**Freescale Semiconductor**
Application Note

AN1933
Rev. 1, 11/2005

# Synchronizing the On-Chip Analog-to-Digital Converter on 56F80x Devices

Synchronization of the Internal Analog-to- Digital Converter on 56F80x Devices to Internal and External Events.

*Daniel Malik*

## Contents

1. Introduction .....1

2. Advantages and Key Features .....1

3. Purpose of ADC Synchronization .....4
   3.1 Sources of Synchronization Events.....4
   3.2 ADC Synchronization to PWM in Motor Control.....4

4. Description of Hardware Features .....6

5. Examples of Synchronization Modes .....8
   5.1 Periodic Sampling.....8
   5.2 Sampling Synchronous to PWM.....9
   5.3 Synchronization of ADC to External Events.....13
   5.4 Using More than One Timer for More Advanced Synchronization.....13

6. Conclusion .....15

7. References .....15

## 1. Introduction

This Application Note describes the synchronization of the 56F80x on-chip Analog-to-Digital Converter (ADC) module to internal and external events. The following sections show examples of the typical ADC operation, settings needed for achieving synchronization and other settings needed to operate the ADC module.

## 2. Advantages and Key Features

The Freescale 56F80x Digital Signal Controller (DSC) family is well suited for digital motor control, combining the DSP's calculation capability with the MCU's controller features on a single chip. These devices offer many dedicated peripherals, such as Pulse Width Modulation (PWM) unit, Analog-to-Digital Converters (ADC), Timers, communication peripherals (SCI, SPI, CAN), on-board Flash and RAM. Generally, all family members are well-suited to motor control.

A typical member of the family, the 56F805, provides the following peripheral blocks:

- Two Pulse Width Modulator modules (PWMA & PWMB), each with six PWM outputs, three Current Sense inputs, and four Fault inputs, fault tolerant design with deadtime insertion, supporting both Center- and Edge- aligned modes
- Twelve bit, Analog-to-Digital Convertors (ADCs), which support two simultaneous conversions with 8-pin multiplexed inputs; ADC can be synchronized by PWM Module
- Two Quadrature Decoders, each with four inputs, or two additional Quad Timers
- Two General Purpose Quad Timers totaling six pins; Timer C with two pins and Timer D with four pins
- CAN 2.0 A/B Module with 2-pin port used to transmit and receive
- Two Serial Communication Interfaces, each with two pins (or four additional GPIO lines)
- Serial Peripheral Interface (SPI), with configurable 4-pin port (or four additional GPIO lines)
- Fourteen dedicated General Purpose I/O (GPIO) pins, 18 multiplexed GPIO pins
- Computer Operating Properly (COP) Watchdog timer
- Two dedicated external interrupt pins
- External reset input pin for hardware reset
- External reset output pin for system reset
- JTAG/On-Chip Emulation (OnCE$^{TM}$) for unobtrusive, processor speed-independent debugging
- Software-programmable, phase lock loop-based frequency synthesizer for the core clock .
-

**Table 2-1. Memory Configuration**

| | 56F801 | 56F803 | 56F805 | 56F807 |
|---|---|---|---|---|
| Program Flash | 8188 x 16-bit | 32252 x 16-bit | 32252 x 16-bit | 61436 x 16-bit |
| Data Flash | 2K x 16-bit | 4K x 16-bit | 4K x 16-bit | 8K x 16-bit |
| Program RAM | 1K x 16-bit | 512 x 16-bit | 512 x 16-bit | 2K x 16-bit |
| Data RAM | 1K x 16-bit | 2K x 16-bit | 2K x 16-bit | 4K x 16-bit |
| Boot Flash | 2K x 16-bit | 2K x 16-bit | 2K x 16-bit | 2K x 16-bit |

The flexible PWM module, the 16-bit Quadrature Timer modules and the ADC module are especially useful in motor control. The flexible hardware configuration enhances the efficiency of motor control.

The PWM module has the following features:

- Three complementary PWM signal pairs, or six independent PWM signals
- Features of complementary channel operation
- Deadtime insertion
- Separate top and bottom pulse width correction via current status inputs or software
- Separate top and bottom polarity control

- Edge-aligned or center-aligned PWM signals
- 15 bits of resolution
- Half-cycle reload capability
- Integral reload rates from one to 16
- Individual software-controlled PWM output
- Programmable fault protection
- Polarity control
- 20mA current sink capability on PWM pins
- Write-protectable registers

The Quadrature Timer modules have the following features:

- Each timer module consists of four 16-bit counters/timers
- Count up/down
- Counters are cascadable
- Programmable count modulo
- Max count rate equals peripheral clock/2 when counting external events
- Max count rate equals peripheral clock when using internal clocks
- Count once or repeatedly
- Counters are preloadable
- Counters can share available input pins
- Each counter has a separate prescaler
- Each counter has capture and compare capability

The ADC module has the following features:

- 12-bit resolution
- Sampling rate up to 1.66 million samples per second[1]
- Maximum ADC clock frequency is 5MHz with 200ns period
- Single conversion time of 8.5 ADC clock cycles (8.5 x 200 ns = 1.7µs)
- Additional conversion time of 6 ADC clock cycles (6 x 200 ns = 1.2µs)
- Eight conversions in 26.5 ADC clock cycles (26.5 x 200 ns = 5.3µs) using simultaneous mode
- Simultaneous or sequential sampling
- Internal multiplexer to select two of eight inputs
- Ability to sequentially scan and store up to eight measurements
- Ability to simultaneously sample and hold two inputs
- Optional interrupts at end of scan, if an out-of-range limit is exceeded or at zero crossing
- Optional sample correction by subtracting a pre-programmed offset value
- Signed or unsigned result
- Single ended or differential inputs

1. Once in loop mode, the time between each conversion is 6 ADC clock cycles (1.2 µs). Using simultaneous conversion, two samples can be obtained in 1.2 µs. Samples per second is calculated according to 1.2 µs per two samples or 1666666 samples per second.

# 3. Purpose of ADC Synchronization

Many applications require the analog-to-digital conversion to be started by a particular (usually repetitive) event. An interrupt service routine can be used to handle the conversion process in cases if an interrupt can be associated with an event. However this approach consumes processing power of the device and the exact time when the analog input channel was sampled is not known accurately, because the interrupt latency time is uncertain. On the other hand, if the ADC conversion process can be started by an independent hardware mechanism, then the sampling time is known with a high degree of accuracy and device core processing time is saved as well.

The 56F80x family of motor control devices offers very flexible means of triggering the ADC conversion process in hardware, independent of the device core operation. It is capable of starting a conversion in response to an internal or an external hardware event, where the latency time is not only deterministic to a high degree of accuracy, but can also be deliberately chosen and changed on the fly.

## 3.1 Sources of Synchronization Events

In motor control, the timebase of the whole application is usually derived from a PWM and it is desirable to synchronize ADC operations at the beginning of a PWM period. In many other applications, the timebase is created by a timer instead of a PWM and it is then necessary to synchronize the ADC to an operation of a timer peripheral. In some applications, the timebase is created by circuits and processes, which are external to the device. In such a case, it is necessary to synchronize the ADC to an external signal.

Members of the 56F80x family are capable of performing all three modes of operation and the synchronization signal for the ADC can be generated by:

- PWM (internal sync signal)
- Timer (internal event)
- External signal

## 3.2 ADC Synchronization to PWM in Motor Control

The circuitry in **Figure 3-1** is an example of the most commonly used power stage topology in motor control applications. When controlling transistors by a PWM signal, the inductive load voltage and the current waveforms can look as shown in **Figure 3-2**. The motor torque is a function of the electrical current, flowing through individual motor phases and the measurement of the phase current is therefore an important feature for almost all motor control applications. As seen in **Figure 3-2**, the phase current has a triangular waveform, synchronous to the PWM operation. By synchronizing the phase current measurement with the PWM operation, we can eliminate the need for filtration of the result. The result is a higher sample rate, which may be needed for fast enough filtering. Otherwise it would be necessary to obtain a reasonably stable and precise value. The figure also shows that the winding current equals the average current twice over the PWM period - centers of $T_{ON}$ and $T_{OFF}$ periods are the best moments to the sample winding currents in order to get precise values.

Another good reason for synchronization of the measurement to a PWM operation in motion control systems is the elimination of noise. Most of the electromagnetic noise, which is generated internally by motor control applications, is originated in switching processes of semiconductor power switches (usually MOSFETs or IGBTs). By synchronizing analog measurements to the PWM we can make sure that none of the measurements will be performed during switching of the power semiconductors and the measurements will therefore be unaffected by the generated noise.

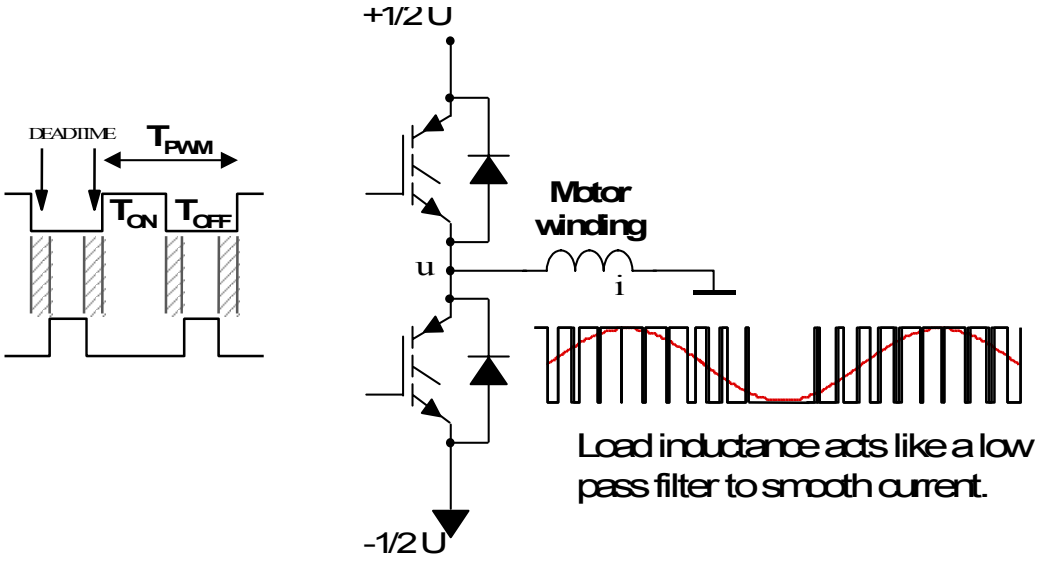


**Figure 3-1. Half Bridge Power Stage**

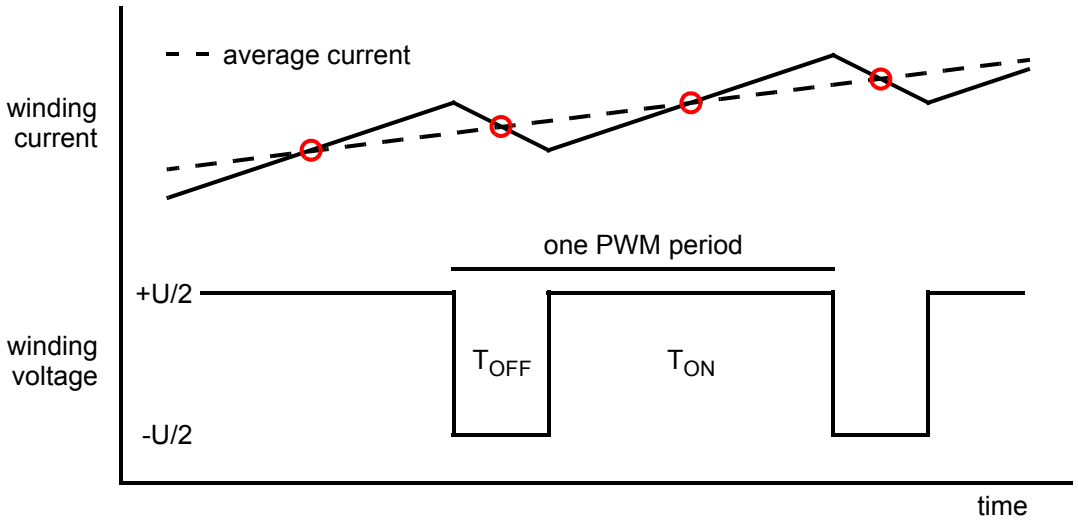


**Figure 3-2. Applied PWM**

# 4. Description of Hardware Features

Figure 4-1 shows the part of the device involved in ADC synchronization (please note that not all blocks are present on all family members).

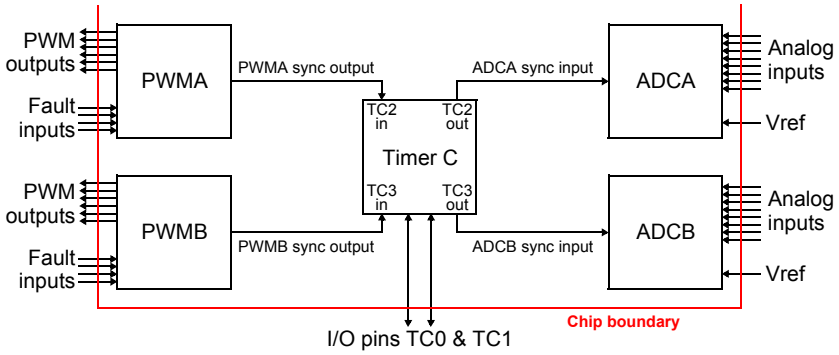


**Figure 4-1. On-chip Synchronization Hardware Blocks**

The heart of the ADC on-chip synchronization hardware features is the Quad Timer block C (see [1.] for details). This timer block has only two I/O pins and the two remaining I/O channels are used for synchronization signals. The internal structure of the Quad Timer C is shown in Figure 4-2. Note how the PWM and the ADC blocks are connected to the timer structure - timer C2 and C3 inputs are provided with synchronization signals from the PWMA and PWMB blocks, and the outputs are used to trigger the conversion of ADCA and ADCB. The internal Timer input multiplexer architecture allows routing of signals from PWMA and PWMB to any Timer channel.

A pulse occurs at PWMA and PWMB synchronization outputs for each reload of the respective PWM, regardless of the state of the LDOK bit. When half-cycle reloads are enabled, the pulse can occur on the half cycle. Details about PWM synchronization outputs are described in Figure 5-3 and Figure 5-4.

The Timer module processes the input signals from the PWM modules or external pins and triggers ADC conversions. The usual function of the Timer is to insert a user-defined delay or to generate a periodic trigger. The function provided by the Timer module is application-dependant and the most commonly used setups are explained in the next section.

The ADCA and ADCB blocks will start the conversion process when a rising edge occurs at their respective synchronization inputs (providing the SYNC bit in ADCR1 register is set and no conversion is in progress).

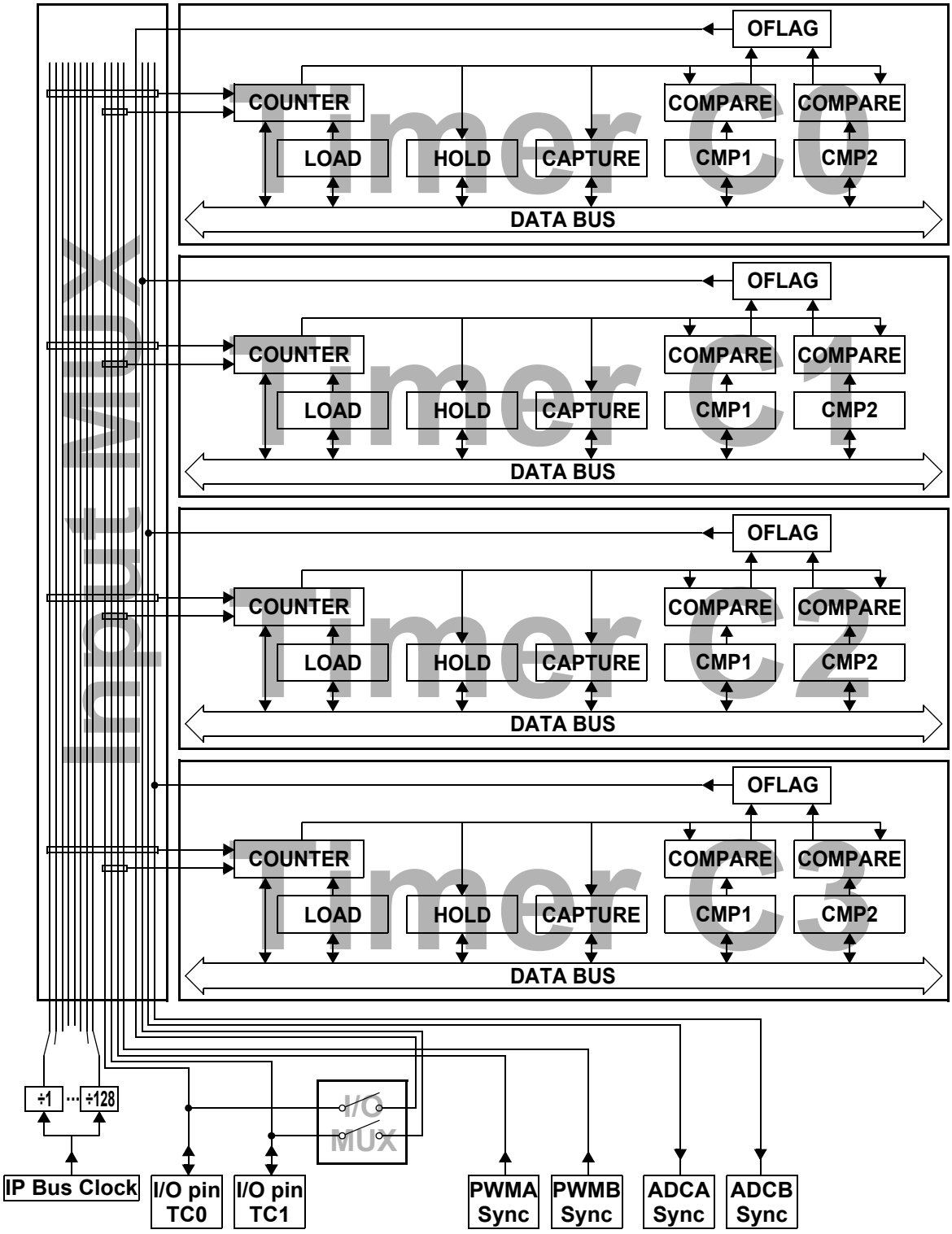


**Figure 4-2. Internal Structure of Quad Timer C**

# 5. Examples of Synchronization Modes

The synchronization modes shown in this chapter are only a very small subset of all possible synchronization setups. Great varieties of synchronization modes are possible due to a flexible hardware architecture, therefore the user can select a setup best suitable for his particular application.

The most commonly used setups, which are described here, are:

- Periodic sampling
- Synchronization to PWM
- Synchronization to external event
- Advanced synchronization utilizing two timer channels

The setup of the ADC module is the same for all examples shown; they only differ in the setup of the timer and the PWM peripherals. The ADC module is initialized to “triggered sequential” mode, the SYNC bit in ADCR1 is set to enable hardware triggering and the end-of-scan interrupt is enabled to acquire data from the result registers in the interrupt service routine.

## 5.1 Periodic Sampling

This section describes the triggering of the ADC in order to digitize the analog input signal at an exact user-defined frequency. In this mode timer C2 is running freely and triggers the ADC conversion by toggling its output on every compare event.

The setup of the timer is shown in **Figure 5-1** - the timer is counting down, starting from the value stored in the Load Register (TMRC2_LOAD), down to the value stored in the Compare Register 2 (TMRC2_CMP2). In an usual setup, the Compare Register 2 will be loaded with 0 and the Load Register will hold one half of the sample period length.

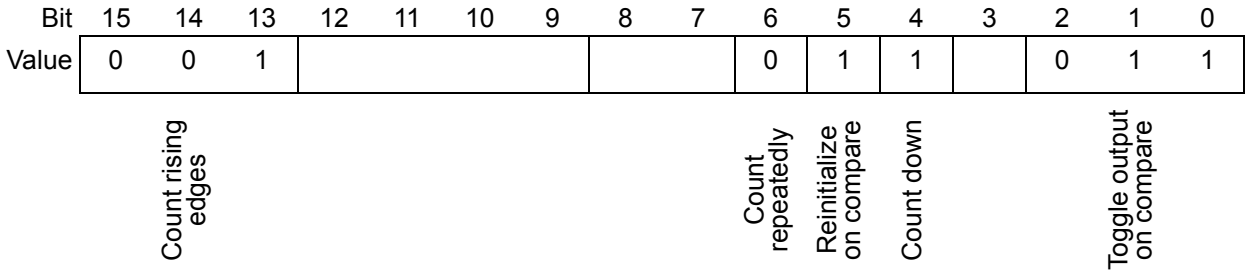


**Figure 5-1. Timer C2 Control Register (TMRC2_CTRL)**

The ADCA conversion sequence is triggered on every rising edge of the Timer C2 output and samples are converted. The conversion process takes 8.5 ADC clock cycles for the first sample and 6 ADC clock cycles for any consecutive samples in the acquisition sequence.

Note that the start of conversions is aligned to the next rising edge of the ADC Clock - at a maximum frequency of 5 MHz, the maximum delay is 200 ns. The timing is shown in **Figure 5-2**.

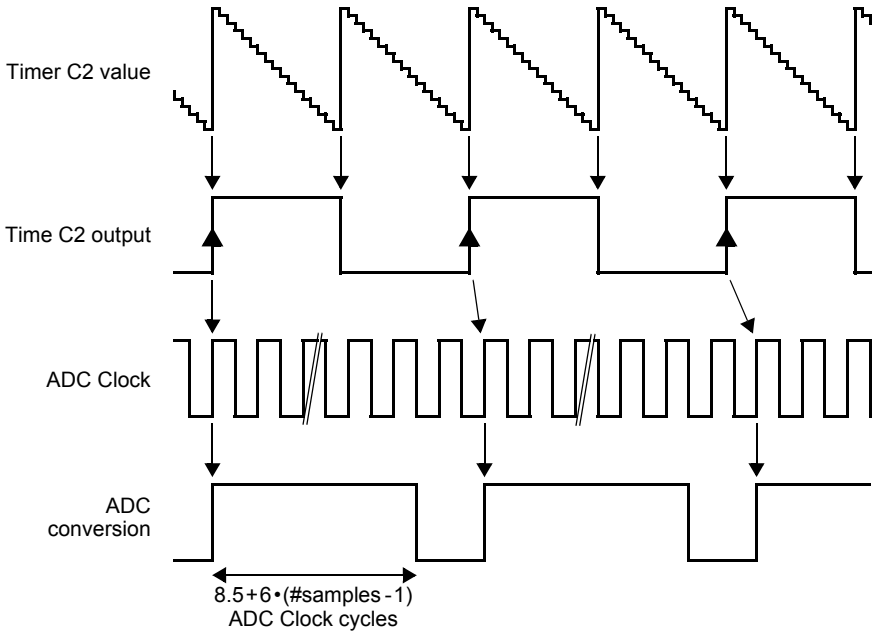


**Figure 5-2. Timing of Periodic Sampling**

## 5.2 Sampling Synchronous to PWM

The simplest form of synchronizing measurements to PWM operation is running one acquisition sequence per PWM period. In this mode the timebase is provided by the PWM peripheral instead of the Timer. There are two basic modes of operation of the PWM peripheral: Center-Aligned and Edge-Aligned. Differences between these modes and the timing of the synchronization output are shown in **Figure 5-3** and **Figure 5-4**. A synchronization pulse appears at the output of the PWM block for each reload of the PWM, regardless of the state of the LDOK bit. Note that the synchronization signal is internal to the chip. It is not directly bonded out to any pin, however, it can be observed indirectly through the operation of the timer with a dedicated input/output pin (Timer C0 or Timer C1).

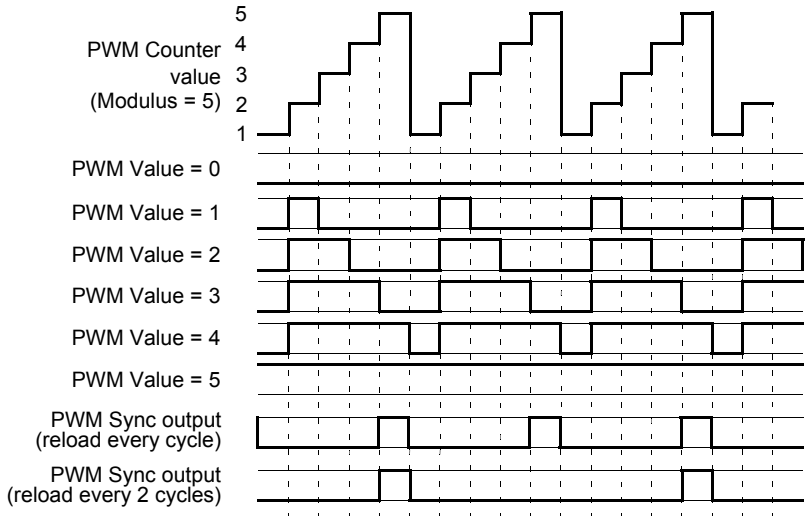


**Figure 5-3. Timing of PWM Signals in Edge-Aligned Mode**

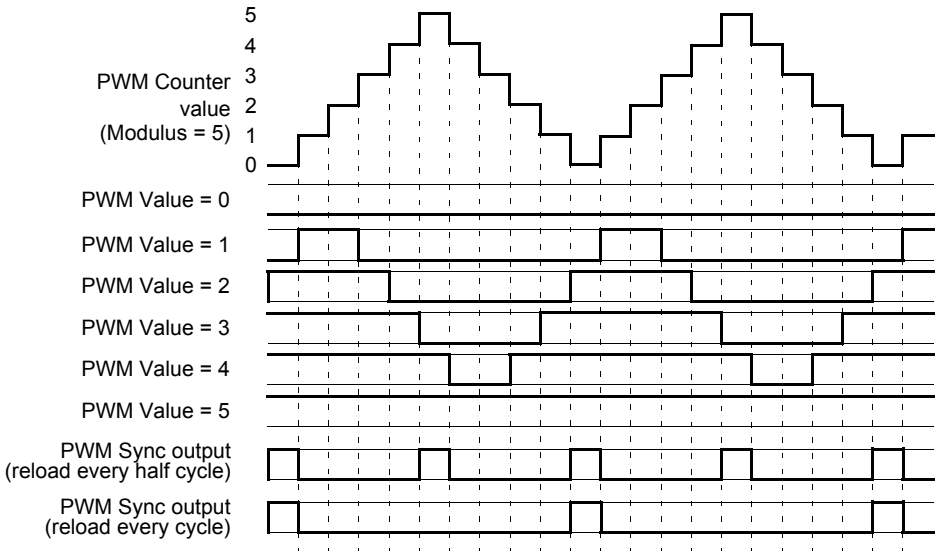


**Figure 5-4. Timing of PWM Signals in Center-Aligned Mode**

Note that the PWM outputs are delayed by one IP Bus Clock with respect to the PWM counter.

This mode requires the Timer C2 to act as a programmable delay line. After detection of the secondary input rising edge (falling edge if IPS bit in TMRC2_SCR register is set), the timer output is cleared and the timer starts counting. When the timer value reaches the pre-programmed compare value a compare event occurs, the timer stops counting, reloads its initial value from the Load Register and the output is set. The setup of the timer is shown in **Figure 5-5**.

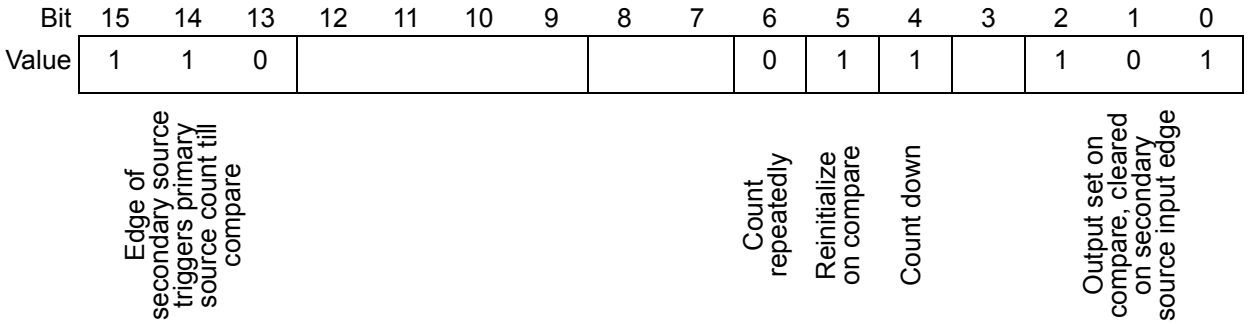


**Figure 5-5. Timer C2 Control Register (TMRC2_CTRL)**

The assertion of the Timer C2 output triggers the ADCA conversion sequence, samples are taken and stored in Result Registers (the conversion process takes 8.5 ADC Clock cycles for the first sample and 6 ADC Clock cycles for any consecutive sample in the acquisition sequence). The timing shown in **Figure 5-6** assumes that the PWM is running in Center-Aligned mode and that both the PWM and the Timer peripherals are clocked by the IPBus Clock with prescaler of 1. The ADC conversion timing is shown only for the case where the IPS bit is set. Please note that the start of ADC the conversion process is aligned to the next rising edge of the ADC Clock (not shown in the figure) - at a maximum frequency of 5 MHz the maximum delay is 200 ns.

Please also note, that, due to the synchronous architecture used for the Timer peripheral and for the signal propagation delay, the Timer will receive the synchronization signal one clock cycle after reloading the PWM module (i.e. one clock cycle after the center of the PWM output pulse in Center-Aligned mode).

The whole situation becomes more complicated when the output pulses are shortened by deadtime. **Figure 5-7** shows the timing in such a situation. At the top of the figure, the PWM counter and PWM synchronization output are shown. Below you can see an example of the complementary PWM outputs for a PWM value of 2. Up to now the situation is the same as in **Figure 5-4**. When deadtime insertion is enabled, all output rising edges are delayed (assuming outputs are active high).

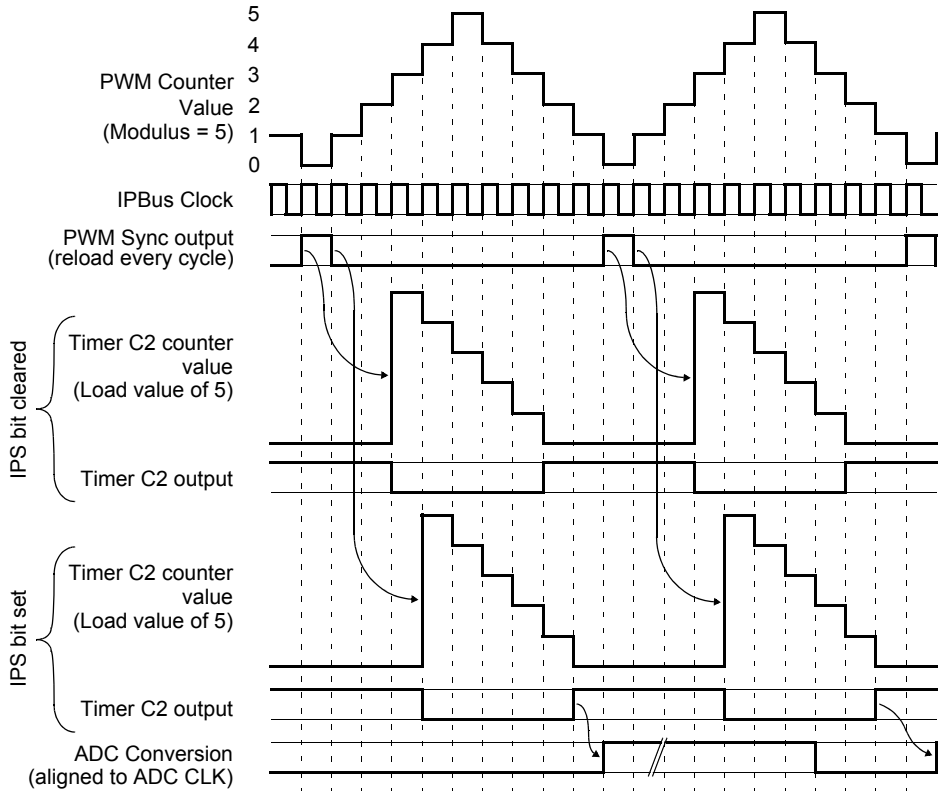


**Figure 5-6. Timing of Sampling Synchronous to PWM**

Let's now consider the whole motor control system, including optoisolation, transistor drivers and the power transistors themselves. All these components cause some signal delays, so the voltage waveform at the motor is shifted by a certain delay.

Let's now assume that the application requires measurement of currents, using shunt resistors, connected between the low side power transistor and the negative DC Bus terminal. Therefore we need to measure the currents only when the low side transistor is in ON state. The best way to achieve such a measurement is to schedule the ADC conversion to take place as near as possible of to the middle of the bottom transistor conduction pulse as possible - this optimum point is marked in **Figure 5-7** with red circles. Now we need to calculate the desired delay for the delay line, implemented by the Timer C2.

As seen in **Figure 5-7**, the center of the PWM top transistor output signal, with dead time deactivated, is at the reload point. The center of the PWM bottom transistor output signal is then at a $T/2$ distance from the reload point, where $T$ is the PWM period. However the timer starts counting one clock cycle after the reload point. The time delay between counter start and center of the PWM bottom transistor signal is therefore $T/2 - t_{PWM}$, where $t_{PWM}$ is the PWM clock period.

When deadtime is enabled, it can be seen that the center point is shifted by half of the deadtime. The center of the PWM bottom transistor output signal is therefore $T/2 - t_{PWM} + T_D/2$, where $T_D$ is the deadtime.

The ADC sampling process is about 3 ADC clock cycles long and the start of the ADC conversion process is delayed up to the next rising edge of the ADC clock. Therefore it is necessary to subtract 2 ADC clock cycles to achieve an even distribution of the sampling point around the desired center of the bottom transistor pulse. The last thing to add to the whole external hardware is the delay time $T_S$ - combined optoisolation, driver and transistor delay. The final formula for the Timer delay is therefore:

$$Timer_{Delay} = T/2 - t_{PWM} + T_D/2 - 2t_{ADC} + T_S$$

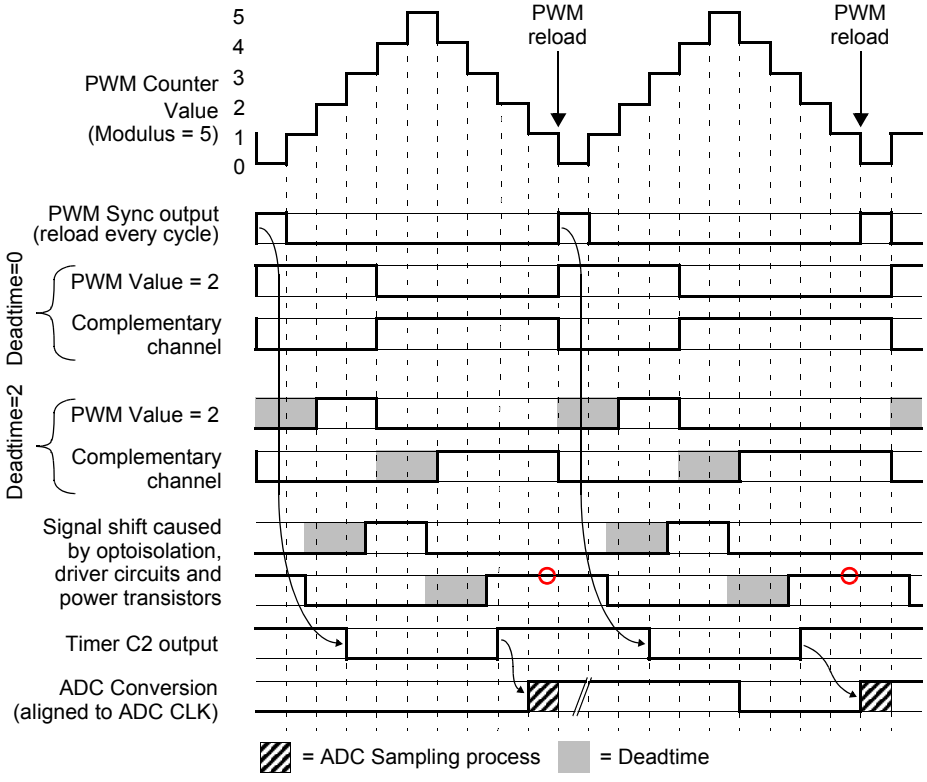


**Figure 5-7. Timing of Sampling Synchronous to PWM with Deadtime**

It is now easy to convert the delay time to the Load Register value:

$$Timer\ Load\ Register\ value = \frac{T/2 - t_{PWM} + T_D/2 - 2t_{ADC} + T_S}{t_{Timer}} - 1$$

Note that for a value of 0 in the Load Register the Timer produces a delay of 1 clock cycle (hence the subtraction of 1 in above equation).

This synchronization mode is fully autonomous and does not require any software interference. Therefore it is well suited for a wide range of applications.

## 5.3 Synchronization of ADC to External Events

The previous section described the ADC synchronization to a signal derived from the PWM module. The same principle applies to an ADC synchronization to an external signal. The setup is different only in that the synchronization signal from an external pin (TC0 or TC1) is used instead of the PWM signal (TMRCx_CTRL Register).

## 5.4 Using More than One Timer for More Advanced Synchronization

The Timer block C contains 4 timers and more than one timer unit can be utilized to achieve complex synchronization algorithms. In **Section 5.2** we have seen how to perform one ADC acquisition sequence per PWM reload period. Let's assume that we would like to perform several series of ADC acquisition sequences to acquire more data points.

To achieve this goal we need to use two timer units. Timer C1 is setup to act as a delay line and synchronizes the whole process of PWM reloads. Timer C2 is setup to count only when the output of Timer C1 is low and generates a synchronizations signal for the ADC. The whole process is shown in **Figure 5-8**.

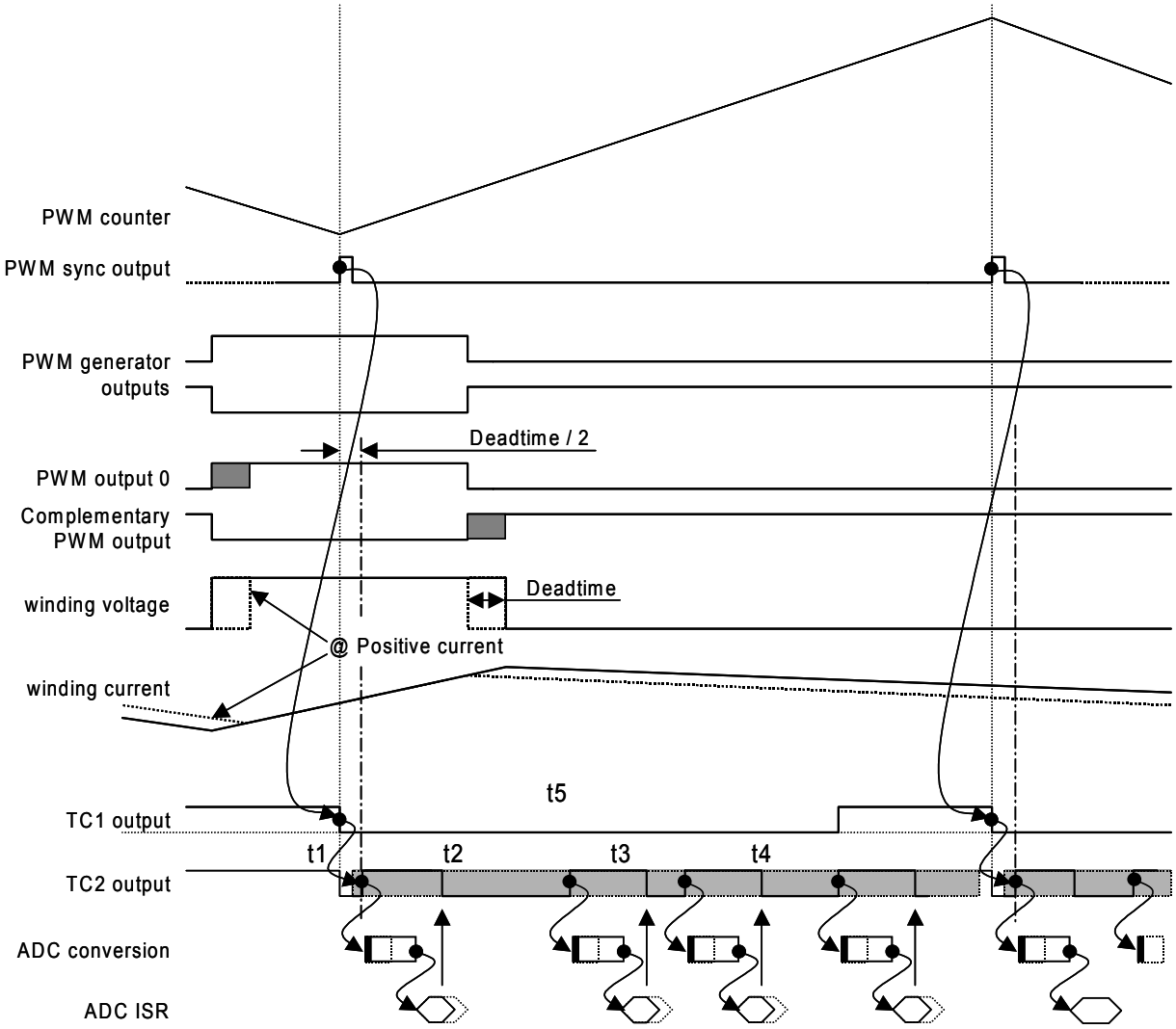


**Figure 5-8. Example of a Complex Synchronizations Setup**

After the PWM module has generated the synchronization signal, Timer C1 starts counting and forces its output low. This makes Timer C2 start counting and, once Timer C2 reaches its compare value (time *t1*), it sets its output triggering ADC conversion. Timer C2 will load a new counter value from its Load Register (corresponding to time *t2*) and continue counting. After the ADC has finished the acquisition sequence, an ADC interrupt is issued and the interrupt service routine is called. This routine picks up results from the ADC Result Registers and reprograms Timer C2 (Timer C2 output is reset and a new value is stored in the Load Register - corresponds to timer *t3*). After Timer C2 has reached the compare value once again (timer *t2* is over), the ADC conversion is triggered and the whole process repeats.

Time *t5* equals *t1+t2+t3+t4* in our example; therefore at the end of the acquisition series, the Timer C1 output is set and Timer C2 stops counting. After the ADC finishes the last conversion, the interrupt service routine is called for the last time in the series and prepares Timer C2 for the next series (Counter Register is loaded with a value corresponding to time *t1* and the Load Register is loaded with a value corresponding to time *t2*).

The number of acquisition sequences (performed in one series) and the delays between them (times *t1*, *t2*,...) can be freely chosen to fit in the application.

The execution of equidistant acquisition sequences (i.e. times *t2*, *t3*,... are equal) can be achieved by pre-programming the timer Load Register with a constant value. This simplifies the interrupt service routine because the user does not need to handle the Load Register updates.

# 6. Conclusion

The new Freescale chips - 56F80x - offer exceptional hardware features with high degree of flexibility. The 56F80x on-chip Analog-to-Digital converter (ADC) module can be synchronized to internal and external events with zero processing overhead. The on-chip timer architecture offers a high versatility, which makes the part suitable for a wide range of applications, starting from servos and control of a large variety of motor types, power inverter and converter applications up to the control of actuators or modems.

# 7. References

[1.] 56F80x 16-bit Digital Signal Controller User's Manual, DSP56F801-7UM, Freescale Semiconductor, Inc.

[2.] 56F80x MC PWM Module in Motor Control Applications, AN1905, Freescale Semiconductor, Inc.

## *How to Reach Us:*

**Home Page:**
www.freescale.com

**E-mail:**
support@freescale.com

**USA/Europe or Locations Not Listed:**
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

**Europe, Middle East, and Africa:**
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

**Japan:**
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

**Asia/Pacific:**
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

***For Literature Requests Only:***
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com






AN1933
Rev. 1
11/2005