C2000 *Teaching Materials*

GETTING STARTED WITH THE TMS320F24x PROCESSOR

Tutorial 2: Controlling the XF LED.

New Instructions Introduced

```
SETC
NOP
BCND
```

New Test Condition Introduced

```
BIO
```

Introduction

In the previous tutorial we saw how the TMS320F243 DSP Starter Kit (DSK) powers up with the XF (external flag) LED on. We also saw how we can turn this LED off under the control of software. In this tutorial we shall use software to control the brightness of the XF LED.

Turning the XF LED Off and On

To turn the XF LED off we use the instruction `CLRC XF`. How do we turn the XF LED back on again?

To do so we need to use the instruction `SETC` (set control bit). The instruction `SETC` takes a single operand, which is the name of the flag, in this case XF.

Example 2-1.

	SETC XF	; Turn on the XF LED.

The instruction `SETC` sets a control bit. By the term *set* we mean make the value logic level ‘1’ which approximates to 5V on the physical pin. This is opposite to the term *clear* which means make the logic level ‘0’, which approximates to 0V on the physical pin.

We shall now write a program to repeatedly turn the XF LED on and off. This is shown in Example 2-2.

Example 2-2.

	.setsect ".text",	8800h
start:	SETC XF	; Turn on XF LED.
	CLRC XF	; Turn off XF LED.
	B start	; Go round again.

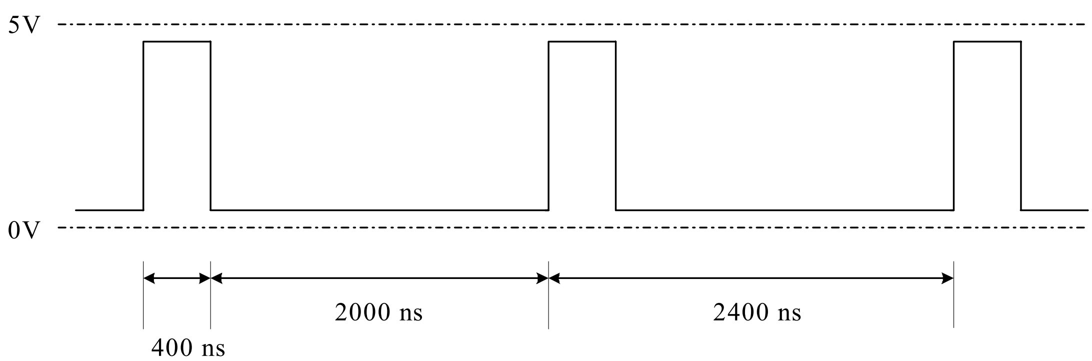
When the code in Example 2-2 is assembled and then run on the DSK, what happens?

The XF LED comes on, but *dimmed* in comparison with the power LED that is full on. Why should this be the case?

Output Wave-form at XF

If we were to look at the output on the XF pin of the TMS320F243 DSK using an oscilloscope, we would see the waveform shown in Figure 2-1.

Figure 2-1. Wave-form at XF Pin



It can be seen from Figure 2-1 that there is a repeating pattern at the XF pin. The output is high (at approximately 5V) for 400 ns, then low (at approximately 0V) for 2000 ns. The pattern repeats every 2400 ns. This corresponds to a frequency of 417 kHz.

The XF LED is in fact being switched on and off 417000 times every second. Because this is faster than the human eye can see and the LED is off more than it is on, the LED appears to be dimmed.

Timings of XF Wave-form

Why are the timings of the waveform at the XF pin as they are?

Each TMS320F243 instruction takes a certain amount of time to execute. This time is derived from the number of *processor machine cycles*. The execution time for every instruction can be expressed in terms of processor machine cycles, more commonly referred to simply as *cycles*. On the TMS320F243 DSK, each cycle takes 400 ns. We can measure this time by measuring the output at the CLKOUT pin on the TMS320F243 DSK using an oscilloscope or a frequency meter.

Let us re-write Example 2-2, but this time putting in the number of cycles.

Example 2-3.

	.setsect ".text",	8800h
start:	SETC XF	; Turn on XF LED. 1 cycle.
	CLRC XF	; Turn off XF LED. 1 cycle.
	B start	; Go round again. 4 cycles.

Example 2-3 shows how the instructions `SETC` and `CLRC` both execute in a single cycle, whereas the instruction `B` requires 4 cycles. The total time to complete the loop is therefore 1 + 1 + 4 = 6 cycles.

When calculating the time to execute a block of code, it is normal practice to express the execution time in cycles rather than in divisions of a second. This is because the number of clock cycles is always the same, regardless of the speed on CLKOUT1. The actual timings can then be calculated by multiplying the number of cycles by the time per cycle.

Note that XF pin stays in the same state until the instruction has finished executing. When XF is at logic '1', it takes 1 cycle (400ns) for the instruction `CLRC XF` to take effect.

A Brighter XF LED

Let us now re-write Example 2-3, but this time reversing the order in which the `SETC XF` and `CLRC XF` instructions occur.

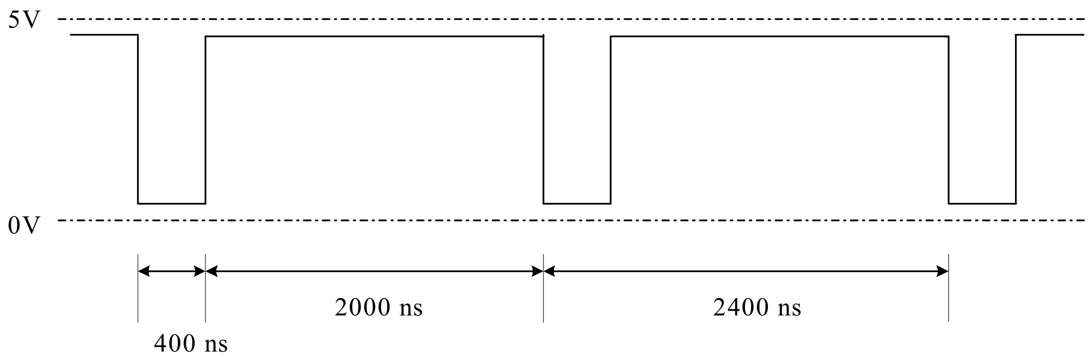
Example 2-4.

	.setsect ".text",	8800h
start:	CLRC XF	; Turn off XF LED. 1 cycle.
	SETC XF	; Turn on XF LED. 1 cycle.
	B start	; Go round again. 4 cycles.

In this case the XF LED will be off for 400 ns and on for 2000 ns. As in Example 2-3, the frequency will be 417 kHz.

If the code is run on the DSK, then the XF LED will be on at almost full brightness, as compared with the power LED. Figure 2-2 shows the waveform at the XF pin.

Figure 2-2. Wave-form at XF Pin



Equal On and Off Times

In Examples 2-3 and Example 2-4, the XF LED is off for a different time to which it is on. How do we make the on and off times identical?

One way is to add some “padding”. We require an instruction that does nothing other than take up execution time. For this we use the instruction `NOP` (no operation), which takes a single cycle to execute.

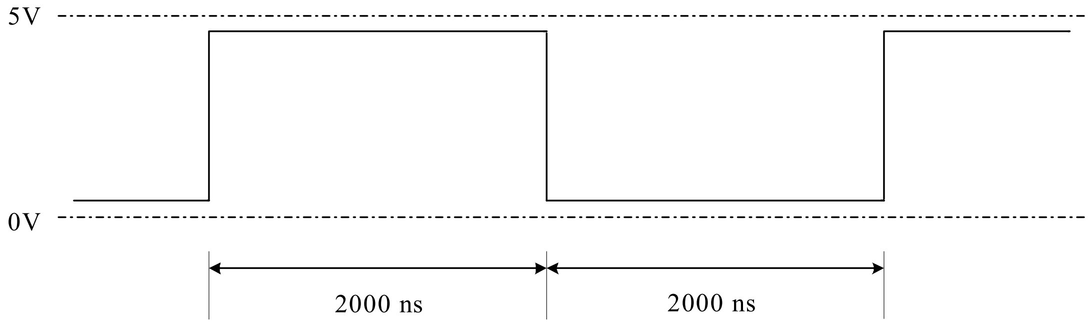
Example 2-5.

	.setsect ".text",	8800h
start:	CLRC XF	; Turn off XF LED. 1 cycle.
	NOP	; No operation. 1 cycle.
	NOP	; No operation. 1 cycle.
	NOP	; No operation. 1 cycle.
	NOP	; No operation. 1 cycle.
	SETC XF	; Turn on XF LED. 1 cycle.
	B start	; Go round again. 4 cycles.

In Example 2-5, the XF LED will be off for 1 + 1 + 1 + 1 + 1 = 5 cycles and on for 4 + 1 = 5 cycles. This corresponds to an on time of 5 *x* 400 ns = 2000 ns and an off time of 5 *x* 400 ns = 2000 ns.

If we were to assemble and run Example 2-5, then using an oscilloscpoe we would see the waveform as shown at Figure 2-3.

Figure 2-3. Equal on and off times



The total cycle time = 10 *x* 400 ns = 4000 ns = 4 µs. This corresponds to a frequency measured at the XF pin of 250 kHz.

External Control of the XF LED

The TMS320F243 has a dedicated physical input pin (BIO) that can be used to control the execution of the program. When the input pin BIO is pulled down to logic ‘0’ (0V), then a branch can be made to occur. When the input pin BIO is left unconnected or connected to 5V (logic ‘1’), then no branch occurs.

The C code to carry out this operation would be of the form:

Example 2-6.

	`for ( ; ; )`	
	`{`	
	`if (BIO == 0 )`	`/* Test if BIO pin is zero */`
	`XF = 0;`	`/* Turn XF LED off */`
	`else`	
	`XF = 1;`	`/* Turn XF LED on */`
	`}`	

To implement Example 2-6 in assembly language we could write:

Example 2-7.

	`.setsect “.text”,`	`8800h`

`start:`	`BCND off, BIO`	`; Test if BIO pin is low`
		`; (logic ‘0’). If so, branch to`
		`; label` *`off`*`.`

	`SETC XF`	`; Turn on XF LED.`
	`B done`	`; Do not execute next` `; instruction.`
`off``:`	`CLRC XF`	`; Turn off XF LED.`
`done``:`	`B start`	`; Go round again.`

The instruction BCND (branch conditionally) takes two operands. The first (left-hand) operand is *where* the branch is to go and is specified by a *label*. It is similar in syntax to the instruction B (branch unconditionally).

The second (right-hand) operand is the *test condition*.

If the *test condition* evaluates to TRUE, then a branch occurs. Here the test *condition* to be evaluated is "the BIO pin is at logic '0' (low)"? If this is true, then the branch occurs.

If the condition under test does not evaluate to true, which will be the case when the BIO pin is at logic '1', then a branch does not occur. The next instruction to be executed is the one immediately following the instruction BCND.

Example 2-8 shows some right and wrong ways to use the instruction BCND:

Example 2-8.

start:	BCND done, BIO	; Correct syntax.
	BCND BIO	; Incorrect. Missing operand.
done:	BCND BIO, start	; Incorrect. Operands reversed.

We can only branch on the condition that the BIO pin is low (logic '0'). There is no conditional branch available on BIO pin high (logic '1').

Timing of the Loop

We can re-write the code in Example 2-7 showing the instruction cycles.

Example 2-9.

	.setsect ".text",	8800h
start:	BCND off, BIO	; 4 cycles when branch occurs, ; otherwise 2 cycles.
	SETC XF	; 1 cycle.
	B done	; 4 cycles.
off:	CLRC XF	; 1 cycle.
done:	B start	; 4 cycles.

Note that the number of cycles taken by the instruction BCND depends on whether the branch occurs. If the branch occurs, then the instruction takes 4 cycles. If the branch does not occur, then the instruction takes only 2 cycles.

When the BIO pin is high (logic '1) then the condition BIO low evaluates to FALSE. Therefore the instruction BCND does not branch but falls through to the next instruction. The loop executes in 2 + 1 + 4 + 4 = 11 cycles.

On the other hand, when the pin `BIO` is low (logic '0') then the condition `BIO` low evaluates to TRUE. The conditional branch `BCND` causes program execution to continue at the label *`off`*. The loop executes in 4 + 1 + 4 = 9 cycles.

In this case the loop executes in different numbers of cycles depending upon the status of the `BIO` pin.

The Importance of Instruction Cycles

In this tutorial, the execution times of the instructions have been stated in cycles. When performing real-time programming, the speed of operation can often be crucial and effort goes into optimizing the code. Instruction cycles give us the speed of throughput through the code. When using a C compiler, we can never be sure of which instructions are actually being used and how well the code is optimized.

DC Outputs

In Example 2-3, Example 2-4 and Example 2-5 we have produced an output waveform on the XF pin, which affects the brightness of the XF LED. So far we have looked at this output as an alternating (A.C.) signal. However, if we were to measure the output at the XF pin with an analog meter set to a 5V DC or 10V DC range, then we would measure a voltage at the XF pin as follows:

Table 2-1.

Program	XF on time	XF off time	D.C. Voltage
Power up	XF full on	None	4.9 V
Example 2-3	on 400 ns	off 2000 ns	0.9 V
Example 2-4	on 2000 ns	off 400 ns	4.1 V
Example 2-5	on 2000 ns	off 2000 ns	2.5V

To calculate the DC output we need to find the average value that the output produces. In the case of Example 2-3, the output is 0V for 400 ns and 4.9V for 2000 ns. The average is:

400/(2000+400) * 4.9V + 2000/(2000+400) * 0.0V = 0.9V.

Similarly:

0.9V = 4.9V *x* 1/6
4.1V = 4.9V *x* 5/6
2.5V = 4.9V *x* 3/6

Digital to Analogue Conversion

By altering the time the XF pin is high / low we have been able to vary the D.C. output. This is a commonly used method used to generate analog outputs from a digital source. The TMS320F243 has a series of timers that can be configured to produce output pulses of different widths for use as D-to-A converters.

TMS320F243 DSK EXPERIMENTS

Equipment required: TMS320F243 DSK.
Digital frequency meter (0 - 2.5 MHz) or oscilloscope.
Analog D.C. meter 0-5V or 0 –10V (or analog multi-meter).
Switch and resistors to short out BIO pin to ground.

Experiment 2-1.

Objective: To Measure the Clock Frequency of the TMS320F243 DSK

Measure the frequency on the CLKOUT pin of the TMS320F243 DSK using a frequency meter or an oscilloscope. Connect one probe to CLKOUT (Connector P2, Pin 29) and the other probe to GND (Connector P2, Pin 39 or Pin 40).

For the TMS320F243 DSK, the frequency on CLKOUT is 2.5 MHz, which corresponds to an instruction time of 400 ns.

Experiment 2-2.

Objective: To Generate 0.9V D.C. on the XF Pin

Enter the code in Example 2-2 into a text editor, assemble it and run it on the TMS320F243 DSK.

The frequency on the XF LED measured between XF (Connector P2, Pin 21 or one end of the XF LED itself) and GND (Connector P2, Pin 39 or 40) should be 471 kHz.

If an oscilloscope is available, then using a Y deflection of 1V/cm and a time-base of 500 ns per division, the output should be as per Figure 2-1.

If an analog D.C. meter is used to measure the output on the XF pin, then the measured voltage should be 0.9V.

Experiment 2-3.

Objective: To Generate 4.1V D.C. on the XF Pin

Enter the code in Example 2-4 into a text editor, save it, assemble it and run the `.dsk` file the TMS320F243 DSK.

The frequency on the XF LED should be 471 kHz and output should be as per Figure 2-2. The D.C value should be 4.1V.

Experiment 2-4.

Objective: To Generate 2.5V D.C. on the XF Pin

Enter the code in Example 2-5 into a text editor, save it, assemble it then run it on the TMS320F243 DSK.

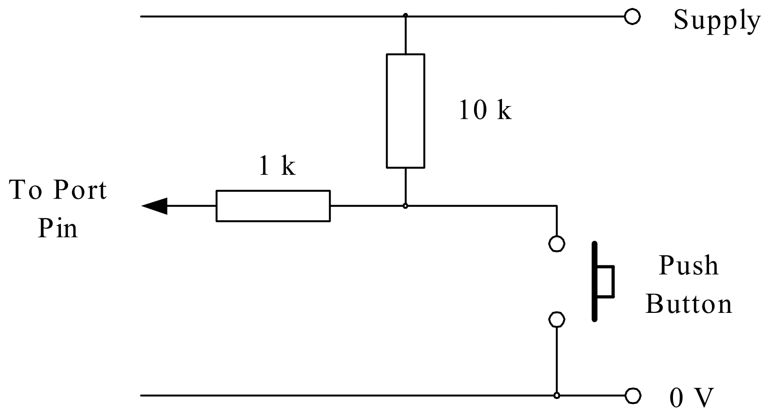
The frequency on the XF LED should be 250 kHz and output should be as per Figure 2-3. The D.C value should be 2.5V.

Experiment 2-5.

Objective: To Control the XF LED using the BIO Pin

Connect a switch to the BIO pin shown in Figure 2-4. For the TMS320F243 DSK, the supply can be up to 5V. However, for devices such as the TMS320LF2407, the supply voltage should not exceed 3.3V, otherwise damage can be done to the device.

Figure 2-4. Suggested Connection of Switch to BIO Pin



Enter the code in Example 2-7 into a text editor, save it, assemble it and run it on the TMS320F243 DSK.

The XF LED should be off. Short the BIO pin to ground to turn on the XF LED. On the TMS320F243 DSK, the BIO pin is Connector P2, Pin 22 and the ground is Connector P2, Pin 39 or 40.

Design Problem 2-1.

Rewrite the code in Example 2-7 so that when the BIO pin is unconnected the XF LED is off, and when the BIO pin is shorted to ground, the XF LED comes on.

Design Problem 2-2.

In Example 2-9, when the BIO pin is high (5V which corresponds to logic ‘1’), the loop executes in 11 cycles. Re-write Example 2-9 so that when BIO pin is high, the loop executes in 7 cycles.

***Self-Test Questions*......................................** *Click Here To View* Answers!

1.	What is meant by the term *set* when applied to a control bit?
2.	What is meant by the term *clear* when applied to a control bit?
3.	Which two of the following instructions are correct? a) `SET XF` b) `SETC XF` c) `CLR XF` d) `CLRC XF`
4.	In order turn on the XF LED on the DSK, which instruction do we use?
5.	In order to turn off the XF LED on the DSK, which instruction do we use?
6.	If we repeatedly flash the XF LED on and off rapidly, why does the XF LED appear dimmed?
7.	Why is the frequency on CLKOUT important?
8.	Why do we express execution time in cycles rather than in divisions of seconds?
9.	What effect does the instruction `NOP` have?
10.	Which one of the following commands is correct? a) `B label, BIO` b) `BCND label, BIO` c) `BCND label,` d) `BCND BIO, label` e) `BCND BIO` f) `BCND label1, label2`
11.	How would we use the BIO pin to control software?
12.	The instruction `BCND` always takes 4 cycles to execute. Is this true or false?
13.	Why are timing cycles important for real-time programming?
14.	How do we use the output on the XF pin as a simple digital-to-analog converter (D-to-A)?

References

TMS320F/C24x DSP Controllers. CPU and Instruction Set. Reference Number: SPRU160.

TMS320C/F240 DSP Controllers. Peripheral Library and Specific Devices. Reference Number: SPRU161

TMS320F243, TMS320F241 DSP Controllers. Reference Number SPRS064.

CLICK TO VIEW
Tutorials
1 2 3 4 5 6 7 8 9 10

Click here to view.......... Route Map