MICROBOTICS APS-SERIES AUTOPILOT PLATFORM HARDWARE OPERATIONAL DESCRIPTION

APS BASIC I/O OPERATIONS

Overview. The Microbotics APS-series autopilot platform consists of three boards: an SSA-series control board, an APX-series expansion board, and an APM-series modem board. The control board forms the programming and control center of the APS-series autopilot platform. While it is the same base board as used in the Microbotics SSC-series Servo Controller/Safety Switch, its microprocessor programming and hardware resources are dedicated to autopilot functionality. The expansion board can physically mount an optional Microbotics MIDG-series INS/GPS unit, as well as providing additional I/O resources to the autopilot, and allowing for the use of an optional Secure Digital Flash memory card. The modem board physically mounts the RF wireless modem, as well as providing the power converter for the modem. An overall block diagram of the APS-series autopilot platform is shown in Figure 1.

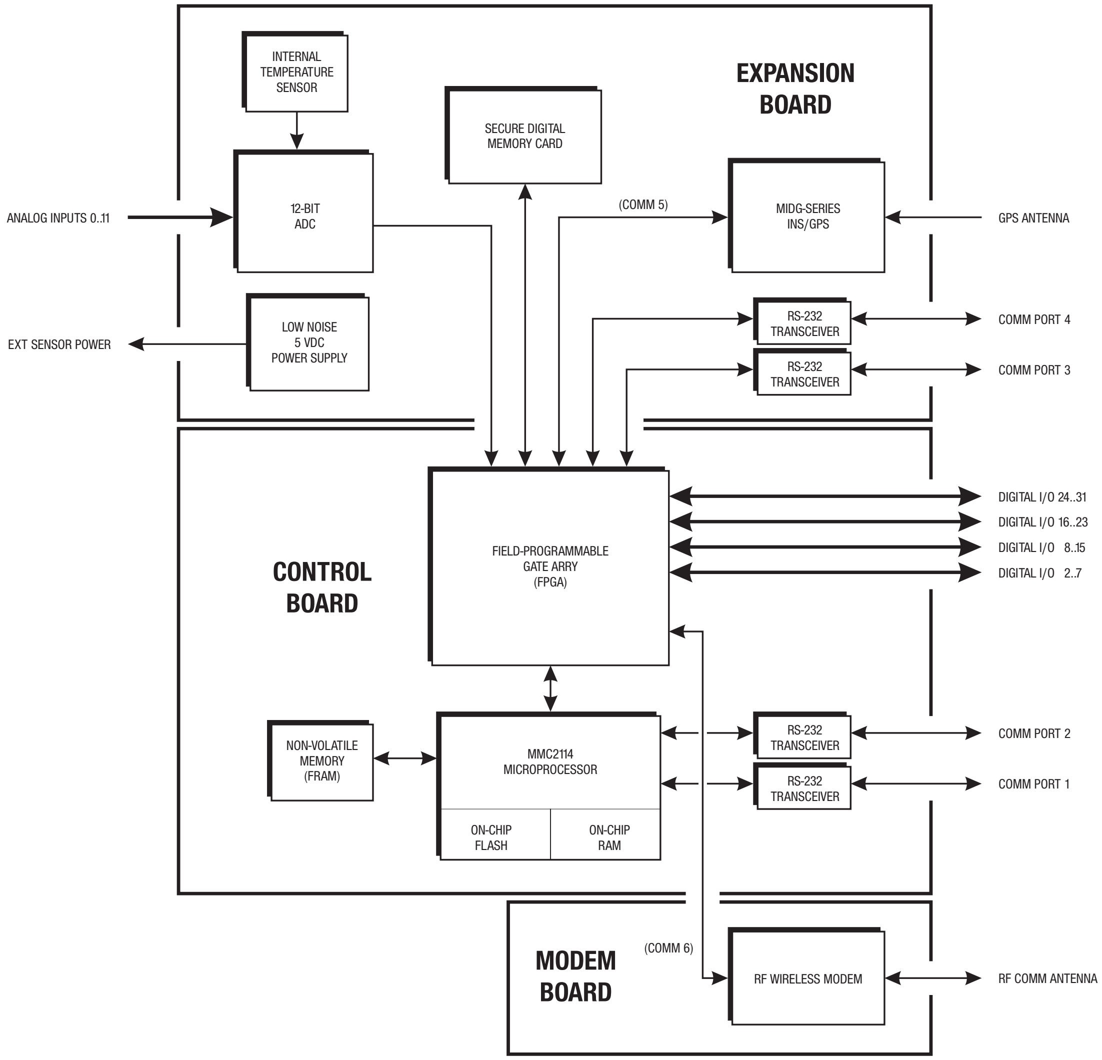

Figure 1. APS Series Autopilot Block Diagram

Control Board. The control board has an M-Core MMC2114 RISC microprocessor for system programming, and an FPGA for the majority of I/O operations. This arrangement off-loads most of the I/O overhead from the microprocessor onto the logic of the FPGA, thus freeing the processor bandwidth for the user operations. As the FPGA programming is stored in the microprocessor Flash memory, the FPGA operations can be changed if needed to suit custom applications. An additional off-chip memory (the FRAM) is included to allow user parameter storage. This device has effectively unlimited erase/re-write cycles, and can be written or erased during system operations.

The 44-pin high-density D-subminiature connector of the control board is used to supply power to the APS-series autopilot platform (nominally 10-32 VDC). This connector also presents the primary I/O of the control board to the user system. These I/O lines consist of 30 digital lines and two asynchronous serial ports. The digital lines are logically divided into four groups of eight lines each. Note, however, that the last group only provides six physical lines to the 44-pin connector – the last two lines are available only within the FPGA itself, providing "buried register" operations for the microprocessor. The two serial ports are the SCI ports of the microprocessor itself, followed by RS-232 level converters.

Expansion Board. The expansion board allows for mounting an optional MIDG-series INS/GPS unit. The MIDG-series unit, complete with its own sensor suite and on-board processing, provides, via a dedicated internal asynchronous serial port, 3-D attitude (pitch, roll, azimuth), 3-D dynamics (angular rates, accelerations, velocities), and location information (GPS coordinates) to the APS-series autopilot platform microprocessor. The serial port is generated by the FPGA, and employs FIFO memories that allow block reading of the MIDG data stream. An external passive or active (+5 VDC) GPS antenna is connected via the SMA connector of the APS autopilot. If a MIDG-series INS/GPS is not used with the APS-series autopilot platform, any external INS and/or GPS system can be connected to the system via the APS-series autopilot platform asynchronous serial ports.

The 26-pin high-density D-subminiature connector of the APX expansion board provides two additional asynchronous serial ports, twelve 0-5VDC analog inputs, and a low noise +5VDC (30 ma maximum) power source for user sensors. The serial ports are generated within the FPGA and are followed by RS-232 level converters. These serial ports employ FIFO memories, thus allowing for block communications operations. The analog inputs are fed to a multiplexed 12-bit analog-to-digital converter which is controlled and read by the FPGA. The +5VDC source is provided as a convenience to the user, and is current-limited and separate from the internal APS supplies.

To provide for data logging or off-line storage, a Secure Digital (SD) Flash memory card may be plugged into the expansion board (access is via a rear panel of the APS autopilot housing). The FPGA handles the I/O operations to and from the SD card. These operations employ FIFO memories, allowing for block read and write operations with the SD card.

Modem Board. The modem board mounts an Aerocomm AC4490 frequency-hopping spread-spectrum RF wireless modem, with a line-of-sight range of approximately 20 miles. An AC4486 single-frequency FSK unit is provided on export models of the APS series autopilot, with a nominal line-of-sight range of approximately 10 miles.. The modem is accessed via a dedicated asynchronous serial port generated by the FPGA. As with the other FPGA serial ports, this serial port employs FIFO memories to allow block communications. A reverse-polarity SMA connector is used to attach the RF modem antenna to the APS series autopilot.

APS DIGITAL I/O OPERATIONS

Basic Pin Operations. The operations of the 32 FPGA digital lines are controlled by FPGA register operations with the microprocessor. Each I/O line has a basic structure as shown in Figure 2. 30 of the I/O pins are directly connected to the the 44-pin connector. Each line can be individually configured as an input or an output. When configured as a normal output, the pin is driven between zero volts (logic '0') and 3.3 volts (logic '1'), presenting a TTL-compatible output (note that these output levels are not 5-Volt CMOS compatible, as 5-Volt CMOS inputs require at least 3.5 volts to guarantee a logic '1'). The output driver of each I/O line can be individually configured as an "open-drain" output. In this mode, a logic '0' actively drives the line to zero volts out, while a logic '1' disconnects the output driver from the I/O. As all FPGA digital lines have a pull-up resistor of approximately 15 Kohms, the open-drain logic '1' will be

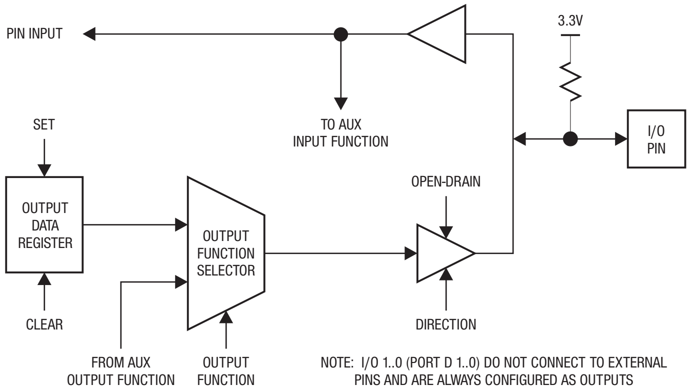

Figure 2. Basic I/O Block.

pulled up to 3.3 volts. If an external pull-up resistor is added (e.g., a 5 Kohm pull-up to 5 volts), the resultant logic ‘1’ will be 5-Volt CMOS compatible. An open-drain output also allows for party-line operations (i.e., more than one output can pull an interconnected line low).

When a line is configured as an input, it reads TTL-compatible input signals (logic ‘0’ < .8 volts, logic ‘1’ > 2.0 volts), and is 5-volt tolerant (i.e., the inputs are compatible with 5-Volt CMOS inputs). The microprocessor always reads the current signal level at the I/O pad (i.e., if the line is configured as an output, the microprocessor reads the value actually being output by the FPGA). Additionally, any signal present at the I/O pin is routed to the Alternate Input Function (if any) for that pin.

FPGA registers control the operation of the I/O lines (Figure 3). Each I/O line (numbered from ‘31’ down to ‘0’) corresponds to the same bit in each register (register bits are numbered ‘31’ for the MSB to ‘0’ for the LSB). Note that I/O channels ‘1’ and ‘0’ do not go to any physical I/O pin of the APS-series autopilot platform. However, these I/O do exist within the FPGA, and may be used by the microprocessor for “buried register” operations. All I/O lines are operated independently of each other. The power-up default for all I/O registers is all bits set to zero.

A digital line becomes an output when the corresponding bit in the Data Direction Register is set to a ‘1’, enabling the output driver for the pin. If an I/O line is set for output, setting the corresponding bit in the Open-Drain Register configures the output driver as open-drain. As I/O lines ‘1’ and ‘0’ do not exit the FPGA and thus have no physical input available, their corresponding Data Direction Register and Open-Drain Register bits have no effect, these lines effectively always being outputs (the microprocessor always reads whichever Data Register value or Alternate Output Function is driving the line).

The microprocessor does not directly write to the Output Data Register, but, rather, writes to a pair of registers to set or clear the current state of the Output Data Register bits. When the microprocessor writes a ‘1’ to a bit in the Set Data Register, the corresponding bit of the Output Data Register is set. Set Data Register bits written with a ‘0’ have no effect. Conversely, when the microprocessor writes a ‘1’ to a bit in the Clear Data Register, the corresponding bit in the Output Data Register is cleared (writing a ‘0’ to a Clear Data Register bit has no effect). This allows the user to control the status of one or more I/O outputs without having to be concerned with the status of the other I/O lines.

DATA DIRECTION REGISTER (WRITE ONLY)

MSB	31	30	29	28	27	26	25	24	23	22	21	21	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB

0 = Corresponding I/O pin is input (power-on default)
1 = Corresponding I/O pin is ouput

OPEN-DRAIN REGISTER (WRITE ONLY)

MSB	31	30	29	28	27	26	25	24	23	22	21	21	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB

0 = Corresponding I/O pin is standard output (power-on default)
1 = Corresponding I/O Pin is open-drain ouput

OUTPUT FUNCTION REGISTER (WRITE ONLY)

MSB	31	30	29	28	27	26	25	24	23	22	21	21	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB

0 = Corresponding I/O output driver is connected to Data Register (power-on default)
1 = Corresponding I/O output driver is connected to Alternate Output Function

DATA REGISTER (READ ONLY FOR INPUTS, WRITTEN BY SET AND CLEAR REGISTER OPERATIONS)

MSB	31	30	29	28	27	26	25	24	23	22	21	21	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB

Power-on default all output register bits zero

SET DATA REGISTER (WRITE ONLY)

MSB	31	30	29	28	27	26	25	24	23	22	21	21	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB

1 = Corresponding Data Register output bit is set
0 = No operation

CLEAR DATA REGISTER (WRITE ONLY)

MSB	31	30	29	28	27	26	25	24	23	22	21	21	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB

1 = Corresponding Data Register output bit is cleared
0 = No operation

Figure 2. FPGA Registers for I/O Operations.

Each output driver is fed by a two-input multiplexer. One input is from the corresponding Digital Output Data Register bit, the other is from the Alternate Output Function for that I/O line. The multiplexer operation is determined by the corresponding bit in the Output Function Register — a ‘0’ selects the Data Register, while a ‘1’ selects the Alternate Output Function.

Alternate Functions. Alternate Input or Output Functions are dependent upon the group of I/O lines. The 32 FPGA I/O lines are divided into four groups of eight lines each: I/O 31..24 forming Port A, I/O 23..16 forming Port B, I/O 15..8 forming Port C, and I/O 7..0 forming Port D. Note that only 30 of the 32 lines are brought out to the 44-pin connector – the lower two bits of Port D (I/O 1..0) are only available within the FPGA, and may be used as “buried registers” by the microprocessor for timer or pulse operations. Alternate Output Functions consist of a 16-output PWM Generator and a 16-output Pulse Generator. Alternate Input Functions consist of a 16-input PWM Resolver and a 16-input Timer Module.

PWM Generator. The PWM Generator creates 16 pulse width modulated outputs (a typical output channel shown in Figure 4). These outputs are only available as Alternate Output Functions at Port A and Port B. Each output channel has its own associated period setpoint and high time setpoint. Both setpoints are 15-bit values representing counts of one microsecond. The period setpoint value determines the repetition rate of the I/O output, allowing a repetition period of up to 32767 microseconds. The high time setpoint value determines the number of one microsecond periods the output will remain high. A value of zero forces the output to remain low, while a value equal to or greater than the value in the corresponding period setpoint forces the output to remain high. This allows a 0-100 percent output pulse width modulation with one microsecond resolution.

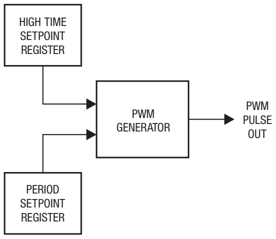

Figure 4. PWM Generator Output.

The PWM Generator outputs are individually controlled by FPGA registers (Figure 5). Sixteen contiguous 16-bit registers are used for the period setpoints, while another sixteen contiguous 16-bit registers are used for the high time setpoints. The lowest addressed register in each case corresponds to the PWM Generator output channel ‘0’, while the highest addressed register corresponds to PWM Generator output channel ‘15’. Writes to Bit 15 of the registers are ignored (the bit value is effectively always set to ‘0’). At power-up, all registers are set to zero. Each register can be accessed independently, allowing for completely independent settings of each channel period and high time.

PERIOD SETPOINT REGISTERS (WRITE ONLY)

HIGHEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 15
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 14
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 13
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 12
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 11
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 10
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 9
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 8
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 7
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 6
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 5
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 4
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 3
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 2
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 1
LOWEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 0

HIGH TIME SETPOINT REGISTERS (WRITE ONLY)

HIGHEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 15
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 14
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 13
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 12
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 11
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 10
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 9
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 8
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 7
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 6
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 5
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 4
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 3
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 2
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 1
LOWEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 0

* Bit 15 (MSB) writes of all setpoint values ignored – set to ‘0’

Figure 5. FPGA Registers for PWM Generator.

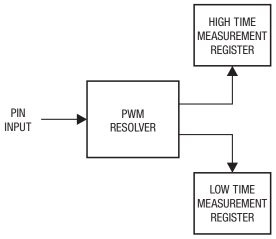

Figure 6. PWM Resolver Input.

PWM Resolver. The PWM Resolver measures the timing of 16 pulse width modulated inputs (a typical input channel shown in Figure 6). These inputs are only received as Alternate Input Functions from Port B and Port C. The Port B pin inputs are channel inputs 15..8 for the PWM Resolver, while Port C pin inputs are channel inputs 7..0 for the PWM Resolver. Each input channel has its own associated high time measurement and low time measurement. Each measurement is a 15-bit value indicating the incoming pulse high time or low time in microseconds, allowing measurements up to 32767 microseconds each. If a high time or low time exceeds the count range of 32767 microseconds, an overrange condition is set for that particular measurement

PWM RESOLVER MEASUREMENT REGISTERS (READ-ONLY)

HIGHEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 15 HIGH TIME	(LOWEST ADDRESS + 15)	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 7 HIGH TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 15 LOW TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 7 LOW TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 14 HIGH TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 6 HIGH TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 14 LOW TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 6 LOW TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 13 HIGH TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 5 HIGH TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 13 LOW TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 5 LOW TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 12 HIGH TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 4 HIGH TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 12 LOW TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 4 LOW TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 11 HIGH TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 3 HIGH TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 11 LOW TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 3 LOW TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 10 HIGH TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 2 HIGH TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 10 LOW TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 2 LOW TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 9 HIGH TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 1 HIGH TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 9 LOW TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 1 LOW TIME
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 8 HIGH TIME		*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 0 HIGH TIME
(LOWEST ADDRESS + 16))	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 8 LOW TIME	LOWEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 0 LOW TIME

* A '1' in Bit 15 (MSB) of value indicates that value has overranged

Figure 7. FPGA Registers for PWM Resolver.

The PWM Resolver measurements are individually presented to the microprocessor by FPGA registers (Figure 7). Sixteen contiguous register pairs are used to read the values for each PWM Resolve channel. The lowest addressed register pair corresponds to measurements at PWM Resolver channel '0', while the highest addressed register pair corresponds to channel '15'. The higher addressed register of the pair is the high time measurement, while the lower addressed register is the low time measurement. Each value is 15 bits, with the Bit 15 (MSB) of each register indicating an overrange condition when it is '1' (at overrange, the value in the lower 15 bits of the register is indeterminant). Each register can be accessed independently, allowing for completely independent reading of each channel high or low time.

Pulse Generator. The Pulse Generator creates 16 pulse outputs (Figure 8). These outputs are only available as Alternate Output Functions at Port C and Port D. Each output has an associated time base selector, a period setpoint, a

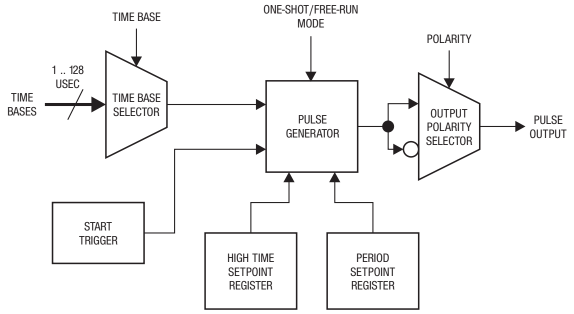

Figure 8. Pulse Generator Output.

high time setpoint, a mode selection for one-shot or free-run operations, an output polarity selector, and a start trigger circuit.

The time base selector is used to set a clocking time base for the channel at 1, 2, 4, 8, 16, 32, 64, or 128 microseconds. The period setpoint is a 15-bit value that determines the number of time base units of a free-running output (up to 32767 counts for up to a 4.194 second repetition rate). The high time setpoint is a 15-bit value that determines the number of time base units the pulse remains high (pulses from 1 microsecond to 4.194 seconds can be generated). If the high time setpoint is zero, the output of the Pulse Generator remains low. As with the PWM Generator, if a channel is set for free-running and its high time setpoint value is equal to or greater than its period setpoint value, that Pulse Generator output will remain high. When the channel is in one-shot mode, the period setpoint has no effect on the channel operation.

Each Pulse Generator channel allows its output signal to be inverted. When the polarity selection determines whether the channel output is inverted. When the signal is inverted, its non-active state in one-shot mode is high, while the pulse is low for the time set by the high time setpoint. When the channel is in free-run mode, the high time setpoint determines the number of time base periods the repetitive signal is low (the period is not affected by the polarity setting).

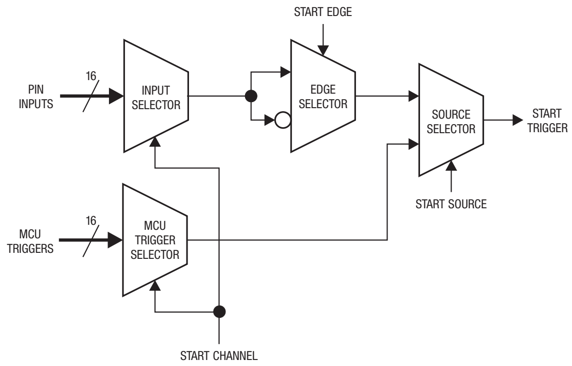

Figure 9. Pulse Generator Start Trigger Circuit.

The start trigger for each channel is active only when the channel is in one-shot mode. This circuit (Figure 9) is essentially a collection of selectors. The start trigger is chosen by the start source from an pin input selector or from an MCU triggers selector. The pin input selector uses the start channel value to choose one of sixteen input pins as the trigger when this selector is used for the start pulse. These pin inputs are a one-for-one selection of the signals currently present on the Port C pins (forming start channel selections ‘15’ to ‘8’) and the Port D pins (forming start channel selections ‘7’ to ‘0’). The start edge selector determines whether the rising egde or the falling edge of the chosen input signal will be the actual start trigger. The MCU triggers selector uses the start channel value to choose from one of sixteen MCU Trigger Register bits (start channel ‘15’ chooses the MSB — see below). Polarity is not relevant when using MCU triggers, so the start edge value is ignored if the the start source selects MCU triggers. If a start trigger re-occurs before the high time of the pulse has completed, the pulse will restart its high time at the new trigger occurrence.

The Pulse Generator outputs are individually controlled by FPGA registers (Figure 10). Three sets of 16 registers each are used, with each lowest addressed register within the set corresponding to channel ‘0’ and the highest addressed register within the set corresponding to channel ‘15’: the Period Setpoint Registers, the High Time Setpoint Registers, and the Channel Control Registers. The Period Setpoint Registers and High Time Setpoint Registers are 16-bit registers. Writes to Bit 15 of these registers are ignored (the bit value is effectively set to ‘0’). When the output is set for free-running mode, the Period Setpoint Register and High Time Setpoint Register for that output work in the same manner as those of the PWM Generator. The only difference is these registers set the period and high time ot the output in the number of time base units selected for that channel, rather than in units of microseconds as in the PWM Generator. When the output is set for one-shot mode, the value in the Period Setpoint Register for that output is ignored, and the value in the High Time Setpoint Register sets the number of time base units the pulse will remain high after a start trigger occurs.

The Pulse Control Registers are each 16 bits wide, with Bits 4..0 shared with the Timer Module. Each Pulse Control Register is fully independent, and the registers are read/write with all bits set to zero at power-up. Bit 15 sets the Pulse Mode for the channel: a ‘0’ sets Free-Run Mode, while a ‘1’ set One-Shot Mode. If the the One-Shot Mode is set, Bits 14..11 set the Start Channel selection (0..15) used in the trigger circuit. If the One-Shot Mode is set, Bit 10 sets the Start Source selection: a ‘0’ sets pin inputs, while a ‘1’ sets the MCU triggers. If the One-Shot Mode is set with pin inputs selected, Bit 9 sets which edge is used for the start trigger: a ‘0’ selects the falling egde, while a ‘1’ selects the rising edge (if MCU triggers are selected, this bit is ignored). Bits 14..9 are ignored if the Free-Run mode is selected. Bit 8 selects the Output Polarity of the channel: a ‘0’ outputs a normal pulse, while a ‘1’ inverts the output (the output idles high, and the value of the High Time Setpoint Register sets the low time of the output pulse). Bits 7..5 sets the channel Time Base: time base values from 1 microsecond to 128 microseconds may be selected. Note that, since the power-up mode forces all Pulse Control Register values to zero, the Pulse Generator will behave like the PWM Generator until the Pulse Control Register values are changed.

PERIOD SETPOINT REGISTERS (WRITE ONLY)

HIGHEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 15
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 14
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 13
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 12
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 11
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 10
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 9
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 8
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 7
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 6
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 5
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 4
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 3
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 2
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 1
LOWEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	HANNEL 0

HIGH TIME SETPOINT REGISTERS (WRITE ONLY)

HIGHEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 15
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 14
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 13
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 12
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 11
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 10
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 9
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 8
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 7
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 6
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 5
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 4
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 3
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 2
	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 1
LOWEST ADDRESS	*	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	HANNEL 0

* Bit 15 (MSB) of all setpoint values ignored – set to ‘0’

PULSE CONTROL REGISTERS (READ/WRITE)

HIGHEST ADDRESS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 15
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 14
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 13
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 12
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 11
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 10
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 9
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 8
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 7
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 6
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 5
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 4
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 3
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 2
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CHANNEL 1
LOWEST ADDRESS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	HANNEL 0

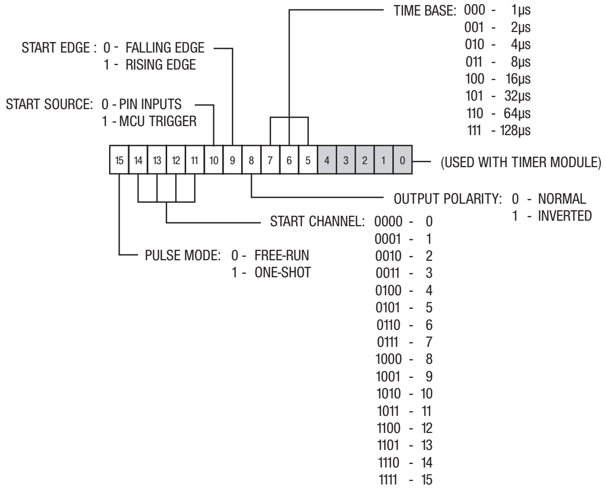

Figure 10. FPGA Registers for Pulse Generator.

Timer Module. The Timer Module measures the timing of 16 inputs (Figure 11). These inputs are only received from Port C and Port D. Each input has an associated time base selector, a count register, a pulse accumulator/timer mode selector, an input polarity selector for pulse accumulation, and start and stop trigger circuits.

The time base selector is used to set a clocking time base for the channel at 1, 2, 4, 8, 16, 32, 64, or 128 microseconds. The number of time base periods counted by the Timer Module channel is stored in the count register.

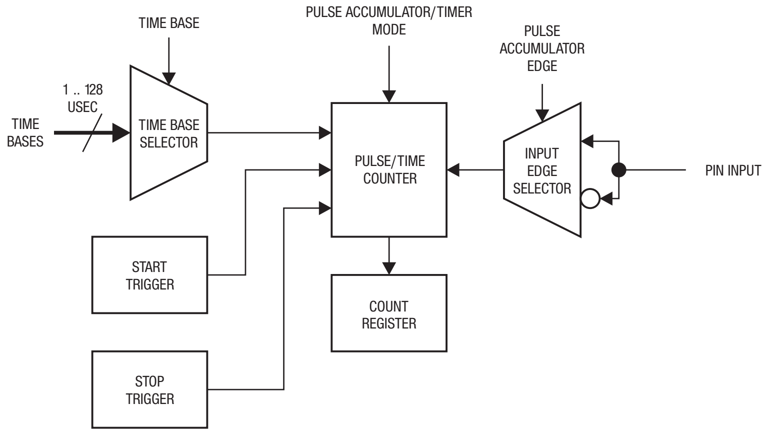

Figure 11. Timer Module.

The Timer Module has two primary modes of operation: Timer Mode and Pulse Accumulator. In the Timer Mode, the input channel will measure the number of time base periods between a start trigger and a stop trigger (this is similar to normal "stop-watch" operations). As the Count Register holds a 14-bit value, the each timer channel can measure a period between one microsecond and 2.948 seconds. In the Pulse Accumulator Mode, the channel will count the number of pulses occurring at the corresponding input pad that occur between a start trigger and a stop trigger (this is essentially a gated event counter). Port C pin inputs are assigned to pulse accumulator inputs 15..8, and Port D pin inputs are assigned to pulse accumulator inputs 7..0. In the Pulse Accumulator Mode, the time base is ignored, and counting of either the rising edges or the falling edges of the pin input signal can be selected. The Count Register holds a 14-bit value, up to 16383 input pulse edges can be counted.

Whether a channel is in Timer Mode or Pulse Accumulator Mode, all counting is controlled by the start and stop trigger circuits. The start trigger will reset the counter, and initiate the counting (time base periods in Timer Mode or pin input pulse edges in Pulse Accumulator Mode). The stop trigger will transfer the current count into the Count Register and stop the counter. If more than one stop trigger occurs after a start trigger, the extra stop triggers are ignored. If the start trigger is the same as the stop trigger (e.g., as in RPM measurements), the current count is savee in the Count Register, and a new count is started.

The start and stop trigger circuits operate in the same manner (Figure 12). A three-input Source selector determines if the trigger comes from a pin input, from one of the Pulse Generator outputs, or from the MCU Trigger Register bits. When pin inputs is selected as a trigger source, any of 16 inputs can be chosen. These inputs the Port C pins inputs for input channel selections 15..8, and Port D pin inputs for input channel selections 7..0. When Pulse Generor outputs is selected as the trigger source, any of the Pulse Generator outputs can be chosen. If the trigger source is a pin input or a Pulse Generator output, either the rising or falling edge can be chosen as the trigger event. When MCU Trigger Register bits are selected (channel '15' chooses the MSB — see below), edge polarity is not relevant, and the start edge value is ignored.

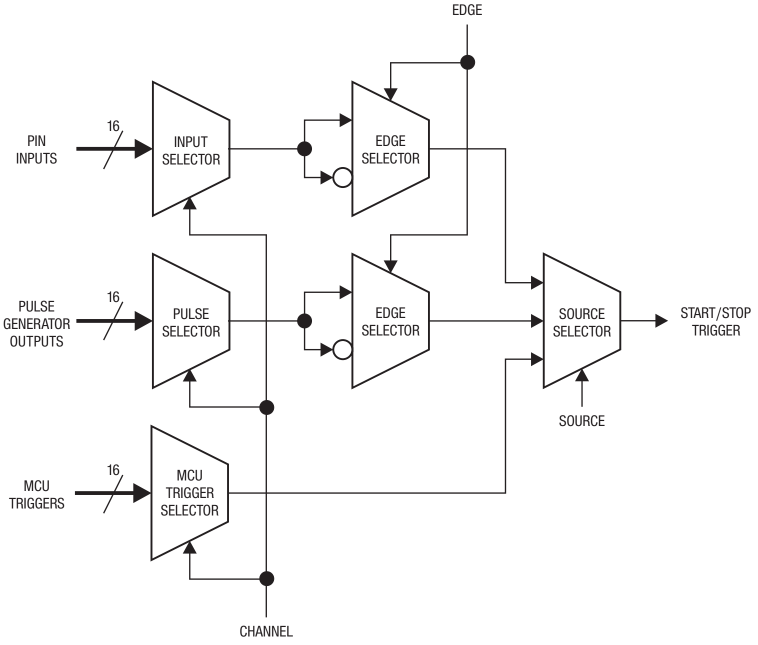

Figure 12. Timer Module Start/Stop Trigger Circuit.

It is important to note that the pin inputs or the Pulse Generator output selected for a particular Timer Module channel start or stop trigger do not have to be the same value as the channel itself. This allows several measurement options: a channel can measure the high time, low time, or period of signals on its own channel or on another channel; a channel can measure the time between signals on two separate channels; a channel can measure time between a pulse driven from one channel to the input signal received at another channel (e.g., as used in range-finder applications); or pulse accumulation can be controlled by a periodic signal from a Pulse Generator output (either the output assigned as an Alternate Output Function for that particular channel, or as an an Alternate Output Function assigned to another channel). Pulse Generator outputs are available to the Timer Module trigger circuit even when the outputs have be selected to drive their assigned output pins. Thus a Pulse Generator output can be used for timing or pulse accumulation operations while its associated pin input can be used for other functions (e.g., a general purpose input, a PWM measurement, a timer trigger, or a pulse accumulation).

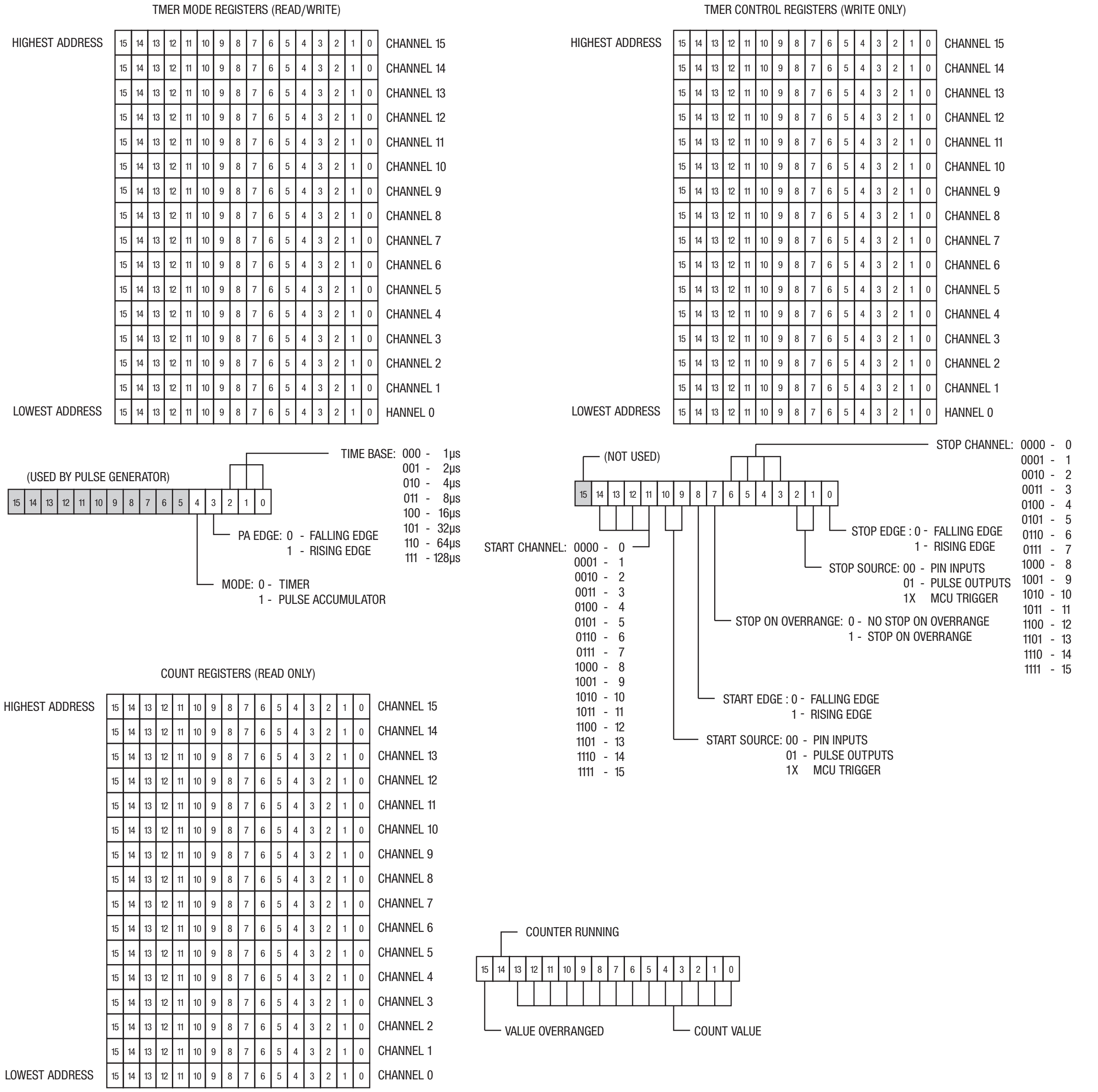

Figure 13. FPGA Registers for Timer Module.

The Timer Module channels are individually controlled by FPGA registers (Figure 13). Three sets of 16 registers each are used, with the lowest addressed register within the set corresponding to channel ‘0’ and the highest addressed register within the set corresponding to channel ‘15’: the Count Registers, the Timer Mode Registers, and the Timer Control Registers. The Count Registers are 16-bit registers that hold the most current count for that channel (whether it a count of time base periods or pulse edges) in the lower 14 bits. If Bit 15 is a ‘1’, then the value is overranged (a count greater than 16383 was encountered) and the actual value is not valid. Bit 14 is a ‘1’ whenever the counter is running — i.e., whenever a start trigger has been received and stop trigger has not yet been received.

The Timer Mode Registers are 16 bits wide, with the upper 11 bits shared with the Pulse Generator. A Bit 4 value of ‘0’ sets Timer Mode (the channel counts time base periods), while a Bit 4 value of ‘1’ sets Pulse Accumulator Mode (the channel counts its corresponding pin input edges). If the Pulse Accumulator is set (Bit 4 is ‘1’), Bit 3 selects the pin input edges to be counted: a ‘0’ counts falling edges, while a ‘1’ counts rising edges. Bit 3 is ignored when the channel is set for Timer Mode. Bits 2..0 set the time base for the channel when it is set for Timer Mode. Bits 2..0 are ignored when the channel is set for Pulse Accumulator Mode.

The Timer Control Registers are 16 bits wide (Bit 15 is not used) and are used to set the start trigger and the stop trigger for each channel. Bits 14..8 of the Timer Control Registers select the start trigger conditions. Bits 14..11 select the Start Channel Number. The value is used for all three start trigger source selectors (pin inputs, Pulse Generator outputs, and MCU Trigger Register bits). Bits 10..9 select the actual trigger source to use for the start trigger. Bit 8 selects the active edge for the start trigger if the trigger source is from an pin input or Pulse Generator output. Bit 8 is ignored if the MCU Trigger Register is selected as the start trigger source.

Bits 7..0 of the Timer Control Registers select the stop trigger conditions. Bit 7 is used to allow an overranged count to become a stop trigger. When Bit 7 is set, the counter will stop immediately upon overrange rather than wait for the selected stop trigger condition (this mode is usually used to prevent “lock-up” in the event the stop condition has been lost such as a stopped shaft during RPM measurements). Bits 6..3 select the Stop Channel Number. The value is used for all three stop trigger source selectors (pin inputs, Pulse Generator outputs, and MCU Trigger Register bits). Bits 2..1 select the actual trigger source to use for the stop trigger. Bit 0 selects the active edge for the stop trigger if the trigger source is from an pin input or Pulse Generator output. Bit 0 is ignored if the MCU Trigger Register is selected as the stop trigger source.

MCU Trigger Register. The 16-bit MCU Trigger Register is used to allow the microprocessor to start Pulse Generator one-shot operations, or to start and/or stop Timer Module operations. The associated MCU Trigger bit is activated by writing a ‘1’ to the corresponding bit of the register. Writing zeros to any bit have no effect. Note that any or all MCU triggers can be written simultaneously.

Alternate Functions Microprocessor Interrupts. The PWM Generator and PWM Resolver are effectively free-running functions and require no microprocessor intervention during normal operation, and thus do not generate interrupts. However, the Pulse Generator and the Timer Module use interrupts and polled flags to inform the microprocessor of functions completions.

Whenever a Pulse Generator output completes (i.e., when the time set by the corresponding High Time Register has finished), the flag for that channel is set. If a Pulse Generator channel is set for Free-Run Mode, the flag can be used to create a periodic interval timer (PIT) for the microprocessor. If the One-Shot Mode is set, the flag informs the microprocessor the pulse period has been completed. All Pulse Generator channels use a 16-bit Pulse Flag Register (channel ‘15’ flag appearing in the MSB) and share the same microprocessor interrupt. To enable a specific Pulse Generator channel interrupt, a 16-bit Pulse Interrupt Enable Register is used with a one-for-one bit correspondence to the Pulse Flag Register (only the channels with a ‘1’ in the corresponding Pulse Interrupt Enable Register bit will interrupt). To clear a channel flag, a ‘1’ is written to the corresponding bit of the Pulse Flag Register (bits written with a ‘0’ have no effect), thus allowing individual interrupts to be processed without affecting the other channels.

Whenever a Timer Module channel receives a stop trigger (whether a normal stop trigger or an overrange condition with the stop on overrange mode set), the flag for that channel is set. All Timer Module channels use a 16-bit Timer Flag Register (channel ‘15’ flag appearing in the MSB) and share the same microprocessor interrupt. To enable a specific Timer Module channel interrupt, a 16-bit Timer Interrupt Enable Register is used with a one-for-one bit correspondence to the Timer Flag Register (only the channels with a ‘1’ in the corresponding Timer Interrupt Enable Register bit will interrupt). To clear a channel flag, a ‘1’ is written to the corresponding bit of the Timer Flag Register (bits written with a ‘0’ have no effect), thus allowing individual interrupts to be processed without affecting the other channels.

I/O Lines Assignments. Figure 14 shows the connections of the Alternate Input and Output Functions for the I/O lines.

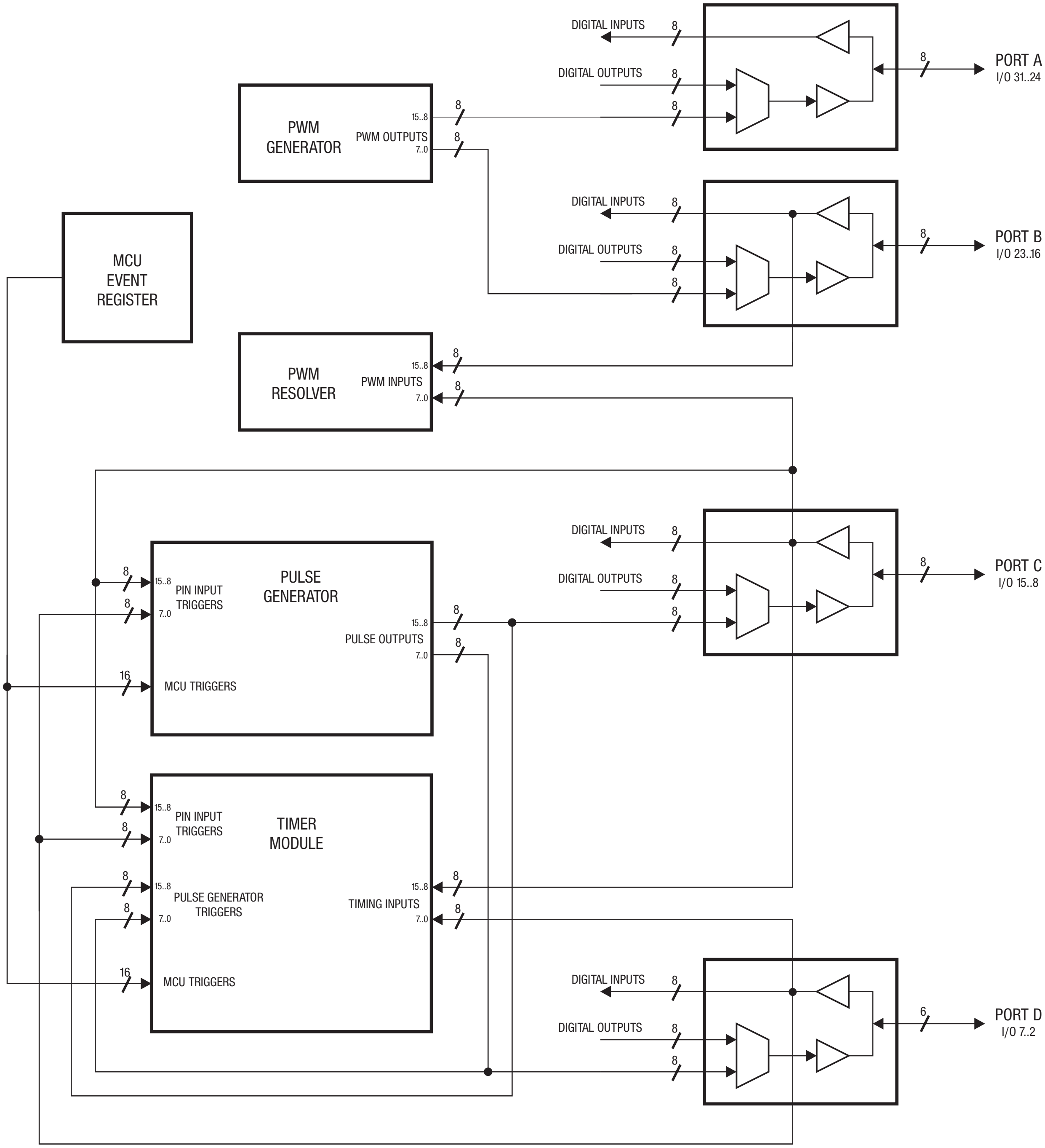

Figure 14. I/O Port Alternate Functions Assignments.

The Alternate Output Functions of Port A are driven by the high channels of the PWM Generator. There are no Alternate Input Functions assigned to Port A.

The Alternate Output Functions of Port B are driven by the low channels of the PWM Generator. The inputs of Port B drive the upper channels of the PWM Resolver.

The Alternate Output Functions of Port C are driven by the high channels of the Pulse Generator. The inputs of Port C drive the low channels of the PWM Resolver, drive the high channels of the Timer Module, and provide the high I/O input pad signals for the start and/or stop events of the Pulse Generator and the Timer Module.

The Alternate Output Functions of Port D are driven by the low channels of the Pulse Generator. The inputs of Port D drive the low channels of the Timer Module, and provide the low I/O input pad signals for the start and/or stop events of the Pulse Generator and the Timer Module. Note that only the upper six lines of Port D are available at the 44-pin connector. Whatever output function is assigned to the lower two lines of Port D (discrete digital outputs or Pulse Generator outputs) are used as the I/O input pad signals for these lines, effectively allowing “buried register” operations (e.g., they can start or stop Pulse Generator or Timing Module operations).

APS SERIAL PORT OPERATIONS

Two types of asynchronous serial ports are available on APS Series Autopilots. All serial ports presented at the I/O connectors use RS-232 voltage level transceivers. The supplied software drivers handle the low-level operation of these ports, creating buffer structures and I/O streams for use with the user code.

COMM1 and COMM2, available at the 44-pin high-density D-subminiature connector, are the two SCI ports of the MMC2114 microprocessor. The transmitters and receivers of these serial ports are double-buffered, and operate on a byte-by-byte basis.

COMM3 and COMM4, available at the 26-pin high-density D-subminiature connector, are created by the FPGA. The transmitters use a 128-byte FIFO, while the receiver uses a variable-length FIFO up to 64 bytes. The FIFOs allow for either byte-by-byte or block operations.

COMM5 and COMM6 are internally dedicated serial ports created by the FPGA. COMM5 is used for communications with an optional internally mounted MIDG INS/GPS, while COMM6 is used for RF modem communications. These ports operate identically to COMM3 and COMM4.

MIDG INS/GPS OPERATIONS

An optional Microbotics MIDG-series INS/GPS can be mounted on the expansion card. The dedicated internal COMM5 is used for serial communications with the internally mounted MIDG. The MIDG has its own dedicated sensor suite and on-board processing to present a full attitude, dynamics, and position solution to the APS processor. This removes the flight dynamics and navigation processing requirements from the APS microprocessor, freeing operational bandwidth for the user programming. The MIDG uses GPS aiding in its navigation solution. An active (+5 volts, 25 ma max) antenna or a passive antenna may be used, and is attached to the SMA connector of the APS autopilot. Refer to the MIDG documentation for detailed descriptions of the messaging used.

APS ADC OPERATIONS

The expansion board allows for 12 channels of analog voltage measurements. These inputs are at the 26-pin high-density D-subminiature connector, and have input ranges of 0-5 VDC. The analog inputs have no anti-aliasing filters, and are routed to a multiplexed 12-bit ADC. Additionally, an LM50 temperature sensor is used to provide a dedicated internal temperature measurement to the ADC. The FPGA handles all ADC operations, and effects conversions on all channels at approximately 1800 times a second. The results of these conversions are placed in a software structure that can be read by the microprocessor. The microprocessor always reads the last converted value for each of the input channels.

A low-noise dedicated +5-volt power supply is available at the 26-pin high-density D-subminiature connector. This power supply is provided for user sensors, and has a current rating of 30 milliamps maximum.

APS SD CARD OPERATIONS

The expansion board has a connector for a standard Secure Digital (SD) flash card. All SD card operations are handled by the FPGA, which employs a FIFO for read/write operations. The supplied software drivers provide SD card operations as MSDOS-compatible file operations (FAT12/FAT16), making data transferred to and from the SD card compatible with DOS and Windows programs.

RF MODEM

An Aerocomm AC4490 RF wireless modem (AC4486 for export versions) mounted on the modem board is used for ground station communications. The modem uses the dedicated internal COMM6 for serial communications, with the supplied software drivers handling the low-level operations and creating buffer structures and I/O streams for use with the user code.