, the following pinout should be used.

Pulse1	56	output	This pin provides the pulse (step) signal to the motor.
Direction1	54	output	This pin indicates the direction of motion, and works in conjunction with the pulse signal.
AtRest1	52	output	This signal indicates the axis is at rest, and that the step motor can be switched to low power or standby mode.

If the output mode is set to **SPI DAC**, the following pinout should be used.

SPIEnable1	54	output	This pin provides the enable signal when SPI DAC output is active.
SPIClock	35	output	This pin provides the SPI Clock signal.
SPIXmt	30	output	This pin holds the transmitted SPI data.

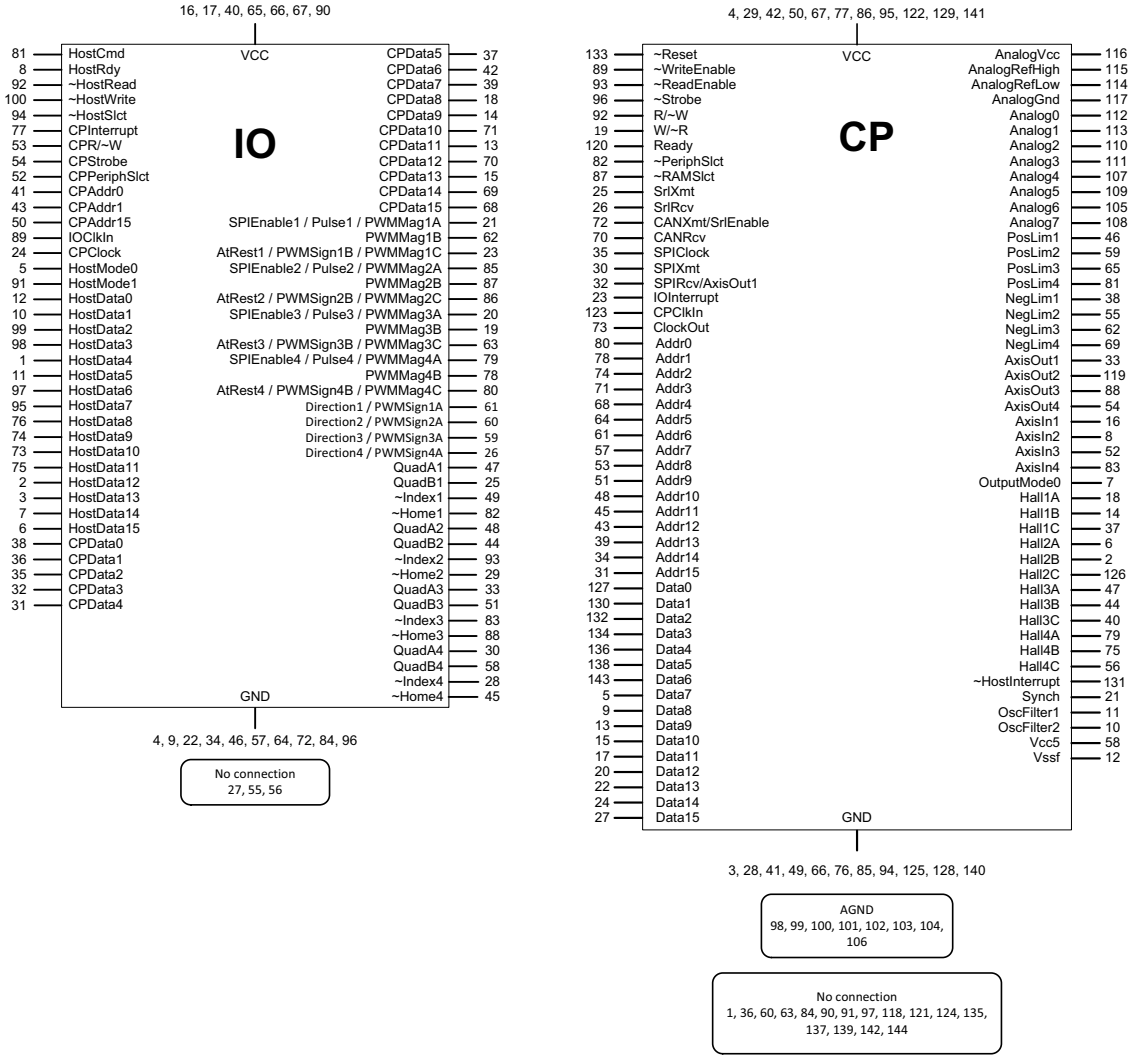
If the output mode is set for **SPI Atlas**, which is the interface used with PMD's Atlas Digital Amplifiers, the following pinouts should be used.

OutputMode0	7	input	This pin should be tied to ground.
SPIClock	35	output	This pin provides the SPI Clock signal.
SPIXmt	30	output	This pin holds the transmitted SPI data sent to Atlas.
SPIRcv	32	input	This pins holds the received SPI data sent by Atlas.
SPIEnable1	54	output	This pin provides the enable signal when SPI Atlas transmissions are active.

5.3 Pinouts for the MC58x20

5.3.1 MC58420 Pinouts

Figure 5-2: MC58420 pinouts



5.3.2 MC58320 Pinouts

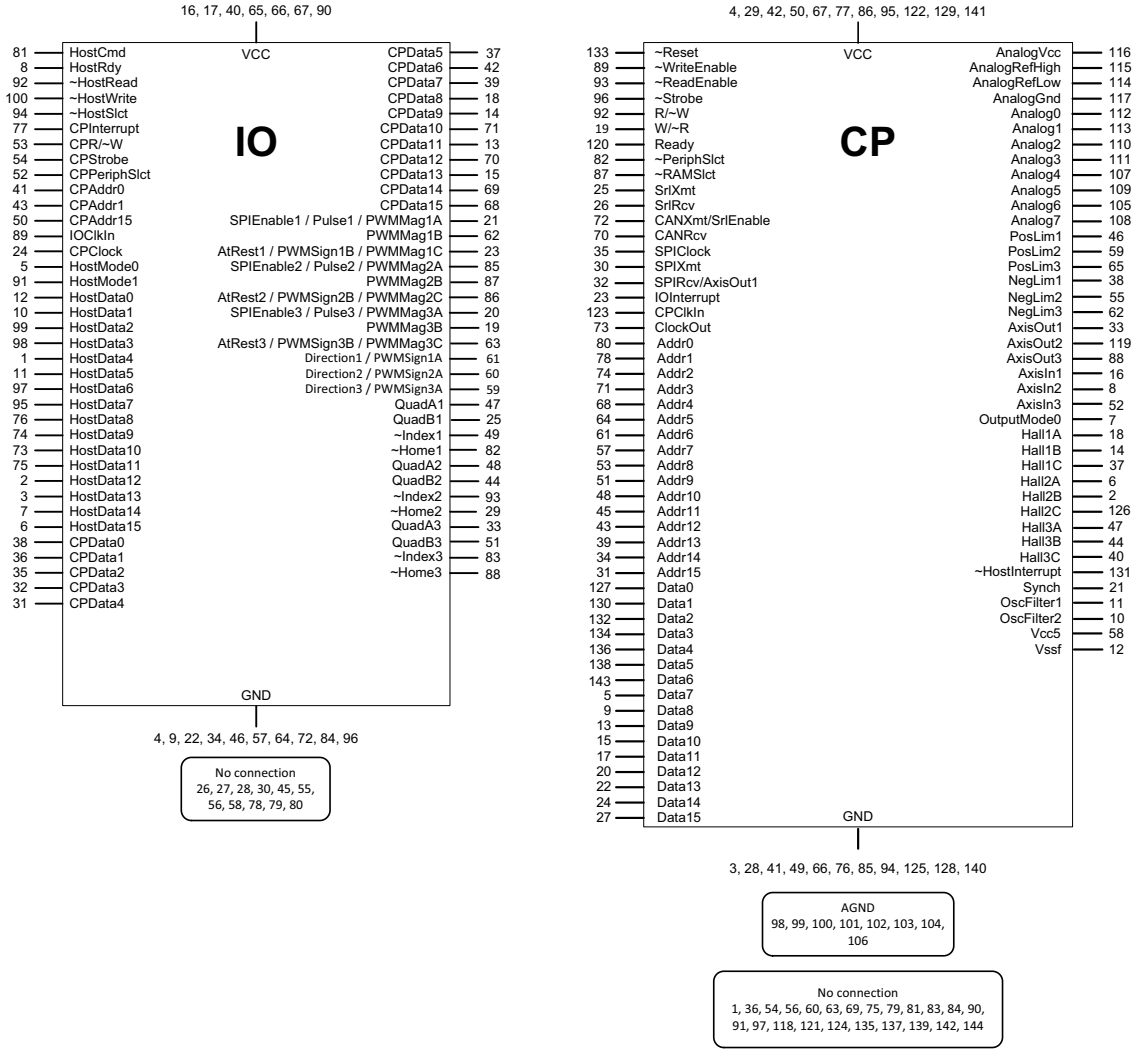
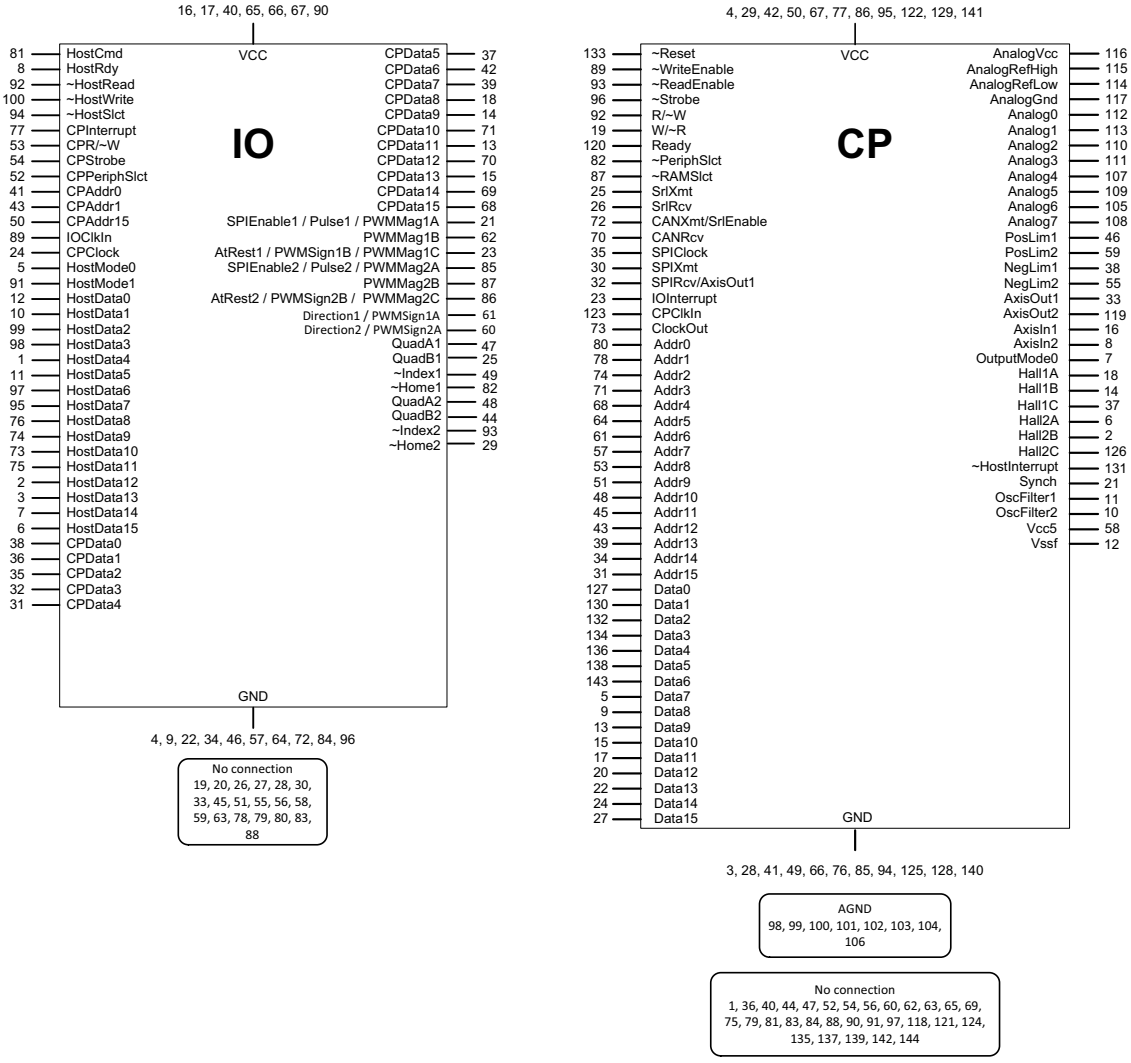


Figure 5-3: MC58320 pinouts

5.3.3 MC58220 Pinouts

Figure 5-4: MC58220 pinouts



5.3.4 MC58120 Pinouts

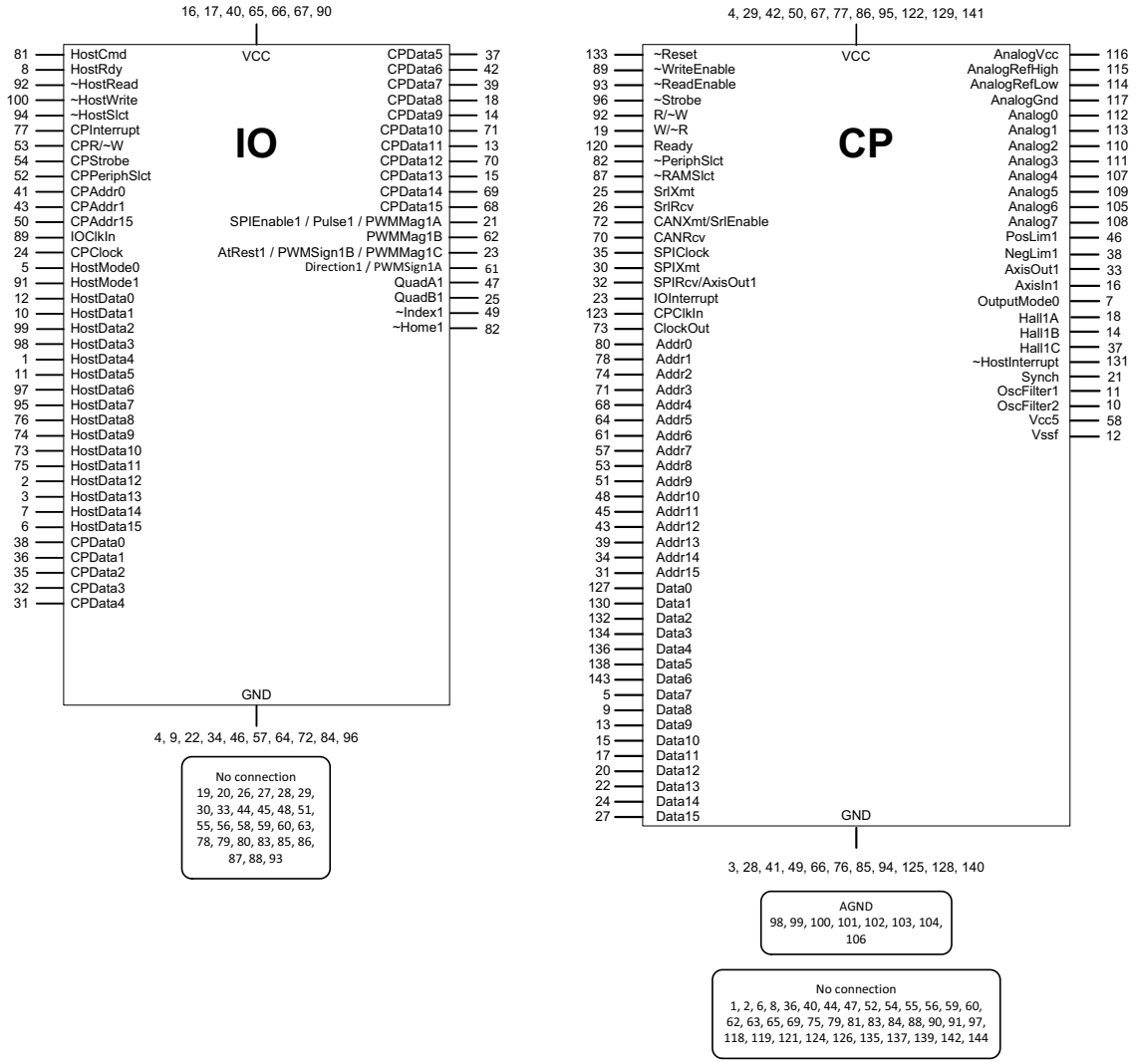


Figure 5-5: MC58120 pinouts

5.4 MC58x20 Pin Descriptions

5.4.1 I/O Chip

MC58x20 IO

Pin Name	Number	Direction	Description
HostCmd	81	input	This signal is asserted *high* to write a host instruction to the motion control IC, or to read the status of the *HostRdy* and *HostInterrupt* signals. It is asserted *low* to read or write a data word.
HostRdy	8	output	This signal is used to synchronize communication between the motion control IC and the host. *HostRdy* (Host Ready) will go *low* indicating host port busy at the end of a read or write operation according to the interface mode in use, as follows: **Interface Mode** — ***HostRdy* goes *low*** 8/16 — after the second byte of the instruction word after the second byte of each data word is transferred 16/16 — after the 16-bit instruction word after each 16-bit data word *HostRdy* will go *high,* indicating that the host port is ready to transmit, when the last transmission has been processed. All host port communications must be made with *HostRdy high* (ready). A typical busy-to-ready cycle is 10 microseconds, but can be significantly longer for certain host command operations.
~HostRead	92	input	When *~HostRead* is *low*, a data word is read from the motion control IC.
~HostWrite	100	input	When *~HostWrite* is *low*, a data word is written to the motion control IC.
~HostSlct	94	input	When *~HostSlct* is *low*, the host port is selected for reading or writing operations.
CPInterrupt	77	output	IO chip to CP chip interrupt. It should be connected to CP chip pin 23, *IOInterrupt*.
CPR/~W	53	input	This signal is *high* when the CP chip is reading data from the IO chip, and *low* when it is writing data. It should be connected to CP chip pin 92, *R/~W*.
CPStrobe	54	input	This signal goes *low* when the data and address become valid during motion control IC communication with peripheral devices on the data bus, such as external memory or a DAC. It should be connected to CP chip pin 96, *~Strobe*.
CPPeriphSlct	52	input	This signal goes *low* when a peripheral device on the data bus is being addressed. It should be connected to CP chip pin 82, *~PeriphSlct.*
CPAddr0 CPAddr1	41 43	input	These signals can be *high* or *low*, and are used when the CP chip is communicating with the IO chip. They should be connected to CP chip pin 80 (*Addr0*), and pin 78 (*Addr1*).
CPAddr15	50	input	This signal is used by the CP chip when communicating with the IO chip. It should be connected to CP chip pin 31 (*Addr15*).
IOClkIn	89	input	This is the master clock signal for the chip set. It is driven at a nominal 40 MHz.
CPClock	24	output	This signal provides the clock pulse for the CP chip. Its frequency is half that of *IOClkIn* (pin 89), or 20 MHz nominal. It is connected directly to the CP chip *IOClock* signal (pin 123).
HostMode0 HostMode1	5 91	input	These two signals determine the host communications mode, as follows: *HostMode1* — *HostMode0* 0 — 0 — 16/16 parallel (16-bit bus, 16-bit instruction) 0 — 1 — *not used* 1 — 0 — 8/16 parallel (8-bit bus, 16-bit instruction) 1 — 1 — Parallel disabled

MC58x20 IO

Pin Name and Number		Direction	Description
HostData0	12	bi-directional, tri-state	These signals transmit data between the host and the motion control IC through the parallel port. Transmission is mediated by the control signals *~HostSelect, ~HostWrite, ~HostRead and HostCmd*. In 16-bit mode, all 16 bits are used (*HostData0-15*). In 8-bit mode, only the low-order 8 bits of data are used (*HostData0-7*). The *HostMode0* and *HostMode1* signals select the communication mode in which this port operates.
HostData1	10		
HostData2	99		
HostData3	98		
HostData4	1		
HostData5	11		
HostData6	97		
HostData7	95		
HostData8	76		
HostData9	74		
HostData10	73		
HostData11	75		
HostData12	2		
HostData13	3		
HostData14	7		
HostData15	6		
CPData0	38	bi-directional	These signals transmit data between the IO chip and pins *Data0-15* of the CP chip.
CPData1	36		
CPData2	35		
CPData3	32		
CPData4	31		
CPData5	37		
CPData6	42		
CPData7	39		
CPData8	18		
CPData9	14		
CPData10	71		
CPData11	13		
CPData12	70		
CPData13	15		
CPData14	69		
CPData15	68		

MC58x20 IO

Pin Name and Number		Direction	Description
SPIEnable1/ Pulse1/ PWMMag1A	21	output	Depending upon the selected motor type and output mode, these signals have the following functions: *PWMMagnA*/n*B*/n*C* signals encode the magnitude of the pulse width modulated output. *PWMSignnA*/n*B*/n*C* signals encode the sign of the pulse width modulated output. *SPIEnable*n provides the enable signal when SPI DAC or SPI Atlas Motor output modes are used. These signals are active *high* for SPI DAC, meaning these signals are high when the SPI DAC channel is being written to, and *low* at all other times. These signals are active low for SPI Atlas mode, meaning they are low when an SPI Atlas communication is occurring, and high at all other times. This default signal sense can be changed using the command **SetSignalSense** by changing the Pulse output bit interpretation. There is one signal per axis. For SPI DAC output these signals can only be used when the axis being controlled is DC brushed, or when the amplifier expects a single-phase input and it performs brushless motor commutation. *Pulse*n provides the pulse (step) signal to the motor. This signal is always a square wave, regardless of the pulse rate. A step occurs when the signal transitions from a *high* state to a *low* state. This default behavior can be changed from a *low* to a *high* state transition using the command **SetSignalSense**. *AtRest*n signal indicates that the axis is at rest, and that the step motor can be switched to low power or standby. A *high* level on this signal indicates the axis is at rest. A *low* signal indicates the axis is in motion. *Direction*n indicates the direction of motion, and works in conjunction with the pulse signal. A *high* level on this signal indicates a positive direction move, and a *low* level indicates a negative direction move. The number of available axes determines which of these signals are valid. Unused or invalid pins should remain unconnected. Refer to the Motor Interfacing section of the *Magellan Motion Control IC User Guide* for more information on PWM encoding schemes.
PWMMag1B	62		
AtRest1/ PWMSign1B/ PWMMag1C	23		
SPIEnable2/ Pulse2/ PWMMag2A	85		
PWMMag2B	87		
AtRest2/ PWMSign2B/ PWMMag2C	86		
SPIEnable3/ Pulse3/ PWMMag3A	20		
PWMMag3B	19		
AtRest3/ PWMSign3B/ PWMMag3C	63		
SPIEnable4/ Pulse4/ PWMMag4A	79		
PWMMag4B	78		
AtRest4/ PWMSign4B/ PWMMag4C	80		
Direction1/ PWMSign1A	61		
Direction2/ PWMSign2A	60		
Direction3/ PWMSign3A	59		
Direction4/ PWMSign4A	26		
QuadA1	47	input	These pins provide the A and B quadrature signals for the incremental encoder for each axis. When the axis is moving in the positive (forward) direction, signal A leads signal B by 90°. **NOTE**: Some encoders require a pull-up to 3.3V resistor on each signal to establish a proper high signal. Check your encoder's electrical specification. The number of available axes determines which of these signals are valid. **WARNING! If a valid axis pin is not used, its signal should be tied high.** Invalid axis pins may remain unconnected, or may be connected to ground.
QuadB1	25		
QuadA2	48		
QuadB2	44		
QuadA3	33		
QuadB3	51		
QuadA4	30		
QuadB4	58		

MC58x20 IO

Pin Name and Number		Direction	Description
~Index1 ~Index2 ~Index3 ~Index4	49 93 83 28	input	These pins provide the Index quadrature signals for the incremental encoders. A valid index pulse is recognized by the chipset when *~Index*, *A*, and *B* are all *low*. The number of available axes determines which of these signals are valid. **WARNING! If a valid axis pin is not used, its signal should be tied high.** Invalid axis pins may remain unconnected, or may be connected to ground.
~Home1 ~Home2 ~Home3 ~Home4	82 29 88 45	input	These pins provide the home signals, which are the general-purpose inputs to the position-capture mechanism. A valid home signal is recognized by the chipset when *~Homen* goes *low*. These signals are similar to *~Index*, but are not gated by the A and B encoder channels. The number of available axes determines which of these signals are valid. **WARNING! If a valid axis pin is not used, its signal should be tied high.** Invalid axis pins may remain unconnected, or may be connected to ground.
Vcc	16, 17, 40, 65, 66, 67, 90		All of these pins must be connected to the IO chip digital supply voltage, which should be in the range of 3.0V to 3.6V.
GND	4, 9, 22, 34, 46, 57, 64, 72, 84, 96		IO chip ground. All of these pins must be connected to the digital power supply return.
Not connected	27, 55, 56		These pins must remain unconnected (floating).

5.4.2 CP Chip

MC58x20 CP			
Pin Name and Number		**Direction**	**Description**
~Reset	133	input/output	This is the master reset signal. This pin must be brought *low* to reset the chipset to its initial conditions. **NOTE**: A software reset will momentarily drive this pin *low*. **NOTE**: For additional important information on *Reset* signal management and timing please see Section 6.4.6, "Reset Signal."
~WriteEnable	89	output	This signal is the write-enable strobe. When *low*, this signal indicates that data is being written to the bus.
~ReadEnable	93	output	This signal is the read-enable strobe. When *low*, this signal indicates that data is being read from the bus.
~Strobe	96	output	This signal is *low* when the data and address are valid during CP communications. It should be connected to IO chip pin 54, *CPStrobe*.
R/~W	92	output	This signal is *high* when the CP chip is performing a read, and *low* when it is performing a write. It should be connected to IO chip pin 53, *CPR/~W*.
W/~R	19	output	This signal is the inverse of *R/~W*; it is *high* when *R/~W* is *low*, and vice-versa. For some decode circuits and devices, this is more convenient than *R/~W*.
Ready	120	input	*Ready* can be pulled *low* to add wait states for external accesses. *Ready* indicates that an external device is prepared for a bus transaction to be completed. If the device is not ready, it pulls the *Ready* pin *low*. The motion control IC then waits one cycle and checks *Ready* again. For a timing diagram and example schematic using the *Ready* signal refer to Section 6.8.1, "Slow Asynchronous SRAM." If used, this signal is typically asserted during RAM Read/Write or Peripheral Read/Write operations. However another use of this signal is to hold off on configuration reads done at startup. Keeping *Ready* asserted at the time of *Reset* release and beyond allows synchronization with FPGAs other external logic that may not yet be ready to receive configuration Peripheral reads. This signal may remain unconnected if it is not used.
~PeriphSlct	82	output	This signal is *low* when peripheral devices on the data bus are being addressed. It should be connected to IO chip pin 52, *CPPeriphSlct*.
~RAMSlct	87	output	This signal is *low* when external memory is being accessed.
SrlXmt	25	output	This pin outputs serial data from the asynchronous serial port.
SrlRcv	26	input	This pin inputs serial data to the asynchronous serial port. If not used, this pin should be tied to Vcc.
CANXmt/ SrlEnable	72	output	When the CAN host interface is used, this pin transmits serial data to the CAN transceiver. When the multi-drop serial interface is used, this pin sets the serial port enable line, and the *CANXmt* function is not available. *SrlEnable* is *high* during transmission for the multi-drop protocol, and always *high* during point–point mode.
CANRcv	70	input	This pin receives serial data from the CAN transceiver. If not used, this pin should be tied to Vcc.
SPIClock	35	output	This pin is the clock signal used for strobing synchronous serial data on the SPI bus to DACs or to Atlas amplifiers. This signal is only active when SPI communications are active.
SPIXmt	30	output	This pin transmits synchronous serial data on the SPI bus to DACs or to Atlas amplifiers.
SPIRcv/ AxisOut1	32	input or output	If OutputMode0 is low at powerup this pin serves as the SPIRcv input, used with the SPI Atlas amplifier interface. if OutputMode0 is left floating at power up this pin provides the AxisOut1 output. AxisOut1 can be programmed to track the state of any bit in the status registers. If unused, this pin may be left unconnected.
IOInterrupt	23	input	This interrupt signal is used for IO to CP communication. It should be connected to IO chip pin 77, *CPInterrupt*.

MC58x20 CP			
Pin Name and Number		**Direction**	**Description**
CPClkIn	123	input	This is the CP chip clock signal. It should be connected to IO chip pin 24, *CPClock*.
ClockOut	73	output	This signal is the reference output clock. Its frequency is the same as the *CPClkIn* signal to the IO chip, nominally 40 MHz. ClockOut will not be active when ~Reset is active.
Addr0 Addr1 Addr2 Addr3 Addr4 Addr5 Addr6 Addr7 Addr8 Addr9 Addr10 Addr11 Addr12 Addr13 Addr14 Addr15	80 78 74 71 68 64 61 57 53 51 48 45 43 39 34 31	output	Multi-purpose address lines. These pins comprise the CP chip's external address bus, and are used to select devices for communication over the data bus. *Addr0*, *Addr1*, and *Addr15* are connected to the corresponding *CPAddr* pins on the IO chip, and are used to communicate between the CP and IO chips. Other address pins may be used for DAC output, parallel word input, external memory, or user-defined I/O operations. See Section 6.2, "Peripheral Device Access," for a complete memory map.
Data0 Data1 Data2 Data3 Data4 Data5 Data6 Data7 Data8 Data9 Data10 Data11 Data12 Data13 Data14 Data15	127 130 132 134 136 138 143 5 9 13 15 17 20 22 24 27	bi-directional	Multi-purpose data lines. These pins comprise the CP chip's external data bus, which is used for all communications with the IO chip and peripheral devices such as external memory or DACs. They may also be used for parallel-word input and for user-defined I/O operations.
AnalogVcc	116	input	Analog input Vcc. This pin should be connected to the analog input supply voltage, which must be in the range of 3.0V to 3.6V. If the analog input circuitry is not used, this pin should be tied to V_{cc}.
AnalogRefHigh	115	input	Analog high voltage reference for A/D input. The allowed range is 2V to *AnalogVcc*. Furthermore, the difference between *Vcc* and *AnalogVcc* should not be larger than 0.3V. If the analog input circuitry is not used, this pin should be tied to V_{cc}.
AnalogRefLow	114	input	Analog low voltage reference for A/D input. The allowed range is *AnalogGND* to *AnalogRefHigh*. If the analog input circuitry is not used, this pin should be tied to *GND*.
AnalogGND	117	input	Analog input ground. This pin should be connected to the analog input power supply return. If the analog input circuitry is not used, this pin should be tied to *GND*.

MC58x20 CP			
Pin Name and Number		**Direction**	**Description**
Analog0 Analog1 Analog2 Analog3 Analog4 Analog5 Analog6 Analog7	112 113 110 111 107 109 105 108	input	These signals provide general-purpose analog voltage levels which are sampled by an internal A/D converter. The A/D resolution is 10 bits. The allowed signal input range is *AnalogRefLow* to *AnalogRefHigh*. Any unused pins should be tied to *AnalogGND*. If the analog input circuitry is not used, these pins should be tied to *GND*.
PosLim1 PosLim2 PosLim3 PosLim4	46 59 65 81	input	These signals provide inputs from the positive-side (forward) travel limit switches. On power-up or after reset, these signals default to active *low* interpretation, but the interpretation can be set explicitly using the **SetSignalSense** instruction. The number of available axes determines which of these signals are valid. Invalid or unused pins may remain unconnected.
NegLim1 NegLim2 NegLim3 NegLim4	38 55 62 69	input	These signals provide inputs from the negative-side (reverse) travel limit switches. On power-up or after reset, these signals default to active *low* interpretation, but the interpretation can be set explicitly using the **SetSignalSense** instruction. The number of available axes determines which of these signals are valid. Invalid or unused pins may remain unconnected.
AxisOut1 AxisOut2 AxisOut3 AxisOut4	33 119 88 54	output	Each of these pins can be conditioned to track the state of any bit in the status registers associated with its axis. The number of available axes determines which of these signals are valid. Invalid or unused pins may remain unconnected.
AxisIn1 AxisIn2 AxisIn3 AxisIn4	16 8 52 83	input	These are general-purpose inputs which can also be used as a breakpoint input. The number of available axes determines which of these signals are valid. Invalid or unused pins may remain unconnected.
OutputMode0	7		OutputMode0 should be tied low when SPI Atlas interfacing will be used, and with Magellan ICs revision 3.0 or higher. For Magellan ICs lower than revision 3.0 this pin should be left floating.
Hall1A Hall1B Hall1C Hall2A Hall2B Hall2C Hall3A Hall3B Hall3C Hall4A Hall4B Hall4C	18 14 37 6 2 126 47 44 40 79 75 56	input	Hall sensor inputs. Each set (A, B, and C) of signals encodes six valid states as follows: A on, A and B on, B on, B and C on, C on, C and A on. A sensor is defined as being on when its signal is *high*. The number of available axes determines which of these signals are valid. Invalid or unused pins may remain unconnected.
~HostInterrupt	131	output	When *low*, this signal causes an interrupt to be sent to the host processor.
Synch	21	input/output	This pin is the synchronization signal. When disabled, the pin is configured as an input, and is not used. In the master mode, the pin outputs a synchronization pulse every 51.2 μsec, which can be used by slave nodes to synchronize with the internal chip cycle of the master. In the slave mode, the pin is configured as an input and should be connected to the *Synch* pin on the master. This signal is falling edge triggered. A *high*-to-*low* transition on the pin synchronizes the internal chip cycle to the signal provided by the master node. The slave expects this signal approximately every 51.2 μsec. If this pin is not used, it may remain unconnected.
OscFilter1 OscFilter2	11 10		These signals connect to the external oscillator filter circuitry. Section 6.4.5, "External Oscillator Filter," details the required filter circuitry.

MC58x20 CP		
Pin Name and Number	**Direction**	**Description**
V_{cc5} 58		This signal can be tied to a 5V supply if available. If 5V is not available this signal must be tied to GND. Being tied to GND will not adversely affect the device performance.
V_{ssf} 12		This signal must be tied to V_{cc}. It must also be tied to pin 28 via a bypass capacitor. A ceramic capacitor with a value between 0.1 µF and 0.01 µF should be used.
V_{cc} 4, 29, 42, 50, 67, 77, 86, 95, 122, 129, 141		CP digital supply voltage. All of these pins must be connected to the supply voltage. V_{cc} must be in the range of 3.0V to 3.6V.
GND 3, 28, 41, 49, 66, 76, 85, 94, 125, 128, 140		CP digital supply ground. All of these pins must be connected to the digital power supply return.
AGND 98, 99, 100, 101, 102, 103, 104, 106		These signals must be tied to *AnalogGND*. If the analog input circuitry is not used, these pins must be tied to *GND*.
No connection 1, 36, 60, 63, 84, 90, 91, 97, 118, 121, 124, 135, 137, 139, 142, 144		These signals must remain unconnected.

5.4.3 MC58x20 Chip Pin Assignments for Multiple Motor Types

The MC58x20 chip supports the output of PWM motor commands in sign/magnitude and 50/50 modes. It can also output pulse and direction signals, and supports external analog signal generation via DAC output, both serial DAC and parallel DACs. Finally, it supports direct connection to PMD's Atlas Digital Amplifiers via the SPI Atlas bus signals.

The following section summarizes the pin connections to the MC58x20 that you will use for all motor output modes other than parallel DAC output. For more information on signaling for parallel DAC output mode see Section 6.15, "Brushless DC Motor Control Using High-Precision Parallel DACs."

For axis 1 of the chipset:

If the output mode is set to **PWM sign/magnitude**, the following pinouts should be used.

PWMMag1A PWMMag1B	IO IO	21 62	output	These pins provide the Pulse Width Modulated signal to the motor. In PWM 50/50 mode, this is the only signal. In PWM sign-magnitude mode, this is the magnitude signal.
PWMSign1A PWMSign1B	IO IO	61 23	output	In PWM sign-magnitude mode, these pins provide the sign (direction) of the PWM signal to the motor amplifier.

If the output mode is set to **PWM 50/50**, the following pinouts should be used.

PWMMag1A PWMMag1B PWMMag1C	IO IO IO	21 62 23	output	These pins provide the Pulse Width Modulated signals for each phase to the motor. If the number of phases is two, only phase A and B are valid. If the number of phases is two, phases A, B, and C are valid. The number of phases is set using the motion control IC command **SetMotorType**. In PWM 50/50 mode, these are the only signals.

If the output mode is set to **Pulse and direction**, the following pinouts should be used.

Pulse1	IO	21	output	This pin provides the pulse (step) signal to the motor.
Direction1	IO	61	output	This pin indicates the direction of motion, and works in conjunction with the pulse signal.
AtRest1	IO	23	output	This signal indicates the axis is at rest, and that the step motor can be switched to low power or standby.

If the output mode is set to **SPI DAC**, the following pinouts should be used.

SPIEnable1	IO	21	output	This pin provides the enable signal when SPI DAC output is active.
SPIClock	CP	35	output	This pin provides the SPI Clock signal.
SPIXmt	CP	30	output	This pin holds the transmitted SPI data.

If the output mode is set for **SPI Atlas**, the following pinouts should be used.

OutputMode0	CP	7	input	This pin should be tied to ground.
SPIEnable1	IO	21	output	This pin provides the enable signal when SPI Atlas transmissions are active.
SPIClock	CP	35	output	This pin provides the SPI Clock signal.
SPIXmt	CP	30	output	This pin holds the transmitted SPI data sent to Atlas.
SPIRcv	CP	32	input	This pins holds the received SPI data sent by Atlas.

For axis 2 of the chipset:

If the output mode is set to **PWM sign/magnitude**, the following pinouts should be used.

PWMMag2A PWMMag2B	IO IO	85 87	output	These pins provide the Pulse Width Modulated signal to the motor. In PWM 50/50 mode, this is the only signal. In PWM sign-magnitude mode, this is the magnitude signal.
PWMSign2A PWMSign2B	IO IO	60 86	output	In PWM sign-magnitude mode, these pins provide the sign (direction) of the PWM signal to the motor amplifier.

If the output mode is set to **PWM 50/50**, the following pinouts should be used.

PWMMag2A PWMMag2B PWMMag2C	IO IO IO	85 87 86	output	These pins provide the Pulse Width Modulated signals for each phase to the motor. If the number of phases is two, only phase A and B are valid. If the number of phases is three, phases A, B, and C are valid. The number of phases is set using the motion control IC command **SetMotorType**. In PWM 50/50 mode, these are the only signals.

If the output mode is set to **Pulse and direction**, the following pinouts should be used.

Pulse2	IO	85	output	This pin provides the pulse (step) signal to the motor.
Direction2	IO	60	output	This pin indicates the direction of motion, and works in conjunction with the pulse signal.
AtRest2	IO	86	output	This signal indicates that the axis is at rest, and that the step motor can be switched to low power or standby.

If the output mode is set to **SPI DAC**, the following pinouts should be used.

SPIEnable2	IO	85	output	This pin provides the enable signal when SPI DAC output is active.
SPIClock	CP	35	output	This pin provides the SPI Clock signal.
SPIXmt	CP	30	output	This pin holds the transmitted SPI data.

If the output mode is set for **SPI Atlas**, the following pinouts should be used.

OutputMode0	CP	7	input	This pin should be tied to ground.
SPIEnable2	IO	85	output	This pin provides the enable signal when SPI Atlas transmissions are active.
SPIClock	CP	35	output	This pin provides the SPI Clock signal.
SPIXmt	CP	30	output	This pin holds the transmitted SPI data sent to Atlas.
SPIRcv	CP	32	input	This pins holds the received SPI data sent by Atlas.

For axis 3 of the chipset:

If the output mode is set to **PWM sign/magnitude**, the following pinouts should be used.

PWMMag3A PWMMag3B	IO IO	20 19	output	These pins provide the Pulse Width Modulated signal to the motor. In PWM 50/50 mode, this is the only signal. In PWM sign-magnitude mode, this is the magnitude signal.
PWMSign3A PWMSign3B	IO IO	59 63	output	In PWM sign-magnitude mode, these pins provide the sign (direction) of the PWM signal to the motor amplifier.

If the output mode is set to **PWM 50/50**, the following pinouts should be used.

PWMMag3A PWMMag3B PWMMag3C	IO IO IO	20 19 63	output	These pins provide the Pulse Width Modulated signals for each phase to the motor. If the number of phases is two, only phase A and B are valid. If the number of phases is three, phases A, B, and C are valid. The number of phases is set using the motion control IC command **SetMotorType**. In PWM 50/50 mode, these are the only signals.

If the output mode is set to **Pulse and direction**, the following pinouts should be used.

Pulse3	IO	20	output	This pin provides the pulse (step) signal to the motor.
Direction3	IO	59	output	This pin indicates the direction of motion, and works in conjunction with the pulse signal.
AtRest3	IO	63	output	This signal indicates the axis is at rest, and that the step motor can be switched to low power or standby.

If the output mode is set to **SPI DAC**, the following pinouts should be used.

SPIEnable3	IO	20	output	This pin provides the enable signal when SPI DAC output is active.
SPIClock	CP	35	output	This pin provides the SPI Clock signal.
SPIXmt	CP	30	output	This pin holds the transmitted SPI data.

If the output mode is set for **SPI Atlas**, the following pinouts should be used.

OutputMode0	CP	7	input	This pin should be tied to ground.
SPIEnable3	IO	20	output	This pin provides the enable signal when SPI Atlas transmissions are active.
SPIClock	CP	35	output	This pin provides the SPI Clock signal.
SPIXmt	CP	30	output	This pin holds the transmitted SPI data sent to Atlas.
SPIRcv	CP	32	input	This pins holds the received SPI data sent by Atlas.

For axis 4 of the chipset:

If the output mode is set to **PWM sign/magnitude**, the following pinouts should be used.

PWMMag4A PWMMag4B	IO IO	79 78	output	These pins provide the Pulse Width Modulated signal to the motor. In PWM 50/50 mode, this is the only signal. In PWM sign-magnitude mode, this is the magnitude signal.
PWMSign4A PWMSign4B	IO IO	26 80	output	In PWM sign-magnitude mode, these pins provide the sign (direction) of the PWM signal to the motor amplifier.

If the output mode is set to **PWM 50/50**, the following pinouts should be used.

PWMMag4A PWMMag4B PWMMag4C	IO IO IO	79 78 80	output	These pins provide the Pulse Width Modulated signals for each phase to the motor. If the number of phases is two, only phase A and B are valid. If the number of phases is three, phases A, B, and C are valid. The number of phases is set using the motion control IC command **SetMotorType**. In PWM 50/50 mode, these are the only signals.

If the output mode is set to **Pulse and direction**, the following pinouts should be used.

Pulse4	IO	79	output	This pin provides the pulse (step) signal to the motor.
Direction4	IO	26	output	This pin indicates the direction of motion, and works in conjunction with the pulse signal.
AtRest4	IO	80	output	This signal indicates that the axis is at rest, and that the step motor can be switched to low power or standby.

Any unused pins may remain unconnected (floating).

If the output mode is set to **SPI DAC**, the following pinouts should be used.

SPIEnable4	IO	79	output	This pin provides the enable signal when SPI DAC output is active.
SPIClock	CP	35	output	This pin provides the SPI Clock signal.
SPIXmt	CP	30	output	This pin holds the transmitted SPI data.

If the output mode is set for **SPI Atlas**, the following pinouts should be used.

OutputMode0	CP	7	input	This pin should be tied to ground.
SPIEnable4	IO	79	output	This pin provides the enable signal when SPI Atlas transmissions are active.
SPIClock	CP	35	output	This pin provides the SPI Clock signal.
SPIXmt	CP	30	output	This pin holds the transmitted SPI data sent to Atlas.
SPIRcv	CP	32	input	This pins holds the received SPI data sent by Atlas.

6. Application Notes — MC58110 and MC58x20

In This Chapter

- General Design Notes
- Peripheral Device Access
- Power Supplies
- Clock Generator, Grounding and Decoupling, and Device Reset
- Serial Communication Interface (SCI)
- CAN Communication Interface
- External Memory
- Asynchronous SRAM
- Using the On-chip ADC
- User I/O Space
- Parallel Word Position Input
- Parallel Communication Interface
- Overcurrent and Emergency Braking Circuits for Motor Drivers
- DC Brush Motor Control Using SPI Interfaced DACs
- Brushless DC Motor Control Using High-Precision Parallel DACs
- Single-Axis Magellan with Brushless DC Atlas
- Multi-Axis Magellan with DC Brush & Step Motor Atlas
- Pulse & Direction Mode Output Connected to Atlas
- Using PWM for DC Brush, Brushless DC and Microstepping Motors
- Using the Allegro A3977 to Drive Microstepping Motors

6.1 General Design Notes

Logic functions presented in the example schematics are implemented by standard logic gates. In cases where specific parameters are of significance (propagation delay, voltage levels, etc…) a recommended part number is given.

One important point of note is that the single and dual chip configurations of Magellan share several signals with the same name that reside on physically different chips. For example, on the MC58x20 the PWM signals are located on the IO chip while on the MC58110 the PWM signals are located on the CP chip. As such, care should be taken to ensure that during the initial schematic layout that the correct CP chip is selected.

In the schematics, pins with multiple functions are referenced by the name corresponding to the specified functionality. For example, pin 54 on the MC58110 CP is named "Direction1 / SPIEnable1 / PWMSign1A" but will be referenced by the name "SPIEnable1" in the SPI DAC example and "PWMSign1A" in the DC brush motor schematics.

The schematic designs presented in this chapter are accurate to the best of PMD's knowledge. They are intended for reference only and have not all been tested in hardware implementations.

6.1.1 Interfacing to Other Logic Families

When integrating different logic families, consideration should be given to timing, logic level compatibility, and output drive capabilities. The Magellan CP and IO chips are 3.3V CMOS input/output compatible and cannot be directly interfaced to 5V CMOS components. In order to drive a 5V CMOS device, level shifters from the 5V CMOS AHCT (or the slower HCT) families can be used. When using a 5V CMOS component to drive the CP, a voltage divider may be used or a member from the CMOS 3.3V LVT family may serve as a level shifter.

6.1.2 Serial Interface

The serial interface is a convenient interface which can be used before host software has been written to communicate through the parallel interface. It is recommended that even if the serial interface is not utilized as a standard communication interface, that the serial receive and transmit signals are brought to test points so that they may be connected during initial board configuration/debugging. This is especially important during the prototype phase. The serial receive line should include a pull-up resistor to avoid spurious interrupts when it is not connected to a transceiver.

If the serial configuration decode logic is not implemented, and the serial interface is used for debugging as previously mentioned, the CP data bus should be tied high. This places the serial interface in a default configuration of 57,600, n, 8, 1 after power on or reset.

6.1.3 Controlling PWM Output During Reset

When the motion control IC is in a reset state (when the reset line is held low), or immediately after a power on, the PWM outputs can be in an unknown state, causing undesirable motor movement. It is recommended that the enable line of any motor amplifier be held in a disabled state by the host processor or some logic circuitry until communication to the motion control IC is established. This can be in the form of a delay circuit on the amplifier enable line after power up, or the enable line can be ANDed with the CP reset line.

6.1.4 Using a Non-standard System Clock Frequency

It is often desirable to share a common clock among several components in a design. In the case of the MC58000 ICs, it is possible to use a clock below the standard value of 40 MHz. In this case, all system frequencies will be reduced as a fraction of the input clock versus the standard 40 MHz clock. The following list details the affected system parameters.

- Serial baud rate
- PWM carrier frequency
- Cycle time
- Commutation rate

For example, if an input clock of 34 MHz is used with a serial baud rate of 9600, the following timing changes will result.

- Serial baud rate decreases to 9600 bps *34/40 = 8160 bps
- PWM frequency decreases to 20 kHz *34/40 = 17 kHz
- Total cycle time increases by a factor of 40/34
- The commutation rate for brushless axes decreases to 10 kHz *34/40 = 8.5 kHz

6.2 Peripheral Device Access

Device addresses on the CP chip's external bus are memory-mapped to the following locations.

Address	Device	Description
0100h	Motor type configuration	Contains the configuration data for the per axis motor type selection
0200h	Serial port configuration	Contains the configuration data (transmission rate, parity, stop bits, etc.) for the asynchronous serial port
0400h	CAN port configuration	Contains the configuration data (baud rate and node ID) for the CAN controller
0800h	Parallel-word encoder	Base address for parallel-word feedback devices
1000h	User-defined	Base address for user-defined I/O devices
2000h	RAM page pointer	Page pointer to external memory
4000h	Motor-output DACs	Base address for motor-output D/A converters
8000h	*reserved*	

6.2.1 Device Initialization and Configuration

Following a hardware or software reset, the motion control IC reads from three external configuration registers to determine the desired settings for the motor type, serial communication, and CAN communication. These reads take place sequentially using the peripheral address read. The timing for this read is shown in Figure 4-1.

6.2.1.1 Motor Type Configuration

When the motor type configuration is read, the 16-bit word is interpreted according to the following table.

Bit Number	Name	Instance	Encoding
0-2	axis1	Brushless DC (3 phase)	0
		Closed loop stepper, 2-phase	1
		Microstepper (3 phase)	2
		Microstepper (2 phase)	3
		Pulse & Direction	4
		reserved	5-6
		DC Brush	7
3	*reserved*	Zero	0
4-6	axis2	Brushless DC (3 phase)	0
		Closed loop stepper, 2-phase	1
		Microstepper (3 phase)	2
		Microstepper (2 phase)	3
		Pulse & Direction	4
		reserved	5-6
		DC Brush	7
7	*reserved*	Zero	0

Bit Number	Name	Instance	Encoding
8-10	axis3	Brushless DC (3 phase)	0
		Closed loop stepper, 2-phase	1
		Microstepper (3 phase)	2
		Microstepper (2 phase)	3
		Pulse & Direction	4
		reserved	5-6
		DC Brush	7
11	*reserved*	Zero	0
12-14	axis4	Brushless DC (3 phase)	0
		Closed loop stepper, 2-phase	1
		Microstepper (3 phase)	2
		Microstepper (2 phase)	3
		Pulse & Direction	4
		reserved	5-6
		DC Brush	7
15	*reserved*	Zero	0

6.2.1.2 Serial Port Configuration

When the serial configuration is read, the 16-bit word is interpreted according to the following table.

Bit Number	Name	Instance	Encoding
0-3	transmission rate	1200 baud	0
		2400	1
		9600	2
		19200	3
		57600	4
		115200	5
		230400	6
		460800	7
4-5	parity	none	0
		odd	1
		even	2
6	stop bits	1	0
		2	1
7-8	protocol	Point-to-point	0
		Multi-drop using idle-line detection	1
		reserved	2
		reserved	3
11-15	multi-drop address	Address 0	0
		Address 1	1
		…	…
		Address 31	31

6.2.1.3 CAN Port Configuration

When the CAN configuration is read, the 16-bit word is interpreted according to the following table.

Bit Number	Name	Instance	Encoding
0-6	nodeID	Address 0	0
		Address 1	1
		…	…
		Address 127	127
7-12	*reserved*	*reserved*	*reserved*

Bit Number	Name	Instance	Encoding
13-15	transmission rate	1,000,000 baud	0
		800,000	1
		500,000	2
		250,000	3
		125,000	4
		50,000	5
		20,000	6
		10,000	7

As an alternative to decoding each configuration address, a special condition occurs when the device powers up and the external bus is pulled high. In this case, the device will read the contents of each configuration register as containing the value 0xffff. When this occurs, the device will configure the serial port as 57,600, n, 8, 1; the CAN port as 20,000 bps with a NodeID of zero; and the motor type will be set to DC Brushed for all available axes.

If the serial or CAN port is not required, the circuitry for decoding the relevant addresses can be omitted, and the *CANRcv* and *SrlRcv* signals can remain disconnected to prevent the chip from responding to either of these communication inputs.

Since CANbus cannot be used when serial multi-drop mode is selected, if using CANbus the serial port configuration register, described in Section 6.2.1.2, "Serial Port Configuration," must have bits 7 and 8 tied low (selecting point to point serial communications).

6.2.2 Parallel-Word Encoder Input

When the parallel-word encoder value is read the word interpretation is as detailed in the table below:

Address	Word
0x810	Upper 16-bits of Axis 1 word
0x811	Lower 16-bits of Axis 1 word
0x812	Upper 16-bits of Axis 2 word
0x813	Lower 16-bits of Axis 2 word
0x814	Upper 16-bits of Axis 3 word
0x815	Lower 16-bits of Axis 3 word
0x816	Upper 16-bits of Axis 4 word
0x817	Lower 16-bits of Axis 4 word

6.2.3 Parallel DAC Output

The following table shows the peripheral address encoding when the parallel DAC mechanism is used to output the motor command:

Address	Command Word	Comment
DC Brush Motors, Other Single-Phase Motors		
0x4000	Axis 1	Single phase output is used with DC Brush motors and Brushless DC motors which are commutated externally
0x4002	Axis 2	
0x4004	Axis 3	
0x4006	Axis 4	
Brushless DC Motors, Other Multi-Phase Motors		
0x4000	Axis 1, Phase A	Multi phase output is used with three-phase Brushless DC motors and two-phase step motors driven in microstepping or closed loop stepper mode
0x4001	Axis 1, Phase B	
0x4002	Axis 2, Phase A	
0x4003	Axis 2, Phase B	
0x4004	Axis 3, Phase A	
0x4005	Axis 3, Phase B	
0x4006	Axis 4, Phase A	
0x4007	Axis 4, Phase B	

6.3 Power Supplies

In the schematic shown in Figure 6-1, the design is powered by an external 5V DC power source. The MC58000 device requires a 3.3V supply and an optional 5V input. The 5V input to the motion control IC can be omitted if 5V is not required elsewhere in the design. The 3.3V digital supply, VCC, is generated by an LT1086-3.3, a 1.5Amp fixed 3.3V low-dropout voltage regulator. Components with a larger power capacity are also available, such as the LT1085-3.3.

If the CP's analog-to-digital converter (ADC) is used, it should be supplied with a filtered 3.3V supply. The +3.3Vs supply is a filtered version of the VCC supply, which is used to supply the ADC and its related conditioning circuitry. The extra filtering is used to provide additional decoupling of the analog elements from the digital elements in the circuitry.

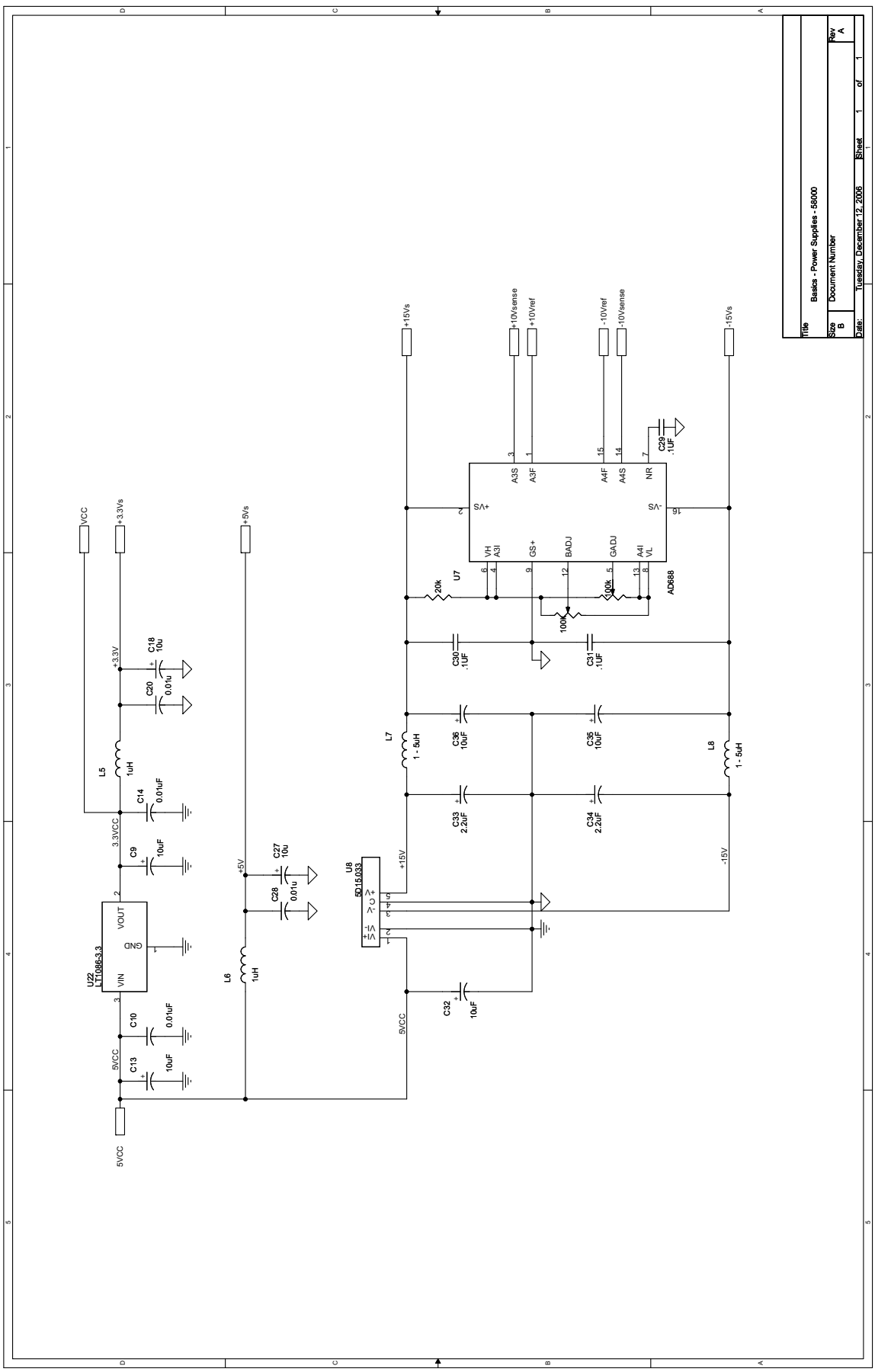
The following is the list of supplies which are referenced in the example schematics:

- +5Vs: a filtered version of 5VCC. This is used for analog components requiring +5V supplies. The extra filtering is used to reduce the voltage ripple, and to generate additional decoupling of the analog elements from the digital elements in the circuitry.
- ±15Vs: ±15V supplies, used for analog components dealing with large input or output voltage swings; usually when interfacing to motors or sensors. The Calex 5D15.033 is a 1W ±15V DC/DC converter which delivers ±33 mA. Depending on the current load of the final design, a larger power capacity DC/DC converter may be required. An LC filter is used to reduce the voltage ripple.
- ±10Vref: two high-precision ±10V reference voltages for driving precise motor commands when using high-precision DACs. The AD688 offers 12-bit absolute accuracy, which may be increased by additional trimming circuitry. The AD688 also offers low noise and a low temperature coefficient, and as such is well suited for the high-precision DAC example.

Notes:

- The schematic in Figure 6-1 should be used for reference only. The actual supplies used should be designed according to the stability and precision requirements of the application. The power supplies presented here are only designed to meet the requirements of the example schematics.
- Power supplies for the motor drivers are not shown. Care should be taken when designing these power supplies, as they should be capable of sinking high switching currents.

Figure 6-1: Basics, power supplies, 58000



6.4 Clock Generator, Grounding and Decoupling, and Device Reset

6.4.1 Clock Generator — MC58110

The nominal clock frequency of the MC58110 CP is 20 MHz. A separate 20 MHz clock may be generated for the device and peripherals or a pre-existing derivative of a clock generated on the board may be used. If an existing clock is to be used, ensure that the input voltages and timing requirements of the MC58110 are met and that the correct frequency is generated.

For any frequency in the range, the bypass capacitor (labeled C3 in Figure 6-2 and Figure 6-3) should be between 0.1 and 0.01 µF, ceramic, and it should have private and as short as possible traces to the pins. This method will reduce noise and jitter, and increase isolation.

6.4.2 Clock Generator — MC58x20

The nominal clock frequency of the MC58x20 IO is 40 MHz. The IO chip generates a nominal 20 MHz clock signal for the CP chip by dividing the input frequency by two. When applying a lower clock frequency to the IO chip, the CP external oscillator filter circuit must adhere to the values listed in Section 6.4.5, "External Oscillator Filter."

6.4.3 Grounding and Decoupling

Each component should be decoupled with the use of large capacitors, usually tantalum 6.7-10 µF in parallel, with a set of 10-100 nF ceramic capacitors placed as close as possible between each one of the power supply pins and ground. This general rule applies to all analog and digital components, although in some of the schematics that follow these capacitors are not shown for reasons of brevity. In some cases, especially in analog parts, it may be beneficial to run a separate power line from the power supply to the component in order to prevent power supply fluctuations from impacting low-level signal components.

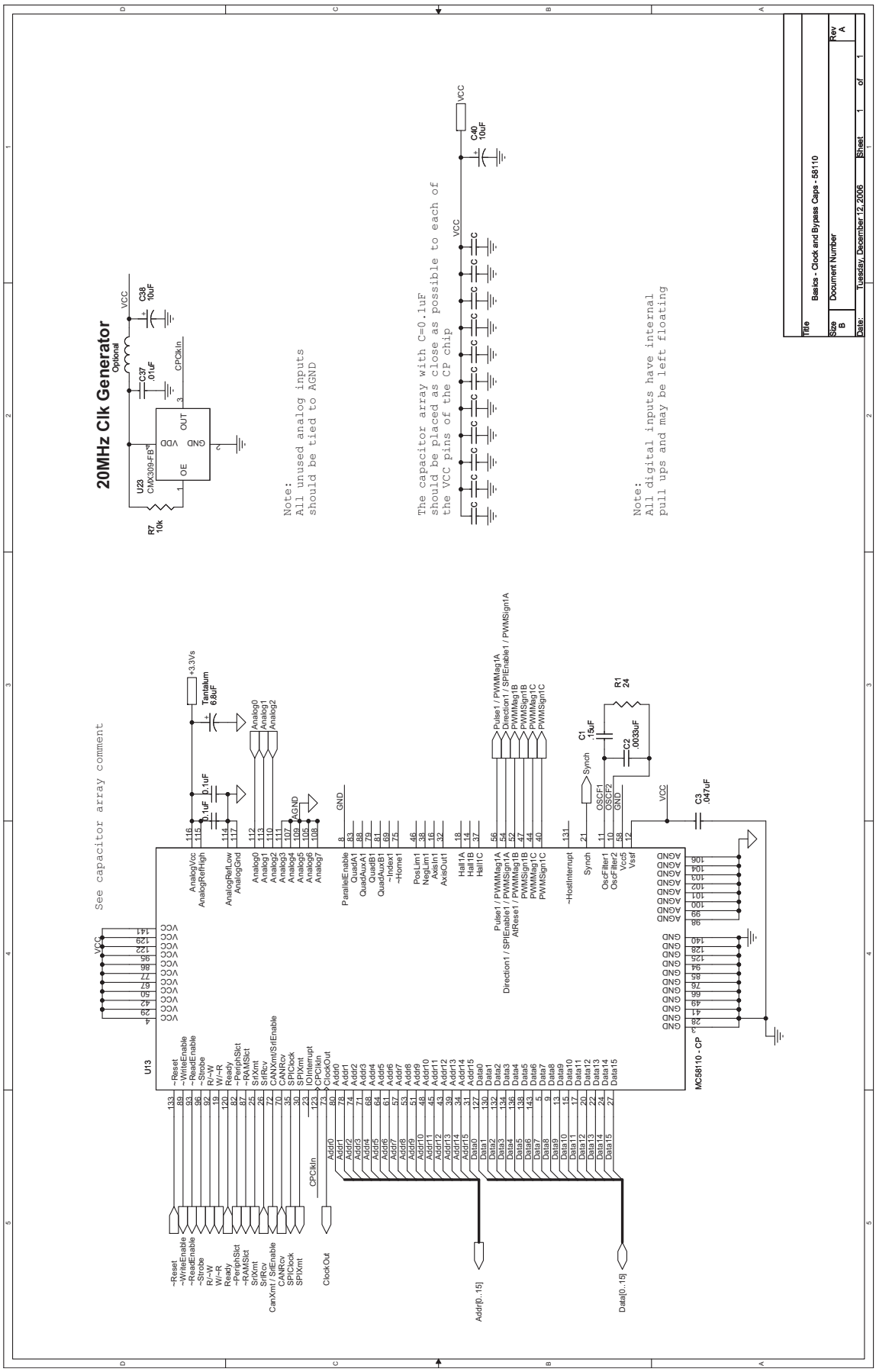
The same points should be considered when designing the ground. The schematics in Figure 6-2 and Figure 6-3 show a star connection at one point in the power supply. Care should be taken to ensure that voltage differences do not accumulate between the grounds, especially in mixed signal components such DACs and ADCs.

Additional isolation, for example ferrite beads, may be inserted between the analog and digital grounds to suppress high frequency ground noise. Some components, such as motor drivers, require special grounding. The system designer should refer to the component data sheets of selected components in order to ensure correct usage of the grounding methods.

6.4.4 Decoupling of the On-chip ADC

The voltage supply to the ADC should be decoupled with the use of a 2.2-6.8 µF tantalum capacitor in parallel with a 0.01-0.1 µF ceramic capacitor placed as close as possible to the power supply and ground pins. For additional isolation purposes, an additional 0.01-0.1 µF ceramic capacitor should be placed across AnalogRefLow and AnalogRefHigh.

Figure 6-2: Basics, clock and bypass caps, 58110



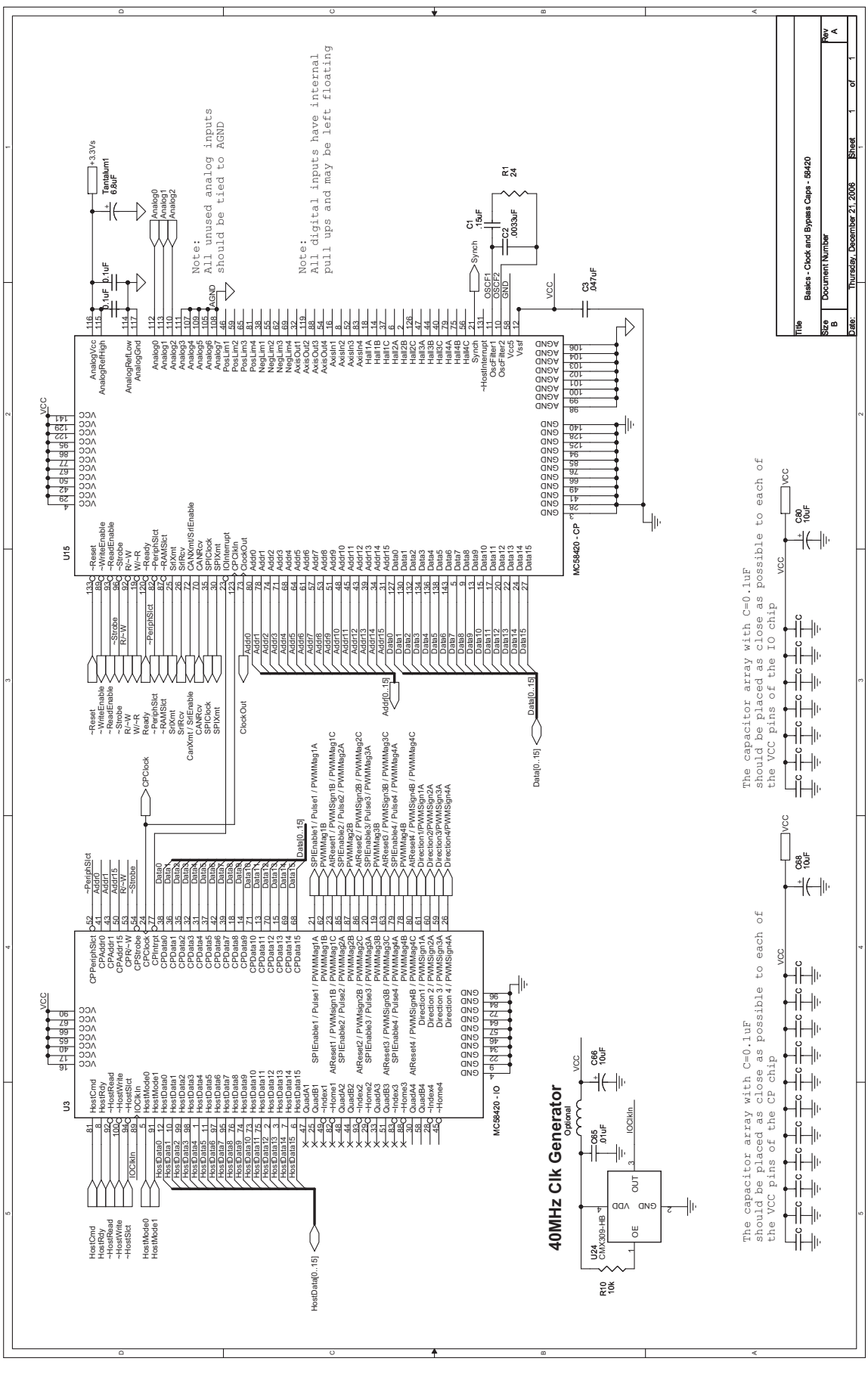
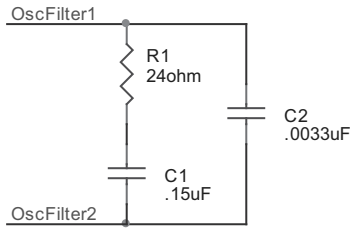


Figure 6-3: Basics, clock and bypass caps, 58420

6.4.5 External Oscillator Filter

The circuit in Figure 6-4 shows the recommended configuration and suggested values for the filter which must be connected to the OscFilter1 and OscFilter2 pins of the CP chip. The resistor tolerance is ±5%, and the capacitor tolerance is ±20%.

Figure 6-4: Oscillator filter circuit



When applying a different clock frequency, the PLL's external loop filter circuit (capacitors C1 and C2, and resistor R1) should be varied as a function of the clock frequency. These reference values are detailed in the following table. C1 and C2 capacitors must be non-polarized.

CPClkIn [MHz]	R1 [Ω] (±5%)	C1 [µF] (±20%)	C2 [µF] (±20%)
5	5.6	2.7	0.056
10	11	0.68	0.015
15	16	0.33	0.0068
20	24	0.15	0.0033

6.4.6 Reset Signal

The CP accepts a reset signal, ***~Reset***, which should go low after power-up or when an external reset event occurs. From the rising edge of this signal, the CP begins an initialization procedure, which is concluded within 1.5 milli-seconds. During this period, the outputs of the CP and IO chips will be in an unknown state. In order to prevent signals in an arbitrary state from driving the motors, a disabling signal, ***~ResetHold***, is generated. The ***~ResetHold*** signal is a ~2.2 msec extension to the active low period of the ***~Reset*** signal, and is used to disable the motor drivers during the initialization period. The ***~ResetHold*** signal is not required if using only Atlas drives.

In addition to the power up delay, the ***~Reset*** signal must stay low (0V) anytime Vcc is below its minimum specification. This is true at all times including power on and power off.

The typical solution implemented to meet this requirement involves the use of a voltage supervisor IC. This IC monitors the Vcc line and forces the ***~Reset*** pin to zero any time Vcc is below specification. Section 3.1, "DC Characteristics for 58110, 58x20 CP," provides details on the specifications for Vcc.

In the schematic shown an optional AND gate is used to allow a microprocessor to control the ***~Reset*** signal when the voltage supervisor is not actively driving it low.

During the initialization period, the CP reads the three external configuration registers to determine the configuration for motor type, serial communication and CAN communication. For the motor type, the CP reads the peripheral address 100h. The circuitry shown in Figure 6-5 includes an example of configuring the motor type with the use of DIPswitches. For configuring the serial and CAN communication refer to Section 6.5.2, "Interfacing to Off-board Hosts Through Asynchronous Serial Communications," and Section 6.6, "CAN Communication Interface."

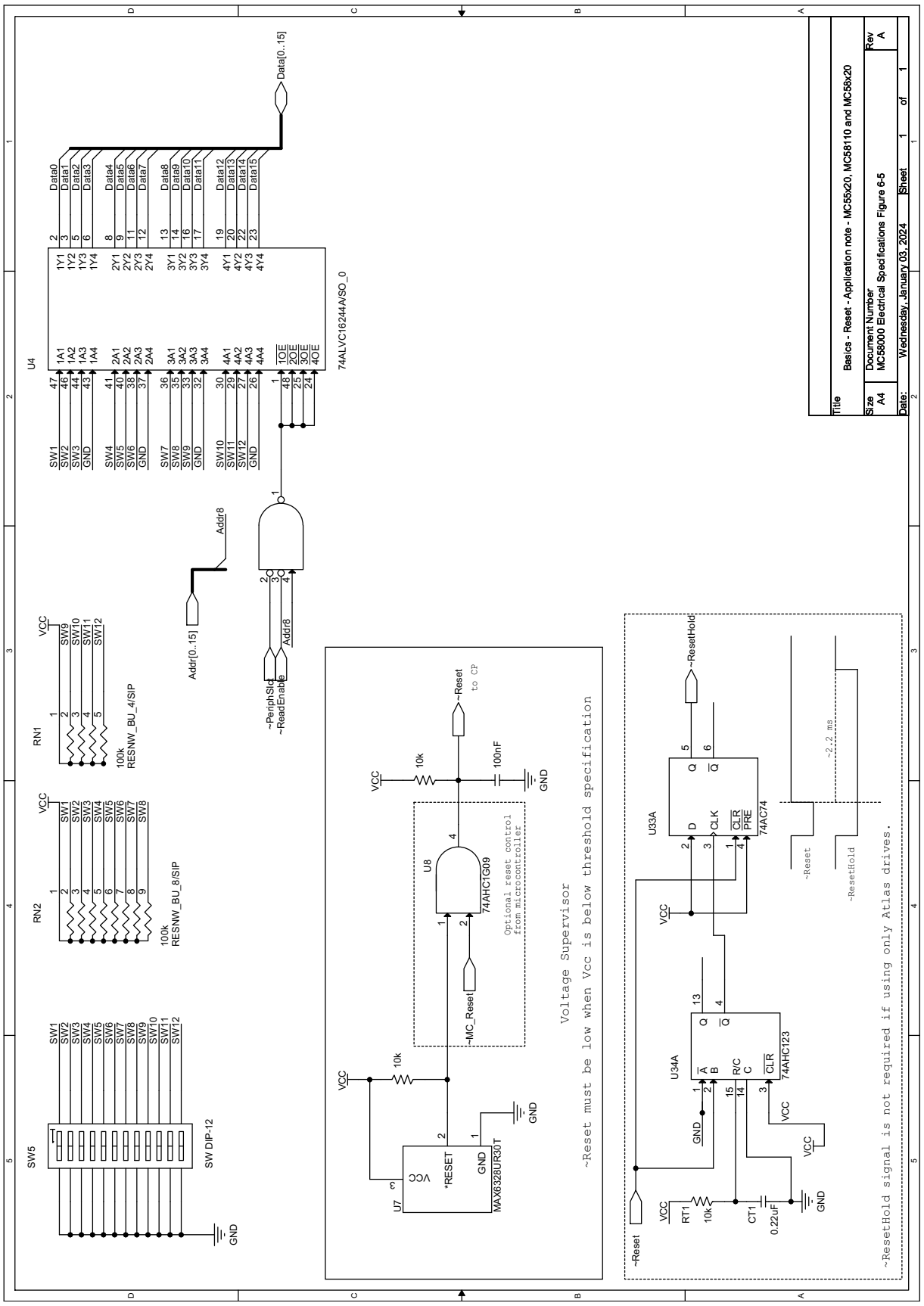


Figure 6-5: Basics, reset

6.5 Serial Communication Interface (SCI)

In this section, the serial communication interface to the host is described. Figure 6-6 shows circuitry used to configure the SCI port on power-up. This circuitry may be omitted if the default configuration values are suitable.

Subsequent sections demonstrate the use of RS-232, RS-422 and RS-485 line-drivers for interfacing to a remote host.

6.5.1 SCI Configuration During Power-up or Reset

On power-up or after a reset, the CP configures the SCI according to the 16-bit value residing at the peripheral address 200h. If a value of FFFFh is read, then the SCI is configured to its default configuration: 57,600 baud, no parity, one stop bit, point-to-point mode. Since the data bus inputs are internally pulled-up, the pins may remain disconnected and the default configuration will take effect at power-up or reset. If a non-default SCI configuration is required the circuitry presented in the reference schematic may be used.

Note that after communication has been established, the SCI configuration may be altered via the SetSerialPortMode command.

The following should be observed when designing the power-up/reset SCI circuitry:

1. The MC58000 peripheral address map is arranged so that Addr9 is dedicated for SCI configuration.
2. The following logic condition for presenting the setup word on the data bus should be used:

   ~SCISetupDataEnable = ~ReadEnable + ~PeriphSlct + ~Addr9

   Where:

~SCISetupDataEnable	When high the tri-state buffer outputs are placed in a high-impedance state.
~ReadEnable	When low the bus is in a read cycle.
~PeriphSlct	When low the peripheral address space is being addressed.

   The logic may be implemented in a PLD, and its propagation delay should not exceed 20 nsec; assuming an enable time for the tri-state buffer of less than 10 nsec.
3. The DIPswitch resistors of ~100K ensure sufficient VIH level. In case of a zero input, a current of ~33 µA will be flowing between VCC and GND. This may result in a worst-case scenario of ~0.55 mA when the all-zero word is encoded.

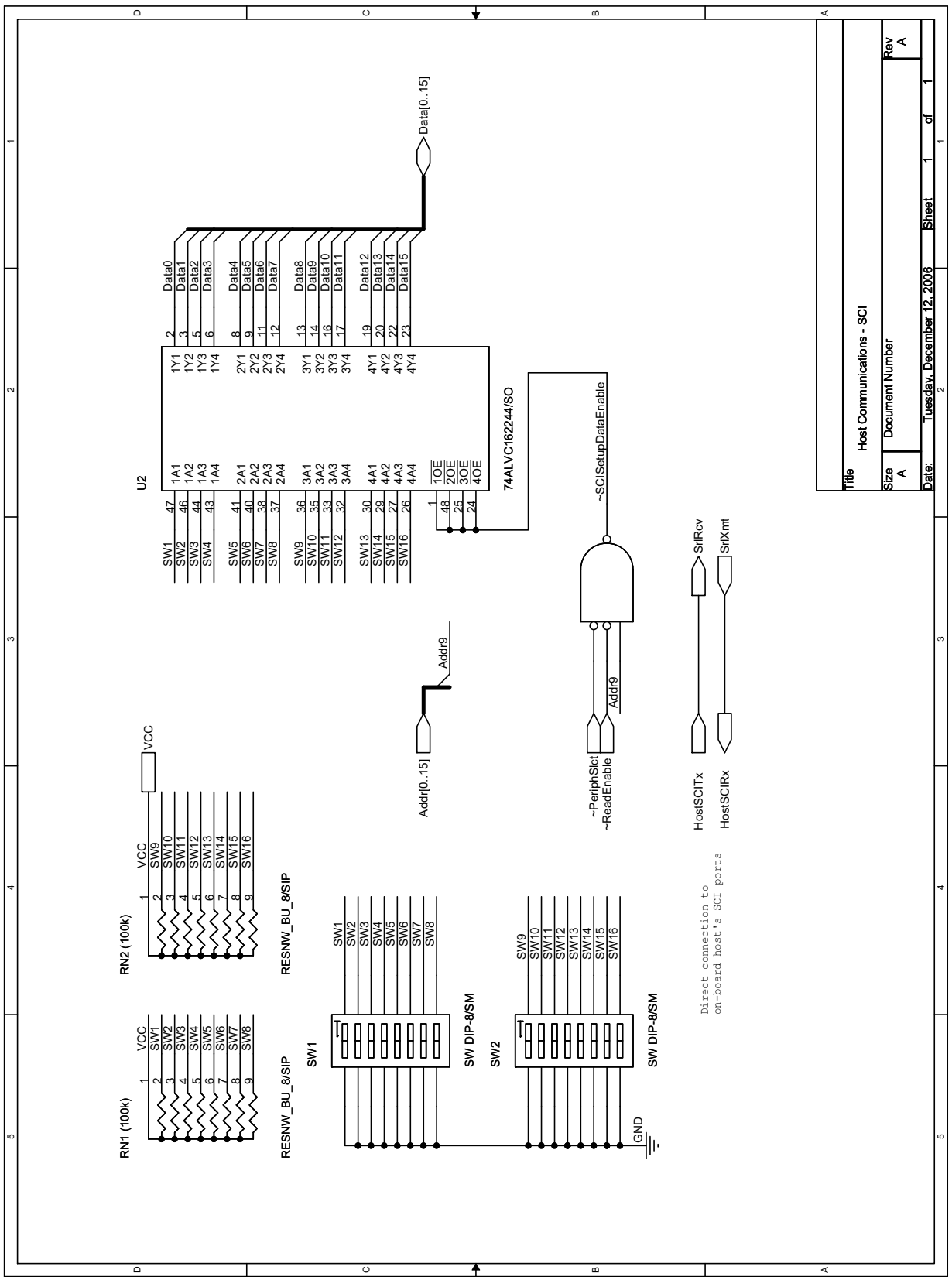


Figure 6-6: Host communication, SCI

6.5.2 Interfacing to Off-board Hosts Through Asynchronous Serial Communications

When the host and motion control IC are located on the same physical board it is most likely that simply wiring the transmit and receive lines directly between the host and CP chip is all that is required (assuming they are both 3.3V CMOS devices). When the host is remote and the interface requires longer communication lines, achieving reliable communication is more involved.

TIA/EIA standards provide reliable communication over varying cable lengths and communication rates. The most commonly used standards are RS-232, RS-422 and RS-485. These standards are separated into two categories: single-ended and differential. RS-232 is a single-ended standard allowing for moderate communication rates over relatively short cables. RS-422 and RS-485 are differential, offering higher data rates and longer cable runs.

Line drivers and receivers (transceivers) are commonly used in order to mediate between the cable interface and the digital circuitry signal levels. Although RS-485 transceivers also support the RS-422 electrical specification (the reverse is not true), there are several design considerations that should be taken into account when deciding which of these two communication methods is the best fit for an application.

- Full-duplex vs. half-duplex

  The terms full-duplex and half-duplex are used to distinguish between a system having two separate physical communications lines from one having one common line for transmission and reception.

- Line contention

  This problem can occur in half-duplex systems. Most line-drivers supply physical protection against such conditions but there is no automatic recovery of lost data in these levels. When interfacing the Magellan to a half-duplex communication system the designer should note that the turn-around time for command processing and response is at least 1 byte at the current baud rate. As a result the host should release the communication line before this time elapses so that contention can be avoided.

- Termination impedance

  Long cables and/or high data rates require termination resistors if the transceiver is located at the end of the transmission lines. One way to determine if termination is required is if the propagation delay across the cable is larger than ten times the signaling transition time. If this condition is satisfied, then termination is required. The RS485 standard specifies the signaling transition time to be less than 0.3 times the signaling period, thus imposing an upper limit on the maximum cable length for a specified baud rate.

  The termination resistor should match the characteristic impedance of the cable with 20% tolerance. Resistors with a value of 80-120Ω are typically used. For RS-422, only the receiver end should use a termination resistor, due to the communication line being unidirectional (full duplex). Note that if the transceiver is not placed at the ends of the cable, no termination resistors are required. However, the stubs should be kept as short as possible to prevent reflections.

The schematic in Figure 6-7 employs the ADM3202 and ADM3491 transceivers as an example of RS232 and RS485/422 interfaces respectively. Other RS232 transceivers may be used, such as Maxim's MAX3321E. The ADM3491 circuitry can be configured for both full-duplex and half-duplex communications, and may include termination resistors. As an alternative, transceivers from the MAX307xE family may be used.

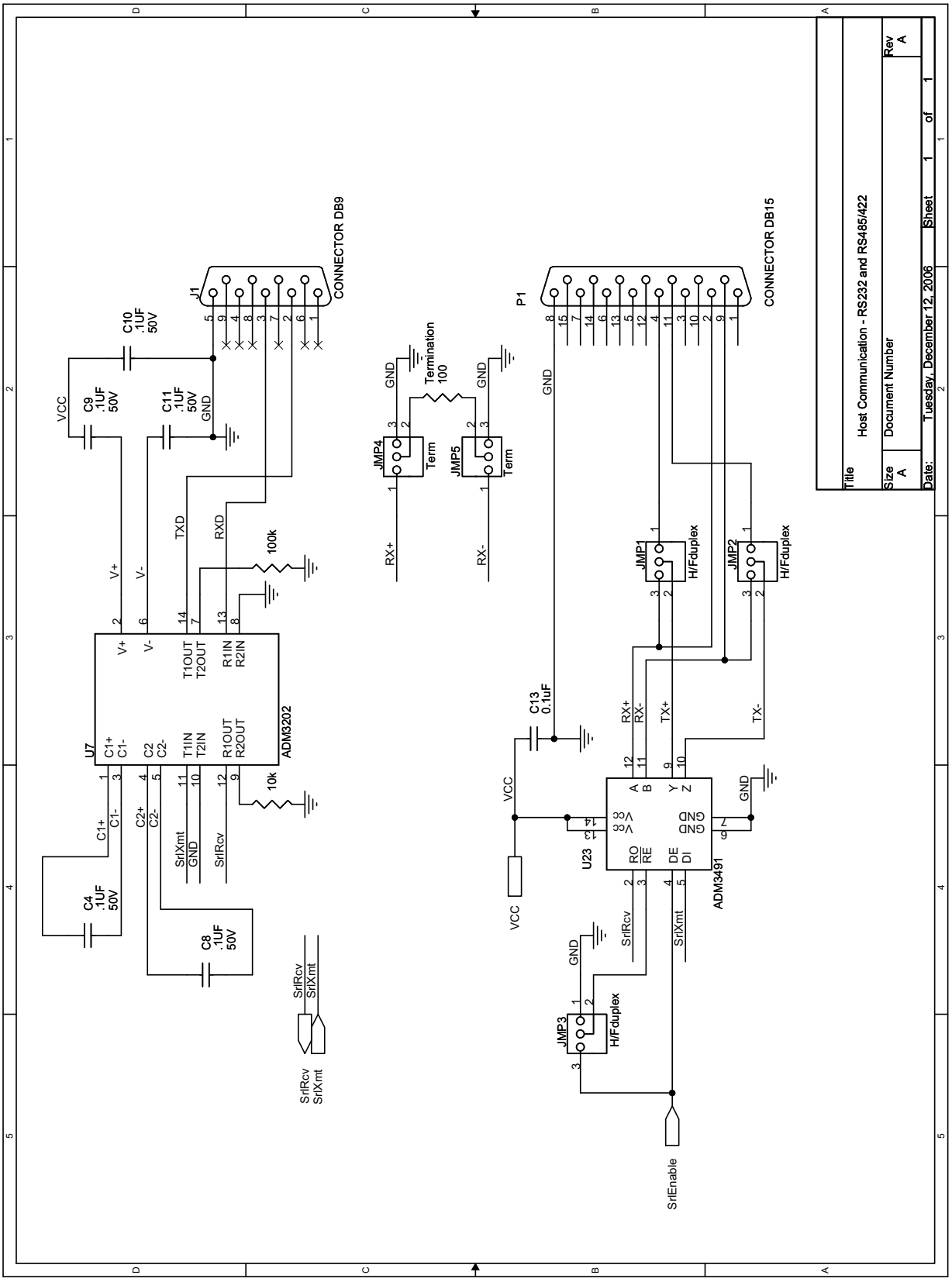
The following table shows configuration options for the RS-485/422 circuitry of Figure 6-7.

Configuration	Jumper Position	Application
Half Duplex[1]	JMP1/2/3 in 2-3	RS-485 in multipoint system
Full Duplex	JMP1/2/3 in 1-2	RS-422 or RS-485 in point to point system

Termination on[2]	JMP4/5 in 1-2	Both RS-485 and RS-422. For high transmission rates and/or long cable. Only when placed at the end of the cable.
Termination off	JMP4/5 in 2-3	Both RS-485 and RS-422. For low transmission rates and short cable. Or when placed at the middle of the cable.

1. JMP3 should only be placed be in the half duplex state (2-3) if multi-point communication is being used.
2. Note that the reference circuitry does not support resistance termination on the transmitting side when operated in full duplex because it is assumed that RS 485 will only be used in the half-duplex configuration.

Figure 6-7: Host communication, RS232 and RS485/422



6.6 CAN Communication Interface

The following example illustrates an interface to a CAN backbone using of TI's SN65HVD232 transceiver, which supports ISO 11898 standard. Generally the CAN high-speed standard ISO 11898 provides a single line structure as network topology. The bus line is terminated at both ends with a single termination resistor of ~120ohms. However in practice some deviation from that topology may be needed to accommodate appropriate drop cable lengths and particular applications. Consult CAN ISO 11898 standard for more information on termination schemes and EMC considerations.

6.6.1 CAN Configuration During Power-up or Reset

On power-up or after reset, the CP configures the CAN controller according to the 16-bit value residing at the peripheral address 400h. If a value of FFFFh is read, then the CAN controller is configured to its default configuration: 20 kbps with a NodeID of 0. Since the data bus inputs are internally pulled-up, the pins may remain disconnected and the default configuration will take effect at power-up or reset. If a non-default CAN configuration is required the circuitry presented in the reference schematic (Figure 6-8) may be used.

Note that after communication has been established, the CAN configuration may be altered via the SetCANMode command.

More advanced CAN bus drivers such as the SN65HVD230 supply a programmable input pin which may be used to adjust the rise and fall times of the transmitter. This may be important in unshielded, low-cost systems in order to reduce electromagnetic interference. The pin may be hard-wired through a resistor to ground (refer to the component data sheet for calculating the resistor's value), or the host may control this pin by introducing additional circuitry attached to the I/O user space of the CP for a more flexible and tunable design.

The following should be observed when designing the power-up/reset CAN circuitry.

1. The MC58000 peripheral address map is arranged so that Addr10 is dedicated for CAN configuration.
2. The following logic condition for presenting the setup word on the data bus should be used.

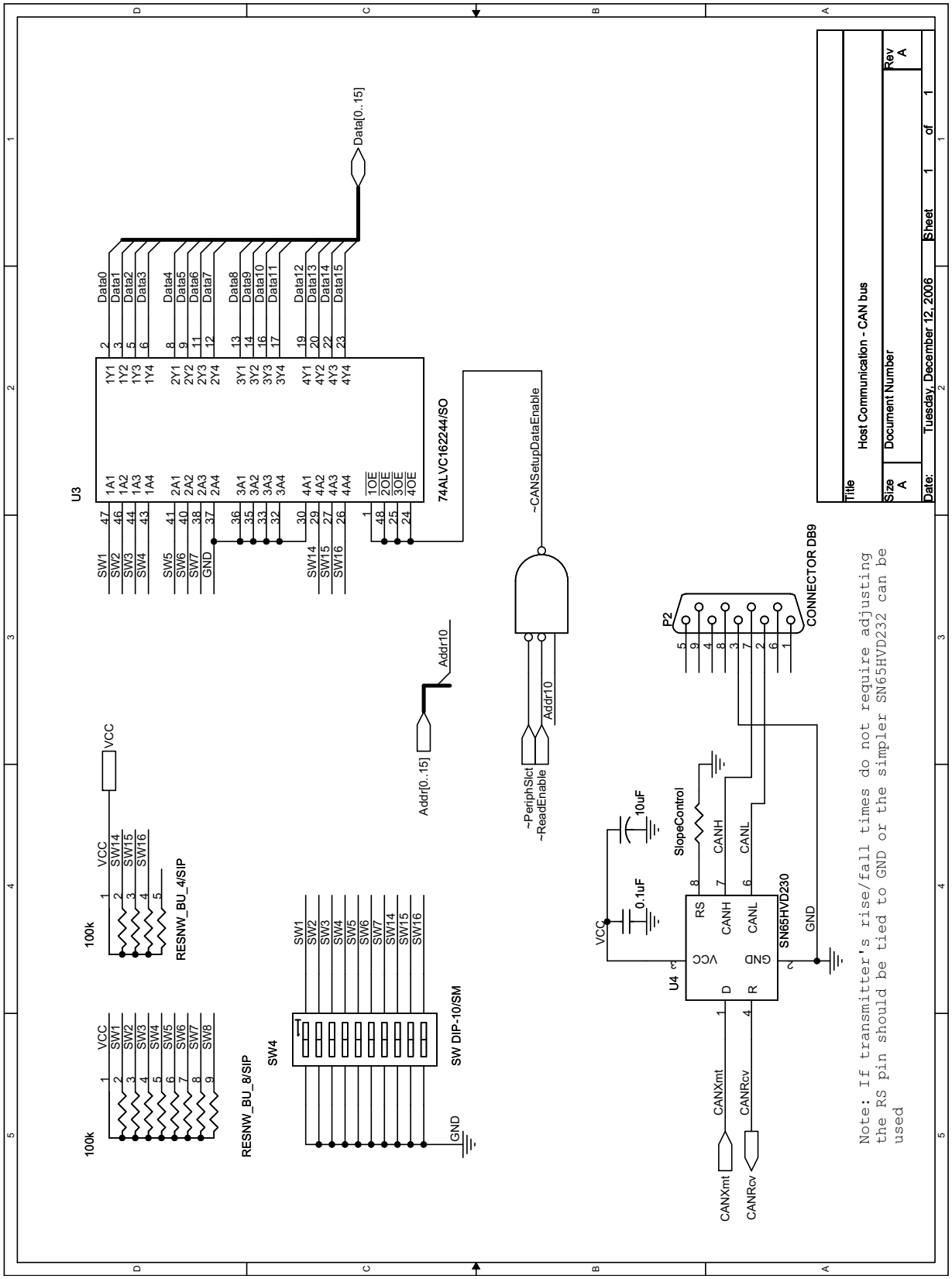
   ~CANSetupDataEnable = ~ReadEnable + ~PeriphSlct + ~Addr10

   Where:

~CANSetupDataEnable	When high the tri-state buffer outputs are placed in a high-impedance state.
~ReadEnable	When low, the bus is in a read cycle.
~PeriphSlct	When low, the peripheral address space is being addressed.

   The logic may be implemented in a PLD, and its propagation delay should not exceed 20 nsec; assuming an enable time for the tri-state buffer of less than 10 nsec.
3. The DIPswitch resistors of ~100K ensure sufficient VIH level. In case of a zero input, a current of ~33 µA will be flowing between VCC and GND. This may result in a worst-case scenario of ~0.55 mA when the all-zero word is encoded.

Figure 6-8: Host communication, CANbus



6.7 External Memory

Utilizing its external bus, the Magellan Motion Control IC can interface with two types of external memory: asynchronous SRAM and synchronous dual port RAM (DPRAM). External memory is used for trace data storage and is optional. SRAM is typically used in designs that do not require real-time access to the data. DPRAM permits high speed downloading of trace data and is most applicable in applications where the data is being downloaded and analyzed on a real-time basis.

6.7.1 CP External Memory Interface

The MC58000 external bus is comprised of the *Addr[0..15]* and *Data[0..15]* signals. The signals *~WriteEnable*, *~ReadEnable*, *~RAMSlct*, *W/~R*, *R/~W* and *~Strobe* are used in conjunction with the address bus signals for controlling access to the attached memory device. The *Ready* input signal may also be used to insert wait-states for accessing slower memory devices. Signal timing information is given in chapters 3 and 4 of this manual.

All signals are time referenced to the *ClockOut* signal, which has a nominal 25 nsec period. The MC58000 can directly access 32Kx16 bits of external memory and uses the 15 least significant bits of the address bus. *Addr15* is not used. Larger external memories may be used by adding a page register, as detailed in the following section.

6.8 Asynchronous SRAM

The following schematic (Figure 6-9) illustrates a pair of IDT71V424SA15 512Kx8 SRAMs with a 15 ns access time interfaced to the MC58000, resulting in a total of 16 pages of storage. Expansion to 32 pages is easily achieved with four IDT71V428 1024Kx8 SRAMs. Memory blocks are accessed with the use of a page register. The IDT71V424SA15 is an asynchronous SRAM which is controlled by three input signals: chip select (~CS), output enable (~OE), and write enable (~WE). The SRAM is interfaced with the MC58000 output signals as shown in the following table.

Device	Signal Name		
MC58000	*~RAMSlct*	*~ReadEnable*	*~WriteEnable*
IDT71V424SA15	*~CS*	*~OE*	*~WE*

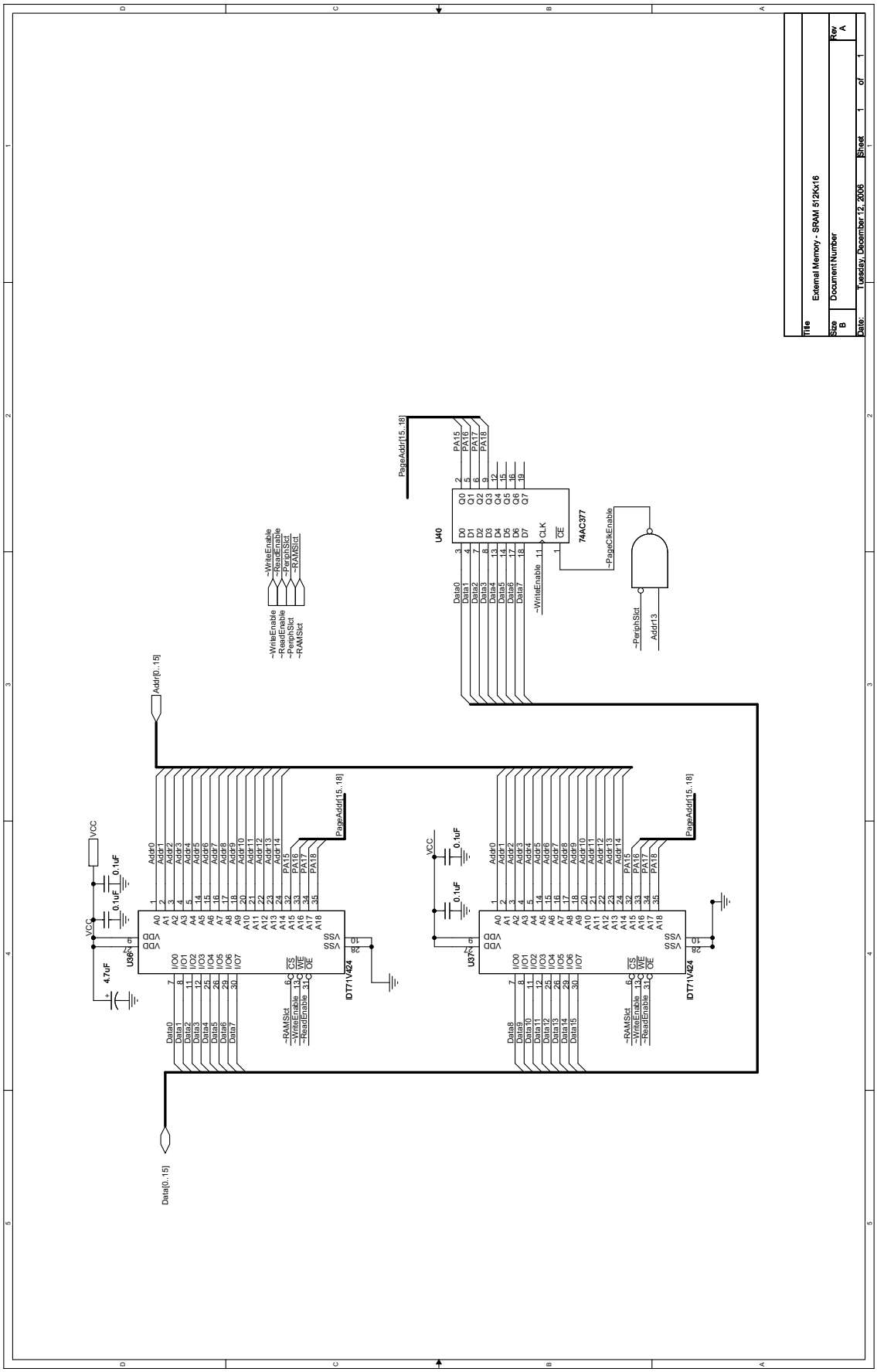
Note that the selected SRAM device should meet the timing requirements of the MC58000 output signals. Usually, asynchronous SRAMs with cycle times of less than 1.5 *ClockOut* cycles will meet this requirement. One example is the CY7C1020CV33-15 32Kx16 SRAM which has a 15 nsec access time. This device may be used to provide one page of storage.

If larger external memory is required, several pages may be addressed (up to 64K pages) by using a page register. The page register may be accessed at the peripheral address 2000h and the *Addr13* signal can be used as a chip-select. The page register operates the extra address lines required to interface with larger external memory devices. Before the external memory write or read cycle, the CP performs a write to the page register to select the required page.

The use of larger SRAM chips is recommended. If the capacity of one SRAM is not adequate, multiple chips may be cascaded. The two common methods for cascading SRAM chips are:

- Using high capacity, lower organization (x8 or x4) chips. In this configuration SRAMs share the same address bus and each SRAM is wired to a different portion of the data bus.
- Using the SRAM's chip-select input(s) as an additional address line. This option requires a decoder to map the address to the appropriate chip select signal. The propagation delay of the decoder must be below 0.5 of the *ClockOut* period. In addition the total access time should not exceed 1.5 *ClockOut* cycles including the SRAM access time.

Figure 6-9: External memory, SRAM 512Kx16



6.8.1 Slow Asynchronous SRAM

If the SRAM does not meet the timing requirements, it may still be connected to the MC58000 by adding wait states. In order to generate the wait-states, the *Ready* signal must be activated during read/write memory accesses. As long as the *Ready* signal is kept low during the rising edge of the *ClockOut* signal, the end of the current read/write cycle will be deferred to the next *ClockOut* rising edge.

The following schematic (Figure 6-10) contains a circuit and timing diagram for generating one wait state. Expanding to two or more wait states is similar. Note that adding wait states slows the operation of the MC58000 and therefore the number of wait states should be kept as low as possible if the device is expected to maintain normal operation. Contact PMD to determine if the use of wait states will affect device operation in your application.

The following timing restrictions apply when the device is operating with the standard 40 MHz *ClockOut* frequency:-

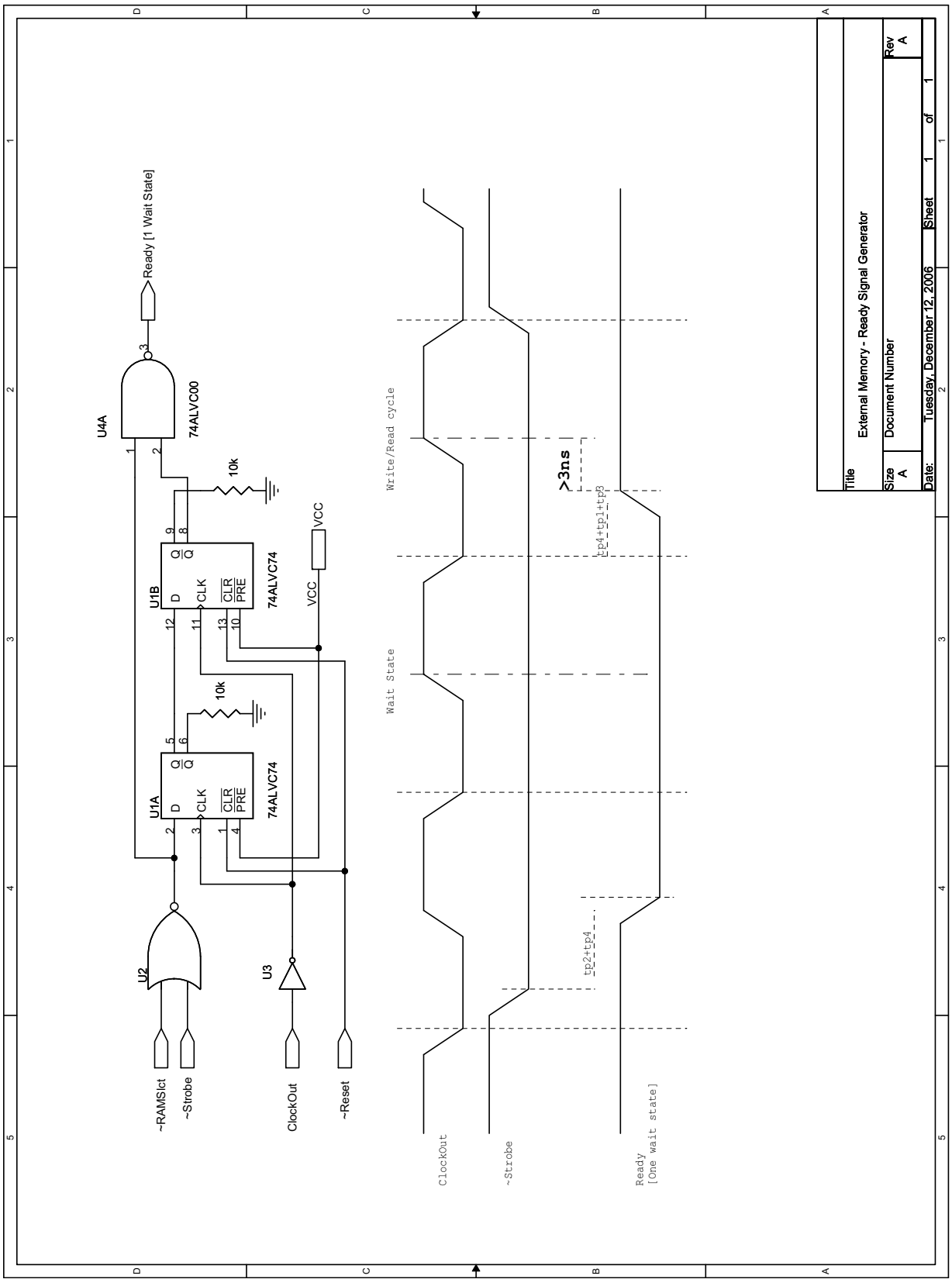
1. tp4 + tp2 < 22 ns
2. tp2+ ts1 < 19 ns
3. tp1 + tp4 + tp3 < 9.5 ns

Where tp1 and ts1 are the propagation delay and setup time of the D flip-flop. TPx is the propagation delay of logic gate Ux.

Notes:

1. In order to meet the above timing requirements high-speed gates may be used or the logic may be implemented in a fast PLD. Timing restriction 3 may be relaxed by the use of fast negative edge JK-FF, such as the 74LCX112, resulting in a very tight constraint on tp4.
2. If read and write cycles do not require the same number of wait states, then either *~WriteEnable* or *~ReadEnable* may replace the *~Strobe* signal. If *~ReadEnable* is used, then restriction 2 becomes more stringent: tp2 + ts1 < 7.5 ns.
3. If there is no need to add wait-states the *Ready* input pin may be left disconnected as it is internally pulled up.

Figure 6-10: External memory, Ready signal generator



6.9 Using the On-chip ADC

In this section two types of conditioning circuits which interface to the on-chip analog-to-digital converter are demonstrated. The first circuit interfaces to a single-ended voltage signal, and the second circuit to a differential voltage signal. The conditioning circuits should be adjusted appropriately in order to meet the system's requirements.

The MC58000 is equipped with an eight-channel 10-bit ADC. The sampling rate of each channel is 57.8K samples per second, and the sample-and-hold time per channel is 1.6 μsec. The sampling capacitor is 30 pF, and the sampling resistor is in the range of 100 - 200Ω. In order to meet the timing requirements, the output impedance of the conditioning circuitry, as seen by the ADC's inputs, should not exceed 6.1 kΩ.

The digital value derived from the input analog voltage is determined using the following formula.

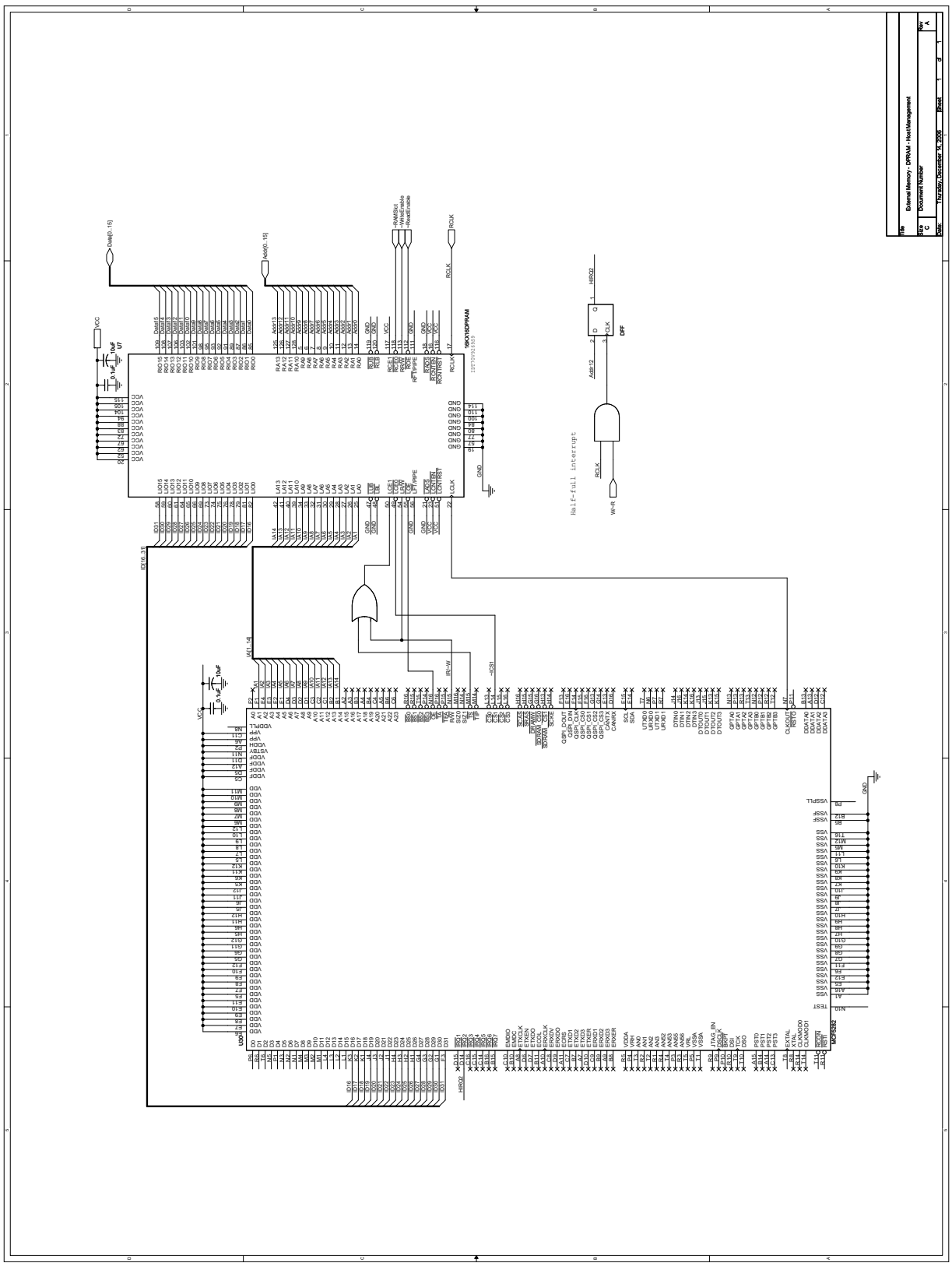
$$\text{Digital value} = 1023 \times (\text{input voltage} - V_{REFLO}) / (V_{REFHI} - V_{REFLO}) \quad (1)$$

Where V_{REFLO} and V_{REFHI} are the voltages applied at ***AnalogRefLow*** and ***AnalogRefHigh*** pins, respectively.

The ADC's performance is guaranteed when V_{REFLO} = AGND and V_{REFHI} = AVCC. Not adhering to these values may result in performance degradation.

The ADC power supply should be decoupled with the use of a 2.2-6.8 μF tantalum capacitor in parallel with a 0.01-0.1 μF ceramic capacitor placed as closely as possible to the power supply and ground pins. An additional 0.01-0.1 μF ceramic capacitor should be placed across ***AnalogRefLow*** and ***AnalogRefHigh***.

Figure 6-11: External memory, DPRAM, host management



6.9.1 Single-ended Interface

The following schematic (Figure 6-12) is a single-ended conditioning circuit that may be used for interfacing an on-board temperature sensor, the RTI ACW-027(refer to application notes). The input signal, V_T, is a single-ended voltage signal with a range of 0.45 - 2.9V and it is assumed to be varying slowly, at no greater than 100 Hz.

The goal of the conditioning circuitry is to match the analog signal to the ADC's voltage range and supply it with the required power. The conditioning circuit should be kept as simple as possible and make use of a single +3.3V supply.

6.9.1.1 Conditioning Circuitry and Op-amp Selection

Because the input is a voltage signal, an inverting amplifier is used to ensure a large input impedance. The operational amplifier should have rail-to-rail inputs/outputs with a unipolar supply. The TLV2471 is recommended as it can swing to within 180 mV of each supply rail while driving a 10 mA load.

The functionality of the circuitry at DC is depicted in equation (2).

$$Vout = V_T\left(1+\frac{Rf}{Rg+R_3\|R_2}\right)-Vs\cdot\frac{R_2}{R_2+R_3}\cdot\frac{Rf}{Rg+R_3\|R_2} \quad (2)$$

The gain of the circuitry is calculated in this manner so as to accommodate the full output swing of the op-amp, and to match it to the input swing of V_T. This is shown in the following equation.

$$\left(1+\frac{Rf}{Rg+R_3\|R_2}\right)=\frac{3.3-2\cdot 0.18}{2.9-0.45}=1.2 \quad (3)$$

Additionally, the circuitry should bias the output so that when V_T reaches the lowest value of interest, the op-amp also reaches its lowest output voltage. Applying (3) and calculating equation (2) at Vout = 0.18, and V_T = 0.45 results in the following.

$$\frac{R_2}{R_2+R_3}=0.5455 \quad (4)$$

Selecting R_2 = 12k (1%), R_3 = 10.0k (1%), Rg = 19.6k (1%), Rf = 4.99k (1%) satisfies equations (3) and (4), while maintaining load currents in the working range of the op-amp and ADC.

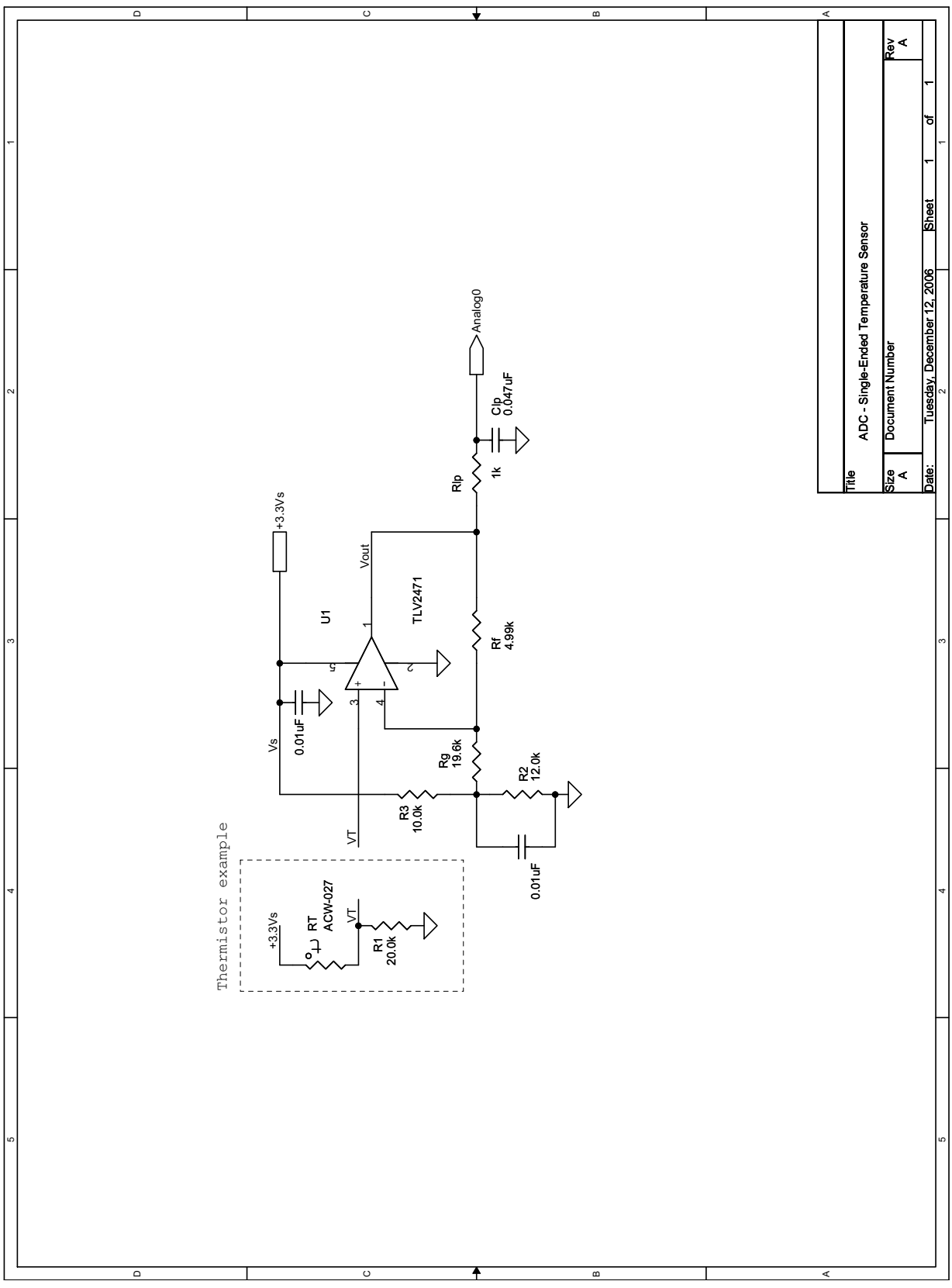
Note that if the input voltage V_T is linear within the supply voltage Vs, then the variation and sensitivity of the circuitry in Vs is relatively small since the same variations will affect the ADC. This will cancel out the variation's effects on the conditioning circuitry.

6.9.1.2 Rlp and Clp values

A low-pass RC filter is used to eliminate noise and prevent aliasing. Additionally, it is used to limit the load on the op-amp, which enables it to swing as close as possible to its rails.

Using Rlp = 1 kΩ and a ceramic Clp = 0.05 µF will result in a low-pass filter with a 3 dB point at f_0~3 kHz (which is assumed to be at least one order larger than the signal's bandwidth). The capacitor should be placed as close as possible to the ADC input pin, as it partially drives the sample capacitor of the ADC.

Figure 6-12: ADC, single-ended temperature sensor



6.9.2 Differential Interface

The input signal is assumed to be differential, Vin+ and Vin-. The voltage signal is in the range of (Vin+ - Vin-) = [-3V,+3V], and slowly varying (not greater than 100 Hz).

The goal is to condition the differential input signal to fit the ADC's voltage range, and supply it with the required drive. For example, the circuitry may be used to interface to a resonator sensor rate such as the RRS75 from Inertial Science, Inc. Additional ADC channels may be used simultaneously in order to expand the dynamic range of the ADC.

6.9.2.1 Conditioning Circuitry

The purpose of this interface is to generate a signal with the following format.

Analog0 = G(Vin+ - Vin-) +Vref, with nominal G = 0.55 and Vref = 1.65.

The interface shown in Figure 6-13 forms an instrumentation amplifier. The high input impedance of the instrumentation amplifier is highly desirable to eliminate voltage drops and CMRR concerns due to the signal source output impedance.

The following equation describes the functionality of the conditional circuitry at DC.

$$RO = (Vin+) \cdot G - (Vin-) \cdot m + Vref \cdot [(R_2+R_1) / R_2] \cdot [R_4 / (R_3+R_4)] \quad (7)$$

where:

$$G = [R_1 / R_2] \cdot [(R_5+R_g) / R_g] + [R_6 / R_g] \cdot [R_3 / (R_3+R_4)] \cdot [(R_2+R_1) / R_2]$$

$$m = [(R_6+R_g) / R_g] \cdot [R_3 / (R_3+R_4)] \cdot [(R_2+R_1) / R_2] + [R_1 / R_2] \cdot [R_5 / R_g]$$

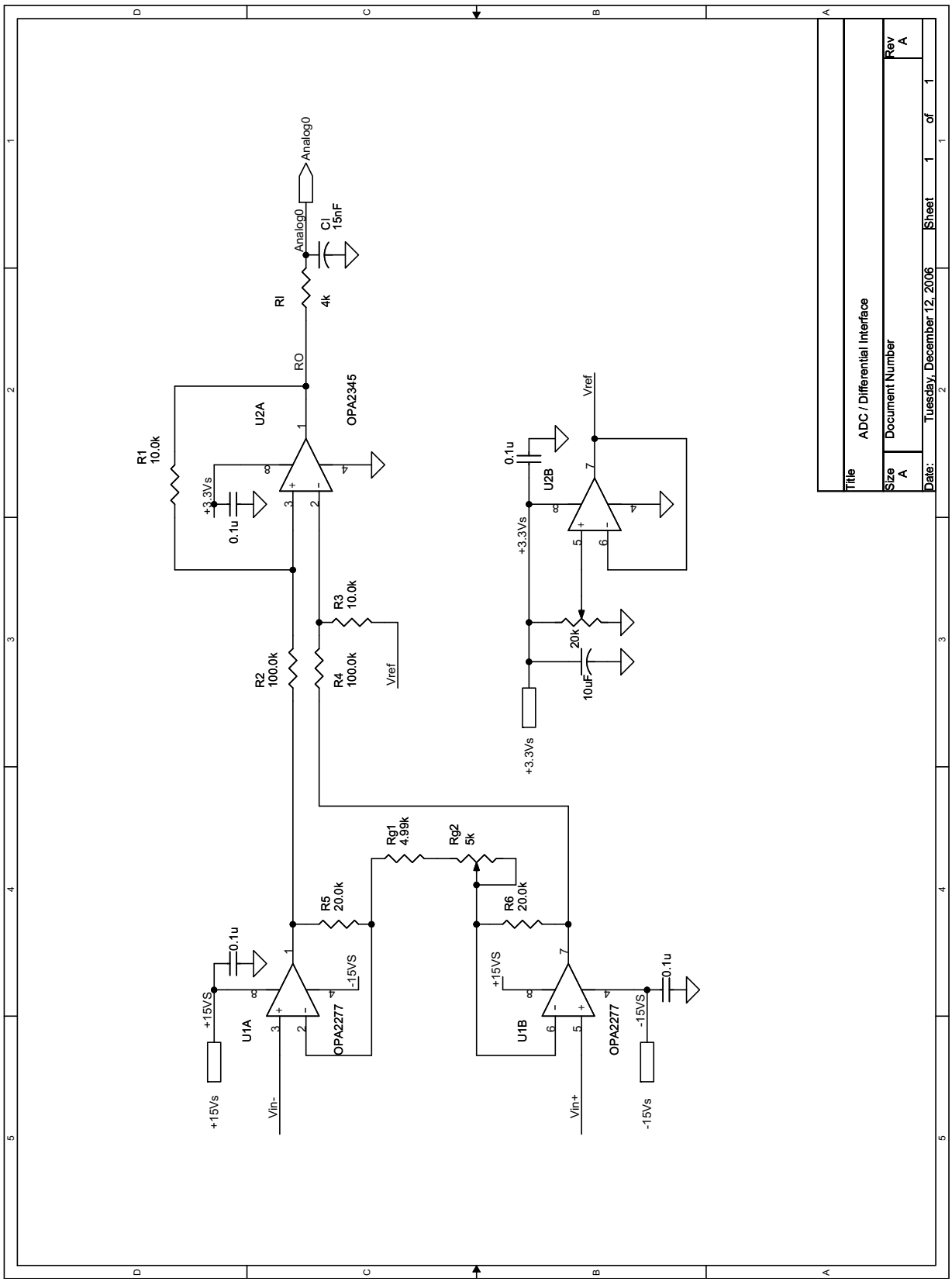
Selecting $R_3 = R_1$ and $R_4 = R_2$ will result in the following simplified version:

$$RO = (Vin+ - Vin-) \cdot [R_1 / R_2] \cdot [1+ (R_5+R_6) / R_g] + Vref \quad (7a)$$

Specifying resistors $R_3 = R_1 = 10.0\ k\Omega$ (%1), $R_4 = R_2 = 100.0\ k\Omega$ (1%), $R_5 = R_6 = 20.0\ k\Omega$ (1%), and nominal $R_g = 8.9\ k\Omega$, will result in the desired G = 0.55. The importance of having matching pairs of resistors should be evident from equation (7). If matching is not done common mode voltage will be introduced.

The OPA2345 is an input/output rail-to-rail operational amplifier with low voltage bias, and high CMRR. It tolerates input common voltages of ± 0.3V from its rails. As a protection measure, the addition of Shottky diodes with 0.3V forward voltage, such as 20L15T, is recommended (but not shown). The output swing of the op-amp is closely related to the load current. In order to make this current as low as possible, a resistor is added at the output of the op-amp. Adding a capacitor forms a LPF, with a 3 dB cut-off at ~3 kHz. The capacitor should be placed as close as possible to the ADC input pins, and is partially used to drive the sampling capacitor.

Figure 6-13: ADC, differential interface



6.10 User I/O Space

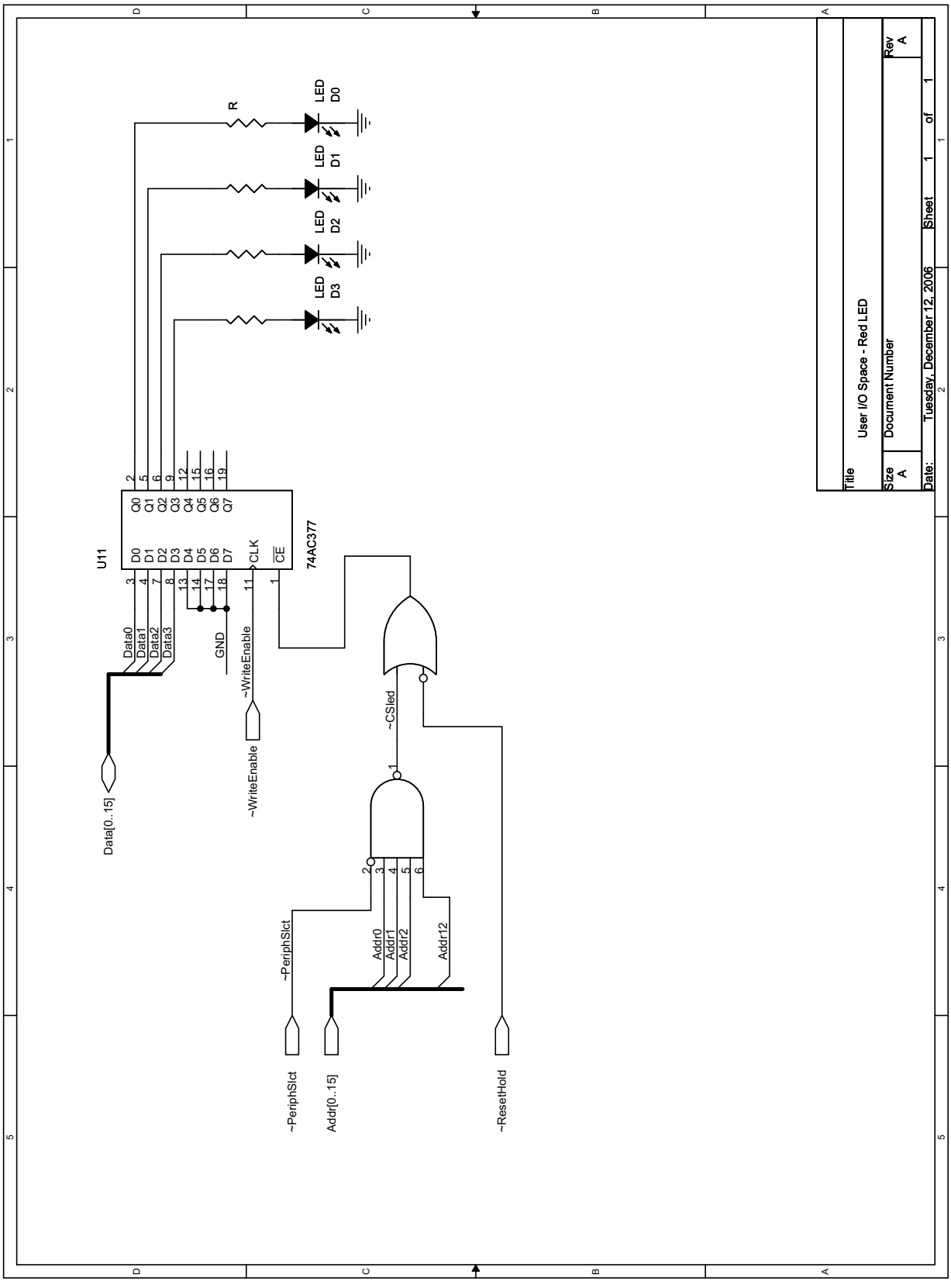
In the following schematic (Figure 6-14), the User I/O space is used to control four LEDs. The MC58000 User I/O accessible address space located from address 1000h to 10FFh on the external bus. The Addr12 bit is reserved in order to serve as a chip select, while the least significant eight bits serve as the active address. In this example, it is assumed that the active I/O space consists of eight registers; using only the three least significant bits of the address word. Additionally, the register to control the LEDs is assumed to reside at address offset 0x7h.

The simplest way to add an LED is to connect it to a high sink/source current port. In this example, a 74AC377 is used to buffer the data and to drive the LEDs. The MV8141 super bright red LEDs have been used, with 1.5V/1.7V/2.4V minimum/typical/maximum forward voltages, respectively, at 20 mA. R = 150Ω ensures that the output current doesn't exceed 20 mA; with a typical current of 8 mA (assuming a supply of V_{CC} = 3.3V±5%). It also ensures that a minimum of 2 mA current will flow through the LED. Note that the variation in the actual current is relatively large and will result in large variations in the luminance.

The 74AC377 was chosen due to the simplicity with which it interfaces to the MC58000. The 74AC377 is not equipped with a master reset and therefore on power-up the LEDs may be in an arbitrary state until the first valid write is received. The master reset version, the 74AC273, may be used providing that logic for generating the clock signal to it is added. The 74ACTQ823 may also be used with 5V supplies and features both clock enable and master reset.

For a User I/O space read example, refer to Section 6.11, "Parallel Word Position Input," noting that the address space is different.

Figure 6-14: User I/O space, red LED



6.11 Parallel Word Position Input

The following schematic (Figure 6-15) shows an interface to the AD2S1200 in order to generate a digital word position feedback input to the MC58000. The AD2S1200 is a programmable 12-bit resolution resolver-to-digital converter. In the following example, the digital interface to the CP's data bus will be demonstrated. The motor interface to the resolver is not shown. The MC58000 maps the peripheral addresses 800h, 801h, 802h, and 803h to the position input registers for axes 1 to 4, respectively. For the MC58110 only address 800h is used.

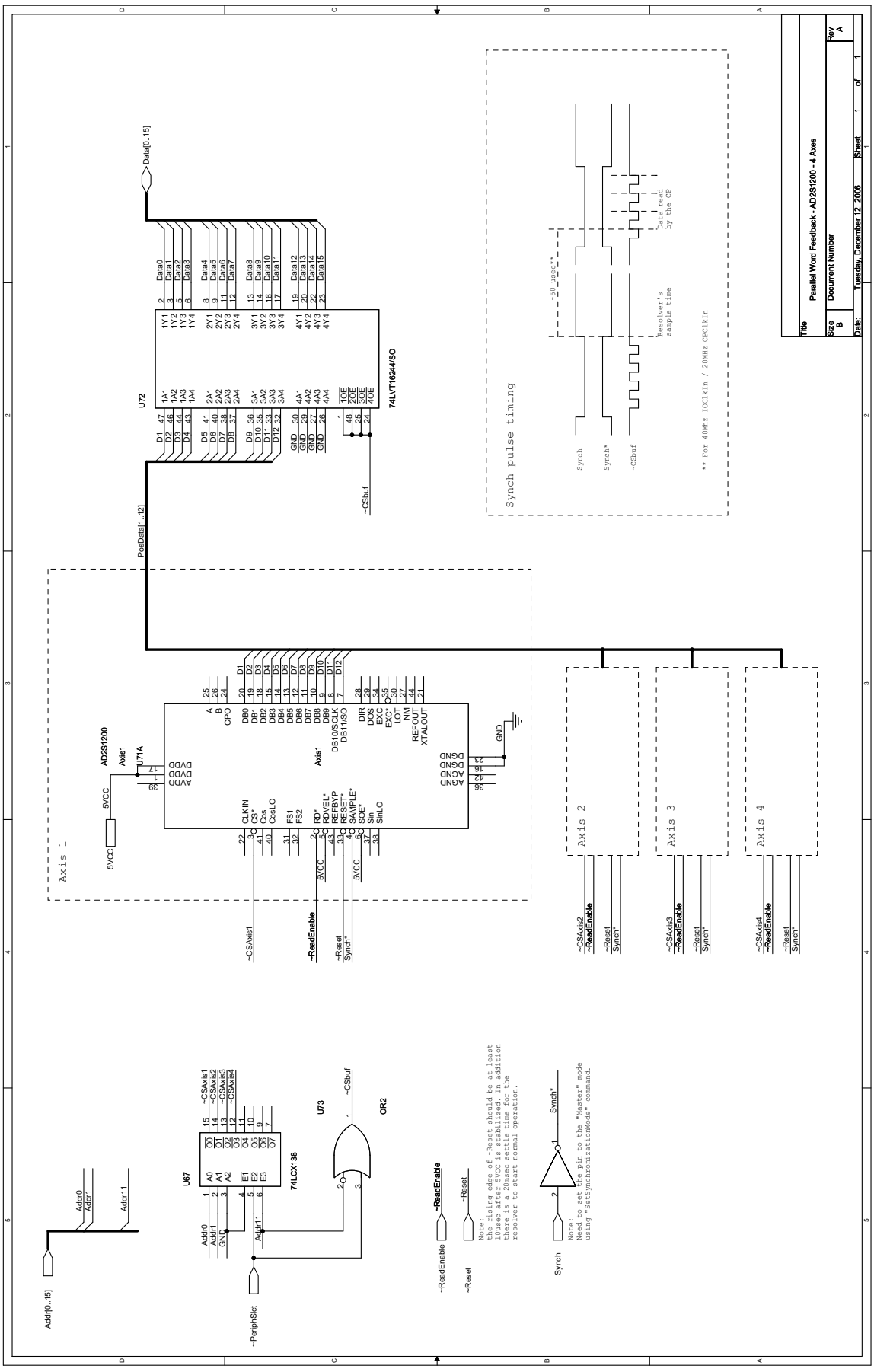
The interface to the AD2S1200 uses the MC58000's output signal, *Synch*. A falling edge of the *Synch* signal indicates that a position inputs are to be read. The rising edge of this signal is used to sample the resolver's position. The actual reads of these values occurs only one cycle later, introducing effective delay of one cycle (51.2 μsec for a 40 MHz *IOClkIn*/20 MHz *CPClkIn*). This delay is necessary to guarantee the timing requirement of the AD2S1200. Consult PMD for ways to avoid this delay.

Note that the *Synch* signal should be activated using the SetSynchronizationMode command.

The TTL outputs of the AD2S1200 are interfaced to a common 3.3V LVT family's 16-bit buffer. The LVT family is a low voltage CMOS with TTL compatible inputs and an input V_{HI} tolerance of up to 5.5V.

Note that the decoder is required only when parallel word input is used by more than one axis. If only one or two axes are used, it is simpler to generate the chip-select signals using discrete components.

Figure 6-15: Parallel word feedback, AD2S1200, 4 axes



6.12 Parallel Communication Interface

Parallel communication supports the highest throughput of any of the communication interfaces. It is most often used when the host processor and MC58000 reside on the same circuit board. In the dual chip configuration (MC58x20) the IO chip provides the parallel interface to the host and care should be taken to ensure that the timing requirements specified in chapters 3 and 4 are observed. In the single chip configuration (MC58x10) if no external logic is providing the parallel interface the *ParallelEnable* input pin should be tied low to indicate that parallel communication is disabled.

In Section 6.12.1, "16/16 Host Interface," and Section 6.12.4, "8/16 Host Interface," the 16/16 (16-bit bus, 16-bit data) and 8/16 (8-bit bus, 16-bit data) host interfaces are demonstrated.

6.12.1 16/16 Host Interface

In this example, the IO chip is interfaced to the Motorola MCF5282 ColdFire microprocessor, as shown in Figure 6-16. The MCF5282's external interface module is used for the interface, which includes data and address buses as well as additional control signals such as *R/~W*, *~TS*, *~OE* and *~CS*. For a detailed description of this device's functionality and timing specifications, refer to the MCF5282 data sheet. The following design notes focus on the interface between the MCF5282 and the IO chip.

6.12.2 Write Cycle

The host should be able to generate valid data at less than T_{15} (refer to Section 3.3, "AC Characteristics,") from the latest falling edge of either *R/~W* or *~CS* signals. Since both *R/~W* and *~CS* become active as much as half of the MCF5282's bus clock cycle prior to the data, the *~TS* (Transfer Start) signal is ANDed with the inverted *R/~W* signal. For the write cycle this achieves a one bus clock cycle delay in the incoming *IR/~W* signal to the IO chip. In this manner, a maximum of 10 nsec is guaranteed between the falling edge of *IR/~W* and valid data.

6.12.3 Read Cycle and Wait States

The host should add wait states in order to meet the timing requirements of the IO chip. The following formula may be used in order to calculate the number of required wait states as a function of the host's bus clock period, t_{CYC}.

$$N_{WS} = \lceil 70/t_{CYC} \rceil - 1$$

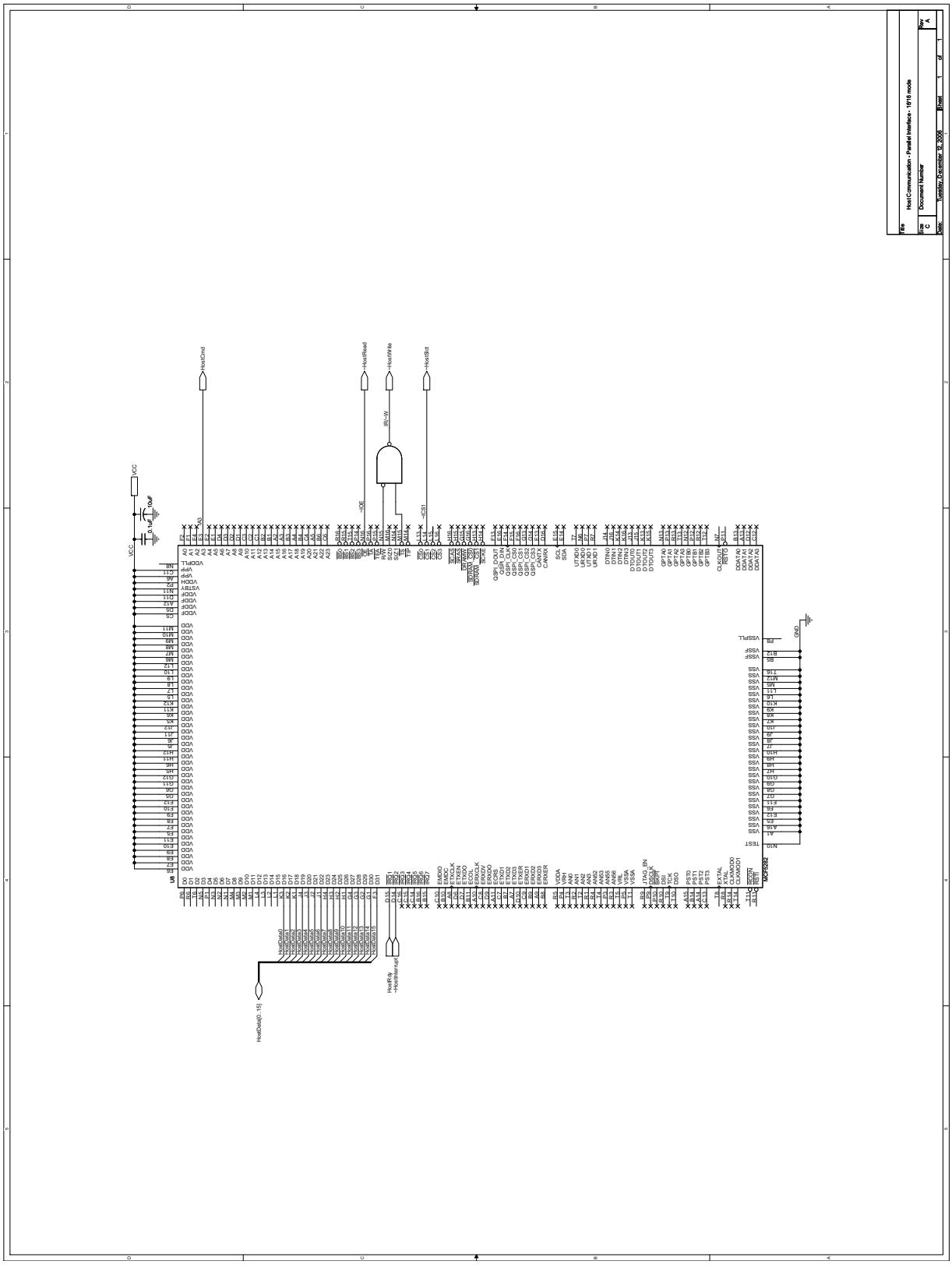
The operator $\lceil \cdot \rceil$ indicates rounding towards the largest integer. N_{WS} is the number of wait states that are required. The selected MC5282 has a 66 MHz input clock resulting in a t_{CYC} of ~15 nsec and thus four wait states are required.

6.12.3.1 Other Control Signals

In the example, address bit 3 is used for signaling to the IO chip whether a command or data word is being written. For a read cycle, this bit may be used for requesting either data or the status word.

HostRdy is used to interrupt the MCF5282. This can be a low priority interrupt used to invoke a communication ISR in the MCF5282. According to conditions that are programmable by the host the MC58000 can also activate the *~HostInterrupt* signal.

Figure 6-16: Host communication, parallel interface, 16/16 mode



6.12.4 8/16 Host Interface

In this example, a PIC microcontroller with limited I/O pins is interfaced with the IO chip. The minimum number of I/O pins required for the parallel 8/16 communications mode is eight bi-directional pins for the data, and five additional pins for the control signals (four outputs and one input). These signals are shown in Figure 6-17.

The PIC16F648 features:

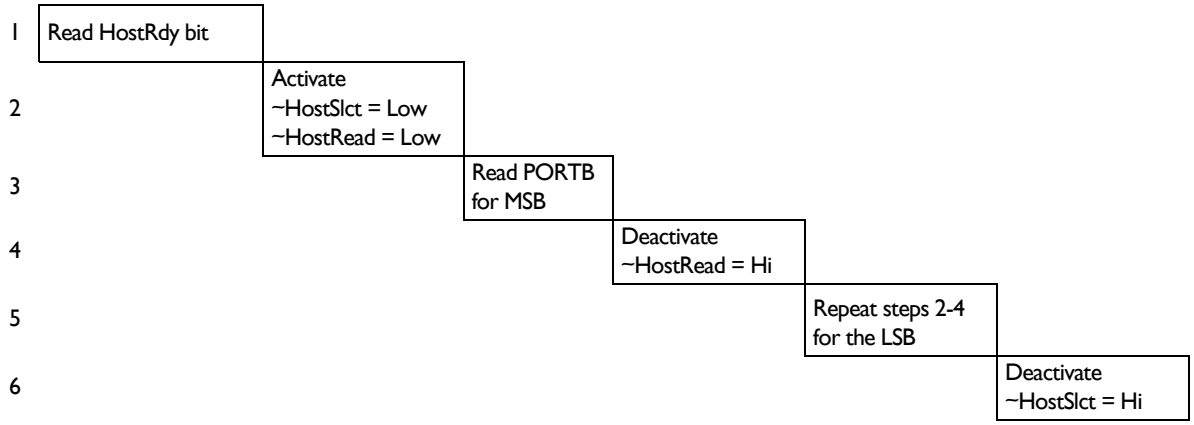
- An 8-bit microcontroller core
- Up to 10 MHz clock at 3.3V voltage supply
- 16 I/O pins divided into two ports, with a high impedance state and weak internal pull-up.

The eight I/O pins of the PORTB are used to connect to the IO chip's *HostData* bus (pins 0 through 7). The five pins of the PORTA are reserved for the control signals, as shown in the following table.

IO Chip Signal	PIC16F648 Signal	Dir	Comments
HostData0-7	PORTB RB0-RB7	I/O	Should be pulled up (Bit 7 of the OPTION register). Should be kept in high impedance state unless written to.
~HostSlct	PORTA RA0	O	
HostCmd	PORTA RA1	O	
~HostWrite	PORTA RA2	O	
~HostRead	PORTA RA3	O	
HostRdy	PORTA RA6	I	

Shown below is a typical sequence of events for performing the host read and write cycle.

6.12.4.1 Typical Data / Status Read Sequence



Where:

Step 1: May be performed using the BTFSS instruction. This step must loop until *HostRdy* is high.

Step 2: Write a byte into PORTA with the lowest four bits set to either 0x4 (data) or 0x6 (status) using the MOVLW and MOVWF instructions.

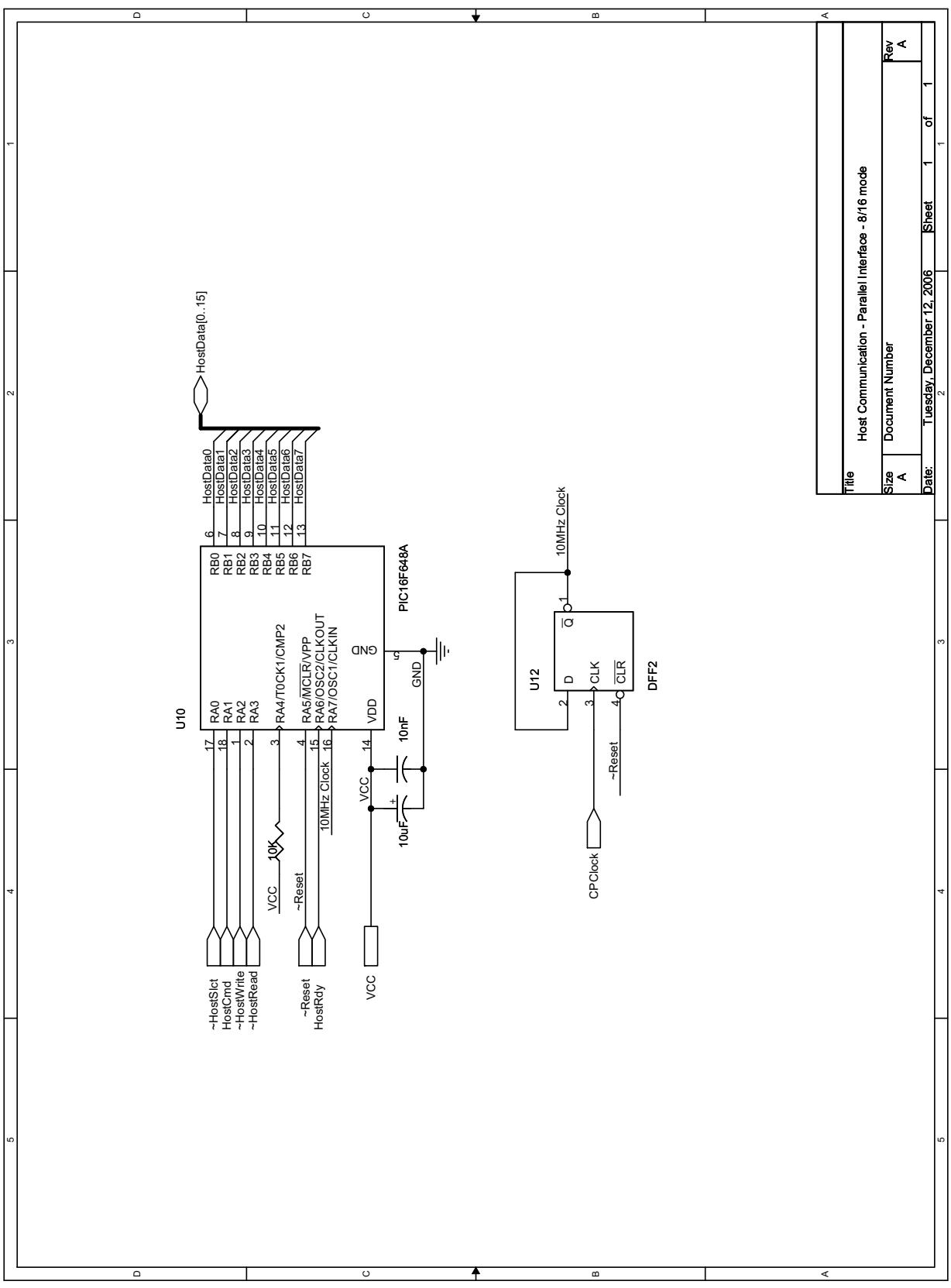
Step 3: Read Port B and store the result.

Step 4: Write a byte into PORTA with the lowest four bits set to either 0xC or 0xE to deactivate ~HostRead.

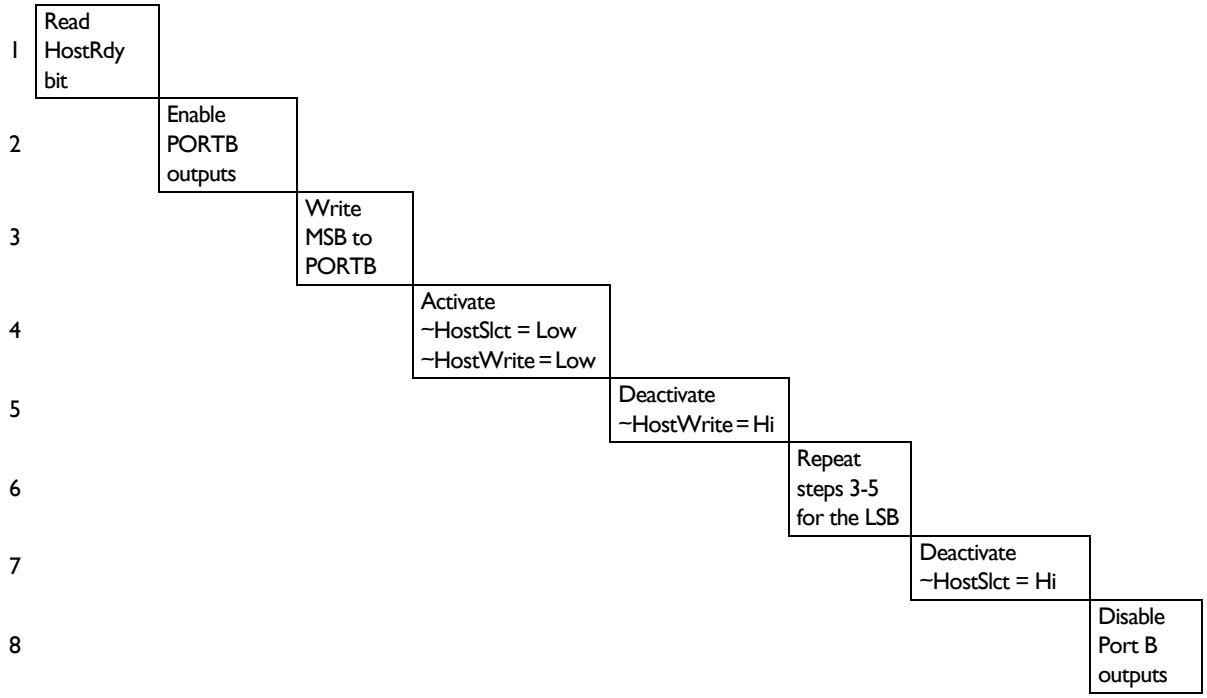
Step 5: Repeat steps 2 - 4 for the LSB.

Step 6: Write a byte into PORTA with the lowest four bits set to 0xF to deactivate ~HostSlct.

Figure 6-17: Host communication, parallel interface, 8/16 mode



6.12.4.2 Typical Data/Instruction Write Sequence



Where:

Step 1: May be performed using the BTFSS instruction. This step must loop until *HostRdy* is high.

Step 2: Enable PORTB outputs by writing the all-zero word to the TRISB register.

Step 3: Write the MSB into PORTB.

Step 4: Write a byte into PORTA, with the lowest 4 bits set to either 0x8(data) or 0xA(instruction) using the MOVLW and MOVWF instructions.

Step 5: Write a byte into PORTA, with the lowest 4 bits set to either 0xC or 0xE to deactivate ~HostWrite.

Step 6: Repeat steps 3 - 5 for the LSB.

Step 7: Write a byte into PORTA with the lowest four bits set to 0xF to deactivate ~HostSlct.

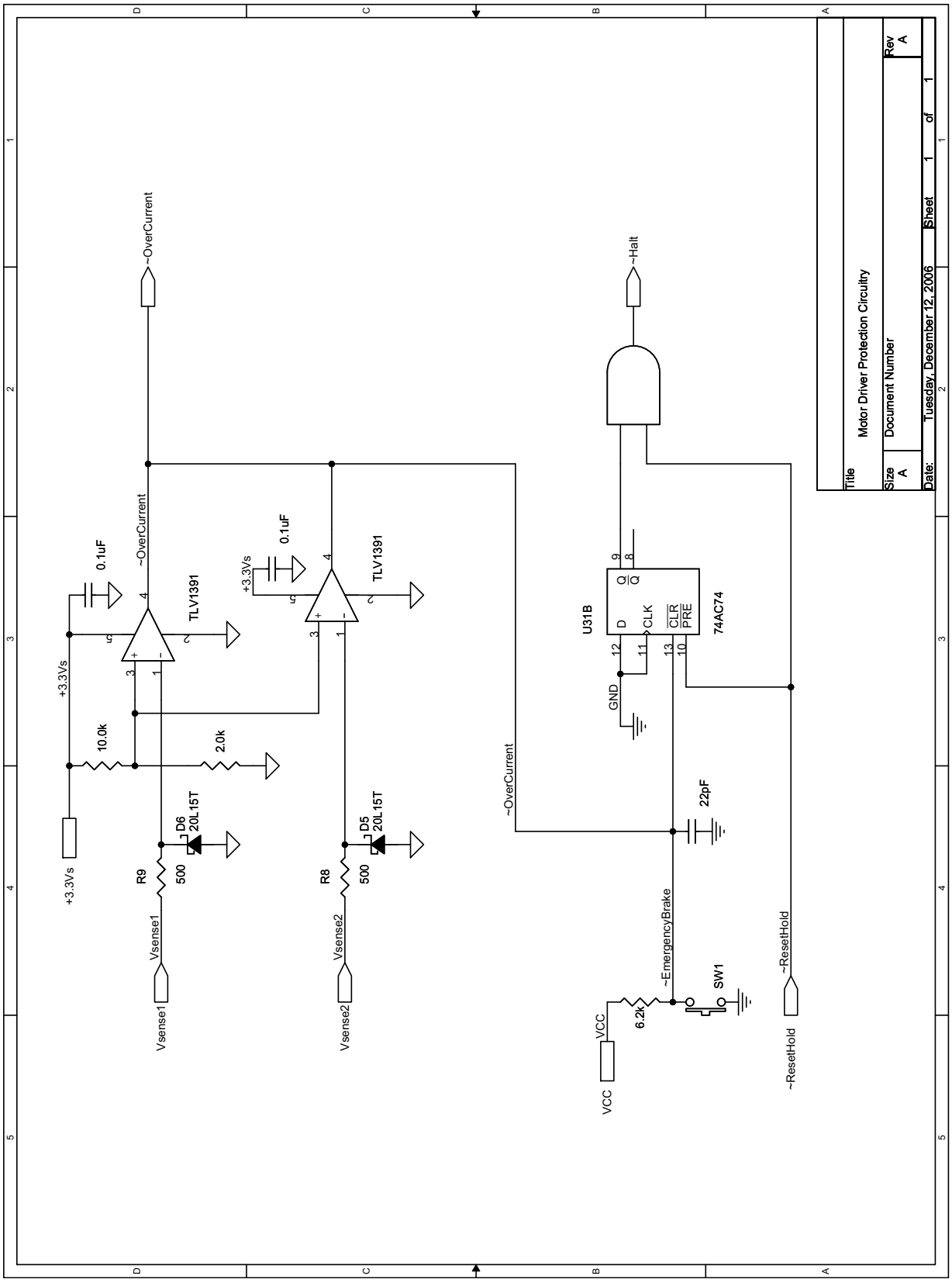
Step 8: Disable PORT B outputs by writing 0xFF to the TRISB register.

Note: Since each step takes one or multiple execution cycles (400 nsec) there are no practical timing constraints. For the write cycle, the data should be present on the data bus prior to enabling the write operation by setting *~HostWrite* to active low.

6.13 Overcurrent and Emergency Braking Circuits for Motor Drivers

Most of the drivers demonstrated in this manual, either full or half bridges, are used with a sense power resistor through which the winding current flows. This results in a voltage drop, which is then sampled by the overcurrent circuitry. Due to switching transients in the driver, the current through the sense resistor is prone to spikes. The following schematic (Figure 6-18) shows protection circuitry based on the voltage developed over *Rsense*, *Vsense1* and *Vsense2*. This circuitry should serve as a protection device and as such in normal operation the circuitry should not reach its threshold voltage. As a protective device, the response time should be determined according to the application requirements. The response time for this circuitry is set to 2 µsec.

Figure 6-18: Motor driver protection circuitry



The comparator is a TLV1391, which is a single supply, fast response, open collector output. The TLV1391 tolerates an input range as low as –0.3V. Because the sense voltage may fall below this range, a protection diode is added, with a V_F = 0.25V. In order to avoid false over-current detection, the sense signal is low pass filtered with a cut off frequency in the MHz region. This is achieved with the use of a 500Ω resistor and the capacitance of the Schottky diode. The nominal reference voltage of the comparator is set to 0.55V, which is selected to be 80% higher than the nominal voltage drop over *Rsense* at the rated current of the motor windings (0.3V). The typical response time of the TLV1391 is 800 nsec; leaving ~1 µsec response time for the driver itself. Additional input branches may be added to the *~OverCurrent* circuitry. In this case, the value of the pull-up resistor should be re-calculated. The resultant *~OverCurrent* signal may either be used for the momentary disabling of the motor's driver, or to halt it completely until the next *~Reset* signal is generated.

An external switch, ~EmergencyBrake, is used as an additional method for halting the motor. Note that the U31B D-FF may momentarily have both Clear and Preset inputs active low.

6.14 DC Brush Motor Control Using SPI Interfaced DACs

The example in Figure 6-19 shows a cost-effective solution for controlling DC brush motors using the SPI interface to an Analog Devices 16-bit DAC AD1866. The AD1866 incorporates 5V CMOS logic with TTL compatible inputs, enabling a direct interface to the 3.3V MC58000 outputs. In all respects the SPI interface requirements of the AD1866 impose no practical limitations. The Magellan SPI port can be configured using the **SetSPIMode** command to either falling edge without phase delay, or rising edge with phase delay, in order to maintain compatibility with the AD1866 SPI port. The AD1866 requires twos-complement data format which can be selected with the Magellan **SetOutputMode** command.

Note that only SPI DACs which support 16-bit packets can be used.

6.14.1 Conditioning Circuitry

The AD1866 analog output, Vo, is a ±1Vpp output, ΔV, centered at the Vref = 2.5V reference voltage; in other words, Vo = Vref + ΔV. The AD1866 has moderate accuracy, with a mid-scale error of ±30 mV at ±3% of the full range, and a gain error of ±3% of the full range. The goals of the conditioning circuitry are to amplify the output to the ±10V range, and to provide a means to disable the AD1866 outputs until the MC58000 generates the first valid DAC word.

There are two methods for interfacing the AD1866 to a motor amplifier. First, when the output voltage is referenced to Vref, and the second is when it is referenced to the signal ground. The first solution is simpler, since it doesn't involve high precision matching resistors, but the appropriate method for a given application will depend on the requirements of the motor amplifier. Note that both interfaces use single-ended transmission. If the system requires differential transmission, then changes to the design will be required.

6.14.2 Referencing to Vref

An operational amplifier (U6) and two resistors (R_1 and R_2), are used to generate a ±10V differential output (V_{DAC}), which is referenced to Vref.

$$V_{DAC} = V \cdot (R_2/R_1) + Vref \quad (1)$$

Selecting R2 = 100 kΩ (%1) and R1 = 10.0 kΩ (%1) results in an amplification gain of A_M = 10.

In order to avoid starting the motors in an unknown state at power-up, the DAC's output voltage is wired through a buffer equipped with shut-down capability. At power-up or reset, this buffer will be in shutdown mode resulting in high

impedance output. In this state, the inverting x10 amplifier output will be Vref. Vref is wired through a buffer, since it is not designed to sink or source the large currents that may be required at the input stage of the motor amplifier.

Since the DAC performance poses no practical requirements on the op-amp selection, the selected components are the TLE2021, and dual TLV2473 with shutdown. The TLE2021 is capable of swinging up to the required worst-case +12.5+(0.6)V, but has a relatively large bias current. $R_3 = R_1 || R_2$ is used in order to reduce the effect of the bias current.

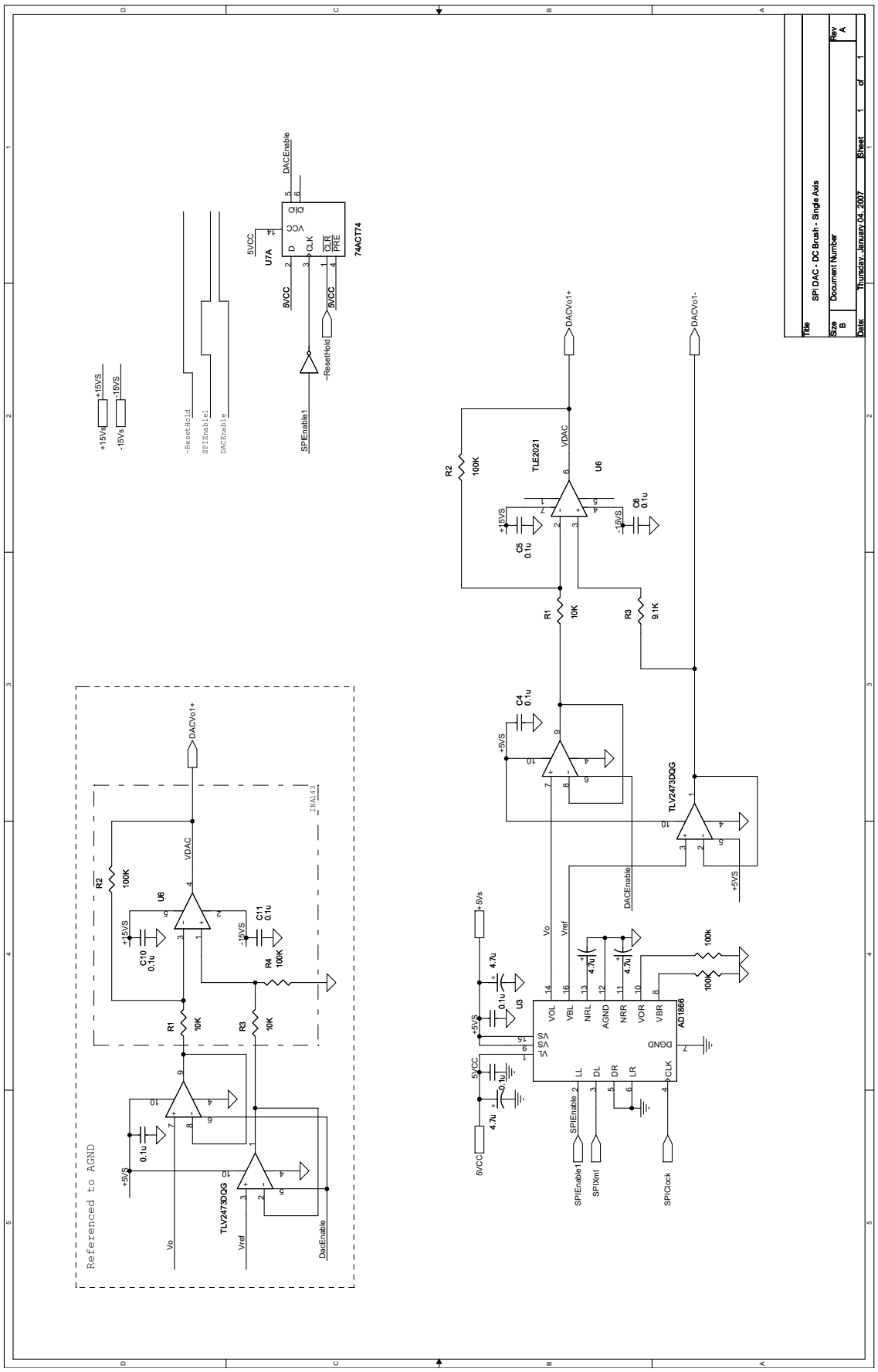


Figure 6-19: SPI DAC, DC brush, single axis

6.14.3 Referencing to AGND

By adding an additional resistor, R_4, and selecting $R_2 = R_4 = 100\ k\Omega$ (1%), and $R_1 = R_3 = 10.0\ k\Omega$ (1%), U6 becomes a differential amplifier with a nominal gain of 10.

$$V_{DAC} = - V \cdot R_2/R_1 \qquad (2)$$

This equation only holds true for matching pairs of resistors; otherwise undesired biases will not be completely canceled out, resulting in common-mode voltage. Using a 1% tolerant resistor will result in ~1% bias in the output signal. This bias is only one third of the AD1866 tolerances, but it should not be ignored. If this accuracy is not adequate, then high precision 0.1% resistors may be used. Alternately, the entire resistor set and amplifier may be replaced by a differential amplifier, such as the INA106 or the INA143, which offer a fixed gain of 10 and high CMRR.

A dual TLV2473 is used in order to ground the output at reset or power-up. The resulting circuitry forms an instrumental amplifier, which also benefits from the small output impedance of the buffers; thus minimizing the CMRR even further.

Notes:

1. In the dual-chip configuration (MC58x20), when more than one motor is to be driven by the SPI DAC, use the *SPIEnable1-4* signals (which are sourced by the IO chip) as chip-selects to enable the appropriate SPI port. For instance, if the right port of the AD1866 is to be used to drive a second motor connect *SPIEnable2* and *SPIXmt* to the AD1866's LR and DR input pins, respectively.
2. In both conditioning circuitries, an inverting amplifier (U6) is used. This may require that the motor output signal be inverted by setting bit 12 in the Magellan signal sense register.
3. The *DACEnable* signal goes active high on the first write to the DACs after the *~ResetHold* active low period has completed. Note that if the AD1866 is used to drive two motors, then the *DACEnable* signal should be generated with the use of the *SPIEnable* signal that corresponds to the last of the two motors being written to.
4. The AD1866 analog signal power supply should be decoupled with capacitors placed as closely as possible to both the supply pins and the signal ground. Refer to the AD1866 data sheet for a complete description.
5. The TLV2473 shutdown input accepts TTL input levels, which can be fed by a 3.3V CMOS D-FF (U7). If other op-amps are selected, then the shutdown input levels should be checked. If there is level incompatibility the D-FF may be selected from the HCT/ACT family.

6.15 Brushless DC Motor Control Using High-Precision Parallel DACs

In this example, TI's high accuracy 16-bit parallel DAC7744 is used to drive four brushless DC motors. The MC58420 uses the peripheral address space to access the correct driver, with the *Addr14* signal used as a chip-select, as detailed in the following table.

Address	4000h	4001h	4002h	4003h	4004h	4005h	4006h	4007h
Output	Axis1A	Axis1B	Axis2A	Axis2B	Axis3A	Axis3B	Axis4A	Axis4B

The DAC7744 is a four channel, digital-to-analog converter (DAC). The four channels are divided into two pairs; each fed by a different reference voltage. In order to minimize the effects of the mismatches, the DAC7744 pair of channels (which share the same reference) are used for driving Phase A and Phase B of the same motor.

The schematic in Figure 6-20 indicates that two DAC7744 components are required. The *Addr2* signal is used to select between the two devices, and the *Addr0* and *Addr1* lines are used to write into the appropriate channel's input buffer register. As the order of writes from the CP is Phase B followed by Phase A, the LSB of the address is used to generate the *LDDACS* signal, which is used by the DAC7744 to latch the input buffer registers. In this manner, the new values for both phases take effect simultaneously. The DAC7744 supports 5V CMOS input voltages. To interface it with the MC58000 data bus, a transparent latch with TTL input voltage compatibility, the 74ACT373, and its 10-bit version 74ACT841, are used. The propagation delay of these latches is less than 11 nsec. All other logic interfacing to the DAC7744 has TTL compatible inputs, and is supplied with +5V.

The digital input stage of the DAC7744 imposes several stringent timing requirements. To meet these requirements, the *~WriteEnable* signal is sampled on the rising edge of *ClockOut*, and a *~CS* pulse of 75 nsec is generated out of it. In order for the data to be ready at least 40 µsec before the rising edge of *~CS*, the propagation delay through the 5V data buffers should be less than 15 µsec. This requirement is satisfied by the 74ACT373 and 74ACT841. In order to satisfy the 15 µsec hold time, the propagation delay through the D-FF and its subsequent logic should be less than 20 µsec. Assuming 10 µsec for the 74AHCT32 OR gates leaves 10 µsec for the D-FF, which is sufficient for the 74LVT74. Note that the U6 OR gate should have a propagation delay of less than 6.5 µsec, minus the setup time of the D-FF (1.7 µsec for 74LVT74). Since *LDDACS* only goes active once per axis, the DAC's requirements for a period of at least 80 µsec between the rising edge of the *LDDACS* signal and the next falling edge of *~CS* is met.

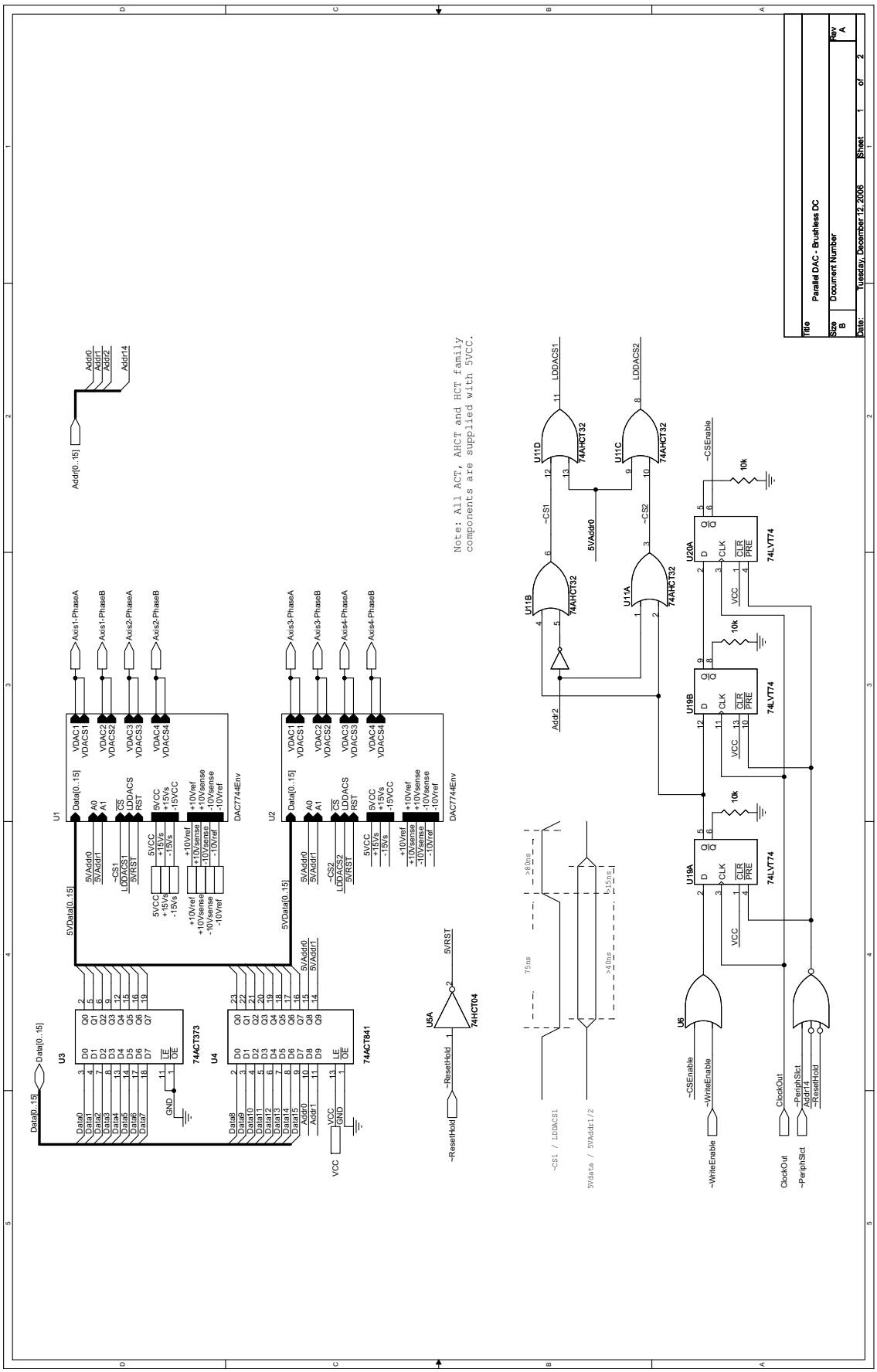
At power-up or reset, the *~CS* signals will be held inactive high while the MC58000 completes its initialization which occurs during the *~ResetHold* active low period. During this time undesired values will not latch into the DACs.

6.15.1 Analog Signals

The schematic in Figure 6-21 demonstrates a high-precision DAC solution. In order to maintain the accuracy of the DAC, it should be supplied with high accuracy reference voltages with tolerances of less than half the LSB. The AD2702 or AD688 voltage regulators may be used (see Figure 6-1).

For improved accuracy, the DAC7744 is equipped with sensing mechanisms on both the reference voltages and the analog output signals. These allow for compensation of the board's wiring resistances. The AD688, which is used to generate the ±10V reference voltage, provides sensing pins. These pins may be tied to the reference sensing inputs in the DAC7744 to compensate for the resistance path. The AD688 may source/sink ±10 mA of current on each of the output pins, and should be able to drive up to two dual DAC7744 components. The AD688 provides calibration pins, which may be used to reduce the initial ±2 mV error as described in the AD688 data sheet. The ±2 mV error will translate to ±8 LSB accuracy for the 16-bit DAC. Achieving full 16-bit resolution implies that the error budget and noise levels should be kept below ±150 µV. If such precision is not required, less accurate components from the same family may be used, such as the DAC7744E.

Figure 6-20: Parallel DAC, brushless DC, sheet 1 of 2



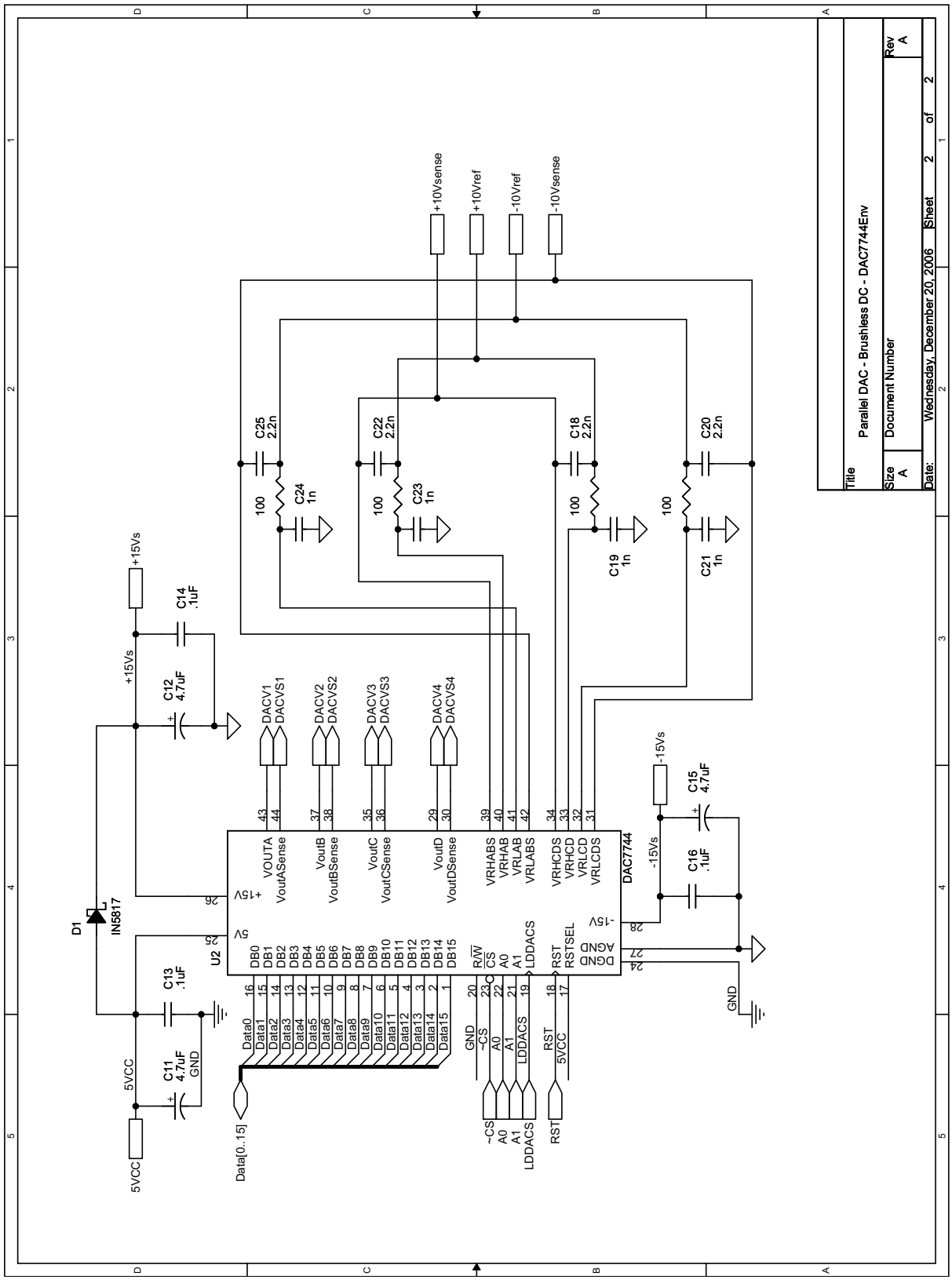


Figure 6-21: Parallel DAC, brushless DC, sheet 2 of 2

6.16 Single-Axis Magellan with Brushless DC Atlas

The following schematic shows a Brushless DC Atlas Amplifier connected to a single-axis Magellan.

6.16.1 Atlas Power Input and Motor Output

Atlas is powered through pin pairs HV and Pwr_Gnd, and the power source is a transformer-isolated DC power supply. When unregulated DC power supply is used the output voltage with respect to its output power/current should meet the full Atlas operating range specification. The power supply should be able to "absorb" the recovered energy when Atlas is in regeneration mode. If a regulated DC power supply is used, but it cannot dump the regenerated energy, a blocking diode between the power supply and HV can be used.

The Pwr_Gnd and GND pins are shorted inside the Atlas, and at a system level they refer to the same ground. Pwr_Gnd, the current return path for the power train, is paired with HV and may therefore be noisy. GND is the reference for the SPI signals and other digital control signals. These signals require a quiet ground reference. To ensure optimal performance, star grounding is recommended for component placement and layout. That is, Pwr_Gnd and GND should be connected to the system ground very close to Atlas, and the two ground paths should be kept away from each other.

There is a third current return path stemming from the high frequency component of the motor winding current. Atlas drives motor windings with pulse-width modulated (PWM) signals. Although the sum of the average winding currents is zero, the high frequency PWM signal may couple to the ground plane and induce noise into other circuits. Therefore, depending on your application, you may consider utilizing a motor shield cable to provide a current return path. If utilized, its ground point should be very close to, or the same, as Pwr_Gnd.

For Brushless DC motors pins MotorA, MotorB and MotorC are wired to motor windings A, B, C, respectively. Pins MotorD are left un-connected.

6.16.2 Atlas SPI Interface

Atlas receives control commands through an SPI interface and functions as an SPI slave. Atlas SPI communication is enabled when ~SPICS is pulled down.

To ensure optimal SPI communication, please consider the following layout recommendations:

1. Keep traces short and use 45 degree corners instead of 90 degree corners.
2. All SPI signal traces should be located next to a continuous ground plane, or if possible, between two continuous ground planes.
3. Keep traces away from other noisy and high speed signal traces. Alternatively, run ground traces along with these signals as a shield.
4. When multiple Atlas modules are used, keep the SPI signal stubs short.

Note that the Atlas Development Kit layout can be used as a layout reference.

6.16.3 Atlas ~Enable and FaultOut Signals

Atlas has one dedicated input signal, ~Enable, which must be pulled low for the Atlas output stage to be active.

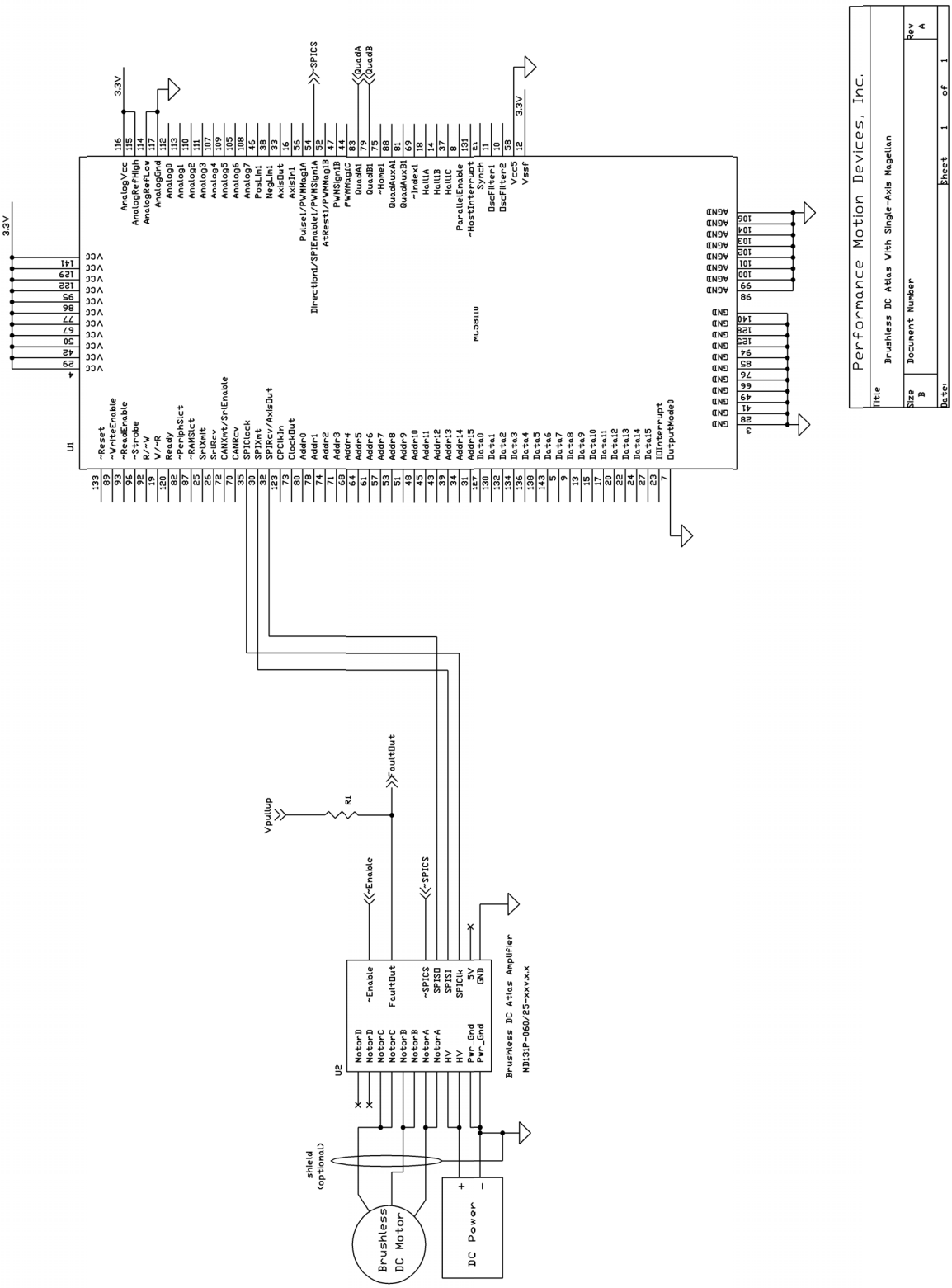
FaultOut is a dedicated output. During normal operation it outputs low. When a fault occurs it will go into a high impedance state. In this example, FaultOut is pulled up by Vpullup through resistor R1. Vpullup can be up to 24V to meet the system requirement. For example, if the fault signal is wired to a 5V TTL input, Vpullup can be 5V.

6.16.4 Magellan MC58110 Configuration

In this schematic the SPI master is a single axis Magellan MC58110. Only the connections with Atlas are shown. For complete Magellan wiring, please refer to the MC58110 electrical specifications.

The MC58110 is configured to default to Atlas motor output by tying pin 7, OutputMode0, to ground. The MC58110 inputs encoder signals, implements motion control and commutation functions, and sends torque commands to Atlas through the SPI interface. Depending on the Magellan commutation method selected the feedback signals HallA, HallB, HallC and ~Index are optional.

Figure 6-22: Brushless DC Atlas With Single-Axis Magellan



6.17 Multi-Axis Magellan with DC Brush & Step Motor Atlas

The following schematic shows a two-axis application with one DC Brush Atlas Amplifier and one step motor Atlas amplifier controlled by a multi-axis Magellan.

6.17.1 Atlas Power Input and Motor Output

Atlas is powered through pin pairs HV and Pwr_Gnd, and the power source is a transformer-isolated DC power supply. In this application the two Atlases share the same power supply. Alternatively they could be powered independently so that different motor voltages could be used.

For DC Brush motors pins MotorA and MotorB are wired to motor windings Motor+ and Motor-, respectively. Pins MotorC and MotorD are left un-connected.

For step motors pins MotorA, MotorB, MotorC and MotorD are wired to motor windings A+, A-, B+ and B-, respectively.

Please refer to Section 6.16.1, "Atlas Power Input and Motor Output," for layout and wiring recommendations on power input and motor outputs.

6.17.2 Atlas SPI Interface

Atlas receives control commands through an SPI interface and functions as an SPI slave. Atlas SPI communication is enabled when ~SPICS is pulled down. Only one Atlas can be enabled at any given time.

Please refer to Section 6.16.1, "Atlas Power Input and Motor Output," for layout recommendation on SPI interface.

6.17.3 Atlas ~Enable and FaultOut Signals

Atlas has one dedicated input signal, ~Enable, which must be pulled low for the Atlas output stage to be active.

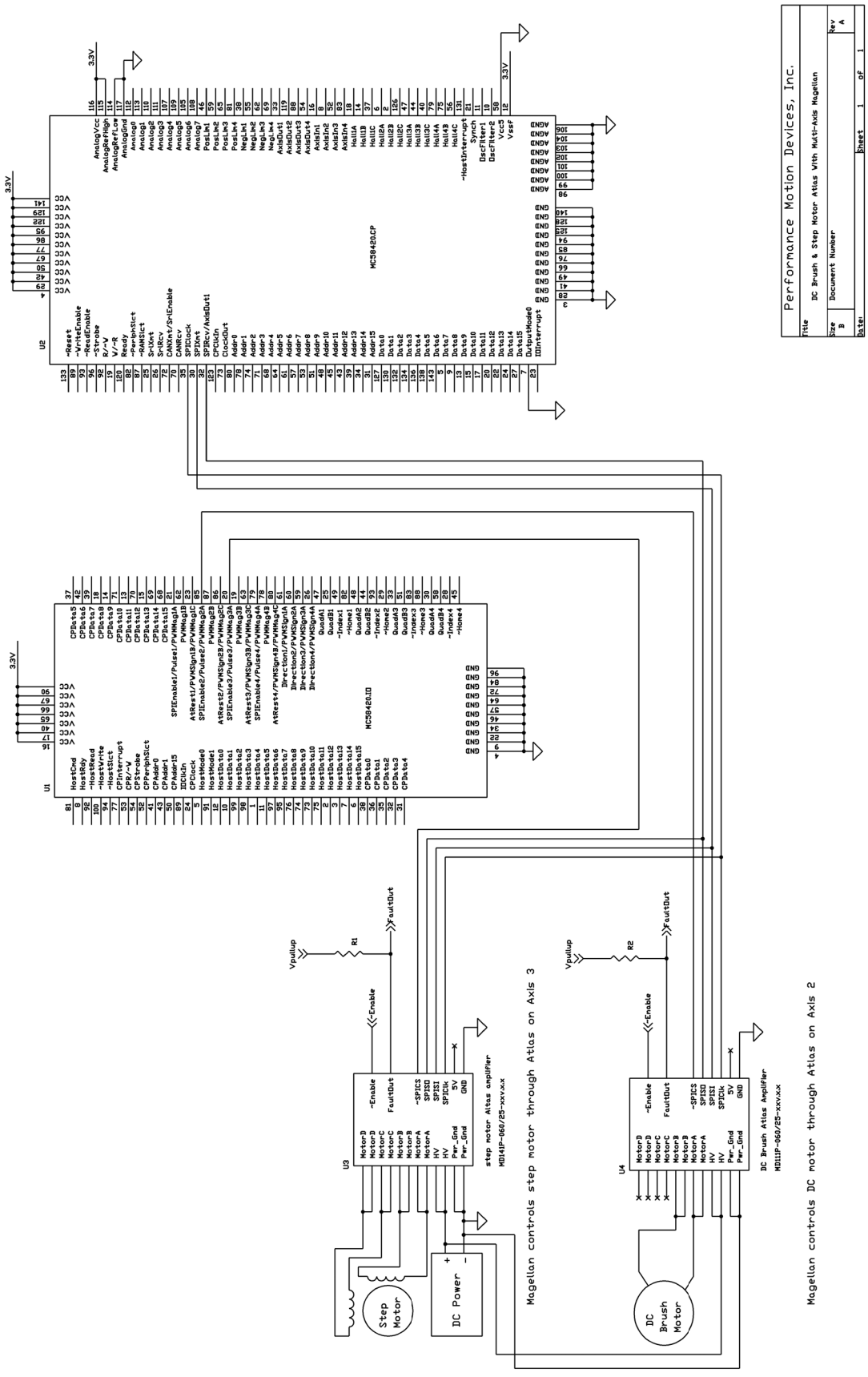
FaultOut is a dedicated output. During normal operation it outputs low. When a fault occurs it will go into a high impedance state. In this example, FaultOut is pulled up by Vpullup through resistor R1. Vpullup can be up to 24V to meet the system requirement. Each Atlas may use a different Vpullup voltage, for example, if the fault signal is wired to a 5V TTL input, Vpullup can be 5V.

6.17.4 Magellan MC58420 Configuration

In this schematic the SPI master is a four-axis Magellan MC58420. Only the connections with Atlas are shown. For complete Magellan wiring, please refer to the MC58420 electrical specifications.

The MC58420 is configured to default to Atlas motor output by tying pin 7, OutputMode0, to ground. In this example axis 2 and axis 3 are under control. The MC58420 sends torque commands to the DC Brush Atlas by pulling SPIEnable2 low, and sends position commands to the step motor Atlas by pulling SPIEnable3 low.

Figure 6-23: DC Brush & Step Motor Atlas With Multi-Axis Magellan



6.18 Pulse & Direction Mode Output Connected to Atlas

The following schematic shows Atlas operated in pulse & direction mode controlled by a single axis Magellan. Note that any source of pulse & direction signals, such as a microprocessor or other dedicated motion control IC, may be substituted for the Magellan in this schematic.

6.18.1 Atlas power input and motor outputs

Atlas is powered through pin pairs HV and Pwr_Gnd, and the power source is a transformer-isolated DC power supply.

For step motors pins MotorA, MotorB, MotorC and MotorD are wired to motor windings A+, A-, B+ and B-, respectively.

Please refer to Section 6.16.1, "Atlas Power Input and Motor Output," for layout and wiring recommendations on power input and motor outputs.

6.18.2 Atlas Pulse & Direction Interface

When in pulse & direction signal mode, Atlas receives pulse, direction and AtRest signals as shown in the schematic. When operated in pulse & direction signal mode SPI communications are not available.

6.18.3 Atlas ~Enable and FaultOut Signals

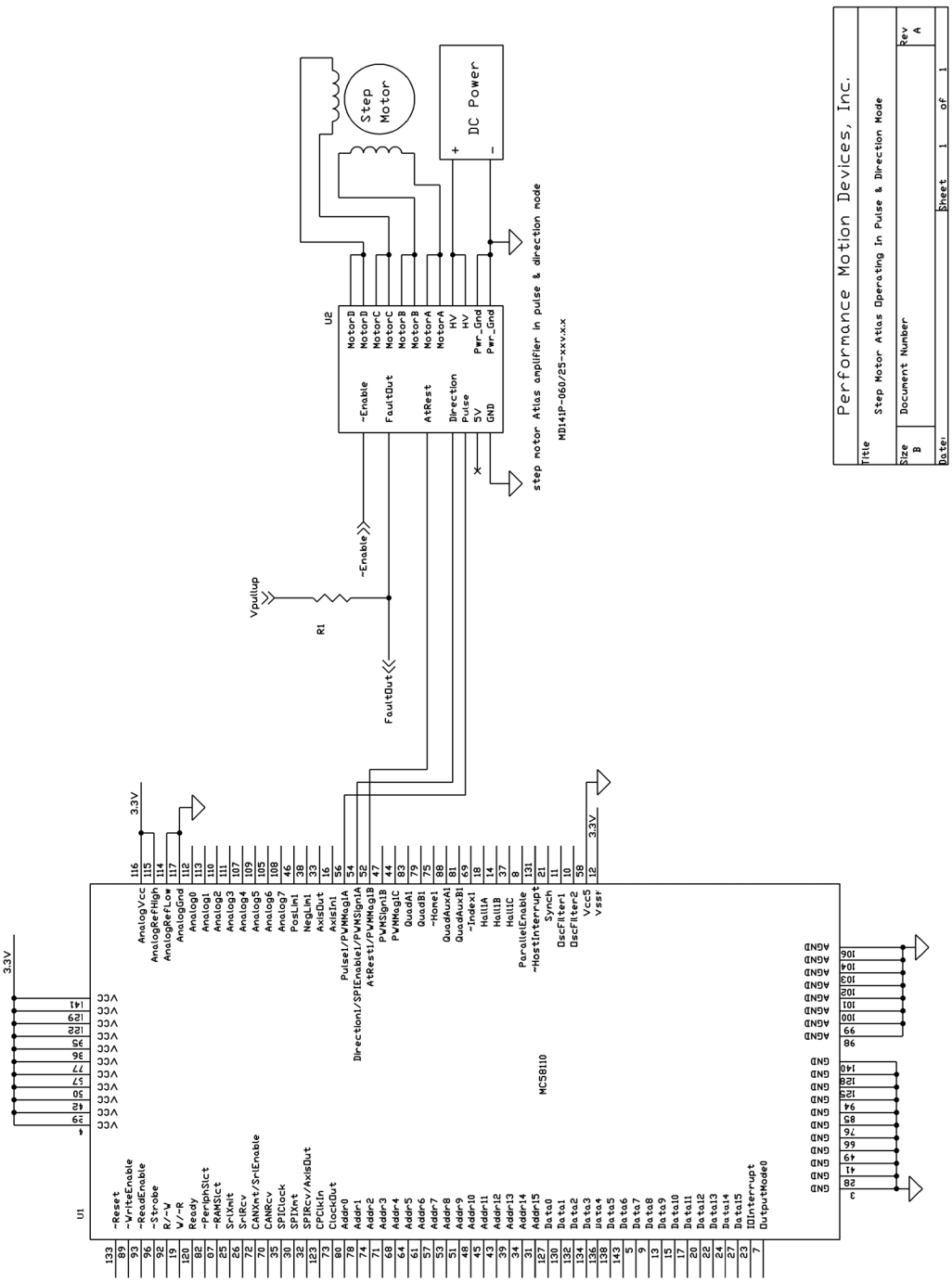
Atlas has one dedicated input signal, ~Enable, which must be pulled low for the Atlas output stage to be active.

FaultOut is a dedicated output. During normal operation it outputs low. When a fault occurs it will go into a high impedance state. In this example, FaultOut is pulled up by Vpullup through resistor R1. Vpullup can be up to 24V to meet the system requirement. For example, if the fault signal is wired to a 5V TTL input, Vpullup can be 5V.

6.18.4 Magellan MC58110 configuration

In this schematic the SPI master is a single-axis Magellan MC58110 configured for pulse & direction signal output. Only the connections with Atlas are shown. For complete Magellan wiring, please refer to the MC581100 electrical specifications.

Figure 6-24: Step Motor Atlas Operating In Pulse & Direction Mode



6.19 Using PWM for DC Brush, Brushless DC and Microstepping Motors

The MC58000 series can drive DC brush and brushless DC motors, with the use of the PWM signals. The PWM magnitude is a symmetric, 10-bit resolution, 19.531 kHz signal. The supplies for the different motor drivers are not shown in the schematics, and it is assumed that the ground is isolated in order to prevent interference from low-signal components.

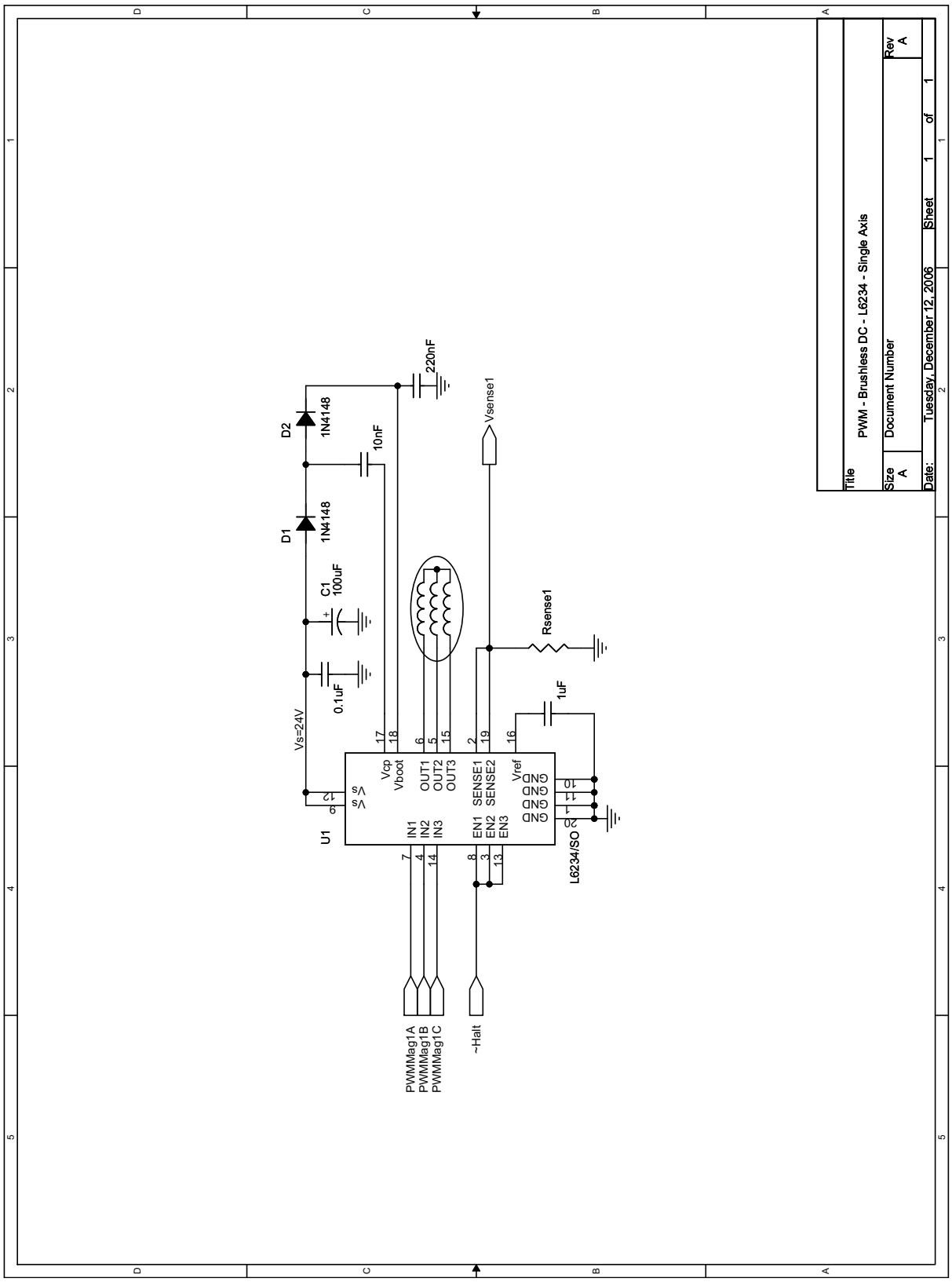
6.19.1 Using the ST L6234 to Drive Three-Phase Brushless DC Motors

In the following schematic (Figure 6-25), PWM 50/50 outputs are used to drive an L6234 three-phase motor driver. The TTL/CMOS input levels of the L6234 digital part are compatible with the outputs of the MC58000. If the power supply cannot sink the switching currents from the motor, a large capacitor should be added between the Vs input pin and ground. The schematic shows a 47 µF capacitor for a nominal 2A motor.

To detect malfunctions, the *Vsense* signal may be used to sense the amount of current flowing through the motor windings. For a nominal 2A driving current, an *Rsense* = 0.15Ω power resistor may be used to generate the halt signal required by the protection circuitry in Section 6.13, "Overcurrent and Emergency Braking Circuits for Motor Drivers," Only half of the *~OverCurrent* circuitry is required. The halt signal shorts the motor winding to ground. Other braking configurations may be implemented by altering the halt signal interface.

The schematic as shown can be duplicated for multi-axis control.

Figure 6-25: PWM, brushless DC, L6234



6.19.2 Using the National LMD18200 to Drive DC Brush Motors

In the following schematic (Figure 6-26), a magnitude and direction PWM signal is used to drive a DC brush motor with a nominal 24V, 2A drive. The H-bridge driver selected for this task is the LMD18200, which can be driven directly from a 3.3V CMOS logic output and as such can be directly interfaced to the MC58000.

There are two methods in which the output current of the H-bridge may be controlled.

First, in the *locked anti-phase control* mode (see the LMD18200 data sheet), a 50/50 PWM signal is applied to the LMD18200 DIR input, while the PWM input is tied high. The current ripple in this mode is relatively high, as the circulating currents are quickly decaying.

Second, in *the sign/magnitude control* mode, sign and magnitude PWM signals are applied to both the PWM and DIR inputs of the LMD18200. In this mode, the resultant current ripple is reduced resulting in smoother operation of the motor. When the acceleration/deceleration requirements for the motor are not high, the sign/magnitude PWM control mode is preferred. This method is demonstrated in the example.

The LMD18200 is equipped with an internal overcurrent circuit, which is tuned to a 10A threshold. External over-current circuitry may be added for currents with a lower threshold by using the sense output. In order to detect malfunctions, the ***Vsense*** signal may be used to sense the amount of current flowing through the motor windings. The sense output of the LMD18200 samples only a fraction of the drive current, with a typical 377 µA sensing per 1A driving current. For a nominal 2A driving current, an ***Rsense*** = 400Ω power resistor may be used with the circuitry in Section 6.13, "Overcurrent and Emergency Braking Circuits for Motor Drivers," (only half of the over current circuitry is required) to generate the halt signal. The halt signal sources both outputs. This is the recommended braking method, as the braking current goes through the upper pair of DMOS, which are connected to the internal over-current circuitry (refer to the LMD18200 data sheet for more information).

The schematic as shown can be duplicated for multi-axis control.

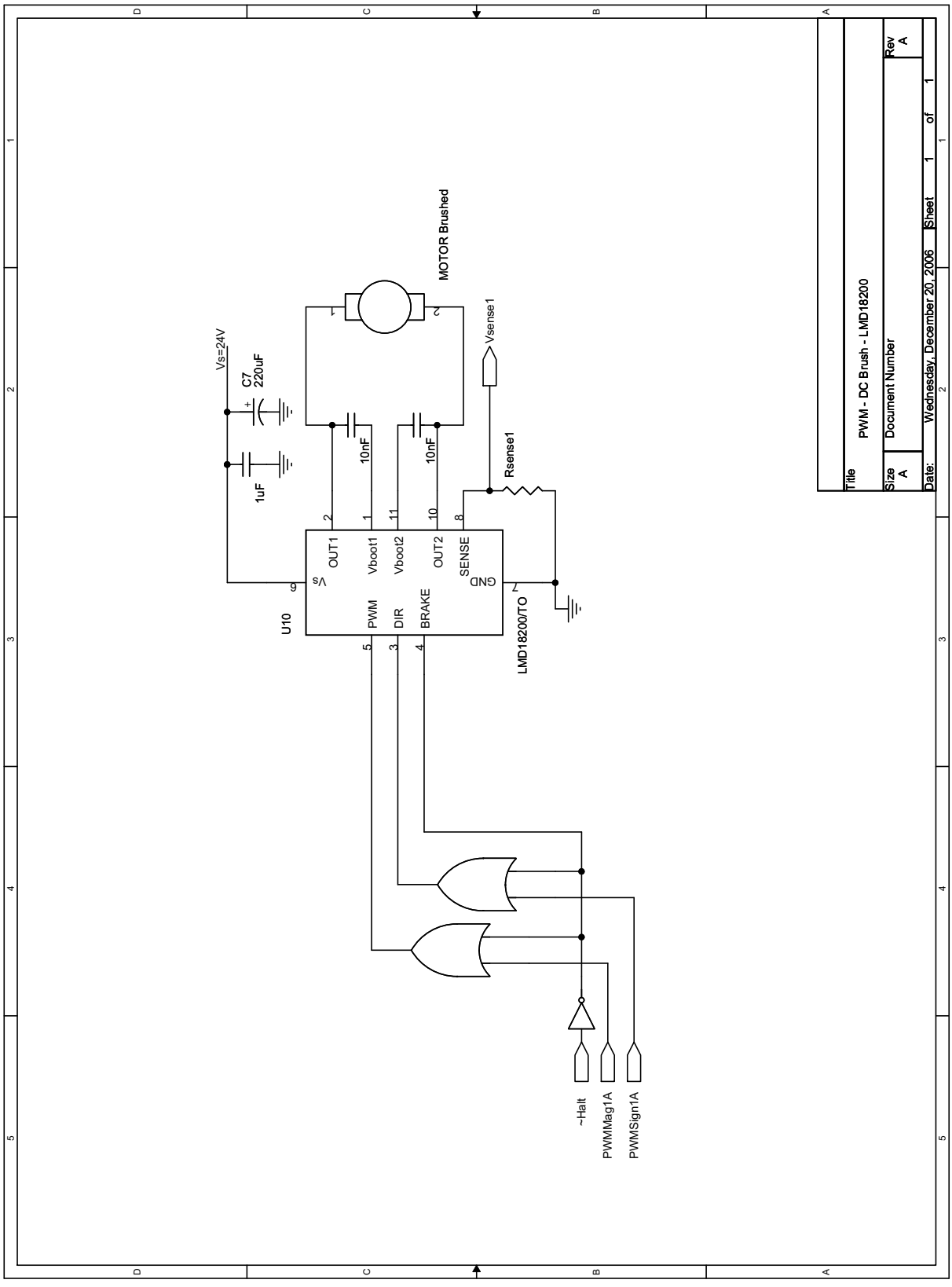


Figure 6-26: PWM, DC brush, LMD18200

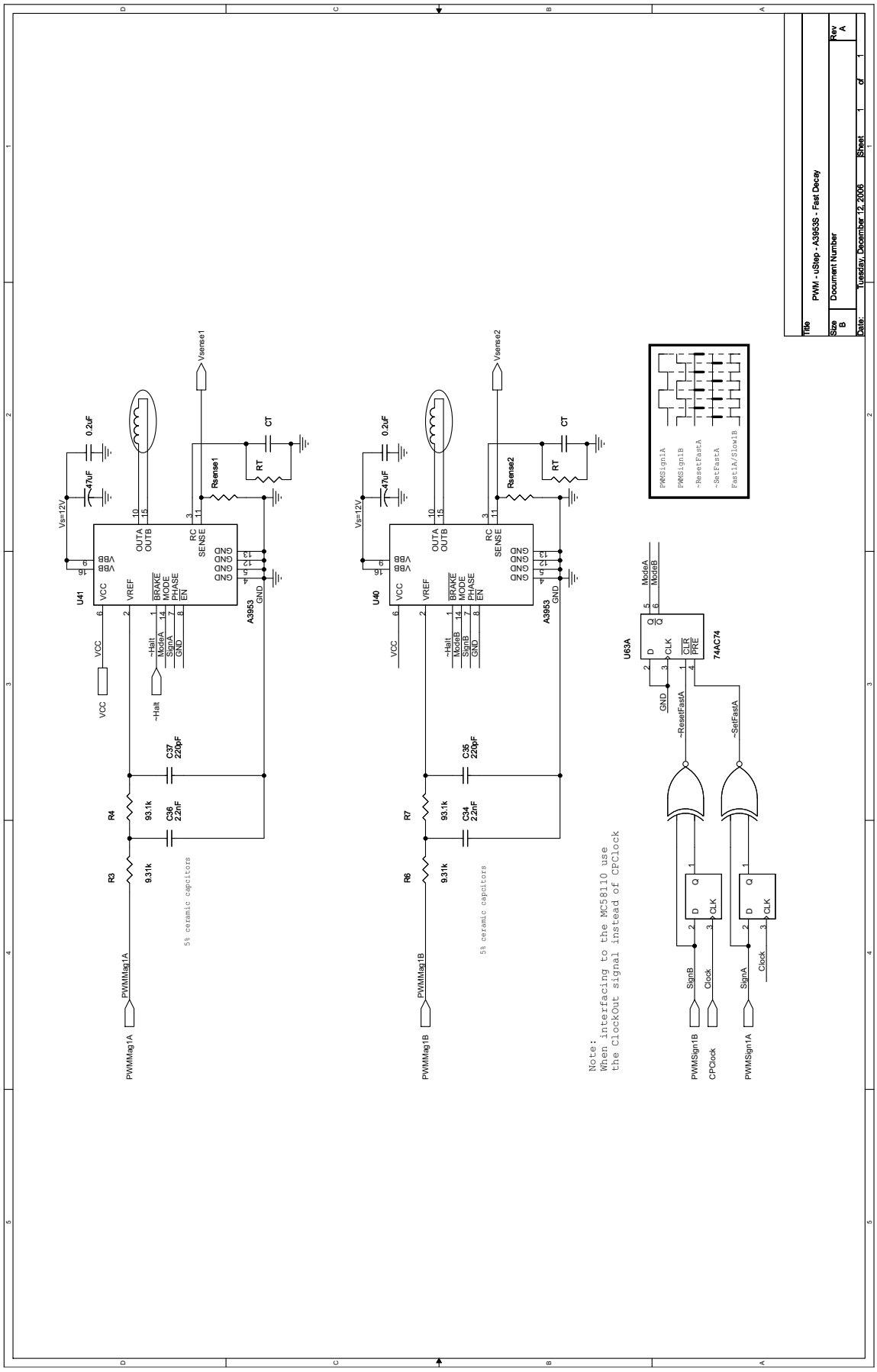
6.19.3 Using the Allegro A4973 to Drive Microstepping Motors

The A4973 is an H-bridge, and is designed to drive full-step motors. In order to achieve a micro step resolution, the reference voltage input to the A4973 is injected with a sinusoidal waveform. This method of interfacing to the driver requires that a low-pass filter be applied to the PWM magnitude signal in order to generate the analog equivalent of the PWM half sine waveform. To achieve a smooth equivalent signal, the PWM cycle frequency should be set to 80 kHz, using the SetPWMFrequency command.

The decay mode, either fast or slow, is controlled via the A4973 MODE input (see Figure 6-27). *PWMSignA* and *PWMSignB* signals are used to generate a decay mode pattern similar to the one shown in Section 6.19.1, "Using the ST L6234 to Drive Three-Phase Brushless DC Motors." Each *PWMSignA/B* signal is differentiated in order to detect its falling and rising edges. The differentiated signals are then applied to the asynchronous reset and set inputs of a D-FF to generate the *FastA/SlowB* signal. The *FastA/SlowB* signal, when high, indicates that Phase A and Phase B are in fast and slow decay modes, respectively.

The A4973 operation may be tuned with the use of external components. *CT* is used to determine the blanking period of the current sense comparator circuitry. The product of *RT* and *CT* is used to determine the PWM constant off period. Refer to the device data sheet for additional details. The sense resistor, *Rsense*, should be selected according to the maximum current intended to be flowing through the windings. Since the output current is controlled through *VREF*, the maximum voltage swing of *VREF* should be considered when the sense resistor value is calculated.

Figure 6-27: PWM, µStep, A3953S, fast decay



6.19.3.1 LPF Design

The PWM signal generated by the MC58000 has an 80 kHz cycle. With 256 resolution steps of 50 µsec each, it can encode sine waveform frequencies up to 500 Hz.

Figure 6-28 shows the spectra of the PWM signal encoded with a 150 Hz electrical cycle signal, superimposed with an ideal analog 150 Hz absolute magnitude sine wave (red). The PWM signal possesses energy at the PWM cycle frequency and its higher order harmonics. This energy is related to the PWM encoding waveform, which should be filtered out; the non-filtered portion of it will appear as ripple. The LPF goal is to pass the energy of the encoding signal, while suppressing the PWM waveform contributions.

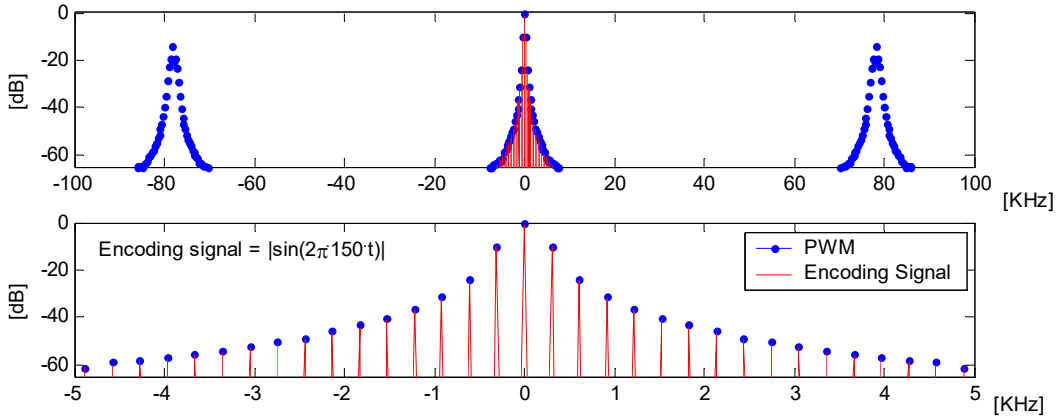


Figure 6-28: Encoded PWM signal spectra

Based on this figure, the filter should have a cut-off frequency at 5 kHz, and suppression of at least 40 dB at 78 kHz.

A second order passive filter is adequate for this task, as indicated in the following figures. Figure 6-32 shows a second-order RC filter frequency response, and Figure 6-33 shows the filter's output for an ideal 150 Hz electrical cycle PWM input.

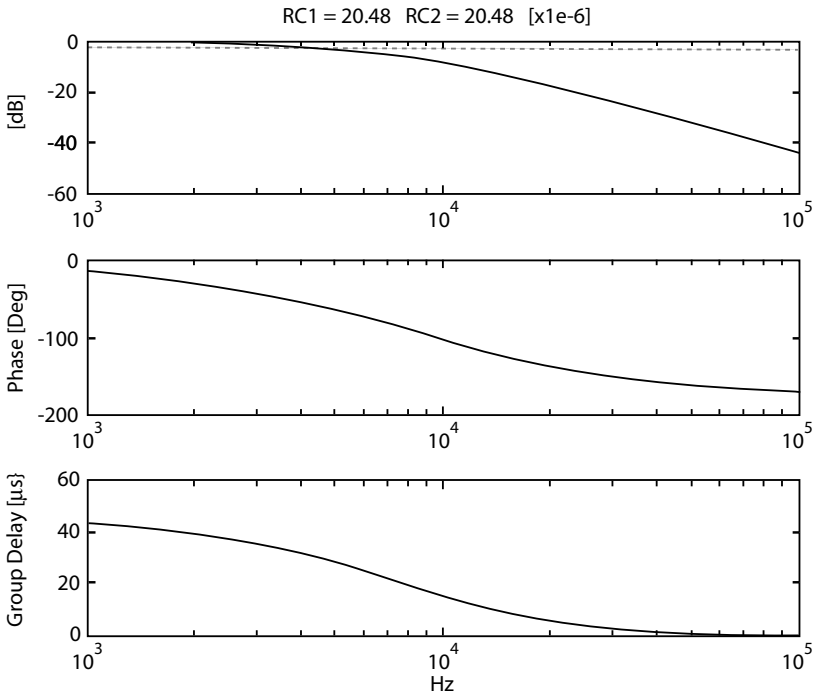
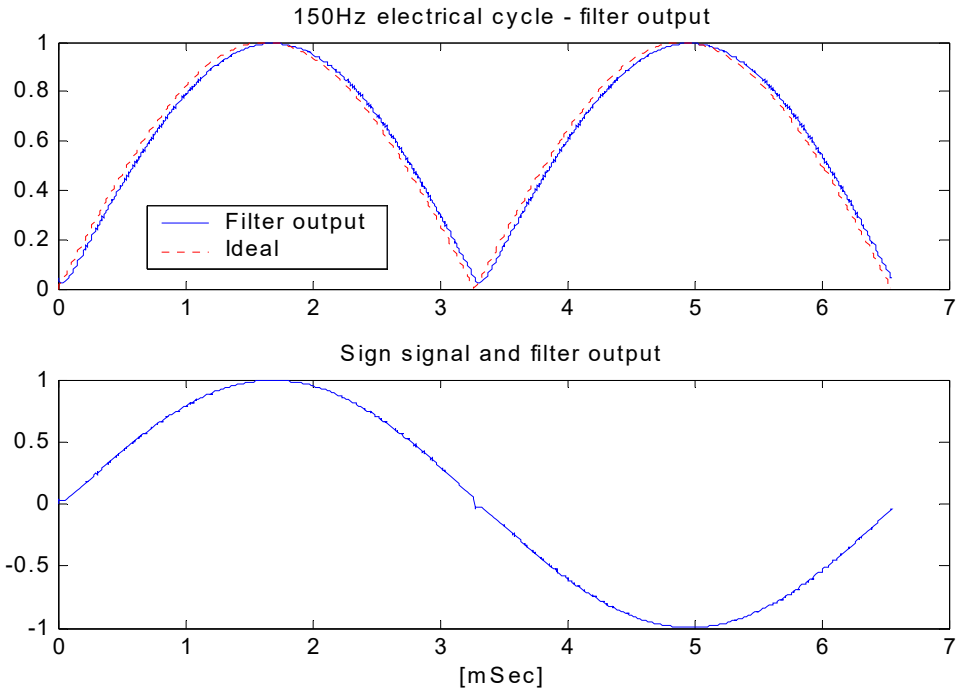


Figure 6-29: Filter frequency response

Figure 6-30: Filter output to 150 Hz electrical cycle



If a different filter is to be designed, the following points should be considered.

1. Reducing the cut-off frequency will result in a larger imperfection at the zero crossing point due to:
   a. The filtered curve at the zero crossing points will experience higher levels.
   b. The filter group-delay will be larger; thus increasing the mismatch between the sign signal and the filtered signal. This can be remedied by delaying the sign signal according to the filter group delay.
2. Increasing the cut-off frequency will reduce the suppression of the PWM waveform, resulting in larger ripple.
3. Increasing the order of the RC filter will result in a better waveform. Due to the slow roll-off of the filter, the improvement will probably be insignificant.

6.20 Using the Allegro A3977 to Drive Microstepping Motors

The A3977 is a complete microstepping motor driver with a built-in translator. The translator is capable of driving bi-polar stepper motors in full-, half-, quad-, and eighth-step modes. When the step input transitions from low to high, the A3977 will advance the motor one full-, half-, quad-, or eighth-step according to the configuration of the MS1 and MS2 pins. In the example the driver is configured for eighth-step resolution.

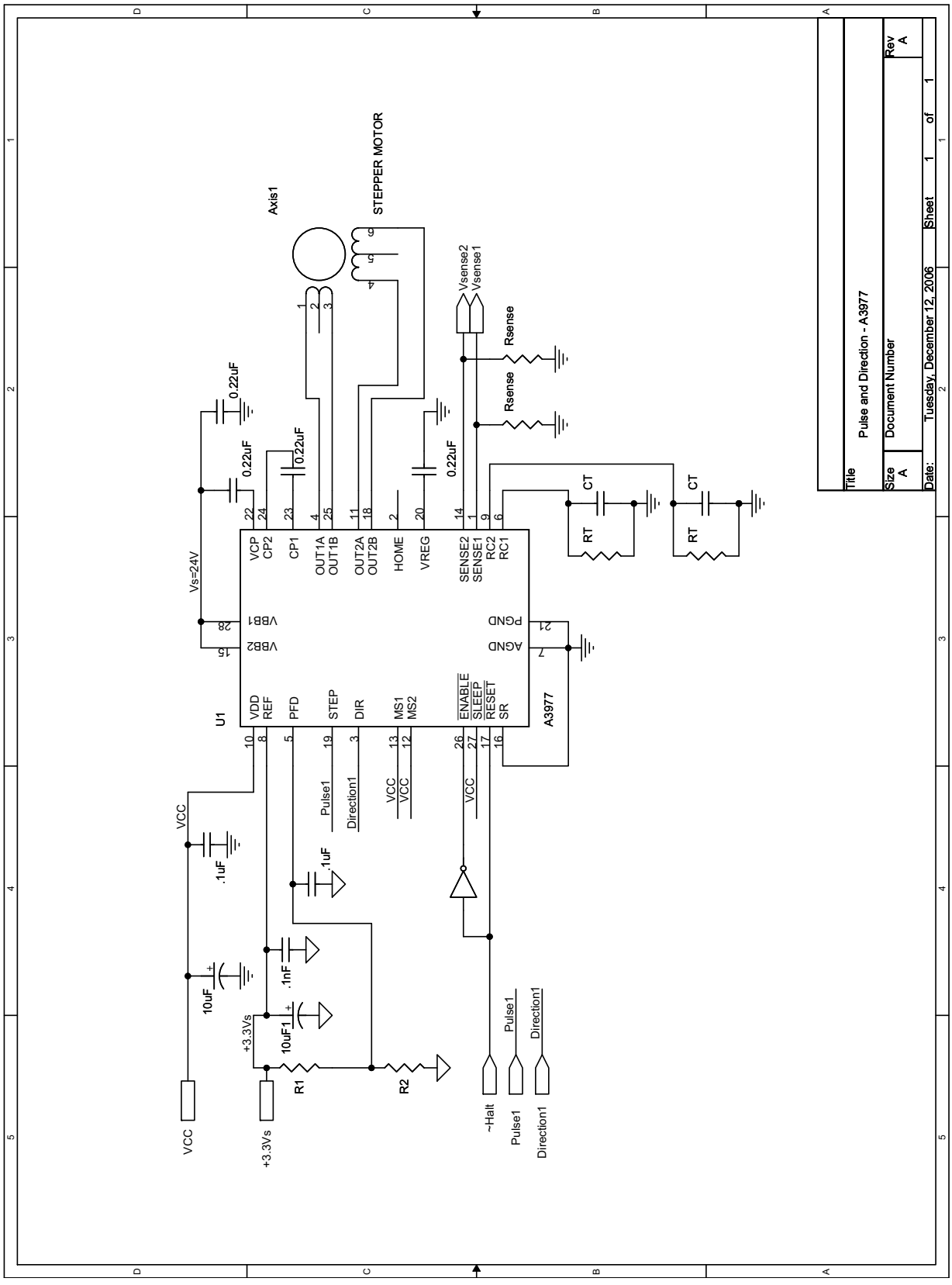
The A3977 operation can be tuned with the use of external components. *CT* is used to determine the blanking period of the current sense comparator circuitry. The product of *RT* and *CT* is used to determine the PWM constant off period. *R1* and *R2*, along with *RT* and *CT*, determine the percentage of the fast decay in mixed decay mode. The sense resistors, *Rsense*, should be selected according to the maximum current and voltage restrictions of the driver. Refer to the device data sheet for further information.

For a direct interface of the pulse signal to the step input, the polarity of the pulse signal must be inverted using the SetSignalSense command. This is required because the A3977 recognizes a step during a low-to-high transition of the step input signal, whereas the non-inverted behavior of the MC58000 is to generate a step on a high-to-low transition.

The pulse and direction outputs will satisfy the A3977 timing requirements if operated at a step rate of 155.625 kHz or less. This can be set using the **SetStepRange** command. The MC58110 has a maximum step rate of 97.6 kHz.

The schematic in Figure 6-31 uses the sense outputs to detect a malfunction by sensing the current through the motor windings. To generate the ***~Halt*** signal, the over-current circuitry should be configured with an ***Rsense*** = 0.15Ω power resistor for a rated 2A motor.

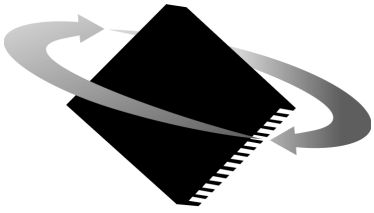
Figure 6-31: Pulse and direction, A3977



For additional information, or for technical assistance, please contact PMD at (978) 266-1210.

You may also e-mail your request to support@pmdcorp.com

Visit our website at https://www.pmdcorp.com



P M D

Performance Motion Devices
80 Central Street
Boxborough, MA 01719