High Performance DSP-Based Servo Motor Controller

ADMC300

FEATURES

25 MIPS Fixed-Point DSP Core
- Single Cycle Instruction Execution (40 ns)
- ADSP-2100 Family Code Compatible
- Independent Computational Units
  - ALU
  - Multiplier/Accumulator
  - Barrel Shifter
- Multifunction Instructions
- Single Cycle Context Switch
- Powerful Program Sequencer
  - Zero Overhead Looping
  - Conditional Instruction Execution
- Two Independent Data Address Generators

Memory Configuration
- 4K × 24-Bit Program Memory RAM
- 2K × 24-Bit Program Memory ROM
- 1K × 16-Bit Data Memory RAM

High-Resolution Multichannel ADC System
- Five Independent 16-Bit Sigma-Delta ADCs
- 76 dB SNR Typical (ENOB > 12 Bits)
- Arranged in Two Independently Clocked Banks
- Differential or Single-Ended Inputs
- Programmable Sample Frequency to 32.5 kHz
- Flexible Synchronization of ADC and PWM Subsystems
- Independent Offset Calibration for Each Channel
- Two Dedicated ADC Interrupts
- Internal 2.5 V Reference
- Three Multiplexer Control Pins for External Expansion
- Hardware or Software Convert Start
- Individual Power-Down for Each Bank

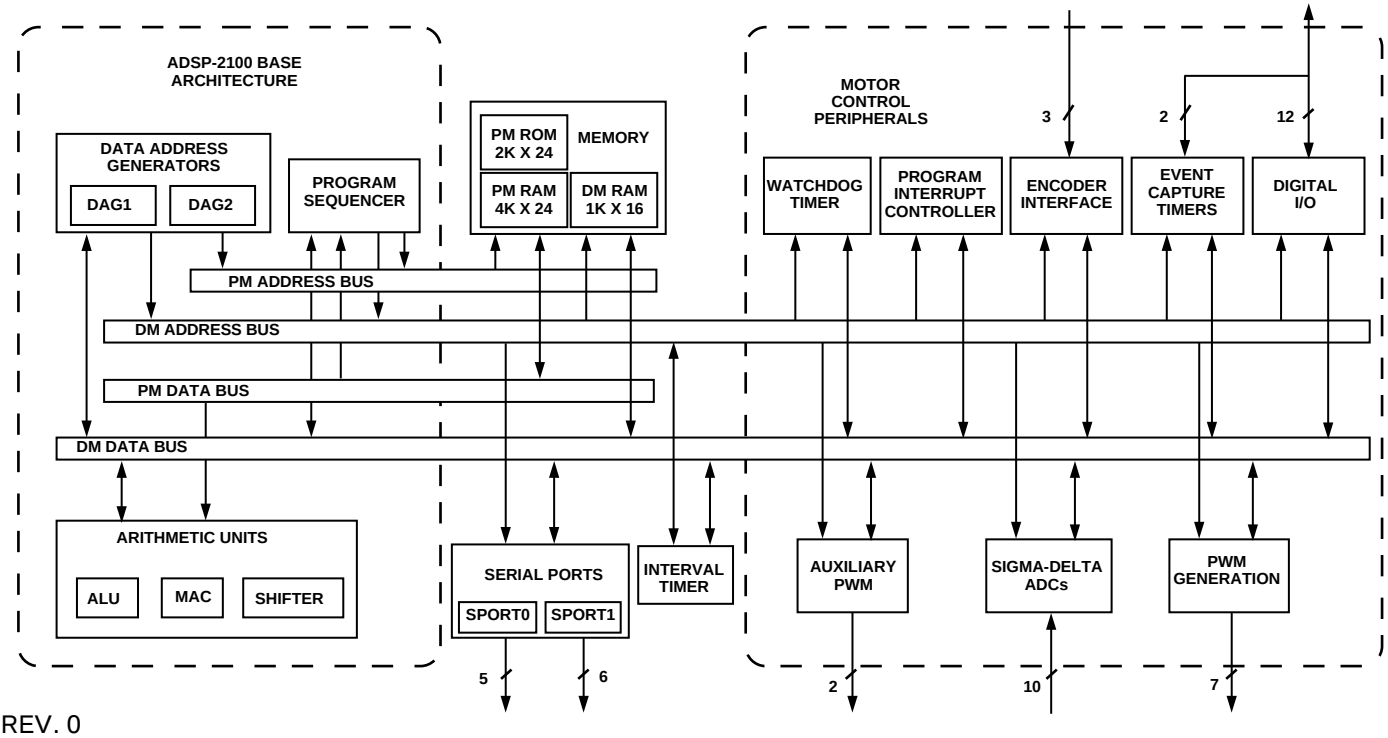
Three-Phase PWM Generation Subsystem
- 12-Bit Center-Based PWM Generator
- Edge Resolution to 80 ns
- Programmable Dead-Time and Narrow Pulse Deletion
- Suitable for AC Induction and Synchronous Motors
- Special Crossover Function for Brushless DC Motors
- Hardwired Polarity Control
- External $\overline{\text{PWMTRIP}}$ Pin
- Additional PWMTRIP Sources in Peripheral I/O System
- Individual Enable and Disable for All PWM Outputs
- High-frequency Chopping Mode for Transformer Coupled Gate Drives

Programmable Interrupt Controller
- Manages Priority and Masking of 11 Peripheral Interrupts

Flexible Encoder Interface Subsystem
- Incremental Encoder Interface
- 16-Bit Quadrature Counter
- Input Signals to 3.1 MHz
- Digital Filtering of Input Signals

(Continued on Page 7)

FUNCTIONAL BLOCK DIAGRAM



REV. 0

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 **World Wide Web Site: http://www.analog.com**
Fax: 781/326-8703

ADMC300–SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS

(V_{DD} = AV_{DD} = 5 V ± 10%, GND = AGND = 0 V, T_{AMB} = –40°C to +85°C, CLKIN = 12.5 MHz, unless otherwise noted)

Parameter		B Grade Min	B Grade Max	Units
V_{DD}	Digital Supply Voltage	4.50	5.50	V
AV_{DD}	Analog Supply Voltage	4.50	5.50	V
T_{AMB}	Ambient Operating Temperature	–40	+85	°C

ELECTRICAL CHARACTERISTICS

Parameter		Test Conditions	Min	Max	Unit
V_{IH}	Hi-Level Input Voltage[1, 2]	@ V_{DD} = Max	2.0		V
V_{IL}	Lo-Level Input Voltage[1, 2]	@ V_{DD} = Min		0.8	V
V_{OH}	Hi-Level Output Voltage[1, 3]	@ V_{DD} = Min, I_{OH} = –1.0 mA	2.4		V
		@ V_{DD} = Min, I_{OH} = –0.1 mA	V_{DD} – 0.3		V
V_{OL}	Lo-Level Output Voltage[1, 3]	@ V_{DD} = Min, I_{OL} = 2.0 mA		0.4	V
I_{IH}	Hi-Level Input Current[4]	@ V_{DD} = Max, V_{IN} = V_{DD} Max		10	µA
I_{IH}	Hi-Level $\overline{\text{PWMTRIP}}$, PIO0–PIO11 Current	@ V_{DD} = Max, V_{IN} = V_{DD} Max		100	µA
I_{IH}	Hi-Level PWMPOL Current	@ V_{DD} = Max, V_{IN} = V_{DD} Max		10	µA
I_{IL}	Lo-Level Input Current[4]	@ V_{DD} = Max, V_{IN} = 0 V		10	µA
I_{IL}	Lo-Level $\overline{\text{PWMTRIP}}$, PIO0–PIO11 Current	@ V_{DD} = Max, V_{IN} = 0 V		10	µA
I_{IL}	Lo-Level PWMPOL Current	@ V_{DD} = Max, V_{IN} = 0 V		100	µA
I_{OZH}	Hi-Level Three-State Leakage Current[5]	@ V_{DD} = Max, V_{IN} = V_{DD} Max		10	µA
I_{OZL}	Lo-Level Three-State Leakage Current[5]	@ V_{DD} = Max, V_{IN} = 0 V		10	µA
I_{DD}	Digital Power Supply Current (Dynamic)[6, 7]	@ V_{DD} = Max		100	mA
I_{DD}	Analog Power Supply Current (Disabled)[8]	@ AV_{DD} = V_{DD} = Max		1.5	mA
I_{DD}	Analog Power Supply Current (Ref Only)	@ AV_{DD} = V_{DD} = Max		6.5	mA
I_{DD}	Analog Power Supply Current (Ref + BankA)	@ AV_{DD} = V_{DD} = Max		11.0	mA
I_{DD}	Analog Power Supply Current (Ref + BankB)	@ AV_{DD} = V_{DD} = Max		13.0	mA
I_{DD}	Analog Power Supply Current (Ref + BankA/B)	@ AV_{DD} = V_{DD} = Max		18.0	mA

NOTES

[1]Bidirectional pins: PIO0–PIO11, RFS0, RFS1, TFS0, TFS1, SCLK0, SCLK1.

[2]Input only pins: $\overline{\text{PWMTRIP}}$, PWMPOL, $\overline{\text{RESET}}$, EIA, EIB, EIZP, DR1A, DR1B, DR0, CLKIN.

[3]Output pins: PWMSYNC, CL, CH, BL, BH, AL, AH, MUX0–MUX2, AUX0, AUX1, CLKOUT, DT0, DT1.

[4]Input only pins: $\overline{\text{RESET}}$, EIA, EIB, EIZP, DR1A, DR1B, DR0, CLKIN.

[5]Three-stateable pins: DT0, DT1, RFS0, RFS1, TFS0, TFS1, SCLK0, SCLK1.

[6]Current reflects device operating with no output loads.

[7]Dynamic condition refers to continuous operation of the DSP core, ADC banks and PWM generation with PWMTM = 0x0480, ADCDIVA = ADCDIVB = 0x180. The encoder inputs are quiescent.

[8]Disabled refers to powering down both ADC banks and the internal reference generation circuit by setting Bits 10, 11 and 12 of the ADCCTRL register. Current is total current from AV_{DD} supply.

Specifications subject to change without notice.

ANALOG-TO-DIGITAL CONVERTER

(V_{DD} = AV_{DD} = 5 V ± 10%, GND = AGND = 0 V, V_{REFIN} = 2.50 V, T_{AMB} = –40°C to +85°C, CLKIN = 12.5 MHz, unless otherwise noted)

Parameter		Test Conditions	Min	Typ	Max	Unit
Signal-to-Noise Ratio[1] (SNR)		@V_{DD} = 5.0 V,	72	76		dB
Total Harmonic Distortion[1] (THD)		@ f_S = 32.55 kHz,			–70	dB
Common-Mode Rejection Ratio[2] (CMRR)		@ f_{IN} = 1.017 kHz,			–82	dB
Channel-Channel Crosstalk[3]		ADCDIVn = 0x180,			–76	dB
Gain Error		V1–V5 = 4.0 V p-p			5	%
Gain		V1N–V5N = V_{REFIN} = 2.5 V		10,600		LSB/V
V_{IN}	Analog Input Range[4]		0		V_{DD}	V
V_{DIFF}	Analog Input Voltage (Differential)[4]				$V_{DD}/2$	V
V_{OFFSET}	DC Offset Voltage[5]				55	mV
$f_{MOD, MAX}$	Maximum Sigma-Delta Modulator Rate	ADCDIVA = 0x180 ADCDIVB = 0x180			2.08	MHz
$f_{S, MAX}$	Maximum ADC Sample Rate[6]	ADCDIVA = 0x180 ADCDIVB = 0x180			32.55	kHz
V_{REFIN}	Reference Input Voltage[7]		2.4	2.5	2.6	V
R_{IN}	Equivalent Input Resistance[8]			25		kΩ

NOTES
[1]SNR measured with ADC channel configured in single-ended mode. SNR measurement does not include harmonic distortion, THD includes first six harmonics. The effective number of bits (ENOB) is related to the SNR by SNR = 6.02 (ENOB) +1.76 dB. Input signal filtered at 1.5 kHz.
[2]Input signal applied to both pins of input differential pair of ADC channel.
[3]Input signal applied to four ADC channels, dc applied to fifth, measurement taken at fifth ADC channel.
[4]Peak-peak input voltage in differential input configuration is half that in single-ended mode.
[5]This offset may be corrected for, using the ADC calibration feature.
[6]At maximum sigma-delta modulator rate of 2.08 MHz.
[7]Input reference pins: REFINA, REFINB.
[8]Analog signal input pins: V1–V5, V1N–V5N.
Specifications subject to change without notice.

VOLTAGE REFERENCE

(V_{DD} = AV_{DD} = 5 V ± 10%, GND = AGND = 0 V, T_{AMB} = –40°C to +85°C, CLKIN = 12.5 MHz, unless otherwise noted)

Parameter		Test Conditions	Min	Typ	Max	Unit
V_{REF}	Voltage Level		2.25		2.75	V
	Source Current				–100	µA
Power Supply Rejection Ratio (PSRR)			–5		5	mV/V

Specifications subject to change without notice.

PULSEWIDTH MODULATOR

(V_{DD} = AV_{DD} = 5 V ± 10%, GND = AGND = 0 V, T_{AMB} = –40°C to +85°C, CLKIN = 12.5 MHz, unless otherwise noted)

Parameter		Test Conditions	Min	Typ	Max	Unit
	Counter Resolution[1]				12	Bits
	Edge Resolution			80		ns
T_D	Programmable Dead Time		0		10.24	µs
	Programmable Dead Time Increments			160		ns
T_{MIN}	Programmable Pulse Deletion		0		10.24	µs
	Programmable Deletion Increments			80		ns
f_{PWM}	PWM Frequency Range[1]		3.05			kHz
T_{SYNC}	PWMSYNC Pulsewidth			1.6		µs
f_{CHOP}	Gate Drive Chop Frequency		0.098		6.25	MHz

NOTES
[1]Resolution varies with PWM switching frequency, 3.05 kHz = 12 bits, 48.8 kHz = 8 bits (12.5 MHz CLKIN).

Specifications subject to change without notice.

ADMC300–SPECIFICATIONS

ENCODER INTERFACE UNIT (V_{DD} = AV_{DD} = 5 V ± 10%, GND = AGND = 0 V, T_{AMB} = –40°C to +85°C, CLKIN = 12.5 MHz, unless otherwise noted)

Parameter		Test Conditions	Min	Typ	Max	Unit
$f_{ENC, MAX}$	Maximum Encoder Pulse Rate				3.1	MHz

Specifications subject to change without notice.

AUXILIARY PWM OUTPUTS (V_{DD} = AV_{DD} = 5 V ± 10%, GND = AGND = 0 V, T_{AMB} = –40°C to +85°C, CLKIN = 12.5 MHz, unless otherwise noted)

Parameter		Test Conditions	Min	Typ	Max	Unit
	Resolution			8		Bits
f_{AUXPWM}	Switching Frequency			48.8		kHz

Specifications subject to change without notice.

TIMING PARAMETERS

Parameter		Min	Max	Unit
Clock Signals				
t_{CK} is defined as 0.5 t_{CKI}. The ADMC300 uses an input clock with a frequency equal to half the instruction rate; a 12.5 MHz input clock (which is equivalent to 80 ns) yields a 40 ns processor cycle (equivalent to 25 MHz). t_{CK} values within the range of 0.5 t_{CKI} period should be substituted for all relevant timing parameters to obtain specification value. Example: t_{CKH} = 0.5 t_{CK} – 10 ns = 0.5 (40 ns) – 10 ns = 10 ns.				
Timing Requirements:				
t_{CKI}	CLKIN Period	80	150	ns
t_{CKIL}	CLKIN Width Low	20		ns
t_{CKIH}	CLKIN Width High	20		ns
Switching Characteristics:				
t_{CKL}	CLKOUT Width Low	0.5 t_{CK} – 10		ns
t_{CKH}	CLKOUT Width High	0.5 t_{CK} – 10		ns
t_{CKOH}	CLKIN High to CLKOUT High	0	20	ns
Control Signals				
Timing Requirement:				
t_{RSP}	RESET Width Low	5 t_{CK}[1]		ns

NOTE
[1]Applies after power-up sequence is complete. Internal phase lock loop requires no more than 2000 CLKIN cycles assuming stable CLKIN (not including crystal oscillator start-up time).

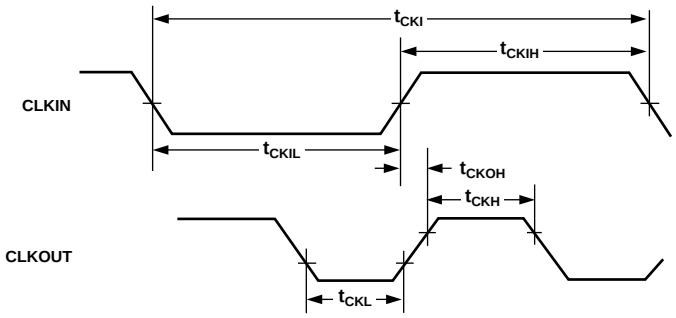


Figure 1. Clock Signals

Parameter		Min	Max	Unit
Serial Ports				
Timing Requirements:				
t_{SCK}	SCLK Period	50		ns
t_{SCS}	DR/TFS/RFS Setup before SCLK Low	5		ns
t_{SCH}	DR/TFS/RFS Hold after SCLK Low	10		ns
t_{SCP}	$SCLK_{IN}$ Width	20		ns
Switching Characteristics:				
t_{CC}	CLKOUT High to $SCLK_{OUT}$	0.25 t_{CK}	0.25 t_{CK} + 15	ns
t_{SCDE}	SCLK High to DT Enable	0		ns
t_{SCDV}	SCLK High to DT Valid		20	ns
t_{RH}	TFS/RFS_{OUT} Hold after SCLK High	0		ns
t_{RD}	TFS/RFS_{OUT} Delay from SCLK High		20	ns
t_{SCDH}	DT Hold after SCLK High	0		ns
t_{SCDD}	SCLK High to DT Disable		20	ns

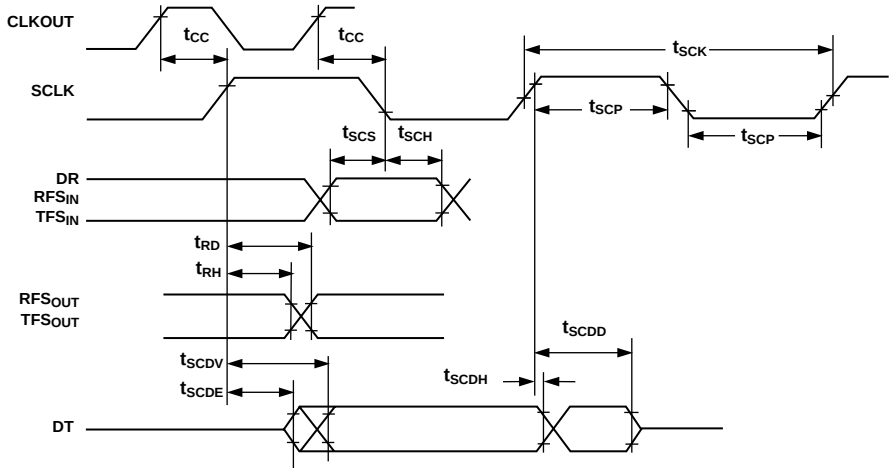


Figure 2. Serial Ports

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (V_{DD}) . . . . . –0.3 V to + 7.0 V
Supply Voltage (AV_{DD}) . . . . . –0.3 V to +7.0 V
Input Voltage . . . . . –0.3 V to V_{DD} + 0.3 V
Output Voltage Swing . . . . . –0.3 V to V_{DD} + 0.3 V
Operating Temperature Range (Ambient) . . . –40°C to +85°C
Storage Temperature Range . . . . . –65°C to +150°C
Lead Temperature (5 sec) . . . . . +280°C

*Stresses greater than those listed above may cause permanent damage to the device. These are stress ratings only; functional operation of the device at these or any other conditions greater than those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ORDERING GUIDE

Model	Temperature Range	Instruction Rate	Package Description	Package Option
ADMC300BST	–40°C to +85°C	25 MHz	80-Lead Plastic Thin Quad Flatpack (TQFP)	ST-80

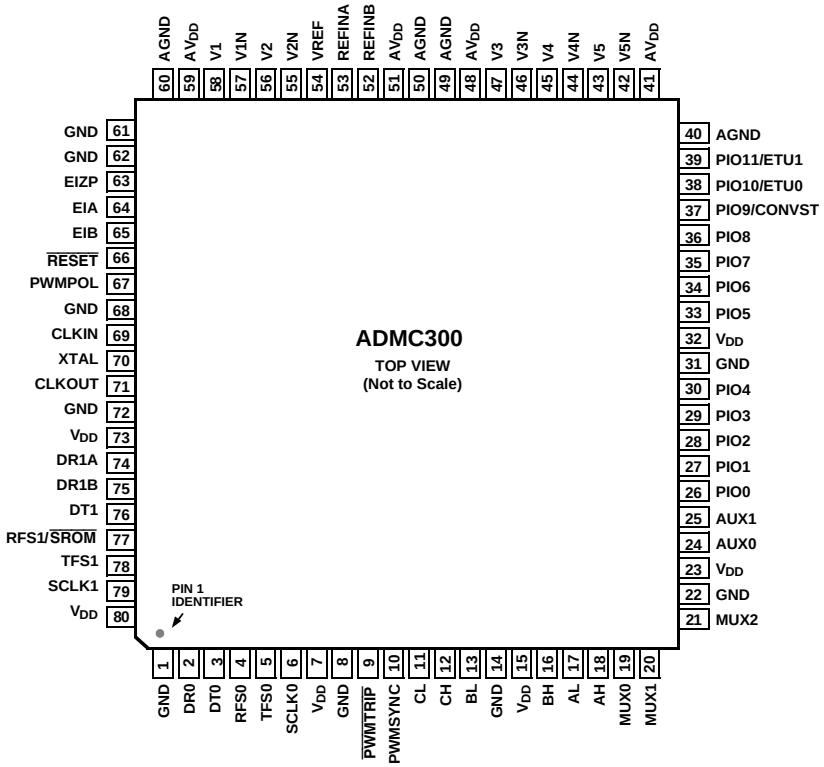
CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADMC300 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

PIN FUNCTION DESCRIPTIONS

Pin No.	Pin Type	Pin Name	Pin No.	Pin Type	Pin Name	Pin No.	Pin Type	Pin Name	Pin No.	Pin Type	Pin Name
1	GND	GND	21	O/P	MUX2	41	SUP	AV_{DD}	61	GND	GND
2	I/P	DR0	22	GND	GND	42	I/P	V5N	62	GND	GND
3	O/P	DT0	23	SUP	V_{DD}	43	I/P	V5	63	I/P	EIZP
4	BIDIR	RFS0	24	O/P	AUX0	44	I/P	V4N	64	I/P	EIA
5	BIDIR	TFS0	25	O/P	AUX1	45	I/P	V4	65	I/P	$\overline{\text{EIB}}$
6	BIDIR	SCLK0	26	BIDIR	PIO0	46	I/P	V3N	66	I/P	$\overline{\text{RESET}}$
7	SUP	V_{DD}	27	BIDIR	PIO1	47	I/P	V3	67	I/P	PWMPOL
8	GND	GND	28	BIDIR	PIO2	48	SUP	AV_{DD}	68	GND	GND
9	I/P	$\overline{\text{PWMTRIP}}$	29	BIDIR	PIO3	49	GND	AGND	69	I/P	CLKIN
10	O/P	PWMSYNC	30	BIDIR	PIO4	50	GND	AGND	70	O/P	XTAL
11	O/P	CL	31	GND	GND	51	SUP	AV_{DD}	71	O/P	CLKOUT
12	O/P	CH	32	SUP	V_{DD}	52	I/P	REFINB	72	GND	GND
13	O/P	BL	33	BIDIR	PIO5	53	I/P	REFINA	73	SUP	V_{DD}
14	GND	GND	34	BIDIR	PIO6	54	O/P	VREF	74	I/P	DR1A
15	SUP	V_{DD}	35	BIDIR	PIO7	55	I/P	V2N	75	I/P	DR1B
16	O/P	BH	36	BIDIR	PIO8	56	I/P	V2	76	O/P	DT1
17	O/P	AL	37	BIDIR	PIO9/CONVST	57	I/P	V1N	77	BIDIR	RFS1/$\overline{\text{SROM}}$
18	O/P	AH	38	BIDIR	PIO10/ETU0	58	I/P	V1	78	BIDIR	TFS1
19	O/P	MUX0	39	BIDIR	PIO11/ETU1	59	SUP	AV_{DD}	79	BIDIR	SCLK1
20	O/P	MUX1	40	GND	AGND	60	GND	AGND	80	SUP	V_{DD}

PIN CONFIGURATION
80-Lead Plastic Thin Quad Flatpack (TQFP)
(ST-80)



(Continued from Page 1)

- **Self-Determination of Encoder Resolution**
- **Position Counter Error Checking**
- **Programmable Direction Reversal of Counter**
- **Companion Encoder Event Timer**

Peripheral I/O (PIO) Subsystem
- **12-Pin Digital I/O Port**
- **Bit Configurable as Input or Output**
- **Each Pin Configurable as Rising Edge, Falling Edge, High Level or Low Level Interrupt**
- **Four Dedicated PIO Interrupts for PIO0 to PIO3**
- **One Combined Interrupt for PIO4 to PIO11**
- **Each I/O Line Configurable as PWM Trip Source**

Two 8-Bit Auxiliary PWM Outputs
- **Synthesized Analog Output**
- **Fixed 48.8 kHz Operation**
- **0 to 99.6% Duty Cycle**

Event Timer Unit
- **Two Event Timer Channels with Dedicated Event Capture Blocks**
- **Permits Timing of Duty-Cycle, Period and Frequency**
- **Configurable Event Definition**
- **Dedicated Event Timer Interrupt**

16-Bit Watchdog Timer

Programmable 16-Bit Interval Timer with Prescaler

Two Double Buffered Synchronous Serial Ports
- **Two Boot Load Protocols via SPORT1**
  - **E^2PROM/SROM Booting**
  - **UART (SCI Compatible) with Autobaud Feature**
- **UART Debugger Interface via SPORT1 with Autobaud**

ROM Utilities
- **Full Debugger for Program Development**
- **Preprogrammed Math Functions**
- **Preprogrammed Motor Control Functions – Vector Transformations**

80-Lead TQFP Package

Industrial Temperature Range –40°C to +85°C

GENERAL DESCRIPTION

The ADMC300 is single-chip DSP-based controller, suitable for high performance control of ac induction motors, permanent magnet synchronous motors and brushless dc motors. The ADMC300 integrates a 25 MIPS, fixed-point DSP core with a complete set of motor control peripherals that permits fast, efficient development of servo motor controllers.

The DSP core of the ADMC300 is the ADSP-2171, which is completely code compatible with the ADSP-2100 DSP family and combines three computational units, data address generators and a program sequencer. The computational units comprise an ALU, a multiplier/accumulator (MAC) and a barrel shifter. The ADSP-2171 adds new instructions for bit manipulation, multiplication (X squared), biased rounding and global interrupt masking. In addition, two flexible, double-buffered, bidirectional, synchronous serial ports are included in the ADMC300.

The ADMC300 provides 4K × 24-bit program memory RAM, 2K × 24-bit program memory ROM and 1K × 16-bit data memory RAM. The program and data memory RAM can be boot loaded through the serial port from either a serial SROM/ E^2PROM or through a UART connection. The program memory ROM includes a monitor that adds software debugging features through the serial port. In addition, a number of preprogrammed mathematical and motor control functions are included in the program memory ROM.

The motor control peripherals of the ADMC300 comprise a high performance, five channel ADC system that uses sigma-delta conversion technology offering a typical signal-to-noise ratio (SNR) of 76 dB, equivalent to 12 bits. In addition, a 12-bit center-based PWM generation unit can be used to produce high accuracy PWM signals with minimal processor overhead. The ADMC300 also contains a flexible encoder interface unit for position sensor feedback, two auxiliary PWM outputs, twelve lines of digital I/O, a two-channel event capture system, a 16-bit watchdog timer, a 16-bit interval timer and a programmable interrupt controller that manages all peripheral interrupts.

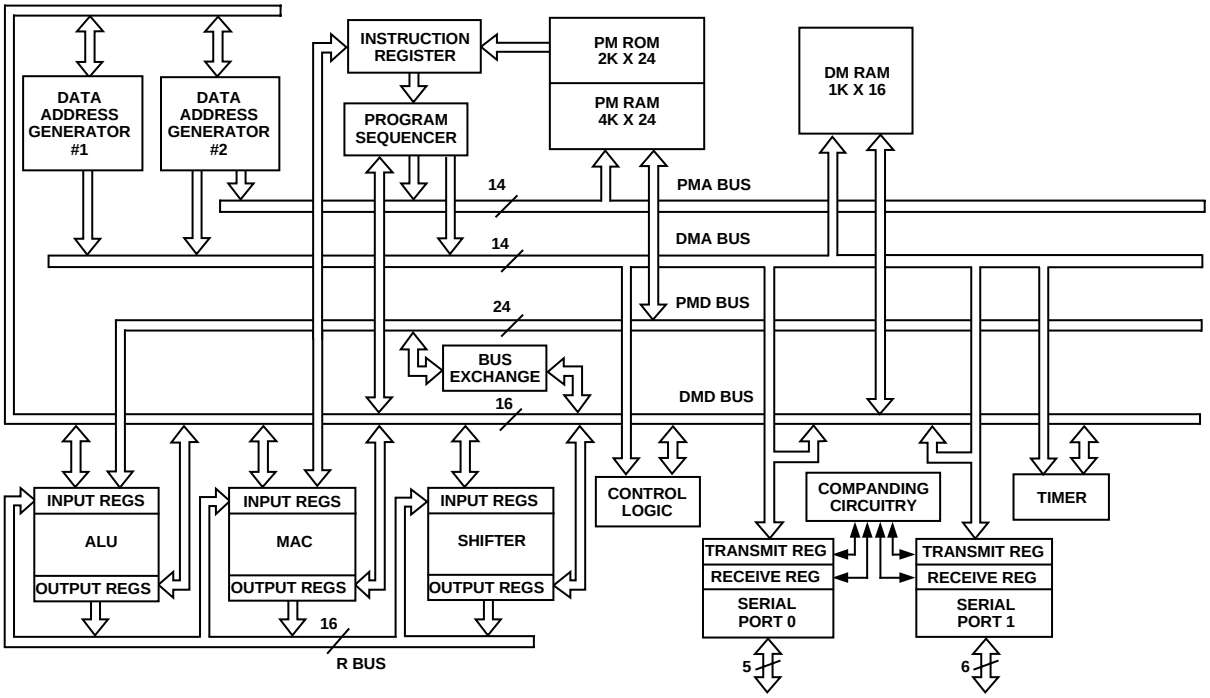


Figure 3. DSP Core Block Diagram

DSP CORE ARCHITECTURE OVERVIEW

Figure 3 is an overall block diagram of the DSP core of the ADMC300, which is based on the fixed-point ADSP-2171. The ADSP-2171 flexible architecture and comprehensive instruction set allows the processor to perform multiple operations in parallel. In one processor cycle (40 ns with a 12.5 MHz CLKIN) the DSP core can:

- Generate the next program address.
- Fetch the next instruction.
- Perform one or two data moves.
- Update one or two data address pointers.
- Perform a computational operation.

This all takes place while the processor continues to:

- Receive and transmit through the serial ports.
- Decrement the interval timer.
- Generate PWM signals.
- Convert the ADC input signals.
- Operate the encoder interface unit.
- Operate all other peripherals including the auxiliary PWM and event timer subsystem.

The processor contains three independent computational units: the arithmetic and logic unit (ALU), the multiplier/accumulator (MAC) and the shifter. The computational units process 16-bit data directly and have provisions to support multiprecision computations. The ALU performs a standard set of arithmetic and logic operations; division primitives are also supported. The MAC performs single-cycle multiply, multiply/add, multiply/subtract operations with 40 bits of accumulation. The shifter performs logical and arithmetic shifts, normalization, denormalization, and derive exponent operations. The shifter can be used to efficiently implement numeric format control including floating-point representations.

The internal result (R) bus directly connects the computational units so that the output of any unit may be the input of any unit on the next cycle.

A powerful program sequencer and two dedicated data address generators ensure efficient delivery of operands to these computational units. The sequencer supports conditional jumps and subroutine calls and returns in a single cycle. With internal loop counters and loop stacks, the ADMC300 executes looped code with zero overhead; no explicit jump instructions are required to maintain the loop.

Two data address generators (DAGs) provide addresses for simultaneous dual operand fetches from data memory and program memory. Each DAG maintains and updates four address pointers (I registers). Whenever the pointer is used to access data (indirect addressing), it is post-modified by the value in one of four modify (M) registers. A length value may be associated with each pointer (L registers) to implement automatic modulo addressing for circular buffers. The circular buffering feature is also used by the serial ports for automatic data transfers to and from on-chip memory. DAG1 generates only data memory address but provides an optional bit-reversal capability. DAG2 may generate either program or data memory addresses, but has no bit-reversal capability.

Efficient data transfer is achieved with the use of five internal buses:

- Program Memory Address (PMA) Bus
- Program Memory Data (PMD) Bus
- Data Memory Address (DMA) Bus
- Data Memory Data (DMD) Bus
- Result (R) Bus

Program memory can store both instructions and data, permitting the ADMC300 to fetch two operands in a single cycle—one from program memory and one from data memory. The ADMC300 can fetch an operand from on-chip program memory and the next instruction in the same cycle.

The ADMC300 writes data from its 16-bit registers to the 24-bit program memory using the PX register to provide the lower eight bits. When it reads data (not instructions) from 24-bit program memory to a 16-bit data register, the lower eight bits are placed in the PX register.

The ADMC300 can respond to a number of distinct DSP core and peripheral interrupts. The DSP core interrupts include serial port receive and transmit interrupts, timer interrupts, software interrupts and external interrupts. The motor control peripherals also produce interrupts to the DSP core.

The two serial ports (SPORTs) provide a complete synchronous serial interface with optional companding in hardware and a wide variety of framed and unframed data transmit and receive modes of operation. Each SPORT can generate an internal programmable serial clock or accept an external serial clock. Boot loading of both the program and data memory RAM of the ADMC300 is through the serial port SPORT1.

A programmable interval counter is also included in the DSP core and can be used to generate periodic interrupts. A 16-bit count register (TCOUNT) is decremented every *n* processor cycles, where n–1 is a scaling value stored in the 8-bit TSCALE register. When the value of the counter reaches zero, an interrupt is generated and the count register is reloaded from a 16-bit period register (TPERIOD).

The ADMC300 instruction set provides flexible data moves and multifunction (one or two data moves with a computation) instructions. Each instruction is executed in a single 40 ns processor cycle (for a 12.5 MHz CLKIN). The ADMC300 assembly language uses an algebraic syntax for ease of coding and readability. A comprehensive set of development tools support program development. For further information on the DSP core, refer to the *ADSP-2100 Family User's Manual, Third Edition*, with particular reference to the ADSP-2171.

Serial Ports

The ADMC300 incorporates two complete synchronous serial ports (SPORT0 and SPORT1) for serial communication and multiprocessor communication. Following is a brief list of capabilities of the ADMC300 SPORTs. Refer to the *ADSP-2100 Family User's Manual, Third Edition*, for further details.

- SPORTs are bidirectional and have a separate, double-buffered transmit and receive section.
- SPORTs can use an external serial clock or generate their own serial clock internally.
- SPORTs have independent framing for the receive and transmit sections. Sections run in a frameless mode or with frame synchronization signals internally or externally generated. Frame synchronization signals are active high or inverted, with either of two pulsewidths and timings.
- SPORTs support serial data word lengths from 3 bits to 16 bits and provide optional A-law and μ-law companding according to ITU (formerly CCITT) recommendation G.711.
- SPORT receive and transmit sections can generate unique interrupts on completing a data word transfer.
- SPORTs can receive and transmit an entire circular buffer of data with only one overhead cycle per data word. An interrupt is generated after a data buffer transfer.
- SPORT0 has a multichannel interface to selectively receive and transmit a 24-word or 32-word, time-division multiplexed, serial bitstream.
- SPORT1 can be configured to have two external interrupts ($\overline{IRQ0}$ and $\overline{IRQ1}$), and the Flag In and Flag Out signals. The internally generated serial clock may still be used in this configuration.
- SPORT1 is the default input for program and data memory boot loading. The RFS1 pin can be configured internal to the ADMC300 as an SROM/E^2PROM reset signal.
- SPORT1 has two data receive pins (DR1A and DR1B). The DR1A pin is intended for synchronous boot loading from the external SROM/E^2PROM. The DR1B pin can be used as the data receive pin for boot loading from an external UART (SCI compatible), as the data receive pin for the debugger communicating over the UART/debugger interface, or as the data receive pin for a general purpose SPORT after booting. These two pins are internally multiplexed onto the one DR1 port of the SPORT. The particular data receive pin selected is determined by a bit in the MODECTRL register.

PIN FUNCTION DESCRIPTION

The ADMC300 is available in an 80-lead TQFP package. Table I contains the pin descriptions.

Table I. Pin List

Pin Group Name	# of Pins	Input/ Output	Function
$\overline{\text{RESET}}$	1	I	Processor Reset Input.
SPORT0	5	I/O	Serial Port 0 Pins (TFS0, RFS0, DT0, DR0, SCLK0).
SPORT1	6	I/O	Serial Port 1 Pins (TFS1, RFS1, DT1, DR1A, DR1B, SCLK1).
CLKOUT	1	O	Processor Clock Output.
CLKIN, XTAL	2	I, O	External Clock or Quartz Crystal Connection Point.
PIO0–PIO11	12	I/O	Digital I/O Port, External Convert Start and Event Timer Pins.
AUX0–AUX1	2	O	Auxiliary PWM Outputs.
AH–CL	6	O	PWM Outputs.
$\overline{\text{PWMTRIP}}$	1	I	PWM Trip Signal.
PWMPOL	1	I	PWM Polarity Pin.
PWMSYNC	1	O	PWM Synchronization Pin.
V1–V5	5	I	Noninverting Inputs of the Differential ADCs' Input Amplifiers.
V1N–V5N	5	I	Inverting Inputs of the Differential ADCs' Input Amplifiers.
REFINA–REFINB	2	I	Voltage reference inputs for ADCs.
VREF	1	O	Voltage Reference Output.
MUX0–MUX2	3	O	Multiplexer Control Lines.
EIA, EIB, EIZP	3	I	Encoder Interface Pins.
AV_{DD}	4		Analog Power Supply.
AGND	4		Analog Ground.
V_{DD}	6		Digital Power Supply.
GND	9		Digital Ground.

INTERRUPT OVERVIEW

The ADMC300 can respond to nineteen different interrupt sources, eight of which are internal DSP core interrupts and eleven interrupts from the motor control peripherals. The eight DSP core interrupts comprise the peripheral ($\overline{\text{IRQ2}}$), SPORT0 receive, SPORT0 transmit, SPORT1 receive (or $\overline{\text{IRQ0}}$), SPORT1 transmit (or $\overline{\text{IRQ1}}$), two software and the interval timer interrupts. In addition, the motor control peripherals add eleven interrupts that include two ADC, two PWM, five peripheral I/O, one encoder interface and one event timer interrupt. The interrupts are internally prioritized and individually maskable. All peripheral interrupts are multiplexed into the DSP core through the peripheral $\overline{\text{IRQ2}}$ interrupt. The programmable interrupt controller manages the masking and vector addressing of all eleven peripheral interrupts. A detailed description of the operation of the entire interrupt system of the ADMC300 is given later, after a more detailed description of the various peripheral systems.

Memory Map

The ADMC300 has two distinct memory types; program memory and data memory. In general, program memory contains user code and coefficients, while the data memory is used to store variables and data during program execution. Both program memory RAM and ROM is provided on the ADMC300. Program memory RAM is arranged in two noncontiguous 2K × 24-bit blocks, one starting at address 0x0000 and the other at 0x1800. Program memory ROM is located at address 0x0800. Data memory is arranged as a 1K × 16-bit block starting at address 0x3800. The motor control peripherals are memory mapped into a region of the data memory space starting at 0x2000. The complete program and data memory maps are given in Tables II and III respectively.

Table II. Program Memory Map

Address Range	Memory Type	Function
0x0000–0x005F	RAM	Interrupt Vector Table
0x0060–0x071F	RAM	User Program Space
0x0720–0x07DF	RAM	Reserved by Debugger
0x07E0–0x07FF	RAM	Reserved by Monitor
0x0800–0x0DD6	ROM	ROM Monitor
0x0DD7–0x0F4C	ROM	ROM Math and Motor Control Utilities
0x0F4D–0x0FFF	ROM	Reserved
0x1000–0x17FF		Unused
0x1800–0x1FFF	RAM	User Program Space
0x2000–0x3FFF		Unused

Table III. Data Memory Map

Address Range	Memory Type	Function
0x0000–0x1FFF		Unused
0x2000–0x20FF		Memory Mapped Registers
0x2100–0x37FF		Unused
0x3800–0x3B5F	RAM	User Data Space
0x3B60–0x3BFF	RAM	Reserved by Monitor
0x3C00–0x3FFF		Memory Mapped Registers

ROM Code

The 2K × 24-bit block of program memory ROM starting at address 0x0800 contains a monitor function that is used to download and execute user programs via the serial port. In addition, the monitor function supports an interactive mode in which commands are received and processed from a host that is configured as a UART device. An example of such a host is the Windows®-based Motion Control Debugger that is part of the software development system for the ADMC300. In the interactive mode, the host can access both the internal DSP and peripheral motor control registers of the ADMC300, read and write to both program and data memory, implement breakpoints and perform single-step and run/halt operation as part of the program debugging cycle.

In addition to the monitor function, the program memory ROM contains a number of useful mathematical and motor control utilities that can be called as subroutines from the user code. A complete list of these ROM functions is given in Table IV. The start address of the function in the program memory ROM is also given. Refer to the *ADMC300 DSP Motor Controller Developer's Reference Manual* for more details of the ROM functions.

Table IV. ROM Utilities

Utility	Address	Function
PER_RST	0x07E4	Reset Peripherals.
UMASK	0x0DD7	Limits Unsigned Value to Given Range.
PUT_VECTOR	0x0DDE	Facilitates User Setup of Vector Table.
SMASK	0x0DF0	Limits Signed Value to Given Range.
ADMC_COS	0x0E50	Cosine Function.
ADMC_SIN	0x0E57	Sine Function.
ARCTAN	0x0E6D	Arctangent Function.
RECIPROCAL	0x0E8F	Reciprocal (1/x) Function.
SQRT	0x0EA5	Square Root Function.
LOG	0x0EE2	Logarithm (Base 10) Function.
LN	0x0EDF	Natural Logarithm Function.
REV_FOR_PARK	0x0EFE	Forward/Reverse Park Transformation.
REV_CLARK	0x0F12	Reverse Clark Transformation.
FOR_CLARK	0x0F28	Forward Clark Transformation.
SDIVQINT	0x0F32	Unsigned Single Precision Division (Integer).
SDIVQ	0x0F3B	Unsigned Single Precision Division (Fractional).

SYSTEM INTERFACE

Figure 4 shows a basic system configuration for the ADMC300 with an external crystal and serial E^{2}PROM for boot loading of program and data memory RAM.

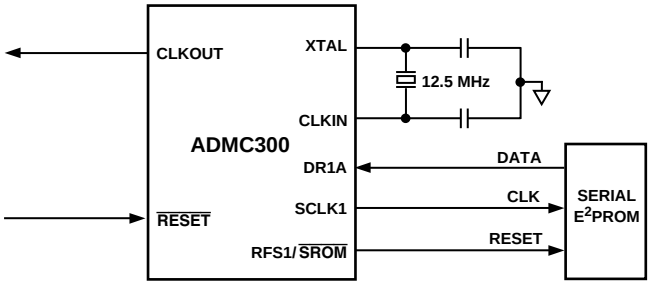


Figure 4. Basic System Configuration

Clock Signals

The ADMC300 can be clocked by either a crystal or a TTL-compatible clock signal. The CLKIN input cannot be halted, changed during operation or operated below the specified minimum frequency during normal operation. If an external clock is used, it should be a TTL-compatible signal running at half the instruction rate. The signal is connected to the CLKIN pin of the ADMC300. In this mode, with an external clock signal, the XTAL pin must be left unconnected. The ADMC300 uses an input clock with a frequency equal to half the instruction rate; a 12.5 MHz input clock yields a 40 ns processor cycle (which is equivalent to 25 MHz). Normally instructions are executed in a single processor cycle. All device timing is relative to the internal instruction rate, which is indicated by the CLKOUT signal.

Because the ADMC300 includes an on-chip oscillator circuit, an external crystal may be used instead of a clock source, as shown in Figure 4. The crystal should be connected across the CLKIN and XTAL pins, with two capacitors as shown in Figure 4. A parallel-resonant, fundamental frequency, microprocessor-grade crystal should be used. A clock output signal (CLKOUT) is generated by the processor at the processor's cycle rate of twice the input frequency. This output can be enabled and disabled by the CLKODIS bit of the SPORT0 Autobuffer Control Register, DM[0x3FF3]. However, extreme care must be exercised when using this bit since disabling CLKOUT effectively disables all motor control peripherals, including the watchdog timer.

Reset

The $\overline{\text{RESET}}$ signal initiates a master reset of the ADMC300. The $\overline{\text{RESET}}$ signal must be asserted during the power-up sequence to assure proper initialization. $\overline{\text{RESET}}$ during initial power-up must be held long enough to allow the internal clock to stabilize. If $\overline{\text{RESET}}$ is activated any time after power-up, the clock continues to run and does not require stabilization time.

The power-up sequence is defined as the total time required for the crystal oscillator circuit to stabilize after a valid V_{DD} is applied to the processor, and for the internal phase-locked loop (PLL) to lock onto the specific crystal frequency. A minimum of 2000 CLKIN cycles ensures that the PLL has locked, but does not include the crystal oscillator start-up time. During this power-up sequence, the $\overline{\text{RESET}}$ signal should be held low. On any subsequent resets, the $\overline{\text{RESET}}$ signal must meet the minimum pulsewidth specification, t_{RSP}.

If an RC circuit is used to generate the $\overline{\text{RESET}}$ signal, the use of an external Schmitt trigger is recommended.

The master reset sets all internal stack pointers to the empty stack condition, masks all interrupts, initializes DSP core registers and performs a full reset of all of the motor control peripherals. When the $\overline{\text{RESET}}$ line is released, the first instruction is fetched from internal program memory ROM at location 0x0800. The internal monitor code at this location then commences the boot-loading sequence over the serial port, SPORT1.

Boot Loading

On power-up or reset, the ADMC300 is configured so that execution begins at the internal PM ROM at address 0x0800. This starts execution of the internal monitor function that first performs some initialization functions and copies a default interrupt vector table to addresses 0x0000–0x005F of program memory RAM. The monitor next attempts to boot load from an external SROM or E^{2}PROM on SPORT1 using the three wire connection of Figure 4. The monitor program first toggles the RFS1/$\overline{\text{SROM}}$ pin of the ADMC300 to reset the serial memory device. If an SROM or E^{2}PROM is connected to SPORT1, data is clocked into the ADMC300 at a rate CLKOUT/26. Both program and data memory RAM can be loaded from the SROM/E^{2}PROM. After the boot load is complete, program execution begins at address 0x0060. This is where the first instruction of the user code should be placed.

If boot loading from an E^{2}PROM is unsuccessful, the monitor code reconfigures SPORT1 as a UART and attempts to receive commands from an external device on this serial port. The monitor then waits for a byte to be received over SPORT1,

locks onto the baud rate of the external device (autobaud feature) and takes in a header word that tells it with what type of device it is communicating. There are two alternatives; a UART boot loader such as an SCI interface to a Motorola 68HC11 or a UART interface to the *Motion Control Debugger*. With the SCI interface, program and data memory is loaded via the SCI interface and execution starts at 0x0060. With the UART debugger interface, the monitor enters interactive mode in which it processes commands received from the external device.

DSP Control Registers

The DSP core has a system control register, SYSCNTL, memory mapped at DM (0x3FFF). SPORT0 is enabled when Bit 12 is set, disabled when this bit is cleared. SPORT1 is enabled when Bit 11 is set, disabled when this bit is cleared. SPORT1 is configured as a serial port when Bit 10 is set, or as flags and interrupt lines when this bit is cleared. For proper operation of the ADMC300, all other bits in this register must be cleared (which is their default).

The DSP core has a wait state control register, MEMWAIT, memory mapped at DM (0x3FFE). For proper operation of the ADMC300, this register must always contain the value 0x8000 (which is the default).

The configuration of both the SYSCNTL and MEMWAIT registers of the ADMC300 is shown at the end of the data sheet.

ANALOG-TO-DIGITAL CONVERSION SYSTEM

A functional block diagram of the ADC system of the ADMC300 is shown in Figure 5. The ADC system provides the high performance conversion required for precision applications. It integrates five completely independent analog-to-digital converters based on sigma-delta conversion technology. Each ADC channel may be configured as either a differential or single-ended input for maximum flexibility in interfacing to external sensors and inputs. The sigma-delta converter consists of two stages, a modulator and a sinc filter, that combine to produce a 16-bit conversion. For each channel, signal-to-noise ratios of 76 dB may be achieved, corresponding to greater than 12 bits of resolution from each converter. Input signals up to 16.27 kHz may be converted.

For maximum flexibility, the five ADCs are arranged as two banks; ADC1 and ADC2 forming Bank A, and ADC3, ADC4 and ADC5 forming Bank B. The characteristics of each bank, such as sampling rate, internal or external conversion, synchronization to the PWM block, operating modes, may be controlled independently. The ADC registers of each bank may be loaded from an internal signal whose frequency may be programmed as a precise fraction of the CLKIN frequency. Alternatively, the ADCs may be updated by an external signal on the CONVST pin. There are two dedicated ADC interrupts; one for each

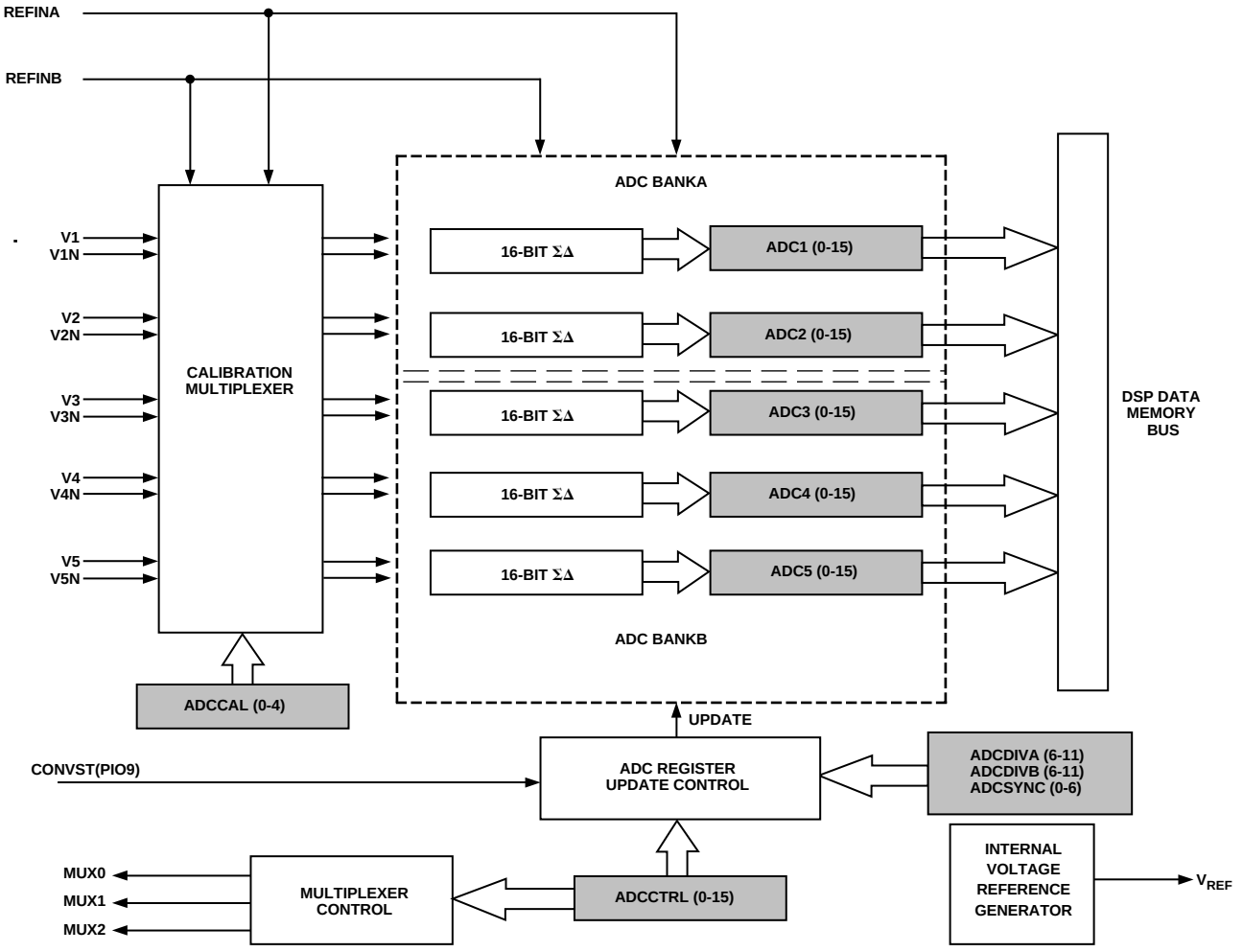


Figure 5. Functional Block Diagram of ADC System of ADMC300

bank of converters that can be used to signal that the ADCs of the particular bank have been updated.

The ADC system also contains a built-in calibration function that may be used to null any offsets within the ADC converters. Each ADC channel may be placed in the calibration state individually or in combination with other channels.

In addition, the ADC system provides three multiplexer control pins that may be used in conjunction with an external multiplexer to permit external signal expansion.

There is a separate reference input for each bank of converters. However, the ADMC300 also provides a reference output that could be buffered and used as a reference source for either or both banks.

Input Configuration

The input to each ADC may be applied to the ADMC300 in either a single-ended or differential configuration. In many cases, a single-ended configuration is easier to provide but the differential connection permits the reduction of common-mode noise from the input signal. Each ADC input may be configured for single-ended or differential inputs as appropriate, completely independent of the other channels. Figure 6 illustrates a typical differential configuration for the inputs of one ADC channel of the ADMC300. The input signals are applied to pins Vx and VxN (for example V1 and V1N). For correct operation and maximum input dynamic range, the inputs signals should be centered on the reference voltage level, V_{REF}. Therefore, the signal applied to the Vx pin should be $V_{IN} + V_{REF}$, where V_{IN} is the analog input voltage. The corresponding signal applied to the inverting terminal of the differential input, VxN, is then $–V_{IN} + V_{REF}$ so that the differential signal applied to the ADC input is actually 2 V_{IN}.

The input RC combination of 100 Ω and 0.047 µF provides a first-order low-pass antialiasing filter with a cutoff frequency of 34 kHz. An advantage to sigma-delta ADCs is that the initial (analog) signal filtering required for antialiasing is much more modest than that required by other ADCs. With the sigma-delta ADC, the input filter needed for the analog signal only has to cut off at one-half of the modulator frequency, rather than the lower effective sampling frequency. For the ADMC300, the modulator runs 64 times faster than the sampling frequency. Thus for a 32.5 kHz sampling rate, the modulator frequency is 2.08 MHz, meaning the needed cutoff for the analog input signal is 1.04 MHz. Therefore, a simple first order filter, such as the RC filter shown in Figure 6, which provides a more than 30 dB attenuation to signals above 1 MHz (3 dB at 34 kHz) is adequate. The additional antialiasing band limiting required by the Nyquist criterion for the 32.5 kHz sampling rate (a cutoff of 16.25 kHz) is supplied by the high order sinc filter in the digital domain.

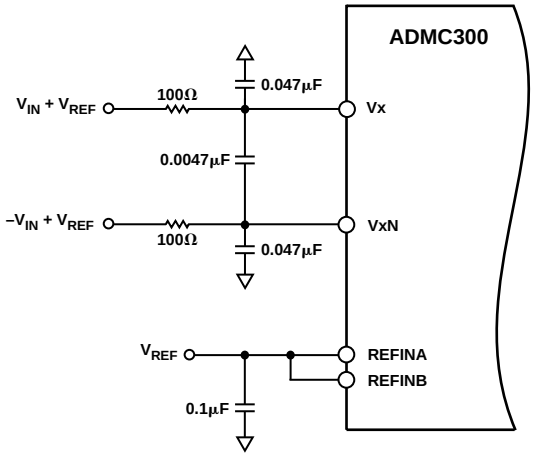


Figure 6. Differential Configuration for ADC Input of ADMC300

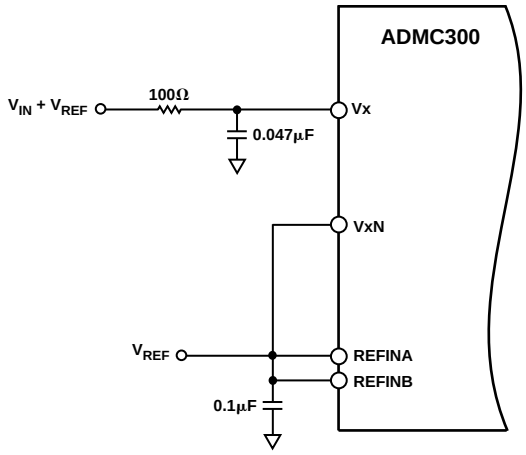


Figure 7. Single-Ended Configuration for ADC Input of ADMC300

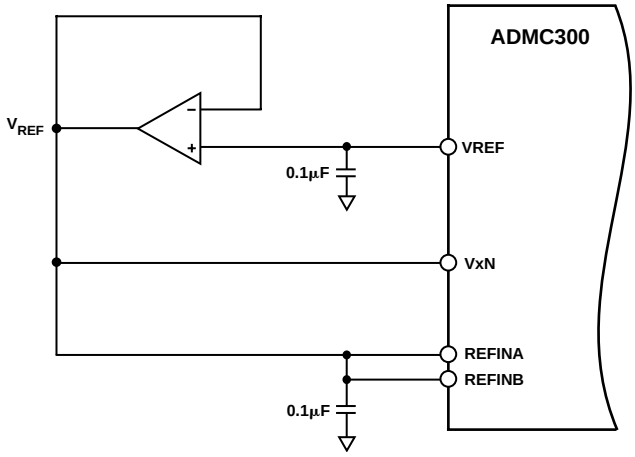


Figure 8. Connection of Internal Voltage Reference of ADMC300

The corresponding configuration for single-ended operation is shown in Figure 7, where the inverting input is now tied directly to the reference voltage level. The noninverting input is V_{IN} + V_{REF}. An antialiasing filter with a cutoff at 34 kHz is included on this input. Clearly, the differential input voltage swing in this configuration is half that of the differential configuration. Therefore, when operating in single-ended mode, the input voltage, V_{IN}, may be twice as large as that when operating differentially.

To ensure correct operation, a 0.1 µF, high-quality capacitor must be included at the reference input pins. A 0.1 µF capacitor should also be used at the VREF pin, even if external references are used.

In both Figures 6 and 7, it is assumed that an external precision reference is used to provide the reference voltage level, V_{REF}. For optimum operation, a high-performance 2.5 V reference such as the AD580 is recommended. This reference is applied directly to both input reference pins, REFINA and REFINB, and is also applied to any external bias circuitry used to produce the ADC input signals. For a lower cost solution, the ADMC300 also provides a reference output that may be used to provide the V_{REF} signal. The reference output is available at the VREF pin. For correct operation, a 0.1 µF capacitor is required at this pin even if the internal reference is not used. In addition, it is recommended that the V_{REF} signal be buffered in a unity-gain stage prior to use, as shown in Figure 8. Of course, since there are two separate reference inputs, pins REFINA and REFINB, the two banks of ADCs may be run differently, if required.

ADC Register Update

The sigma-delta converters of the ADMC300 operate at a highly oversampled rate. A number of control bits in the ADCCTRL register are used to control when and how data is latched into the ADC data registers of each bank. The data registers of each bank may be latched at a regular rate based on either an internal or an external convert start signal. Bits 0–1 of the ADCCTRL register determine whether the convert start signal is internal or external. Bit 0 controls the operation of Bank A and Bit 1 controls the operation of Bank B. If either of these bits is set to 1, the data registers of the corresponding ADC bank are latched within two CLKIN cycles of the occurrence of a rising edge on the CONVST pin (PIO9). Alternatively, if either bit is set to zero, internal mode is selected and the ADCs of the particular bank are updated at a regular rate, determined by the contents of the appropriate ADC sample frequency division register, ADCDIVA or ADCDIVB.

Bits 2–3 of the ADCCTRL register can be used to place the ADC banks in an alternative read mode. The read mode is enabled by setting these bits to 1, effectively disabling the convert start mode. In the read mode, the ADC registers of the particular bank are continuously updated at a rate equal to half the DSP clock rate, so that data is effectively available on demand. Bit 2 of the ADCCTRL register is used to place ADC Bank A in read mode, while Bit 3 is used for Bank B.

ADC Sample Rate Selection

Internal convert start mode is selected by clearing Bits 0 and 2 of the ADCCTRL register for Bank A and Bits 1 and 3 for Bank B. In this mode, the ADC data registers are updated at a regular rate that is determined by the ADC clock divide registers, ADCDIVA and ADCDIVB. Therefore, each bank can be configured for independent update rates by writing different values to the two registers. When Bank A and Bank B are configured for independent update rates, it is important that the REFINA and REFINB pins are driven by separate voltage reference sources to avoid excessive crosstalk between banks. The ADCDIVA and ADCDIVB registers are 6-bit registers aligned in Bits 6–11 and the value written to these registers is used to divide the CLKIN frequency to provide the ADC update rate. A 12-bit value is written to these registers, but since Bits 0–5 are ignored, the value should be an integer multiple of 64 or 0x040. The resultant ADC update rates for banks A and B may be expressed as:

$$f_{S,A} = \frac{f_{CLKIN}}{ADCDIVA}$$

$$f_{S,B} = \frac{f_{CLKIN}}{ADCDIVB}$$

where f_{CLKIN} is the CLKIN frequency, equal to half the DSP instruction rate. Therefore, writing a value of 0x180 (= 384) gives an ADC update rate of 32.55 kHz with a CLKIN frequency of 12.5 MHz. The maximum value that can be written to these registers is 0xFC0, corresponding to an update rate of 3.1 kHz. Since the maximum update rate is limited to 32.55 kHz, the permissible range of ADC divide values is 0x180 to 0xFC0 in steps of 0x040.

Each ADC channel contains an input modulator that oversamples the input signal at a high rate. The modulator sample frequency is automatically set by the internal ADC control to be exactly 64 times the ADC update rate determined by the ADC divide registers. This corresponds to an oversample ratio of 64. Therefore, in the case where ADCDIVA = 0x180, the input modulators of ADC bank A sample the input signal at 2.08 MHz and the data registers ADC1 and ADC2 are updated at the 32.55 kHz rate.

Synchronization of ADC and PWM Systems

In motor control applications, it is advantageous to synchronize the operation of the ADC system to the PWM pulse generation. The ADMC300 permits separate control of such synchronization for each ADC bank and permits sophisticated definition of the particular way that the ADC and PWM systems are synchronized. Operation of bank A of the ADC may be synchronized to the PWM by setting Bit 7 of the ADCCTRL register. Similarly, setting Bit 8 of the ADCCTRL register enables synchronization of Bank B to the PWM.

At its simplest, the ADC and PWM systems may be programmed to operate at the same frequency and be synchronized to one another so that the ADC data registers are automatically updated at the start of each PWM period. This mode of operation is illustrated in Figure 9(a) and is enabled by writing identical values to the PWM period register, PWMTM, and the ADC divide register, ADCDIVA or ADCDIVB. Synchronization is subsequently enabled by setting Bit 7 (for Bank A) or Bit 8 (for Bank B) of the ADCCTRL register.

Additionally, a separate control register, ADCSYNC may be used to phase shift the update of the ADC registers to some suitably defined instant within the PWM period. In this mode, the frequencies of the PWM and ADC register updating are still the same but phase shifted relative to one another. This mode is illustrated in Figure 9 (b). Again, the PWM period register and ADC divide registers are loaded with the same value. However, the offset of the ADC update within the PWM period is programmed using the ADCSYNC register. The ADCSYNC

register is a 7-bit register so that the ADC sample period is effectively subdivided into 128 equal time slices. The value written to the ADCSYNC register is the number of such time slices before the PWMSYNC pulse that the CONVST pulse is active. In other words, the occurrence of the CONVST pulse lags the PWMSYNC pulse of Figure 9 (b) by a time, T_{OFFSET}, that can be expressed as a fraction of the ADC update period:

$$T_{OFFSET} = \frac{(128 - ADCSYNC)}{128}\left(\frac{ADCDIVn}{f_{CLKIN}}\right)$$

Therefore, for the case where ADCDIVA is 0x180 and ADCSYNC is 0x060, the CONVST pulse will lag the PWMSYNC pulse by a quarter of the ADC update period, or 7.68 μs, with a 12.5 MHz CLKIN.

It is also possible to operate the ADCs at a faster update rate than the PWM switching frequency and still maintain synchronism, as illustrated in Figure 9 (c). In this example, the value written to the ADCDIV registers is three times larger than the value written to the PWMTM register, so that the ADC update rate is three times faster than the PWM switching frequency. Synchronism is maintained by setting Bits 7 and 8 of the ADCCTRL register. In addition, it is possible to introduce a phase shift between the ADC update and PWMSYNC pulses while operating at different frequencies, as illustrated by Figure 9 (d). The offset is defined by the ADCSYNC register as a fraction of the ADC update period in an identical manner to before.

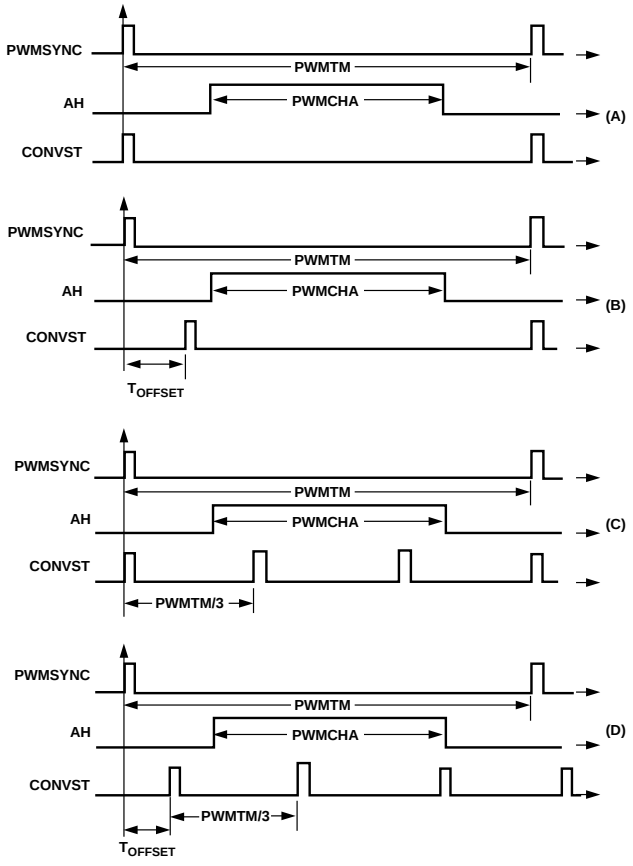


Figure 9. (a) Synchronization of ADC and PWM at Same Frequency with No Offset, (b) Synchronization of ADC and PWM at Same Frequency with Offset, (c) Synchronization of ADC and PWM with No Offset and ADC Update at Three Times the PWM Frequency (d) Synchronization of ADC and PWM at Different Frequencies with Offset

ADC Transfer Characteristics

Each ADC converter of the ADMC300 consists of an input modulator stage and a decimation filter stage that produces the final conversion result. The output of the decimation filters are 16-bit, left-aligned, two's complement representation of the input signal, V_{IN}. The ideal ADC transfer characteristics for both single-ended and differential modes are shown in Figure 10. The transfer characteristics of the ADC when operated in the differential configuration are shown in Figure 10 (a) and for the single-ended configuration in Figure 10 (b). The peak-peak input voltage is 4 V.

The output code of the ADCs is typically given by:

$$ADCx = \left(10{,}600 \times \frac{2.5}{V_{REFIN}}\right)(Vx - VxN)$$

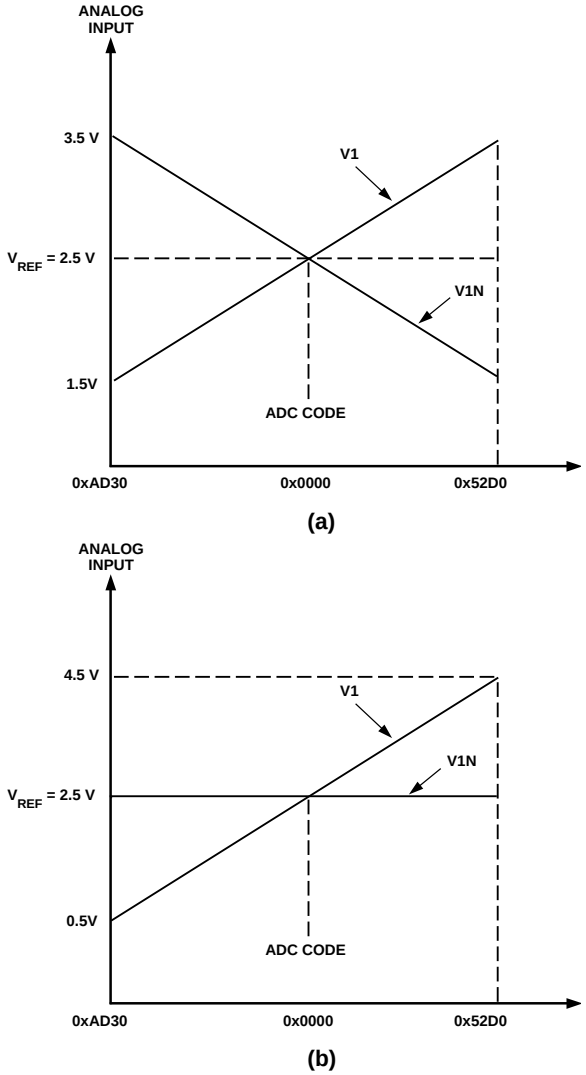


Figure 10. (a) Typical Transfer Characteristic of the ADC in Differential Input Configuration (b) Typical Transfer Characteristic of the ADC in Single-Ended

ADC Group Delay

The digital filters of the ADCs carry out two important functions. First, they remove the out-of-band quantization noise, which has been suitably shaped by the noise-shaping circuits of the input modulator stages. The digital filters then decimate the high frequency bitstream from the modulators to a lower rate 16-bit word. The lower rate is set by the ADC divide registers for the respective ADC banks as described previously. The antialiasing decimation filter may be modeled by the Z-domain transfer function:

$$H_{SINC}(Z)=\left[\frac{1}{64}\left(\frac{1-z^{-64}}{1-z^{-1}}\right)\right]^3$$

Associated with the sinc filters is a group delay that may be approximated by:

$$t_\phi = \frac{1.5}{f_S}$$

where f_S is the update rate of the particular ADC channel. In order to minimize the impact of the group delay on the overall performance of the control system, it is advantageous to oversample the ADCs at a rate faster than the PWM frequency.

ADC Calibration

The ADC system of the ADMC300 has a calibration feature that may be used to null any offsets in the ADC channels. There is a 5-bit ADC calibration register, ADCCAL that has a dedicated bit for each ADC channel. Setting the appropriate bit of the ADCCAL register will place the respective ADC channel in the offset calibration state. Bit 0 controls ADC1, Bit 1 controls ADC2, etc. When the appropriate bit of the ADCCAL register is set, the two input pins associated with that ADC channel are effectively disconnected from the pins of the ADMC300 and connected internally to the reference voltage. After waiting for the settling time of the decimation filters, the resultant ADC code is a measure of the offset for that particular ADC channel. This number should be saved in memory and used to correct all further measurements from that channel.

ADC Interrupt Generation

Two dedicated interrupts are associated with the ADC system of the ADMC300, one for each of the ADC banks. The interrupts are generated after the ADC data registers of the particular bank have been updated by either an internal or external CONVST pulse. Interrupts are not generated when the ADCs are in the read mode. There are separate interrupt vector locations associated with each of the interrupt sources. The ADC Bank A interrupt is the highest priority interrupt with its vector address at program memory location 0x0030. The Bank B interrupt is the third highest priority interrupt with a vector address of 0x0038. Each interrupt has a four word space in the vector table. The sequencing and masking of these interrupts is managed by the Programmable Interrupt Controller (PIC) block described later.

ADC Multiplexer Control

The ADMC300 has three digital output pins, MUX2, MUX1 and MUX0, that can be used to drive an external multiplexer to feed additional analog inputs to the ADCs if required. Using these control lines, up to eight analog signals could be externally multiplexed into each ADC channel, allowing expansion up to 40 analog inputs. The state of the three multiplexer pins is directly controlled from the ADCCTRL register, using Bits 4–6. Bit 4 of the ADCCTRL register directly controls the MUX0 pin, so that setting this bit will place a HI level on the MUX0 pin. Similarly, Bit 5 directly controls the MUX1 pin and Bit 6 controls the MUX2 pin.

Because of the finite impulse response of the decimation filters of the ADCs, it is usually only slower dynamic signals that are multiplexed into the ADCs. In a typical motor control system, such signals may comprise the dc link voltage, the output of various temperature sensors, reference inputs, etc.

ADC Status

Because of the dynamic characteristics of the decimation filters of the sigma-delta converters, it is necessary to allow the impulse response of the filters to decay before meaningful, accurate data is available. There is a one-bit status register, ADCSTAT, in the ADMC300 that indicates whether or not valid data is available from the ADC. Bit 0 of the ADCSTAT register is asserted while the decimation filters are settling to indicate that data is not yet valid. This BUSY bit can be programmed to represent the status of the decimation filters of either Bank A or Bank B by programming Bit 9 of the ADCCTRL register. The BUSY bit will go active for four ADC sample periods any time that ADCCTRL, ADCCAL or ADCSYNC are written to. If Bit 9 of the ADCCTRL register is cleared, the BUSY bit will go active any time ADCDIVA is written to, and the four ADC sample period width of the BUSY pulse will be four ADC Bank A sample periods. If Bit 9 of the ADCCTRL register is set, the BUSY bit will go active any time the ADCDIVB register is written to, and the four ADC sample period width of the BUSY pulse will be four ADC Bank B sample periods. It is still possible to read the ADC data registers while the BUSY signal is asserted. However, care must be taken in the interpretation of such data.

ADC Power-Down and Reset Features

The ADC section of the ADMC300 has certain power-down features that may be used to reduce the overall power consumption of the part. Each bank of the ADC system may be individually powered down if all channels of that bank are not used in a particular application. Setting Bit 10 of the ADCCTRL register will power down the input modulators of both ADC channels of ADC Bank A. Similarly, setting Bit 11 of the ADCCTRL register will power down the three ADC channels of Bank B. Clearing these bits will enable the input modulators of the respective banks. On power-up, both Bits 10 and 11 of the ADCCTRL register are set by default, so that all five input modulator stages are disabled. To operate the required ADC channels, the appropriate bits in the ADCCTRL register must be cleared.

In addition, setting Bit 12 of the ADCCTRL register will power down the internal reference circuitry. Further power reduction is possible if this reference circuitry is powered down. However, this bit has an effect only if both ADC banks are also powered down. Clearing Bit 12 of the ADCCTRL register will enable the internal reference circuitry.

It is also possible to force a reset of all five input modulators of the ADC system by setting Bit 14 of the ADCCTRL register. Setting Bit 15 of the ADCCTRL register will force a reset of the decimation filters in all five ADC channels. In order to come out of input modulator reset or decimation filter reset, the respective bit must be cleared. On power-up, these bits are

cleared so that both the modulators and the decimation filters come up in the normal mode. It is recommended that prior to use, a full reset be performed.

ADC Registers

The composition of all the data registers associated with the ADC system of the ADMC300 is shown at the end of the data sheet. The reset values are shown for certain bits, where appropriate.

THREE-PHASE PWM GENERATOR

The ADMC300 PWM controller is a self-contained programmable waveform generator that produces PWM switching signals for a three-phase power inverter. It includes a waveform timing edge calculation unit which allows the generation of six center-based PWM signals based on only three duty cycle registers updated every switching cycle. This minimizes the DSP software required to service the PWM controller and frees up processor time for the motor control law implementation. In the default configuration, it produces the three-phase center-based PWM waveforms required for a three-phase sinusoidal inverter. However, it can also be configured for space vector modulation schemes, or for controlling brushless dc motors (sometimes known as electronically commutated motor). It also has functions that simplify the interface to the power inverter gate drive and protection circuits.

The PWM controller operates at the CLKIN frequency, which is half the DSP clock frequency, giving a PWM timing resolution of 80 ns with a 25 MHz DSP clock. There are four configuration registers (PWMTM, PWMDT, PWMPD and PWMGATE), which define basic waveform parameters such as the switching frequency, deadtime, minimum pulsewidth and gate drive chopping. The PWM output signals on the pins AH through CL are controlled by the input registers (PWMCHA, PWMCHB, PWMCHC and PWMSEG) and the control pins $\overline{\text{PWMTRIP}}$ and PWMPOL. In addition, all of the digital I/O lines (PIO0 to PIO11) can be configured as PWM trip sources.

PWM Controller Overview

The PWM controller consists of three units: the center-based timing unit, output control unit and the gate drive units as shown in Figure 11.

- The center-based PWM timing unit is the core of the PWM controller and produces three pairs of complemented and deadtime adjusted PWM waveforms.
- The Output Control Unit is a signal switching unit that selects the appropriate PWM signals to be connected to the output pins based on the bits set in the segment register (PWMSEG).
- The Gate Drive Unit sets the logic polarity of the PWM "on" signal according to the polarity of the PWMPOL pin to match the gate drive requirement. It can also modulate the PWM "on" signal with a high frequency carrier (98 kHz–6.25 MHz) for a transformer coupled gate drive circuit.

The DSP-based control algorithm can be synchronized to the PWM generator by a hardware interrupt signal that is generated at the start of every PWM switching cycle. This same PWMSYNC signal is also available at an output pin. The hardware $\overline{\text{PWMTRIP}}$ pin can be used to shut down the PWM controller in the event of a fault. A low-going pulse on the $\overline{\text{PWMTRIP}}$ will immediately turn off all PWM outputs. In addition, all of the digital I/O lines can be programmed to operate as additional PWM trip sources, if required. The PIOPWM flag register can be used to enable or disable this mode at each I/O line.

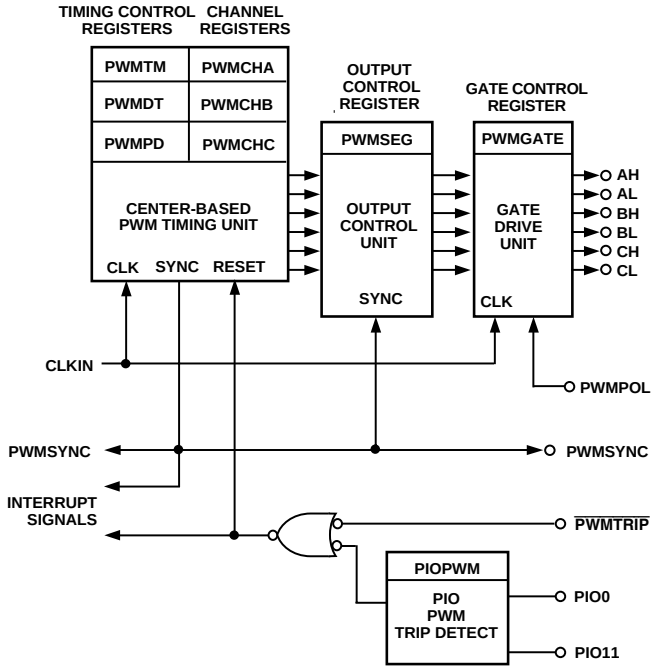


Figure 11. PWM Controller Overview

Center-Based Timing Unit

The center-based PWM Timing Unit is a programmable timer that generates three pairs of fixed frequency PWM waveforms suitable for controlling a three phase power inverter. The unit contains arithmetic circuits that calculate the PWM signal timing edges from waveform parameters such as the PWM period, deadtime, and the duty cycle for each inverter phase. There is no extra DSP software overhead once the duty cycle for each phase has been calculated and loaded into the PWM channel registers.

The PWM Timing Unit produces three pairs of complemented variable duty cycle waveforms symmetrical about a common axis of the form shown in Figure 12. They are complemented waveforms, which means that for any pair of PWM waveforms (AH and AL), both can never be ON at the same time. They are deadtime adjusted, which means that for any pair of PWM waveforms, there is a delay between switching from being ON in one waveform to being ON in the complemented waveform. A pulse deletion function is implemented, which means that very narrow PWM pulses will not be generated. There is an active high PWMSYNC pulse produced at the beginning of each PWM cycle to synchronize the operation of other peripherals with the switching of the power inverter. This signal is applied to the DSP core to generate an interrupt and is used to synchronize the ADC channels to the PWM, if required.

The master switching frequency can range from 3.05 kHz (12-bit resolution) to 48.8 kHz (8-bit resolution) and is an integral fraction of the CLKIN frequency. It is set by the value in the 12-bit PWMTM period register, which sets the total number

of clock cycles in a PWM cycle. The required PWMTM value as a function of the desired master switching frequency (f_{CLKIN}) is given by:

$$PWMTM = \frac{f_{CLKIN}}{f_{PWM}}$$

For example, with f_{CLKIN} = 12.5 MHz, if 10 kHz PWM waveforms are required, then PWMTM should be loaded with 12.5 MHz/10 kHz = 1250. A value must be written to the PWMTM register before the PWM block can be used.

The ON time of each pair of PWM waveforms, e.g., AH and AL, is set by the integer value in the duty cycle registers PWMCHA, PWMCHB and PWMCHC. The deadtime between the active portions of complementary waveforms is set by the value in the deadtime register PWMDT and subtracted from the value in the duty cycle register. The final deadtime adjusted fractional duty cycle for Channel A is given by:

$$d_A = \frac{t_{AON}}{t_{PWM}} = \frac{PWMCHA - PWMDT}{PWMTM}$$

All three duty cycle registers must be written at least four CLKIN periods before the PWMSYNC pulse to update the PWM system. The minimum pulsewidth delivered is set by the value in the pulse deletion register PWMPD. When the calculated high or low pulsewidth for any channel is less than PWMPD, the switching pulse is eliminated and the outputs are saturated, one to 100% high, and the other to 100% low.

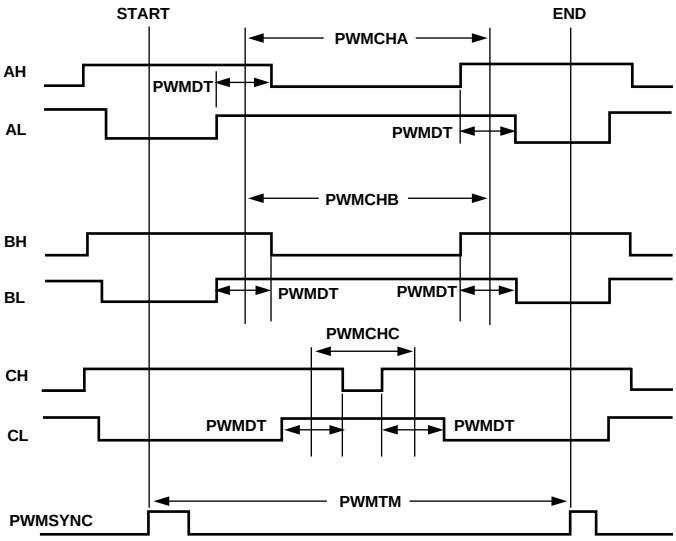


Figure 12. Three-Phase Center-Based Active Low PWM Waveforms

Output Control Unit

The Output Control Unit contains special features that allow the ADMC300 to be easily applied for the control of electronically commutated motors (ECM) or brushless dc motors (BDCM). In these machines, only two motor phases are required to conduct simultaneously so that at most two power switches are turned on at any time. In order to build up current in the motor phases, it is necessary to turn on the upper switch in one phase and the lower switch in another phase of the inverter, simultaneously.

The PWMSEG register of the ADMC300 PWM block allows modification of the pulsewidth modulation signals from the center-based block in order to meet the requirements for ECM control. Three bits of the PWMSEG register (Bits 6–8) permit individual crossover of the three PWM signal pairs. For example, setting Bit 8 will crossover the signals for Phase A such that the high-side signal from the center-based block will ultimately appear at the low-side output pin (AL). Conversely, the low side signal from the center-based block will appear at Pin AH. Similar modifications can be made to Phases B and C using Bits 7 and 6, respectively, of the PWMSEG register. Six bits of the PWMSEG register (Bits 0–5) are used to independently enable/disable any individual PWM output pins. For example, setting Bits 0 and 1 disable PWM outputs CH and CL; which keeps these outputs off over the full PWM period regardless of the value in the PWMCHC register. This feature is not only useful for ECM control, but is also required in some space vector modulation schemes. Modifications to the PWMSEG register only become effective at the start of each PWM cycle. Following a reset, all bits in PWMSEG are cleared.

Consider the situation shown in Figure 13 for operation of an ECM with the AH and BL power devices active. The PWM duty cycle registers, PWMCHA and PWMCHB, are programmed with the appropriate on-time value. Since all three PWM registers must be written to trigger an update of the PWM, it is necessary to write also to PWMCHC. For this example, the particular value written to this register is unimportant. Subsequently, crossover bit of the PWMSEG register for Phase B (Bit 7) is set to enable crossover of the Phase B signals. The PWM outputs on AL, BH, CL and CH are disabled by setting Bits 0, 1, 2 and 5 of the PWMSEG register. Thus, for example, the appropriate value for the PWMSEG register is 0x00A7. In addition, in this example, high side chopping of the signal AH is enabled by setting Bit 8 of PWMGATE.

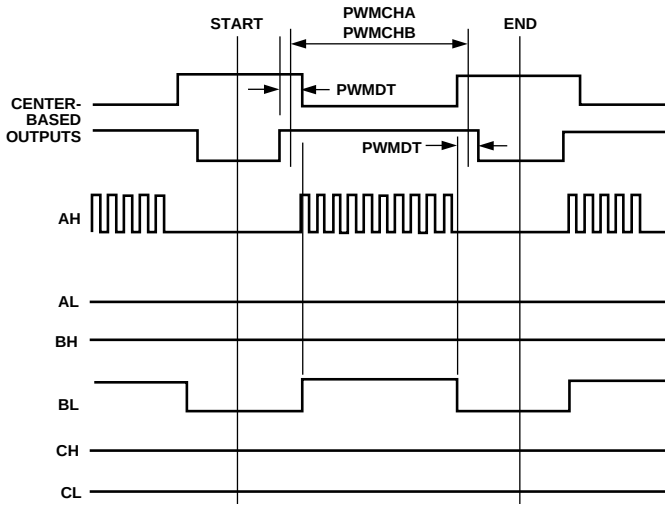


Figure 13. PWM Output Waveforms for an ECM with Inverter Devices AH and BL Active Gate Drive Unit

Gate Drive Unit

The gate drive unit adds features that simplify the interface to a variety of gate drive circuits for PWM inverters. If a transformer coupled gate drive is used, the active PWM signal can be chopped at a high frequency of up to 6.25 MHz. The chopped PWM signals may be required for the high side drivers only or for both high side and low side.

The gate drive chopping feature is enabled by Bits 8 and 9 of the PWMGATE register. Setting Bit 8 enables a chopped PWM signal on all high side output pins AH, BH and CH. Setting Bit 9 enables a chopped PWM signal on all low side output pins AL, BL and CL. The gate chopping frequency is programmed using Bits 0–5 of the PWMGATE register. The gate drive chopping frequency is given by the following equation:

$$f_{CHOP} = \frac{f_{CLKIN}}{2 \times (GATETM + 1)}$$

where *GATETM* is the 6-bit value in Bits 0-5 of the PWMGATE register.

Depending on the type of power device gate drive circuit used, either active high or active low PWM signals will be required, so an external PWM polarity pin is provided. The polarity of the PWMPOL pin determines the active polarity of the PWM output signal (i.e., a low PWMPOL pin means active low PWM). This must be set by hardware because even though the ADMC300 will power up with all PWM outputs off, the correct polarity of an off PWM signal is a function of the gate drive circuit only. The level on the PWMPOL pin is available in Bit 2 of the SYSSTAT register. The PWMPOL pin has an internal pull-up, so if left unconnected, active high PWM is produced.

External PWM Trip

In fault conditions the power devices must be switched off as soon as possible after the fault has been detected; hence an external hardware PWM trip input is provided. A low going $\overline{\text{PWMTRIP}}$ pulse will instantaneously reset the PWM block which will disable all PWM outputs. This will also generate a $\overline{\text{PWMTRIP}}$ interrupt signal and cause a DSP interrupt. The state of the $\overline{\text{PWMTRIP}}$ pin is accessible through Bit 0 of SYSSTAT so that the DSP can determine when the external fault has been cleared. At this point, a full initialization of the PWM controller will be required to restart the PWM. The $\overline{\text{PWMTRIP}}$ pin has an internal pull-down resistor so that if this pin becomes disconnected a PWM trip will be activated.

In addition, each of the digital I/O lines, PIO0 to PIO11, can be configured to operate as extra PWM trip pins. This feature can be enabled for each PIO line by setting the corresponding bit of the PIOPWM register. A low-going pulse on any PIO, with its corresponding PIOPWM bit enabled, will shut down all PWM outputs, similar to the $\overline{\text{PWMTRIP}}$ pin. Following a reset, all PIO lines are configured as inputs, and all PIOPWM bits are set. Because all PIO lines are internally pulled down, any unconnected PIO lines will cause a PWM trip of the ADMC300. It is important, therefore, that the PIO pins be properly configured prior to configuration of the PWM unit of the ADMC300.

The advantage of this feature of the digital I/O lines is that multiple sources may be used to shut down the PWM unit, separately. By combining this feature with the flexible interrupt structure of the digital I/O system, it is possible to generate unique interrupts following a PWM trip from multiple sources, such as over current, over temperature and under voltage etc.

All PWM trip functions, including the $\overline{\text{PWMTRIP}}$ pin and the digital I/O lines operate independently of the DSP clock. Therefore, the PWM system can be safely disabled even in the event of a loss of the DSP clock.

PWM Registers

The configuration of all registers of the PWM system are shown at the end of the data sheet.

ENCODER INTERFACE UNIT

The ADMC300 incorporates an encoder interface to incremental shaft encoders that are often used for position feedback in high performance motion control systems. The Encoder Interface Unit (EIU) includes a 16-bit quadrature up/down counter, input noise filters on the encoder input signals and the zero marker, and has three dedicated pins on the ADMC300. The quadrature encoder signals are applied at the EIA and EIB pins and the optional zero marker may be applied at the EIZP pin. The EIU will operate correctly with input frequencies up to slightly less than a quarter of the CLKIN frequency (3.1 MHz for a 12.5 MHz CLKIN). The EIU may be programmed to use the zero marker for auto-detection of the encoder resolution and/or for position counter error checking. A dedicated encoder interrupt is produced in the event of an encoder counter error. The count direction may be reversed by a control bit, if required. The EIU includes various status bits that indicate encoder feedback loss, the direction of rotation and the initialization state.

The Encoder Interface Unit also includes a high performance Encoder Event Timer (EET) block that permits the accurate timing of successive events of the encoder inputs. The EET can be programmed to time the duration between up to 256 successive encoder pulses and can be used to enhance velocity estimation, particularly at low speeds of rotation.

Encoder Interface Operation

The functional block diagram of the entire encoder interface system of the ADMC300 is shown in Figure 14. The ZERO bit (Bit 1) of the EIUCTRL register determines if the encoder zero marker is used in the operation of the EIU. The EIUCNT register is a 16-bit register that stores the output of the encoder quadrature counter. The EIUMAXCNT register is a 16-bit register that stores the maximum count value.

When the ZERO bit of the EIUCTRL register is cleared, the zero marker is not used by the encoder interface circuitry. In this mode, prior to correct operation, the user *must* configure the EIU with the number of quadrature pulses in one revolution of the incremental encoder. To do this, the user writes EIUMAXCNT with (4N-1) where N is the number of lines on the encoder. Therefore, for a 1024-line encoder, a value of 0x0FFF (= 4095) would be written to the EIUMAXCNT register. However, since absolute position information is not available in this mode, due to the absence of the zero marker, the full 16-bit range of the quadrature counter may be employed by writing a value of 0xFFFF to the EIUMAXCNT register.

The contents of the quadrature counter are updated on each edge of both encoder signals applied on the EIA and EIB pins. Prior to application to the quadrature counter, these signals are filtered to eliminate possible glitches. The direction of counting is determined by Bit 0 (REV) of the EIUCTRL register. If the REV bit is cleared and the EIA-encoder signal leads the EIB-signal, the quadrature counter is incremented on each edge, as shown in Figure 15. This is defined as the forward direction of motion. Conversely, if the EIB signal leads the EIA signal, the counter is decremented on each encoder edge and this denotes the reverse direction. The direction of counting can be reversed by setting Bit 0 of the EIUCTRL registers.

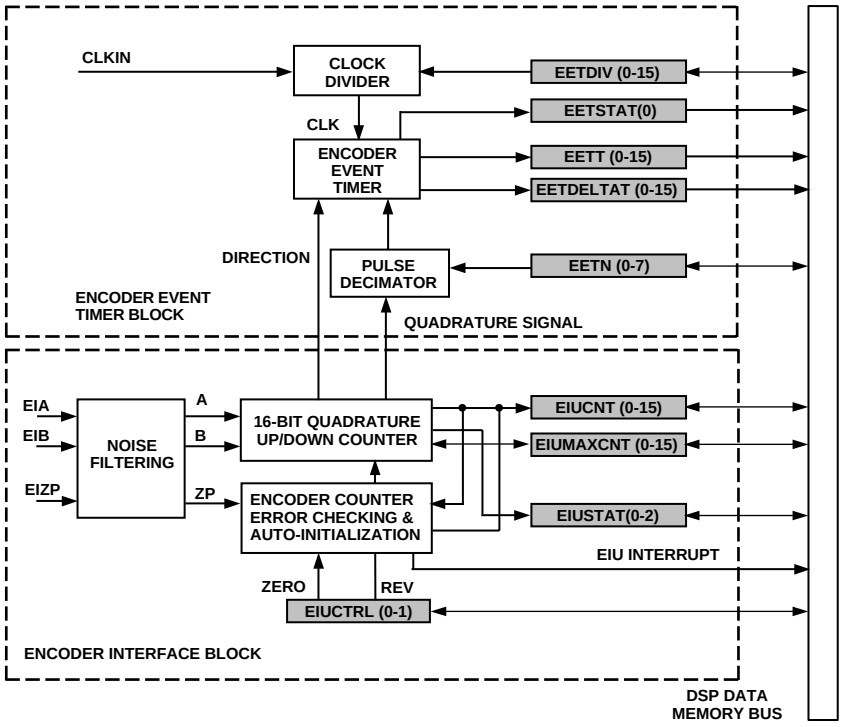


Figure 14. Configuration of Encoder Interface System of ADMC300

As shown in Figure 15, the two encoder signals are used to derive a quadrature signal that is used, in conjunction with a direction bit, to increment or decrement the encoder counter and also the encoder event timer. The status of the direction signal is indicated at Bit 1 of the EIUSTAT register. While the encoder counter is incrementing, Bit 1 is set. Alternatively, when the encoder counter is decrementing, Bit 1 of the EIUSTAT register is cleared.

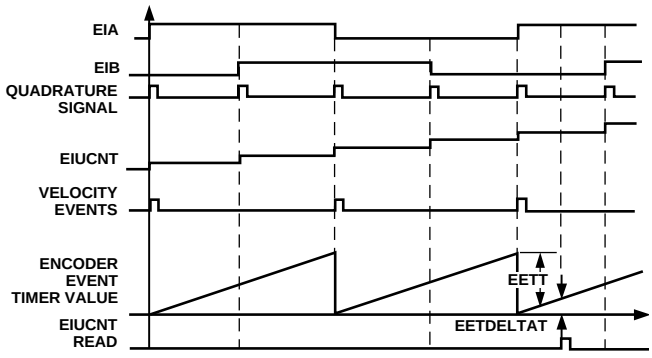


Figure 15. Operation of Encoder Interface Unit and Encoder Event Timer of ADMC300 in Forward Direction with EETN = 2

The alternative operating mode of the Encoder Interface Unit is enabled when Bit 1 of the EIUCTRL register is set. In this mode, the encoder interface circuitry makes use of the encoder zero marker that is applied at the EIZP pin of the ADMC300. The zero marker is used for two distinct purposes. Provided the user does not write directly to the EIUMAXCNT register, the zero marker may be used to automatically determine the encoder resolution. The first occurrence of the zero marker is used to clear the EIUCNT register to zero. On the next occurrence of the zero marker, the contents of the EIUCNT register are transferred to the EIUMAXCNT register. However, if the user writes directly to the EIUMAXCNT register, that value is not overwritten by this auto-initialization. While the EIU is not initialized, Bit 2 of the EIUSTAT register is set.

The zero marker is also used to detect an encoder count error when bit 1 of EIUCTRL is set. Whenever the zero marker is detected, the contents of the EIUCNT register are compared with 0x0000. If all encoder edges have been correctly counted and the EIUMAXCNT register is correctly set (either by the user or by auto-determination) the value of the EIUCNT register should be zero when the zero marker occurs. If this is not the case and a value other than 0x0000 is detected in the EIUCNT register, then Bit 0 of the EIUSTAT register is set to indicate an encoder count error. In addition, an EIU interrupt may be generated when an encoder position count error is detected. This dedicated interrupt is also managed and may be masked by the Programmable Interrupt Controller (PIC) block. The encoder continues to count encoder edges after an error has been detected.

Encoder Interface Unit Registers

The registers associated with the Encoder Interface Unit are shown at the end of the data sheet.

Encoder Event Timer Operation

The Encoder Event Timer (EET) block forms an integral part of the EIU of the ADMC300. The EET accurately times the duration between encoder events. The information provided by the EET may be used to make allowances for the asynchronous timing of encoder and DSP-reading events. As a result, more accurate computations of the position and velocity of the motor shaft may be performed.

The EET consists of a 16-bit encoder event timer, an encoder pulse decimator and a clock divider as shown in Figure 14. The EET clock frequency is selected by the 16-bit EETDIV clock divide register, whose value divides the CLKIN frequency. The contents of the encoder event timer are incremented on each rising edge of the divided clock signal. An EETDIV value of zero gives the maximum divide value of 0x10000 (= 65,536), so that the clock frequency to the encoder event timer is at its minimum possible value.

The quadrature signal from the EIU is decimated at a rate determined by the 8-bit EETN register. For example, writing a value of 4 to EETN, decimates the quadrature signal by four and then provides this decimated signal to the EET. The rising edge of this decimated signal is termed a velocity event. Therefore, for an EETN value of 4, a velocity event occurs every four encoder edges, or once every period of one of the encoder signals. An EETN value of zero gives an effective pulse decimation value of 256.

On the occurrence of a velocity event, the contents of the encoder event timer are stored in a temporary interval time register. Under normal operation, the interval time register stores the elapsed time between successive velocity events. After the timer value has been latched to the interval time register at the velocity event, the contents of the encoder event timer are reset to one.

Whenever the EIUCNT register is read, the current value of the encoder event timer is latched to the 16-bit EETDELTAT register and the current value of the interval time register is latched into the 16-bit EETT register. At that time, the contents of the three registers, EIUCNT, EETT and EETDELTAT may be considered an instantaneous snapshot of the state of the encoder system. The EETDELTAT register is the time from the last velocity event to the instant at which the EIUCNT register is read by the DSP. The information in these three registers may then be used to accurately compute rotor velocity and position. Operation of the encoder event timer with EETN of 2 is shown in Figure 15.

There is a 1-bit EETSTAT register that indicates whether or not an overflow of the encoder event timer has occurred. If the time between successive velocity events is sufficiently long, it is possible that the encoder event timer will overflow. When this condition is detected, Bit 0 of the EETSTAT register is set and the EETT register is fixed at 0xFFFF. Reading the EETSTAT register clears the overflow bit and permits the EETT register to be updated at the next velocity event.

If an encoder direction reversal is detected by the EIU, the encoder event timer is set to zero and the EETT register is set to its maximum 0xFFFF value. Subsequent velocity events will cause the EETT register to be updated with the correct value. If a value of 0xFFFF is read from the EETT register, Bit 0 of the EETSTAT register can be read to determine whether an overflow or direction reversal condition exists. In the case of a direction reversal, the contents of the EETDELTAT register is valid, representing the time from the direction reversal to the instant at which the EIUCNT register is read.

On reset, the EETN, EETDIV, EETDELTAT and EETT registers are all cleared to zero. Whenever either the EETN or EETDIV registers are written to, the encoder event timer is reset to zero.

Encoder Event Timer Registers

The registers associated with the Encoder Event Timer are summarized at the end of the data sheet.

PROGRAMMABLE INPUT/OUTPUT

The ADMC300 has 12 programmable digital input/output pins called PIO0 to PIO11. Each pin may be individually configured as either an input or an output. An associated data register may be used to read data from pins configured as inputs and write data to pins configured as outputs. In addition, each I/O line may be configured as an interrupt source. Both edge (rising and falling) and level (high and low) interrupts may be detected. Four of the PIO lines (PIO0 to PIO3) have dedicated vector addresses in the interrupt table. The remaining eight interrupts (PIO4 to PIO11) are multiplexed into a single additional interrupt vector location. The PIOFLAG register is used to determine which line caused the interrupt.

In addition, all PIO lines may be alternatively configured as PWM trip sources. The PIOPWM register has dedicated bits that may be used to enable this function on each PIO line. In this mode, a low level on any pin configured as a PWM trip source shuts down the PWM in a manner identical to the $\overline{\text{PWMTRIP}}$ pin.

PIO Configuration

Each of the 12 programmable input/output lines may be configured as either an input or an output by programming the appropriate bits of the PIODIR register. This 12-bit read/write register has one bit associated with each I/O line; Bit 0 corresponds to PIO0, etc. Clearing a bit in the PIODIR register will configure the corresponding pin as an input pin. Conversely, setting a bit configures the pin as an output pin. On reset, bits of the PIODIR register are cleared so that all 12 PIO pins are configured as inputs. In addition, all PIO lines are internally pulled down in the ADMC300 so that unconnected lines are seen as low level inputs.

Three of the PIO lines also serve alternate functions. PIO9 is multiplexed as the external convert start signal for the ADC system. Signals on this pin can be used to trigger updating of the ADC data registers, if required. Also, PIO10 and PIO11 may be used as inputs to the Event Timer Unit (ETU) to accurately time the period, frequency or duty cycle of external signals. If these functions are not required, the three pins may be used as general purpose I/O lines.

PIO Data Reading/Writing

Associated with the PIO system is a data register, PIODATA, that also has a bit associated with each I/O line. Data written to the PIODATA register will appear on those pins configured as outputs. In addition, reading the PIODATA register will read the data from those pins configured as inputs.

PIO Interrupt Generation

Each of the twelve PIO lines may be configured as an interrupt source. Four of the PIO lines, PIO0 to PIO3, have dedicated interrupt vector locations while the remaining eight are multiplexed into an additional interrupt vector. The PIOINTEN enable function is used to enable or disable interrupts on the PIO4 to PIO11 lines. The PICMASK register of the programmable interrupt controller is used to enable interrupts on the four dedicated PIO lines, PIO0 to PIO3. Interrupts may be generated on either edge (rising or falling) or level (high or low) events by programming the appropriate bits of both the PIOMODE and PIOLEVEL registers. Both registers have a

dedicated bit for each of the twelve PIO lines. Setting the appropriate bit of the PIOMODE register configures the interrupt as level-sensitive, while clearing the bit configures the corresponding PIO to be edge sensitive. In level-sensitive mode (PIOMODE bit is 1), setting the corresponding bit in the PIOLEVEL register configures the interrupt as active high, while clearing the bit in the PIOLEVEL register configures it for active low. In edge-sensitive mode (PIOMODE bit is 0), setting the corresponding bit of the PIOLEVEL register configures the interrupt for rising edge, while clearing the bit configures the interrupt for falling edge. On reset, all PIO interrupts are disabled. The appropriate register settings for correct PIO interrupt configuration is shown in Table V.

Table V. PIO Interrupt Configuration

	PIOMODE	
PIOLEVEL	**0**	**1**
0	Falling Edge	Active Low
1	Rising Edge	Active High

The four dedicated PIO interrupts from PIO0 to PIO3 have interrupt vector addresses at program memory addresses 0x0048 for PIO0, 0x004C for PIO1, 0x0050 for PIO2 and 0x0054 for PIO3. In the event of an interrupt on PIO4 to PIO11, the corresponding bit of the PIOFLAG register is set and the general PIO interrupt is activated. This interrupt has a dedicated vector address at location 0x003C. In the interrupt service routine for this interrupt, the user must poll the PIOFLAG register to determine which of the PIO4 to PIO11 lines, that have interrupts enabled, caused the interrupt. Of course, if only one of the PIO4 to PIO11 lines have interrupts enabled, no polling is necessary. Reading the PIOFLAG register clears all bits of the register.

PIO lines that are configured as outputs may also be used to generate interrupts. If, for example, one of the PIO lines is configured simultaneously as an output and as an interrupt source, writing the appropriate data to the PIODATA register will trigger an interrupt.

PIO as PWM Trip Sources

By setting the appropriate bits of the PIOPWM register, each of the twelve PIO lines can be configured as a PWM trip source. In this mode, a low level on the PIO pin will cause a PWM trip that will disable all six PWM outputs on AH to CL. The disabling of the PWM is independent of the DSP clock, so that the PWM stage can be fully protected even in the event of a loss of clock signal to the DSP.

A $\overline{\text{PWMTRIP}}$ interrupt will be generated when the PWM is reset (whether the PWM is reset via a PIO configured as a trip source, or via the $\overline{\text{PWMTRIP}}$ pin). It is also possible to generate the normal PIO interrupts on the occurrence of a falling-edge on the PIO line. The advantage of this highly flexible structure for PWM shutdown is that multiple fault signals could be applied to the ADMC300 at different PIO lines. The occurrence of a falling-edge on any of them will instantaneously shut down the PWM. However, based on the particular PIO interrupt that is flagged, the user can easily determine the source of the trip. This permits the action of the interrupt service routines following a PWM trip to be tailored to the particular fault that occurred.

On reset, all PIO lines are configured as PWM trip sources. Because all PIO lines are also configured as inputs and have internal pull-down resistors, any unconnected PIO lines will cause a PWM trip. Therefore, prior to using the PWM unit of the ADMC300, it is imperative that the PIO state be correctly configured for the particular application.

PIO Registers

The configuration of all registers associated with the PIO system are shown at the end of the data sheet. Each of the registers has a bit directly associated with one of the PIO lines. For example, Bit 0 of all registers affects only the PIO0 line of the ADMC300.

AUXILIARY PWM OUTPUTS

The ADMC300 provides two auxiliary, fixed-frequency, variable duty cycle PWM outputs that may be used to drive auxiliary switching circuits in the motor control system. Alternatively, by adding appropriate filtering at the output, these signals can be used as to provide a simple digital-to-analog converter. These output signals appear on the AUX0 and AUX1 pins and are controlled by the duty cycle registers, AUXTIM0 and AUXTIM1.

The auxiliary PWM outputs operate at a fixed frequency that is $f_{CLKIN}/256$. This gives an auxiliary PWM switching frequency of 48.8 kHz for a 12.5 MHz CLKIN. The output duty cycle at the auxiliary PWM output pin is controlled by comparing the 8-bit auxiliary PWM duty-cycle registers, AUXTIM0 and AUXTIM1 with the contents of a timer. The value written to these registers may range from 0 to 255 so that duty cycles from 0 to 99.6% may be produced at the output pins. A simple filter at the output could then be used to produce a corresponding analog output from 0 to 0.996 V_{DD}.

The outputs of the two auxiliary PWM timer circuits are synchronized on their rising edges. When the auxiliary timer registers are written to, the value becomes effective immediately. Therefore, if the value is smaller than the present timer value, the outputs go low immediately. The correct duty cycle appears for the subsequent auxiliary PWM period. On reset, the AUXTIM0 and AUXTIM1 registers are cleared so that no auxiliary PWM signals are produced and the AUX0 and AUX1 pins are low until these registers are programmed. The format of the AUXTIM0 and AUXTIM1 registers is shown at the end of the data sheet.

WATCHDOG TIMER

The ADMC300 incorporates a watchdog timer that can perform a full reset of the DSP and motor control peripherals in the event of software error. The watchdog timer is enabled by writing a timeout value to the 16-bit WDTIMER register. The timeout value represents the number of CLKIN cycles required for the watchdog timer to count down to zero. When the watchdog timer reaches zero, a full DSP core and motor control peripheral reset is performed. In addition, Bit 1 of the SYSSTAT register is set so that after reset the ADMC300 can determine that the reset was due to the time out of the watchdog timer and not a power-on reset. Following a reset, Bit 1 of the SYSSTAT register may be cleared by writing zero to the WDTIMER register. This clears the status bit but does not enable the watchdog timer.

On reset, the watchdog timer is disabled and is only enabled when the first timeout value is written to the WDTIMER

register. To prevent the watchdog timer from timing out, the user must write to the WDTIMER register at regular intervals (shorter than the programmed WDTIMER period value). On all but the first write to WDTIMER, the particular value written to the register is unimportant since writing to WDTIMER simply reloads the first value written to this register. The WDTIMER register is memory mapped to data memory at location 0x2018.

EVENT TIMER UNIT

The ADMC300 contains a dual channel Event Timer Unit (ETU) that may be used to accurately measure the elapsed time between defined events on a particular channel. The ETU uses two input pins, ETU0 and ETU1, that are multiplexed with the PIO10 and PIO11 pins. The ETU system contains a set of 16-bit data registers that are used to store the value of the dedicated ETU timer on the occurrence of the defined events on the input pins. A configuration register is used to define the nature of the events on each of the input pins. In addition, a control register is used to initiate event capture on the inputs. A status register may be read to determine the state of the two capture channels. A dedicated ETU interrupt may be generated upon completion of a capture sequence on either the ETU0 or ETU1 channel.

An event may be defined as either a rising or falling edge on the associated ETU0 and ETU1 inputs pins. Therefore, the ETU system can be used to compute the frequency, period, duty cycle or on-time of signals applied at the inputs. A functional block diagram of the ETU system of the ADMC300 is shown in Figure 16.

ETU Event Definition

The ETU system of the ADMC300 contains a dedicated 16-bit timer whose clock frequency may be programmed using the ETUDIVIDE register. This register divides the CLKIN frequency to provide the clock signal for the ETU timer. The clock frequency of the ETU timer may be expressed as f_{CLKIN}/ ETUDIVIDE and is common to both channels.

Two events are used to trigger the ETU, termed Event A and Event B. By setting the appropriate bits of the ETUCONFIG register, it is possible to define both Events A and B as either rising or falling edges on the appropriate pin. For example, setting Bit 0 of the ETUCONFIG register defines Event A of the ETU0 channel as a rising edge on the ETU0 pin. Similarly, setting Bit 4 of the ETUCONFIG register defines Event A of the ETU1 channel as a rising edge on the ETU1 pin. Event A defines the start of the event capture sequence. Associated with each ETU channel are three data registers, ETUA0, ETUB0 and ETUAA0 for ETU Channel 0 and ETUA1, ETUB1 and ETUAA1 for ETU Channel 1. These data registers store the ETU timer value on the occurrence of the first A Event, the first B Event and the second A Event respectively. For example, for ETU Channel 0, ETUA0 stores the timer value on the first occurrence of Event A on the ETU0 pin, ETUB0 stores the timer value on the first occurrence of Event B on the ETU0 pin and ETUAA0 stores the timer value on the second occurrence of Event A on the ETU0 pin. Registers ETUA1, ETUB1 and ETUAA1 perform the same function for events on ETU Channel 1.

Because the ETU0 and ETU1 pins are multiplexed with the PIO10 and PIO11 pins, it is possible to configure these lines as digital outputs using the PIODIR register. In this mode, writing suitable patterns to the PIODATA register will trigger the corresponding event capture on the ETU channels.

ETU Interrupt Generation

The completion of the event capture sequence can be defined as either the occurrence of Event B or the second occurrence of Event A by setting the appropriate bits of the ETUCONFIG register. At the end of the capture sequence, the ETU generates an interrupt. For example, if Bit 2 of the ETUCONFIG register is set, ETU Channel 0 will generate an ETU interrupt on the occurrence of Event B on the ETU0 pin. On the other hand, if Bit 6 of the ETUCONFIG register is cleared, ETU Channel 1 will generate an ETU interrupt on the occurrence of the second Event A on the ETU1 pin. Both ETU channels generate the same interrupt to the DSP when capture is complete. If both ETU channels are used simultaneously, the ETUSTAT register can be polled to determine which caused the interrupt. If capture on ETU Channel 0 is complete, Bit 0 of the ETUSTAT register is set; if capture on ETU Channel 1 is complete, Bit 1 is set. Reading the ETUSTAT register automatically clears all bits of the register.

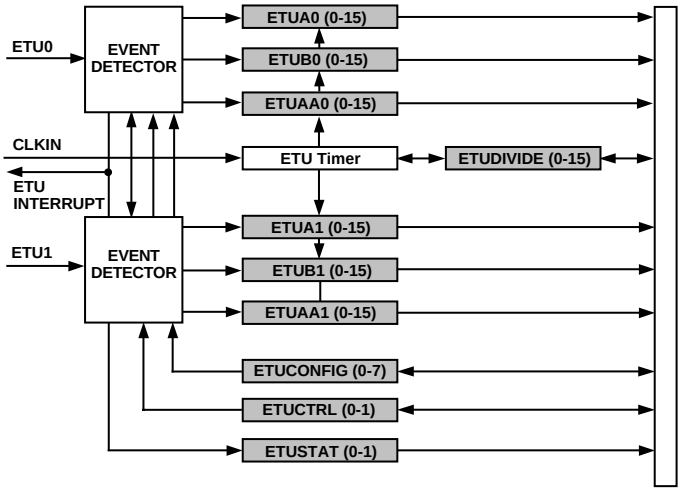


Figure 16. Functional Block Diagram of Event Timer Unit

ETU Operating Modes

The ETU channels of the ADMC300 can operate in two distinct modes; single shot and free-running. The particular mode may be selected for ETU Channel 0 by programming Bit 3 of the ETUCONFIG register and for ETU channel 1 by programming Bit 7 of the ETUCONFIG register. Setting these bits puts the respective ETU channel in free-running mode while clearing the bits enables the single-shot mode. In single-shot mode, upon completion of the capture sequence, further event capture is disabled until the appropriate bit of the ETUCTRL register has been set. Setting Bit 0 of the ETUCTRL register restarts the capture for ETU Channel 0, while Bit 1 restarts capture for Channel 1. In the free-running mode, the bits of the ETUCTRL register remain set and the ETU channel continues to capture following the generation of the interrupt.

ETU Registers
The configuration of the ETU registers is shown at the end of the data sheet.

INTERRUPT CONTROL
Operation and control of the various interrupt sources is managed by a combination of the internal interrupt controller of the DSP core and a dedicated programmable interrupt controller (PIC) that managers all interrupts from the motor control peripherals. Eight internal DSP core interrupts comprise the peripheral ($\overline{IRQ2}$), SPORT0 transmit and receive, two software, SPORT1 transmit and receive (or alternatively $\overline{IRQ1}$ and $\overline{IRQ0}$), and the timer interrupts. The ADMC300 includes eleven additional interrupts that are interfaced to the DSP core through the $\overline{IRQ2}$ interrupt. The eleven peripheral interrupts include two ADC interrupts (one per bank), the PWMSYNC interrupt, the five PIO interrupts (four dedicated to PIO0 to PIO3 and the combined PIO4 to PIO11 interrupt), the EIU interrupt, the ETU interrupt and the $\overline{PWMTRIP}$ interrupt. Each of the nineteen interrupts of the ADMC300 has a dedicated four word section in the interrupt vector table. The start address in the interrupt vector table for each of the nineteen ADMC300 interrupt sources is tabulated in Table VI. The interrupts are listed from the highest priority to the lowest priority.

The entire interrupt control system of the ADMC300 is configured and controlled by the IFC, IMASK and ICNTL registers of the DSP core and the PICMASK and PICVECTOR registers of the PIC block.

Table VI. Interrupt Vector Addresses

Interrupt Source	Interrupt Vector Address
Peripheral Interrupt ($\overline{IRQ2}$)	0x0004 (Highest Priority)
ADC Bank A Update	0x0030
PWMSYNC	0x0034
ADC Bank B Update	0x0038
PIO Interrupt (PIO4 to PIO11)	0x003C
Encoder Interface Interrupt	0x0040
Event Timer Unit Interrupt	0x0044
PIO0 Interrupt	0x0048
PIO1 Interrupt	0x004C
PIO2 Interrupt	0x0050
PIO3 Interrupt	0x0054
$\overline{PWMTRIP}$ Interrupt	0x0058
SPORT0 Transmit Interrupt	0x0010
SPORT0 Receive Interrupt	0x0014
Software Interrupt 1	0x0018
Software Interrupt 0	0x001C
SPORT1 Transmit Interrupt or $\overline{IRQ1}$	0x0020
SPORT1 Receive Interrupt or $\overline{IRQ0}$	0x0024
Timer	0x0028 (Lowest Priority)

Interrupt Masking
Interrupt masking (or disabling) is controlled by the IMASK register of the DSP core and the PICMASK register. These registers contain individual bits that must be set to enable the various interrupt sources. It is important to remember that if any peripheral interrupt is to be enabled both the $\overline{IRQ2}$ interrupt enable bit (Bit 9) of the IMASK register and the appropriate bit of the PICMASK register must be set. The configuration of both the IMASK and PICMASK registers of the ADMC300 is shown at the end of the data sheet.

Interrupt Configuration
The IFC and ICNTL registers of the DSP core control and configure the interrupt controller of the DSP core. The IFC register is a 16-bit register that may be used to force and/or clear any of the eight DSP interrupts. Bits 0 to 7 of the IFC register may be used to clear the DSP interrupts while Bits 8 to 15 can be used to force a corresponding interrupt. Writing to Bits 11 and 12 in IFC is the only way to create the two software interrupts.

The ICNTL register is used to configure the sensitivity (edge or level) of the $\overline{IRQ0}$, $\overline{IRQ1}$ and $\overline{IRQ2}$ interrupts and to enable/disable interrupt nesting. Setting Bit 0 of ICNTL configures the $\overline{IRQ0}$ as edge sensitive while clearing the bit configures it for level sensitive. Bit 1 is used to configure the $\overline{IRQ1}$ interrupt and Bit 2 is used to configure the $\overline{IRQ2}$ interrupt. It is recommended that the $\overline{IRQ2}$ interrupt be always configured for level sensitive as this ensures that no peripheral interrupts are lost. Setting Bit 4 of the ICNTL register enables interrupt nesting. The configuration of both IFC and ICNTL registers is shown at the end of the data sheet.

Interrupt Operation
Following a reset, the ROM code monitor of the ADMC300 copies a default interrupt vector table into program memory RAM from address 0x0000 to 0x005F. Since each interrupt source has a dedicated four word space in this vector table, it is possible to code short interrupt service routines (ISR) in place. Alternatively, it may be required to insert a JUMP instruction to the appropriate start address of the interrupt service routine if more memory is required for the ISR.

On the occurrence of an interrupt, the program sequencer ensures that there is no latency (beyond synchronization delay) when processing unmasked interrupts. In the case of the timer, SPORT0, SPORT1 and software interrupts, the interrupt controller automatically jumps to the appropriate location in the interrupt vector table. At this point, a JUMP instruction to the appropriate ISR is required.

In the event of a motor control peripheral interrupt, the operation is slightly different. For any of the eleven peripheral interrupts, the interrupt controller automatically jumps to location 0x0004 in the interrupt vector table. In addition, the required vector address (between 0x0030 and 0x0058) associated with the particular interrupt source is placed in the PICVECTOR register of the PIC block. Code loaded at location 0x0004 by the monitor on reset subsequently performs a JUMP from location 0x0004 to the address specified in the PICVECTOR register. This operation with the PICVECTOR register results in a slightly longer latency associated with processing any of the peripheral interrupts, as compared with the latency of the internal DSP core interrupts.

The code located at location 0x0004 by the monitor on reset is as follows:

```
0x0004: DM (I4_SAVE) = I4;
        I4 = DM (PICVECTOR);
        JUMP (I4);
```

The default code for each of the motor control peripherals is:

```
        I4 = DM (I4_SAVE);
        RTI;
```

Note that this default restores I4 to its value before the interrupt. The user should replace the RTI with a JUMP to their ISR. The PUT_VECTOR ROM subroutine can be used to replace the RTI with the JUMP.

The PIC block manages the sequencing of the eleven motor control peripheral interrupts. In the case of multiple simultaneous interrupts, the PIC will load the PICVECTOR register with the vector address of the highest priority pending interrupt. The contents of the PICVECTOR register will remain fixed until read by the DSP. This action is performed by the default DSP code at location 0x0004. The PIC block only asserts a new interrupt after the PICVECTOR register has been read.

SYSTEM CONTROLLER

The system controller block of the ADMC300 performs a number of distinct functions:

1. Manages the interface and data transfer between the DSP core and the motor control peripherals.
2. Controls the multiplexing of the SPORT1 pins to select either the DR1A or DR1B data receive pins. It also allows configuration of SPORT1 as a UART interface.
3. Manages the software flags of the DSP core.
4. Contains a status register (SYSSTAT) that indicates the state of the $\overline{\text{PWMTRIP}}$, PWMPOL pins and the watchdog timer.
5. Performs a reset of the motor control peripherals and control registers following a hardware, software or watchdog initiated reset.

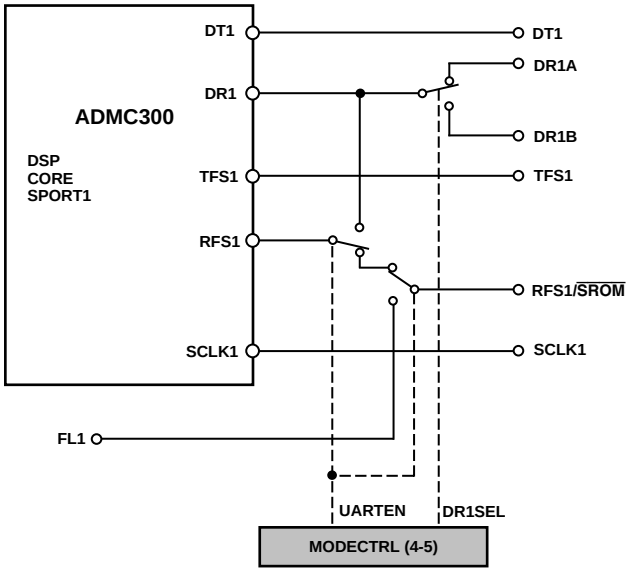


Figure 17. Internal Multiplexing of SPORT1 Pins

SPORT1 Control

The ADMC300 uses SPORT1 as the default serial port for boot loading and as the interface to the development environment. There are two data receive pins, DR1A and DR1B, on the ADMC300. This permits DR1A to be used as the data receive pin when interfacing to serial ROM or E^2PROM for boot loading. Alternatively, if connecting through a UART for either boot loading or interface to the development environment, the DR1B pin can be used. Both data receive pins are multiplexed internally into the single data receive input of SPORT1. Two control bits in the MODECTRL register control the state of the SPORT1 pins by manipulating internal multiplexers in the ADMC300. The configuration of SPORT1 is illustrated in Figure 17.

Bit 4 of the MODECTRL register (DR1SEL) selects between the two data receive pins. Setting Bit 4 of MODECTRL connects the DR1B pin to the internal data receive port DR1 of SPORT1. Clearing Bit 4 connects DR1A to DR1.

Setting Bit 5 of the MODECTRL register (UARTEN) configures the serial port for UART mode. In this mode, the DR1 and RFS1 pins of the internal serial port are connected together. Additionally, setting the UARTEN bit connects the FL1 flag of the DSP to the external RFS1/$\overline{\text{SROM}}$ pin. In this mode, this pin is intended to be used to reset the external serial ROM device.

The monitor code in ROM automatically configures the SPORT1 pins during the boot sequence. Initially, the DR1SEL bit is cleared and the UARTEN bit is set so that the ADMC300 first attempts to perform a reset of the external memory device using the RFS1/SROM pin. This is accomplished by toggling the FL1 flag using the following code segment:

```
SROMRESET: SET FL1;
           TOGGLE FL1;
           TOGGLE FL1;
           RTS;
```

If successful, data will be clocked from the external device in a continuous stream. The start of the data stream is detected by the serial port on the RFS1 pin, which is connected internally to the DR1 pin in this mode. If the serial load is successful, code is downloaded and execution begins at the start of user program memory (address 0x0060). Following a synchronous boot load, SPORT1 could be configured for normal synchronous serial mode by setting the DR1SEL pin to select the DR1B data receive pin and by clearing the UARTEN bit to return to SPORT mode.

Failing a synchronous boot load, the ADMC300 monitor automatically sets the DR1SEL bit to select the DR1B pin and remains in UARTEN mode. The monitor code then waits for a header byte that tells it whether a UART (SCI) boot load or a UART debugger interface is to be performed. Obviously, if a debugger interface is required on SPORT1, it is not possible to use SPORT1 as a general purpose synchronous serial port. If such a serial port is required, it is recommended that SPORT0 be used.

Flag Pins

The ADMC300 provides flag pins. The alternate configuration of SPORT1 includes a Flag In (FI) and Flag Out (FO) pin. This alternate configuration of SPORT1 is selected by Bit 10 of the DSP system control register, SYSCNTL at data memory address, 0x3FFF. In the alternate configuration, the DR1 pin (either DR1A or DR1B depending on the state of the DR1SEL bit) becomes the FI pin and the DT1 pin becomes the FO pin. Additionally, RFS1 is configured as the $\overline{\text{IRQ0}}$ interrupt input and TFS1 is configured as the $\overline{\text{IRQ1}}$ interrupt. The serial port clock, SCLK1, is still available in the alternate configuration. Following boot loading from a serial memory device, it is possible to reconfigure the SPORT1 to this alternate configuration. However, if a debugger interface is used, this configuration is not possible as the normal serial port pins are required for debugger communications.

The ADMC300 also contains two software flags, FL1 and FL2. These flags may be controlled in software and perform specific functions on the ADMC300. The FL1 pin has already been described and is used to perform a reset of the external memory device via the RFS1/$\overline{\text{SROM}}$ pin. The FL2 flag is used specifically to perform a full peripheral reset of the chip (including the watchdog timer). This is accomplished by toggling the FL2 flag in software using the following code segment:

```
PRESET:  SET FL2:
         TOGGLE FL2;
         TOGGLE FL2;
         RTS;
```

System Controller Registers

The system controller includes two registers, the MODECTRL register used to control the multiplexing of the SPORT1 pins and the SYSSTAT register that displays various status information. The format of these registers is shown at the end of the data sheet.

Bit 0 of the SYSSTAT register indicates the state of the $\overline{\text{PWMTRIP}}$ pin. If this bit is set, the $\overline{\text{PWMTRIP}}$ pin is high and no PWM trip is occurring. If this bit is cleared, then the PWM is shut down. Bit 1 of the SYSSTAT register is set following a watchdog timeout. This bit is cleared in normal operation. Finally, Bit 2 indicates the status of the PWMPOL pin. If this bit is set, the PWMPOL pin is high and active high PWM outputs will be produced.

Register Memory Map

The address, name, used bits and function of all motor control peripheral registers of the ADMC300 are tabulated in Table VII. In addition, the relevant DSP core registers are tabulated in Table VIII. Full details of the DSP core registers can be obtained by referring to the ADSP-2171 sections of the *ADSP-2100 Family User's Manual, Third Edition*.

Development Kit

To facilitate device evaluation and programming, an evaluation kit (ADMC300-EVAL KIT) is available from Analog Devices. The evaluation kit consists of an evaluation board and the Motion Control Debugger software. The evaluation kit contains latest programming and device information. It is recommended that the evaluation kit be used for initial program development.

Table VII. Peripheral Register Map of ADMC300

ADDRESS	NAME	BITS	FUNCTION
0x2000–0x2007			Reserved
0x2008	PWMTM	[0 . . 11]	PWM Period Register
0x2009	PWMDT	[1 . . 6]	PWM Deadtime Register
0x200A	PWMPD	[0 . . 6]	PWM Pulse Deletion Register
0x200B	PWMGATE	[0 . . 5, 8, 9]	PWM Gate Drive Configuration
0x200C	PWMCHA	[0 . . 11]	PWM Channel A Duty Cycle
0x200D	PWMCHB	[0 . . 11]	PWM Channel B Duty Cycle
0x200E	PWMCHC	[0 . . 11]	PWM Channel C Duty Cycle
0x200F	PWMSEG	[0 . . 8]	PWM Segment Select Register
0x2010	AUXTIM0	[0 . . 7]	Aux. PWM Channel 0 Duty Cycle
0x2011	AUXTIM1	[0 . . 7]	Aux. PWM Channel 1 Duty Cycle
0x2012–0x2014			Reserved
0x2015	MODECTRL	[4 . . 5]	Mode Control Register
0x2016	SYSSTAT	[0 . . 2]	System Status Register
0x2017			Reserved
0x2018	WDTIMER	[0 . . 15]	Watchdog Timer Register
0x2019–0x201B			Reserved
0x201C	PICVECTOR	[0 . . 15]	Peripheral Interrupt Vector Address
0x201D	PICMASK	[0 . . 10]	Peripheral Interrupt Mask Register
0x201E-0x201F			Reserved
0x2020	EIUCNT	[0 . . 15]	Encoder Count Register
0x2021	EIUMAXCNT	[0 . . 15]	Encoder Maximum Count Register
0x2022	EIUSTAT	[0 . . 2]	Encoder Interface Status Register
0x2023	EIUCTRL	[0 . . 1]	Encoder Interface Control Register
0x2024–0x2027			Reserved
0x2028	EETN	[0 . . 7]	Encoder Event Timer Pulse Decimator
0x2029	EETDIV	[0 . . 15]	Encoder Event Timer Clock Divide
0x202A	EETDELTAT	[0 . . 15]	Encoder Event Timer Delta Time
0x202B	EETT	[0 . . 15]	Encoder Event Timer Period Register
0x202C	EETSTAT	[0]	Encoder Event Timer Status Register
0x202D–0x202F			Reserved
0x2030	ADC1	[0 . . 15]	ADC Channel 1 Data Register
0x2031	ADC2	[0 . . 15]	ADC Channel 2 Data Register
0x2032	ADC3	[0 . . 15]	ADC Channel 3 Data Register
0x2033	ADC4	[0 . . 15]	ADC Channel 4 Data Register
0x2034	ADC5	[0 . . 15]	ADC Channel 5 Data Register
0x2035			Reserved
0x2036	ADCCTRL	[0 . . 15]	ADC Control Register
0x2037	ADCSTAT	[0]	ADC Status Register
0x2038	ADCSYNC	[0 . . 6]	ADC Synchronization Register
0x2039	ADCDIVA	[6 . . 11]	ADC Bank A Clock Divide Register
0x203A	ADCDIVB	[6 . . 11]	ADC Bank B Clock Divide Register
0x203B	ADCAL	[0 . . 4]	ADC Calibration Register
0x203C–0x203F			Reserved
0x2040	PIOLEVEL	[0 . . 11]	PIO Interrupt Configuration
0x2041	PIOMODE	[0 . . 11]	PIO Interrupt Mode Control
0x2042	PIOPWM	[0 . . 11]	PIO PWM Trip Control
0x2043			Reserved
0x2044	PIODIR	[0 . . 11]	PIO Direction Control
0x2045	PIODATA	[0 . . 11]	PIO Data Register
0x2046	PIOINTEN	[4 . . 11]	PIO Interrupt Enable Register
0x2047	PIOFLAG	[4 . . 11]	PIO Interrupt Flag Register
0x2048–0x204F			Reserved
0x2050	ETUA0	[0 . . 15]	Event Timer Event A—Channel 0
0x2051	ETUB0	[0 . . 15]	Event Timer Event B—Channel 0
0x2052	ETUAA0	[0 . . 15]	Event Timer Event AA—Channel 0
0x2053	ETUA1	[0 . . 15]	Event Timer A—Channnel 1

Table VII. Peripheral Register Map of ADMC300 (Continued)

ADDRESS	NAME	BITS	FUNCTION
0x2054	ETUB1	[0 . . 15]	Event Timer Event B—Channel 1
0x2055	ETUAA1	[0 . . 15]	Event Timer Event AA—Channel 1
0x2056–0x205B			Reserved
0x205C	ETUCONFIG	[0 . . 7]	Event Timer Configuration
0x205D	ETUDIVIDE	[0 . . 15]	Event Timer Clock Divide Register
0x205E	ETUSTAT	[0 . . 1]	Event Timer Status Register
0x205F	ETUCTRL	[0 . . 1]	Event Timer Control Register
0x2060–0x20FF			Reserved

Table VIII. DSP Core Registers

ADDRESS	NAME	BITS	FUNCTION
0x3FFF	SYSCNTL	[0 . . 15]	System Control Register
0x3FFE	MEMWAIT	[0 . . 15]	Memory Wait State Control Register
0x3FFD	TPERIOD	[0 . . 15]	Interval Timer Period Register
0x3FFC	TCOUNT	[0 . . 15]	Interval Timer Count Register
0x3FFB	TSCALE	[0 . . 7]	Interval Timer Scale Register
0x3FFA	SPORT0_RX_WORDS1	[0 . . 15]	SPORT0 Multichannel Word 1 Receive
0x3FF9	SPORT0_RX_WORDS0	[0 . . 15]	SPORT0 Multichannel Word 0 Receive
0x3FF8	SPORT0_TX_WORDS1	[0 . . 15]	SPORT0 Multichannel Word 1 Transmit
0x3FF7	SPORT0_TX_WORDS0	[0 . . 15]	SPORT0 Multichannel Word 0 Transmit
0x3FF6	SPORT0_CTRL_REG	[0 . . 15]	SPORT0 Control Register
0x3FF5	SPORT0_SCLKDIV	[0 . . 15]	SPORT0 Clock Divide Register
0x3FF4	SPORT0_RFSDIV	[0 . . 15]	SPORT0 Receive Frame Sync Divide
0x3FF3	SPORT0_AUTOBUF_CTRL	[0 . . 15]	SPORT0 Autobuffer Control Register
0x3FF2	SPORT1_CTRL_REG	[0 . . 15]	SPORT1 Control Register
0x3FF1	SPORT1_SCLKDIV	[0 . . 15]	SPORT1 Clock Divide Register
0x3FF0	SPORT1_RFSDIV	[0 . . 15]	SPORT1 Receive Frame Sync Divide
0x3FEF	SPORT1_AUTOBUF_CTRL	[0 . . 15]	SPORT1 Autobuffer Control Register

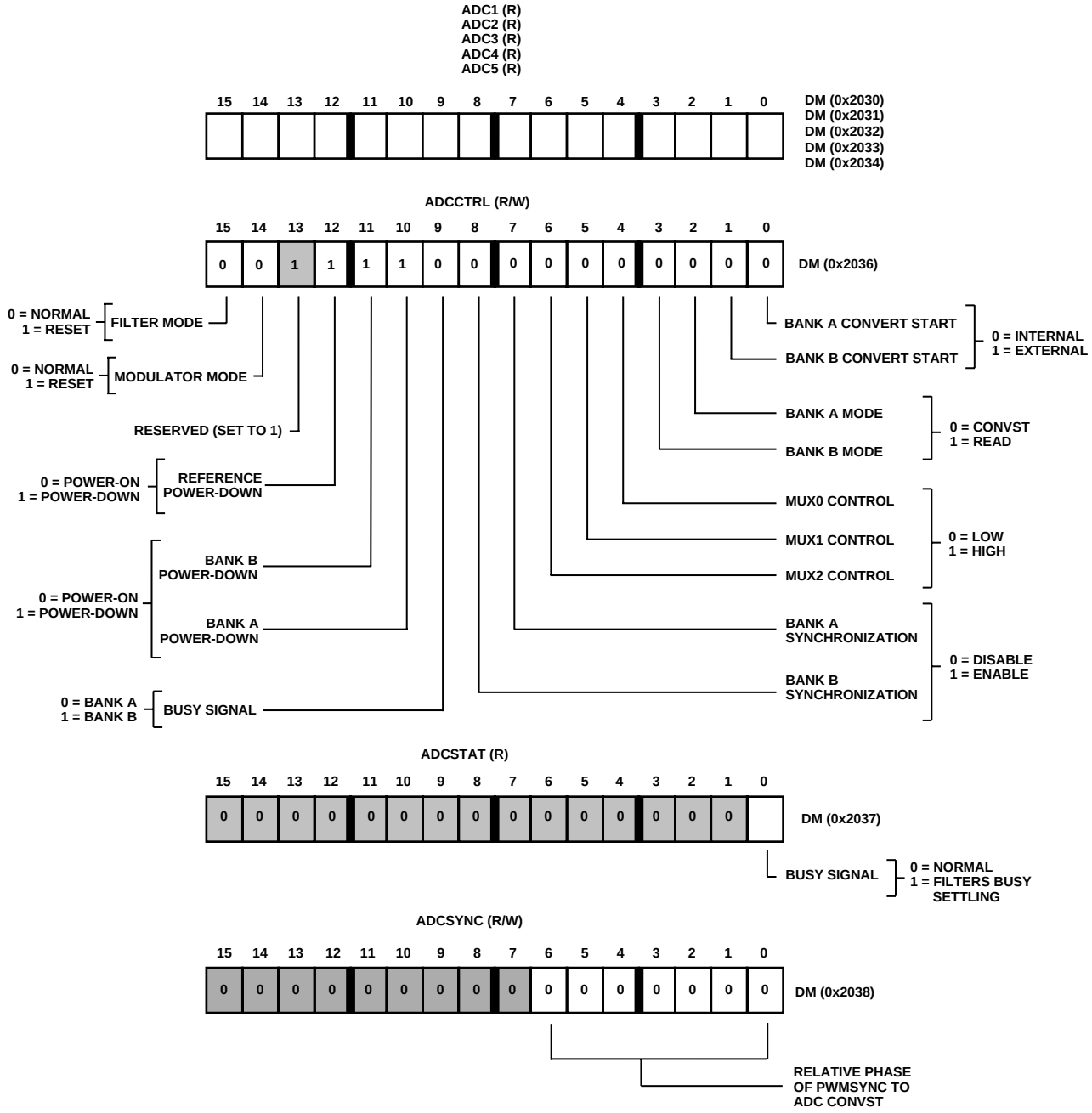


Figure 18. Configuration of ADMC300 Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field – these bits should always be written as shown.

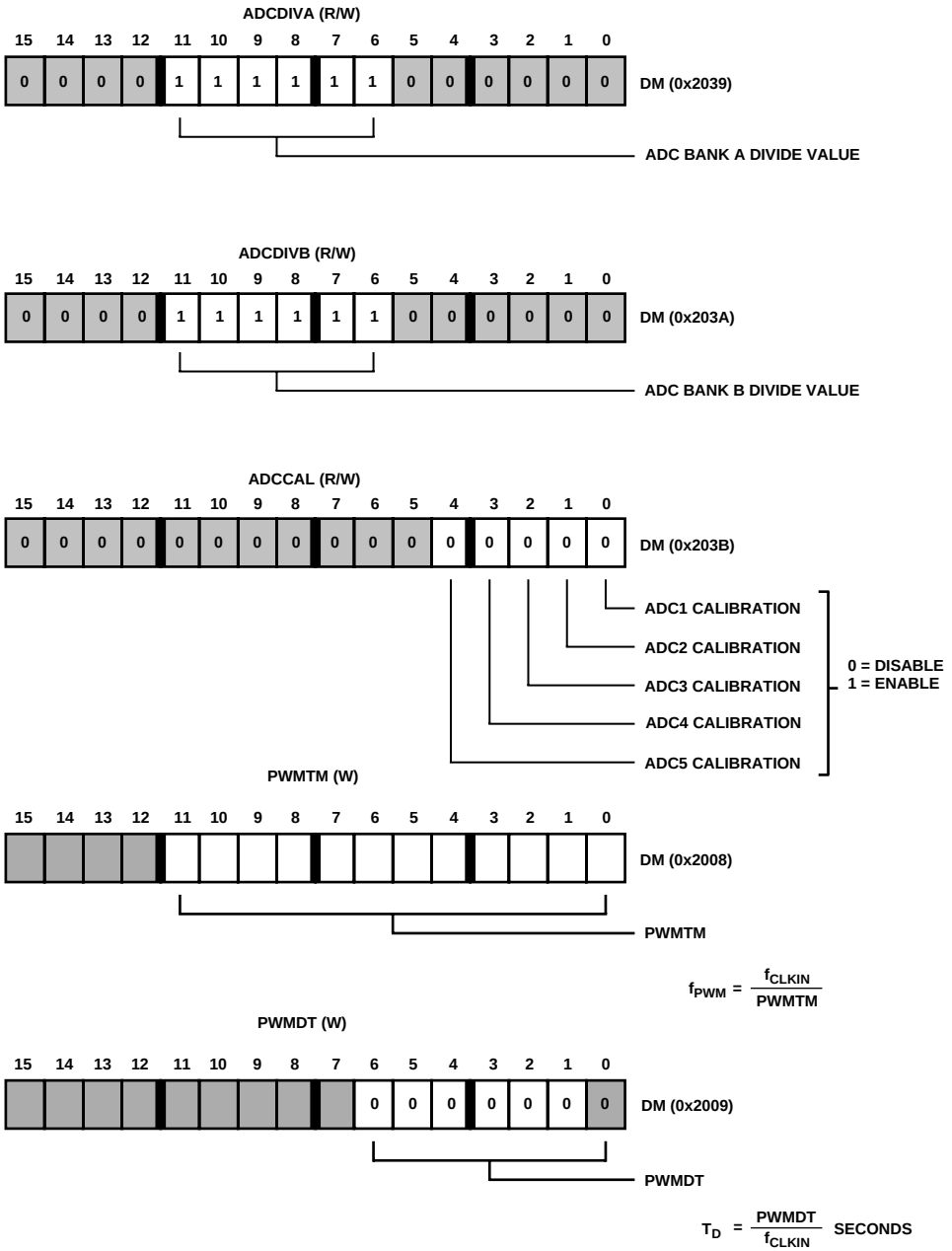


Figure 19. Configuration of ADMC300 Registers

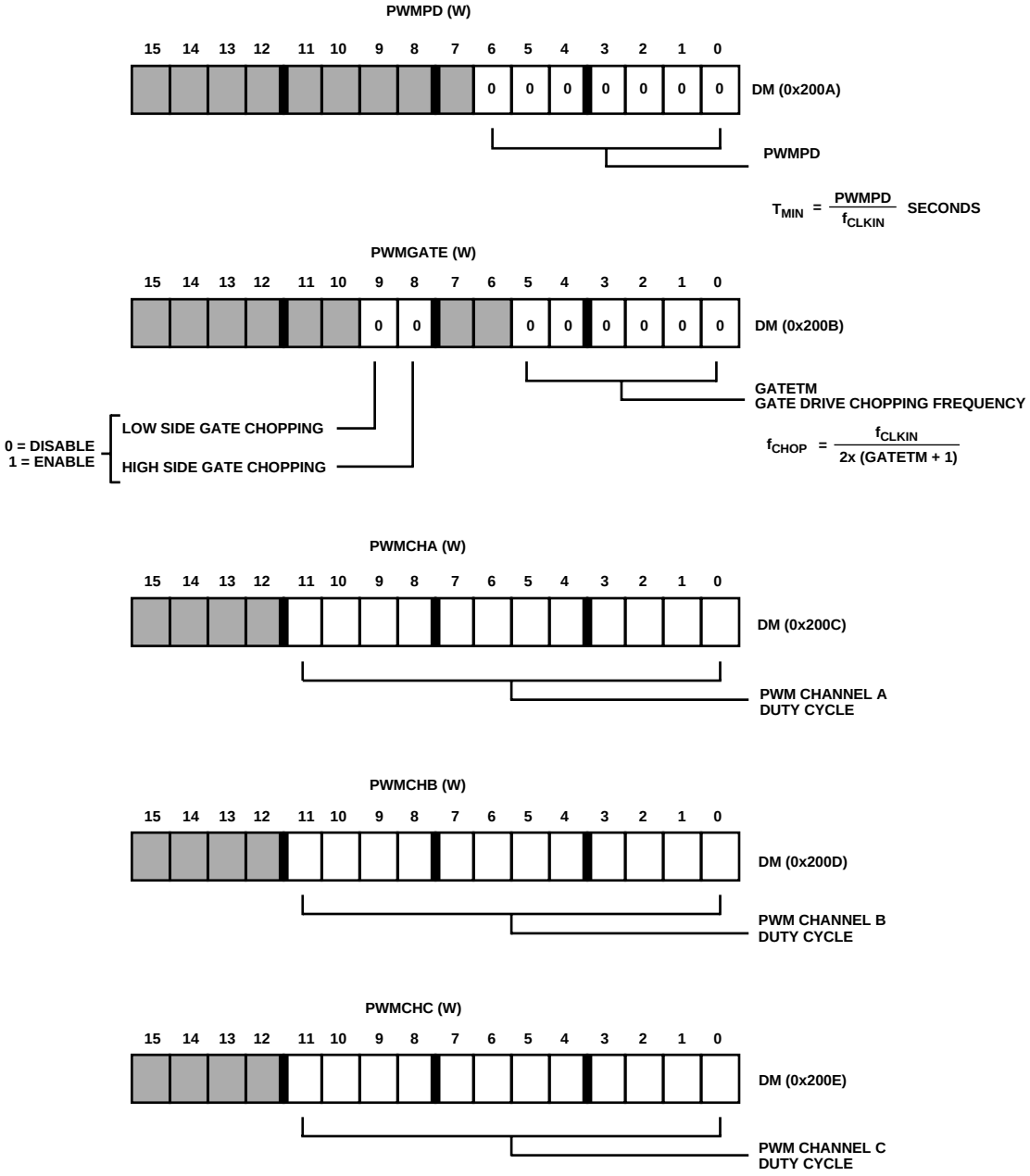


Figure 20. Configuration of ADMC300 Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field – these bits should always be written as shown.

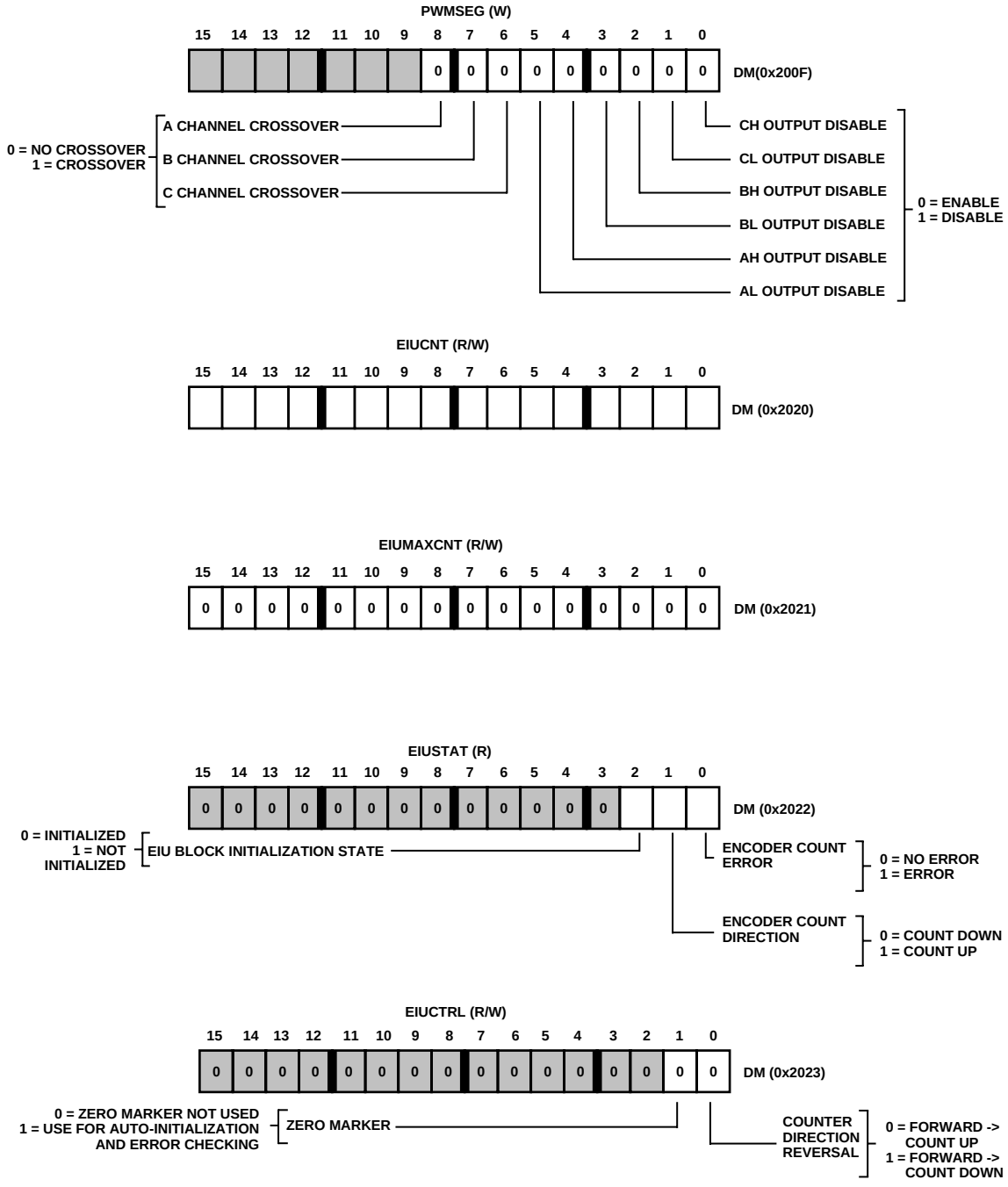


Figure 21. Configuration of ADMC300 Registers

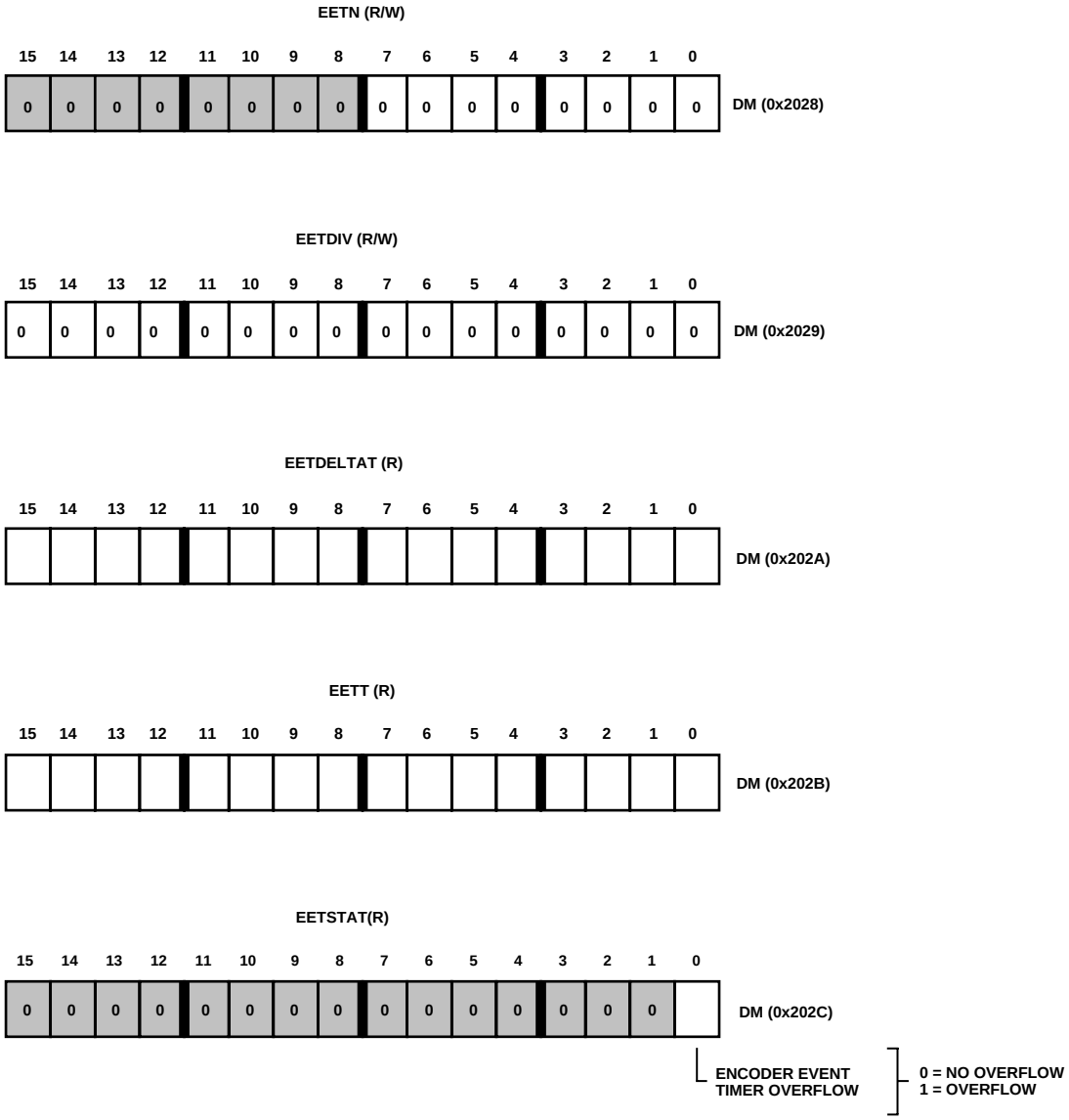


Figure 22. Configuration of ADMC300 Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field – these bits should always be written as shown.

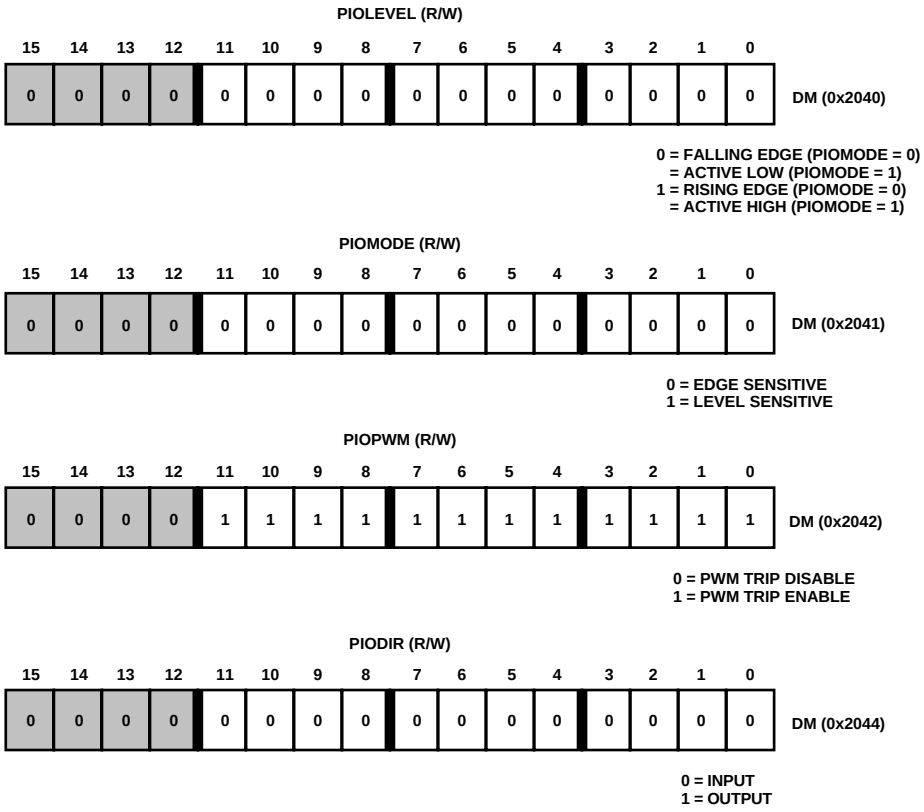


Figure 23. Configuration of ADMC300 Registers

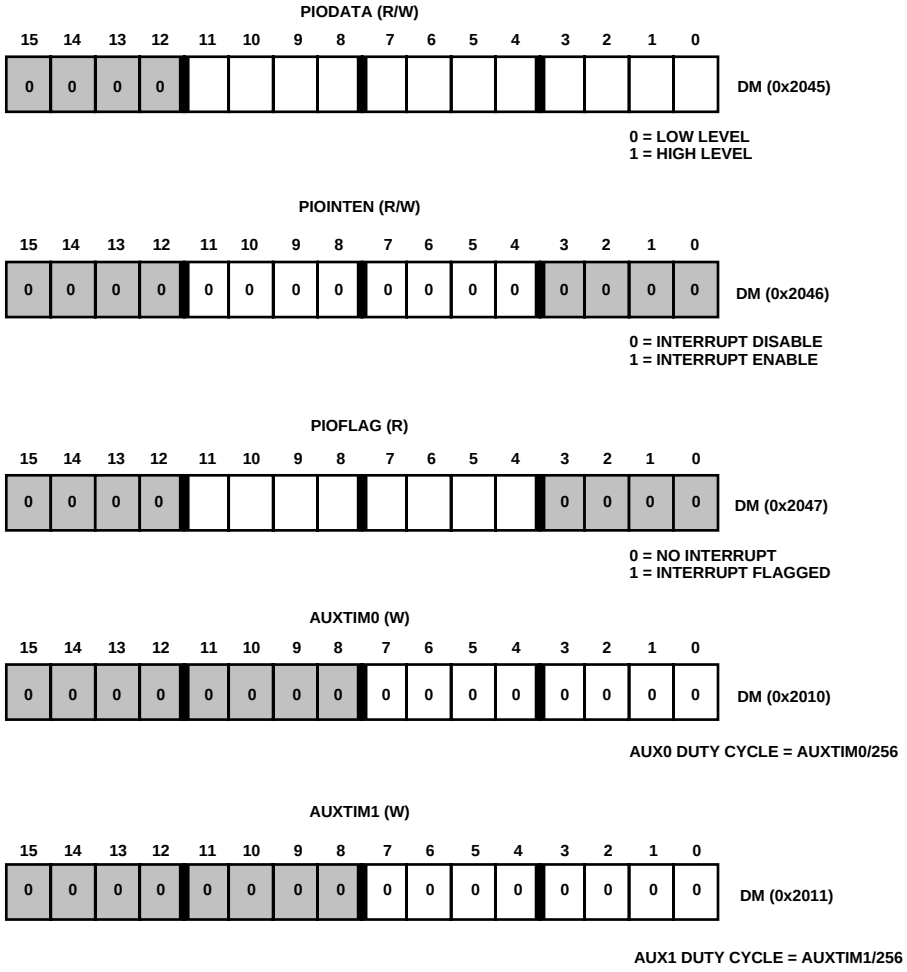


Figure 24. Configuration of ADMC300 Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field – these bits should always be written as shown.

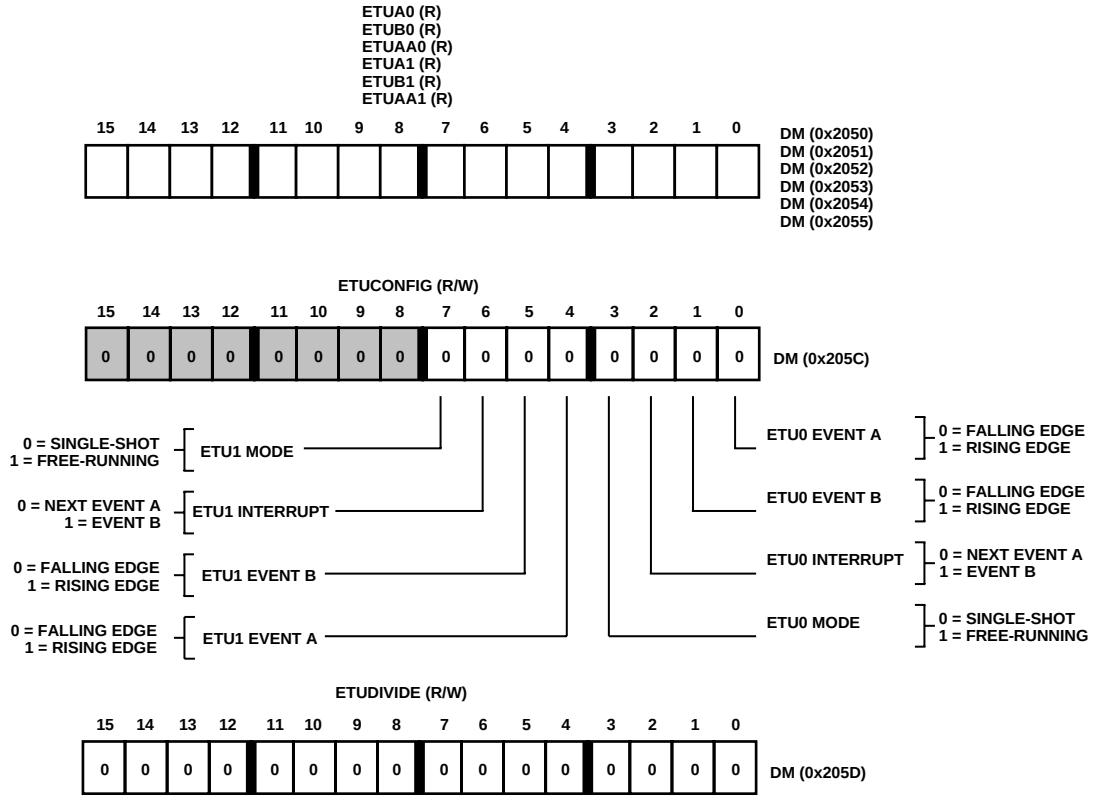


Figure 25. Configuration of ADMC300 Registers

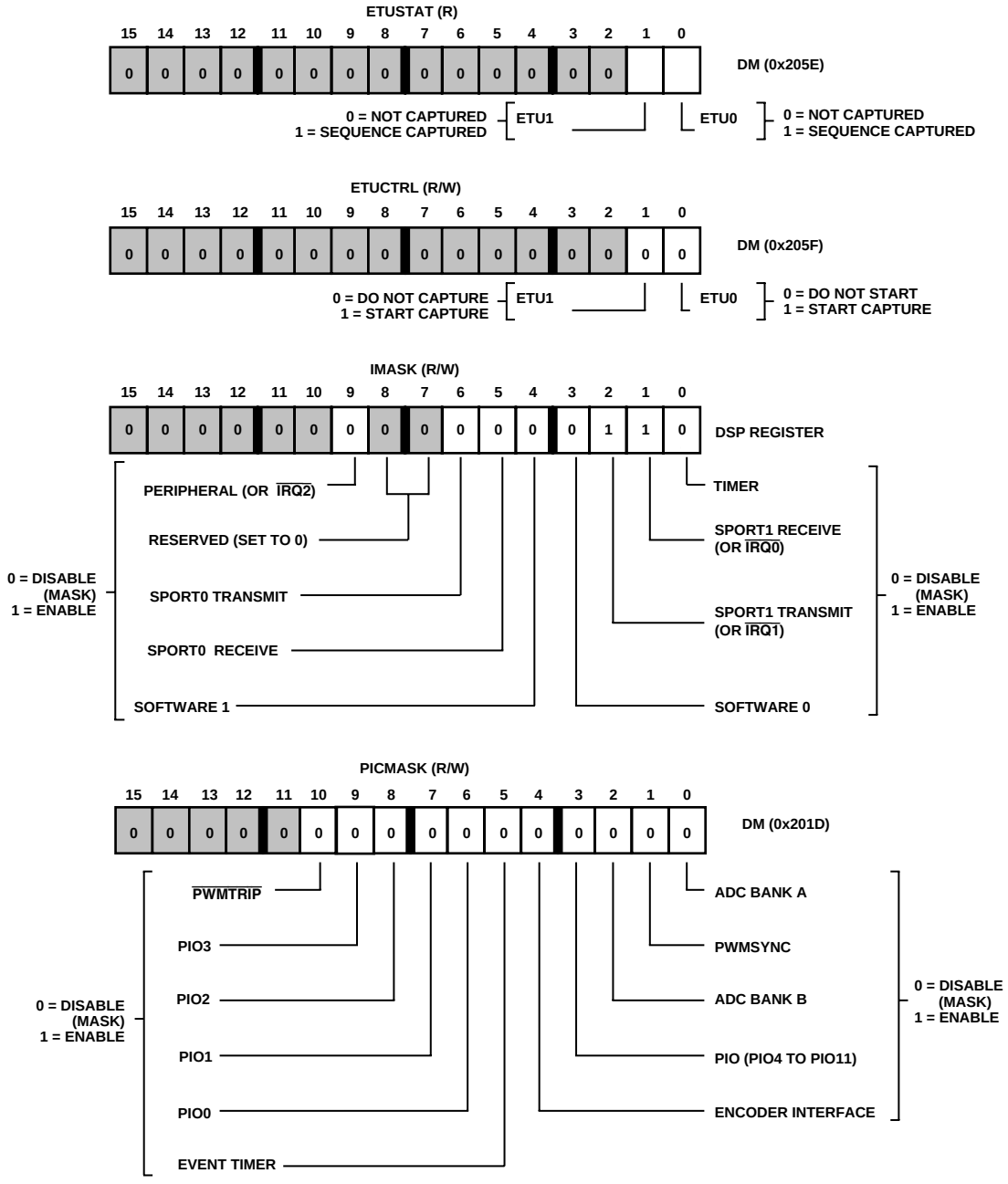


Figure 26. Configuration of ADMC300 Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field – these bits should always be written as shown.

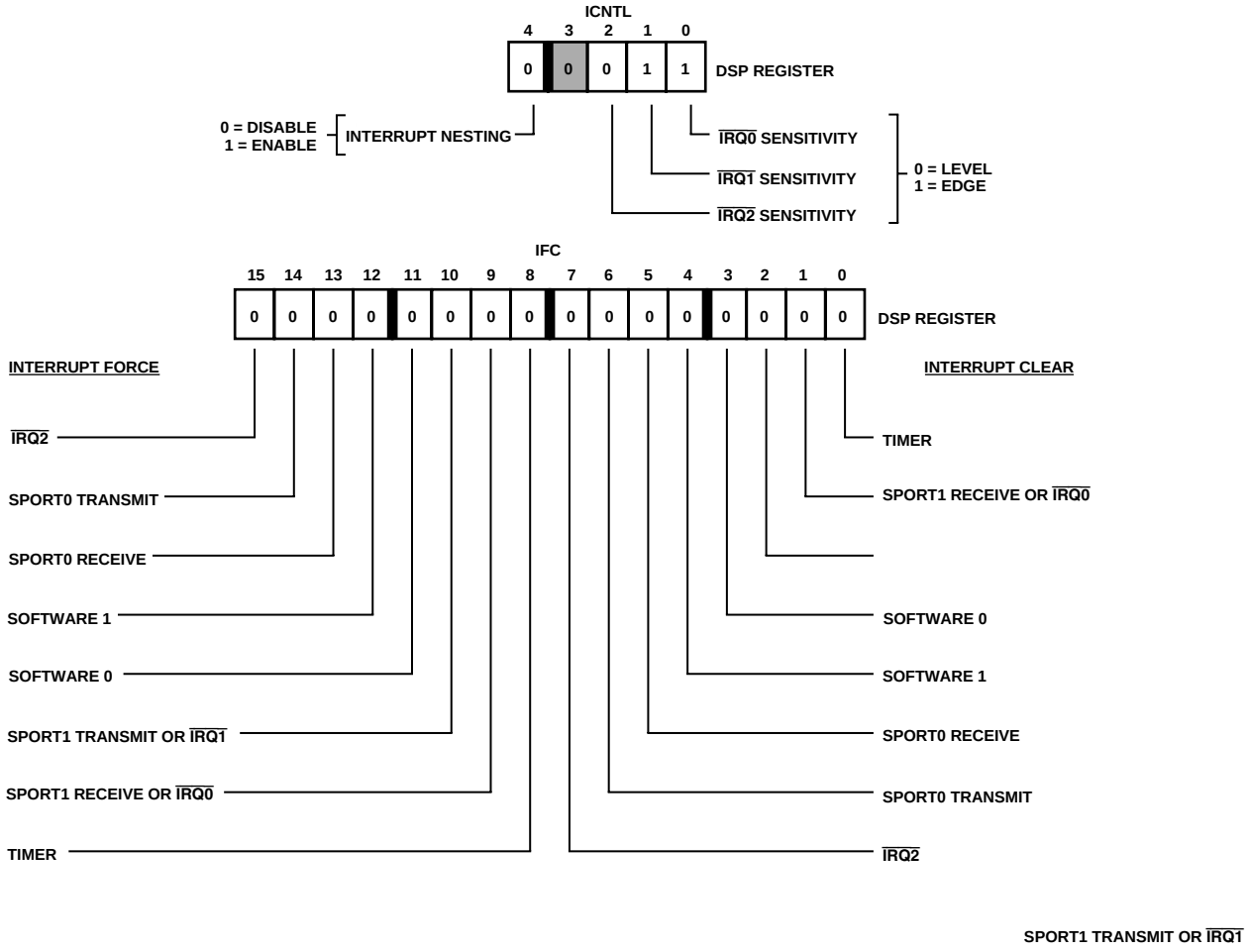


Figure 27. Configuration of ADMC300 Registers

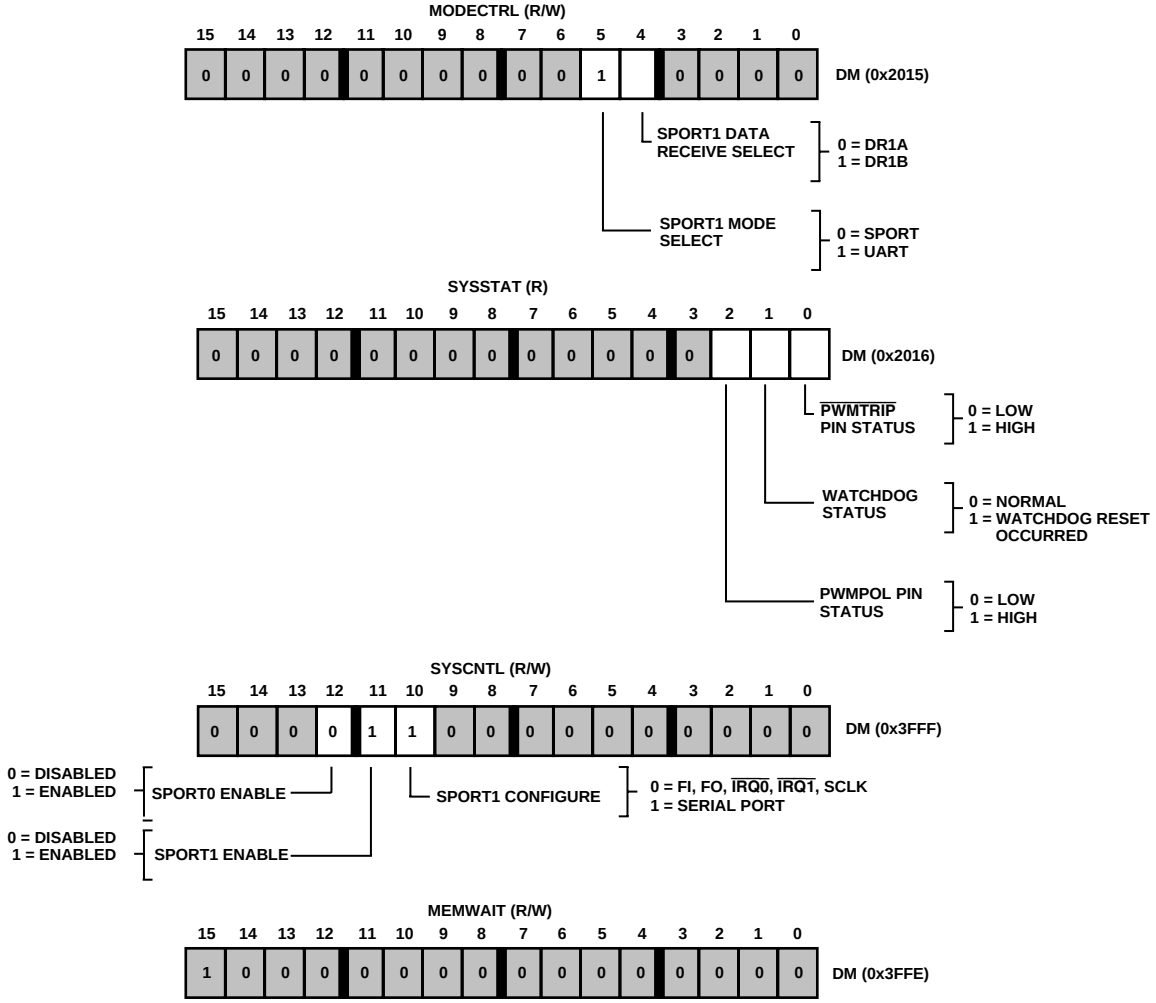


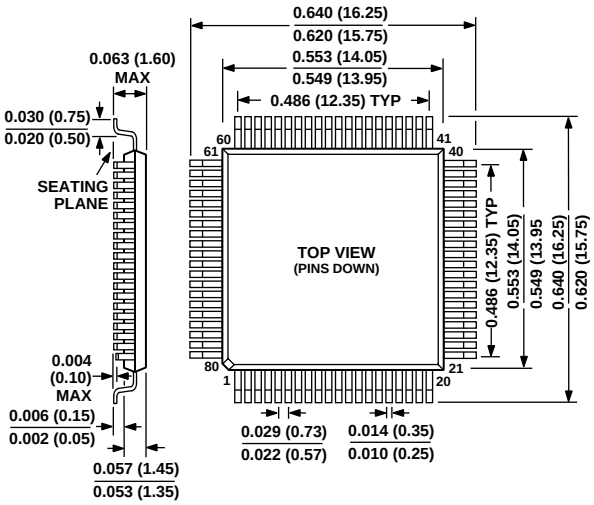
Figure 28. Configuration of ADMC300 Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field – these bits should always be written as shown.

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

80-Lead TQFP
(ST-80)



C3205–8–10/97

PRINTED IN U.S.A.