AN204959

F²MC-8FX Family 8-Bit Microcontroller Low Power Consumption Strategy

Associated Part Family: F²MC-8FX Family 8-Bit Microcontroller Series

This application note describes strategy for low power consumption in all F²MC-8FX Family 8-Bit Microcontroller Series

Contents

1 Introduction

This application note describes strategy for low power consumption in all F²MC-8FX Family 8-Bit Microcontroller Series MCU.

This application note mainly introduces method to reduce power consumption and provides investigation of LPC MCU power consumption.

2 Low Power Consumption Strategy

Low power consumption mainly targets to reduce system cost and increase battery life in the embedded application, especially for portable device. For embedded application based on MCU, minimizing system power consumption can be implemented by both hardware and software method.

3 HW Strategy

3.1 Select Low Power Supply Voltage

The power consumption can be implemented efficiently by reducing the power supply voltage of MCU. According to LPC MCU, it can work at a wide voltage range between 2.4V and 5.5V, as shown in Table 3-1.

Table 3-1 Operation Conditions Description

Parameter	Symbol	Value		Unit	Remarks	
		Min	Max			
Power supply voltage	V_{CC}	2.4	5.5	V	In normal operation	Other than on-chip debug mode
		2.3	5.5		Hold condition in stop mode	
		2.9	5.5		In normal operation	On-chip debug mode
		2.3	5.5		Hold condition in stop mode	
Operating temperature	T_A	-40	+85	℃	Other than on-chip debug function	
		+5	+35		On-chip debug function	

System current will be reduced largely if low power supply is adopted, and the lower the power supply voltage is, the lower the power consumption will be. Therefore it's better to select lower power supply voltage only if system requirement can be achieved.

Now, power supply voltage of many MCU systems is 5 V, in the last 5 years, quantity of 3 V MCU system is increasing so much, quantity of 2 V MCU system is also increasing, in future, quantity of low power supply voltage MCU system may be more than 5 V. It's an important trend to reduce the power supply voltage of MCU.

3.2 Select Low Power Consumption Clock

3.2.1 Adopt Suitable Clock

For F²MC-8FX Family 8-Bit Microcontroller Series MCU, four kinds of clock source are available: Main clock, Main CR clock, Sub clock, Sub CR clock.

- Main Clock Mode

Main clock is used as the machine clock for the CPU and peripheral functions. The timebase timer operates using the main clock. The watch prescaler operates with the subclock (on the dual external clock product).

- Sub Clock Mode (on Dual External Clock Product)

Main clock oscillation is stopped and the sub clock is used as the machine clock for the CPU and peripheral functions. In this mode, the timebase timer stops as it requires the main clock for operation.

- Main CR Clock Mode

Main CR clock is used as the machine clock for the CPU and peripheral functions. The timebase timer and the watchdog timer operate using the main clock. The watch prescaler operates with the sub clock (on the dual external clock product).

- Sub CR Clock Mode (on Dual External Clock Product)

Main clock oscillation is stopped and the sub-CR clock is used as the machine clock for the CPU and peripheral functions. In this mode, the timebase timer stops as it requires the main clock for operation. The watch prescaler operates using the sub-CR clock.

Using Main CR clock and Sub CR clock, the advantage is to economize external crystal， so it can reduce HW cost; the disadvantage is that the precision may be not high, and the power consumption may be high. For more details about how to select suitable clock, refer to related hardware manual and datasheet.

3.2.2 Switch among Clock Modes

Sometimes, it may not satisfy requirement of system function and low power consumption if just adopting one clock mode. Therefore all four clock modes or part of four clock modes can be adopted by Cypress MCU, and MCU can switch among these modes to satisfy different requirements. Figure 3-1 and Figure 3-2 show switching between these modes.

Figure 3-1 Clock Mode State Transition Diagram (Dual External Clock Product)

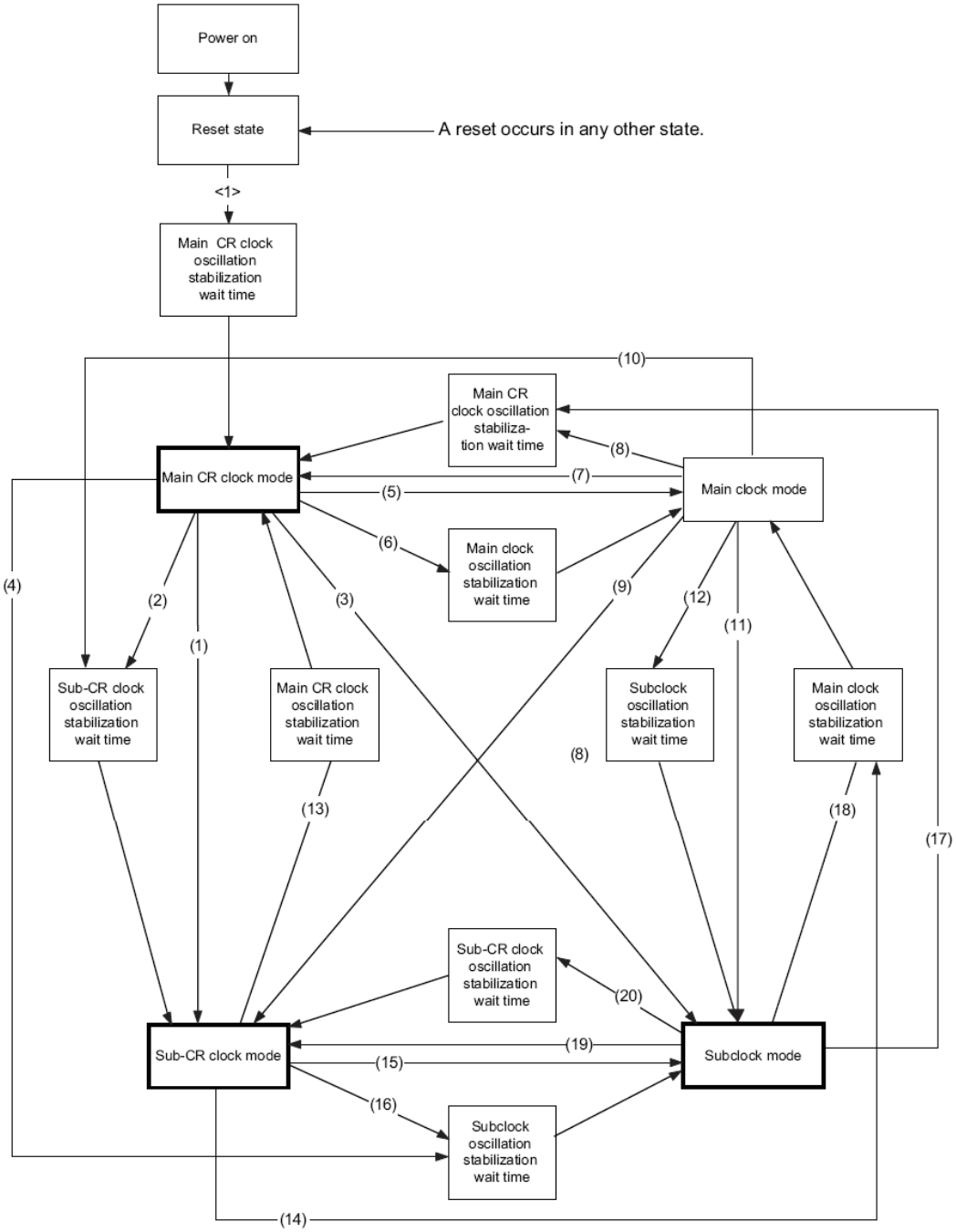
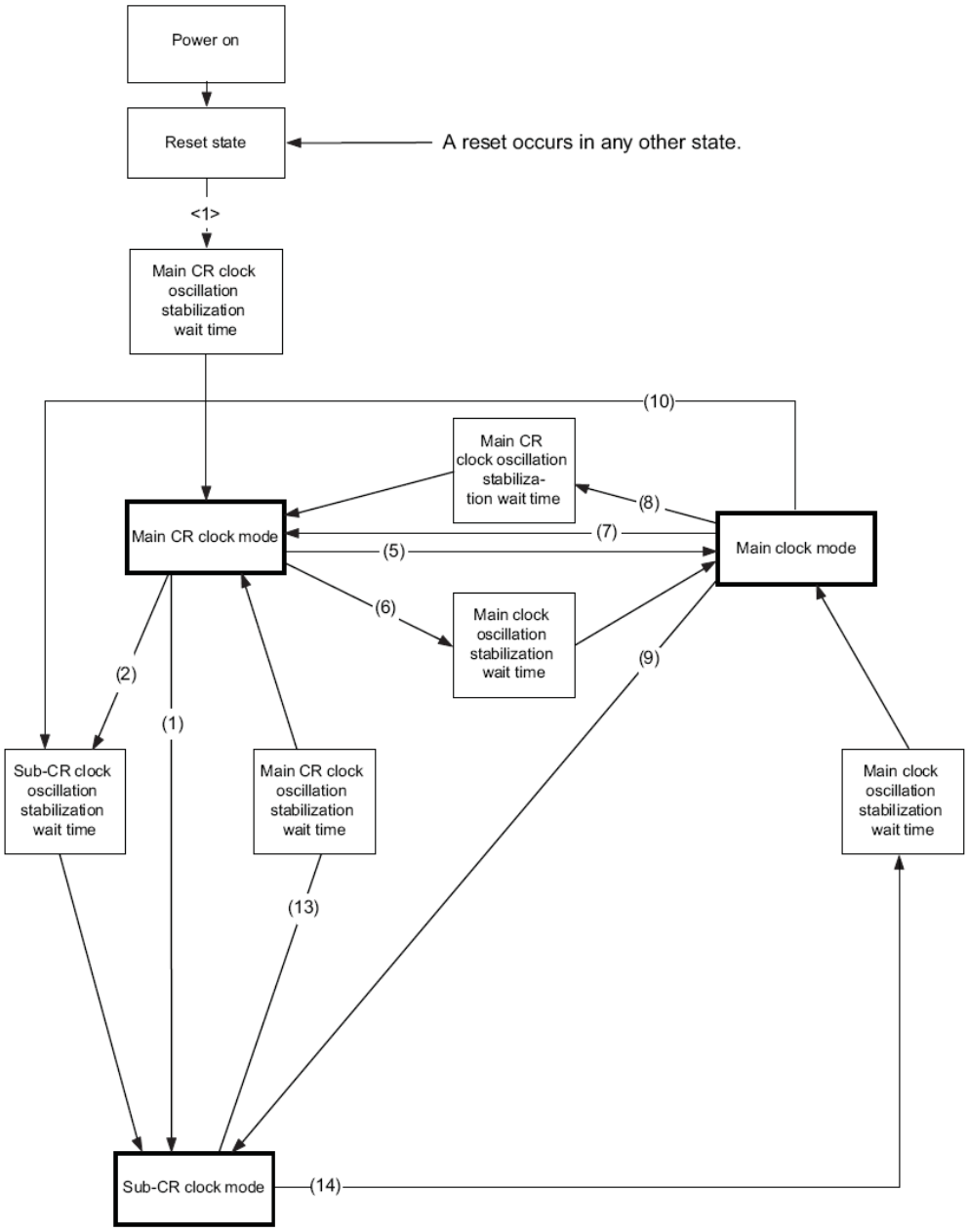


Figure 3-2 Clock Mode State Transition Diagram (Single External Clock Product)



3.3 Select Low Power Consumption Mode

For F²MC-8FX Family 8-Bit Microcontroller Series MCU, low power consumption mode (Standby mode), includes sleep mode, timebase timer mode, watch mode and stop mode. Power consumption in standby mode is less than in run mode.

MCU can be simply divided into below modules: main clock, main CR clock, sub clock, sub CR clock, CPU, ROM, RAM, I/O ports, timebase timer, watch prescaler, external interrupt, hardware watchdog timer, software watchdog timer, low-voltage detection reset, and other peripheral Functions.

3.3.1 Internal Operating States for Standby Mode and Clock Mode

When all these modules are working, the power consumption is the highest. In run mode, many of these modules are working, so the power consumption is high. Oppositely in standby mode, just parts of these modules are working, so the power consumption is low.

Take MB95200/210 Series for example, Table 3-2 and Table 3-3 show operation states of different clock modes in both RUN and sleep mode.

Table 3-2 Combinations of Standby Mode and Clock Mode and Internal Operating States (1)

Function	RUN				Sleep			
	Main clock mode	Main CR clock mode	Subclock mode (Dual external clock prod-uct)	Sub-CR clock mode	Main clock mode	Main CR clock mode	Subclock mode (Dual external clock prod-uct)	Sub-CR clock mode
Main clock	Operating	Stopped*1	Stopped		Operating	Stopped*1	Stopped	
Main CR clock	Stopped*2	Operating	Stopped		Stopped*2	Operating	Stopped	
Subclock	Operating*3		Operating	Operating*3	Operating*3		Operating	Operating*3
Sub-CR clock	Operating*4		Operating*4	Operating	Operating*4		Operating*4	Operating
CPU	Operating		Operating		Stopped		Stopped	
ROM	Operating		Operating		Value held		Value held	
RAM								
I/O ports	Operating		Operating		Output held		Output held	
Timebase timer	Operating		Stopped		Operating		Stopped	
Watch prescaler	Operating*3, *4		Operating		Operating*3, *4		Operating	
External interrupt	Operating		Operating		Operating		Operating	
Hardware watchdog timer	Operating		Operating		Operating*5		Operating*5	
Software watchdog timer	Operating		Operating		Stopped		Stopped	
Low-voltage detection reset	Operating		Operating		Operating		Operating	
Other peripheral functions	Operating		Operating		Operating		Operating	

Table 3-3 Combinations of Standby Mode and Clock Mode and Internal Operating States (2)

Function	Timebase timer		Watch prescaler		Stop			
	Main clock mode	Main CR clock mode	Subclock mode (Dual external clock prod-uct)	Sub-CR clock mode	Main clock mode	Main CR clock mode	Subclock mode (Dual external clock prod-uct)	Sub-CR clock mode
Main clock	Operating	Stopped*1	Stopped		Stopped			
Main CR clock	Stopped*2	Operating	Stopped		Stopped			
Subclock	Operating*3		Operating	Operating*3	Operating*3		Stopped	
Sub-CR clock	Operating*4		Operating*4	Operating	Operating*4		Stopped	
CPU	Stopped		Stopped		Stopped			
ROM / RAM	Value held		Value held		Value held			
I/O ports	Output held / Hi-Z		Output held		Output held/Hi-Z			
Timebase timer	Operating		Stopped		Stopped			
Watch prescaler	Operating*3, *4		Operating		Operating*3, 4		Stopped	
External interrupt	Operating		Operating		Operating			
Hardware watchdog timer	Operating*5		Operating*5		Operating*5			
Software watchdog timer	Stopped		Stopped		Stopped			
Low-voltage detection reset	Operating		Operating		Operating			
Other peripheral functions	Stopped		Stopped		Stopped			

3.3.2 Power Consumption of Standby Modes

Take MB95F264 as example, the power consumption is not the same in different standby modes. Table 3-4 describes the power consumption of standby modes.

Table 3-4 Power Consumption of Standby Modes

Mode Name	Power Consumption		Voltage	Clock Frequency	Temperature
	Typ	Max			
Sleep mode	5.5 mA	9 mA	5.5 V_{CC}	32MHz	+25℃
Stop mode	3.5 μA	22.5 μA	5.5 V_{CC}	32kHz	+25℃
Timebase timer mode	1.1 mA	3 mA	5.5 V_{CC}	32MHz	+25℃
Watch mode	5 μA	30 μA	5.5 V_{CC}	32kHz	+25℃

3.3.3 Switch among Different Standby Modes

Sometimes, it may not satisfy requirement of system function and low power consumption if just adopting one standby mode. So except run mode, all four standby modes or part of four clock modes can be adopted, and MCU can switch among these modes to satisfy different requirements. Figure 3-3 and figure 3-4 show switching between these modes.

Figure 3-3 Standby Mode State Transition Diagram (Dual External Clock Product)

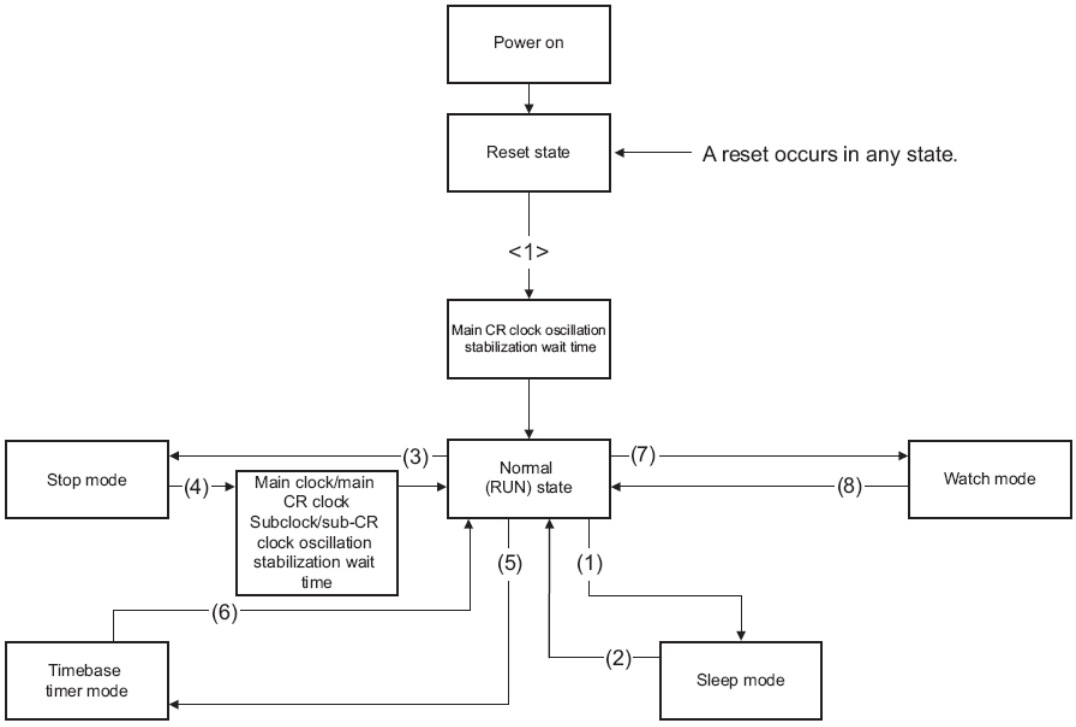
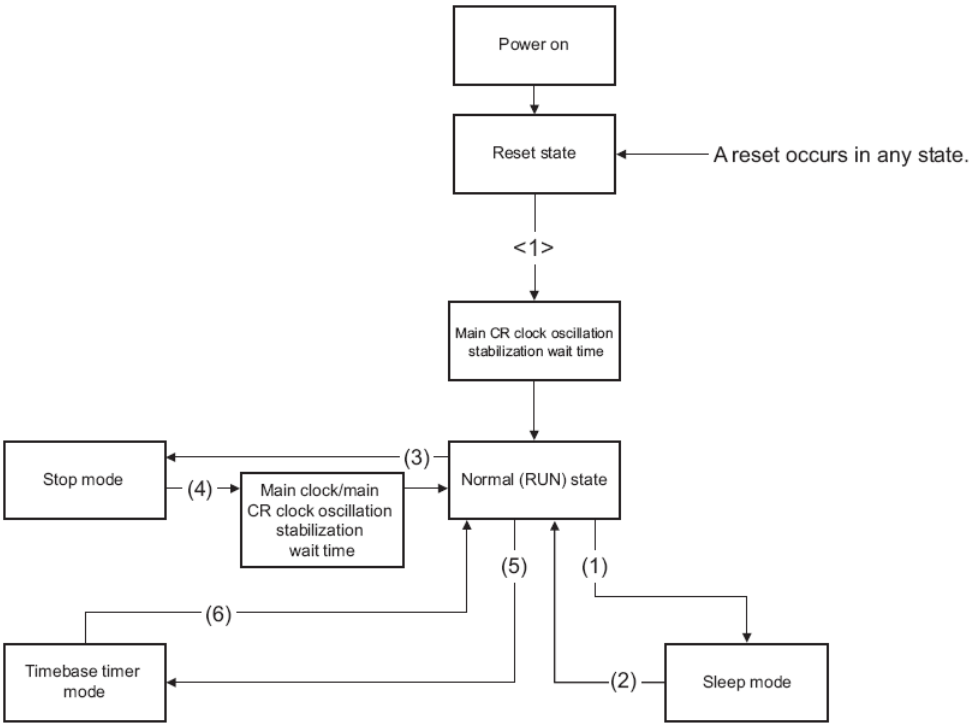


Figure 3-4 Standby Mode State Transition Diagram (Single External Clock Product)



3.3.4 Clock Setting Example for Low Power Consumption

Following are two simple examples:

In car alarm, select main clock as MCLK when it works at normal status, and change to sub clock (32.768 KHz) when it enters into sleep mode.

In smoke detector or other low frequency system, select a 200K sub CR as MCLK is enough. It is based on the fact that this kind of system only needs to work at very low frequency and doesn't care real time function.

4 Software Strategy

Suitable software can reduce power consumption, but it's easy to ignore by users. For reducing power consumption, it's not easy to find the defect on FW, and there is no strict standard to judge if FW has the feature of low power consumption. However, there are some ways to avoid unnecessary power consumption due to FW defect.

4.1 Interrupt Instead of Inquire

For a simple application, it doesn't matter to adopt interrupting or inquiring, however it's important for lower power consumption system. When using interrupt, CPU needs to do nothing, even enter stop mode (lowest power consumption mode); while using inquire, CPU must access I/O registers continuously, so more additional power is consumed.

4.2 Macro Instead of Subroutine

It's well known that more power is consumed to access RAM than to access Flash. When using subroutine, four steps are necessary:

1. Store parameters in stack
2. Save registers in stack
3. Save result
4. Restore registers

What's more, all these four steps are to access RAM. To solve this problem, macro can be used instead of subroutine. In this way, the four steps to access RAM are not implemented any more. But there is a problem that the code size is increased. Fortunately for F²MC-8FX Family 8-Bit Microcontroller Series MCU, the Flash size is from 4K bytes to 60K bytes, and diversified MCU can be selected to satisfy different applications.

4.3 Reduce Calculation

There are many ways to reduce MCU calculation.

4.3.1 Inquire Table

Get the calculation data results beforehand, make them into a data table, and store the data table in Flash of MCU, then MCU will get them by inquiring the data table. In this way, many real time calculations will be reduced, so the power consumption is reduced.

4.3.2 Stop calculation when precision is satisfied

Sometimes some real time calculations must be implemented, if there is requirement for precision, stop the calculation when the precision is satisfied. In this way, many unnecessary real time calculations will be reduced, so the power consumption is reduced.

4.3.3 Adopt Suitable Data Type

Try to adopt suitable data type, such as using 8 bit data to instead of 16 bit data, using fraction calculation to instead of float data calculation. In this way, some unnecessary calculations will be reduced, so the power consumption is reduced.

4.3.4 Adopt Cypress Math API

Try to adopt Cypress Math API instead of original Cypress Math Lib.

Cypress Math APIs are available, which can be applied to all series F²MC-8FX 8-Bit Microcontroller. This math API can calculate more efficiently. Compared to F²MC-8FX compiler's math arithmetic, only multiplication and division APIs which need to be advanced are implemented in this Math API.

For details, refer to mcu-an-500073-e-14 and related sample project.

Table 4-1 shows the performance of Cypress Math API. In this way, many unnecessary calculations will be reduced, so the power consumption is reduced.

Table 4-1: Performance Compare between Math API and C Compiler Math Lib

	C compiler Math Lib				Math API			
Calculation	Times	MCLK	ROM	RAM	Times	MCLK	ROM	RAM
UChar*Uint	10	252	127	10	2	87	40	8
UInt*Uint	10	252	127	10	4	168	88	10
ULong*Uchar	10	252	127	10	4	157	79	12
ULong*Uint	10	252	127	10	7	212	107	12
ULong/Uchar	0	1614	238	12	0	1327	80	12
ULong/Uint	0	1614	238	12	0	1320	76	12

Note: Times: how many times math API uses MULU/DIVU

MCLK: how many machine clock used by math API

ROM: how many bytes ROM used by math API

RAM: how many bytes RAM (stack) used by math API

4.4 Close unused Module

To reduce power consumption, close MCU I/Os and periphery which are not used, and close MCU I/Os and periphery in time which are used discontinuity.

Some periphery modules such as RS232 modules, it will cost power, so 1 I/O pin can be used to control it, when module is not used sometimes, power of module can be turned off by MCU.

Set unused I/O pins to input or output, and pull up to Vcc through pull up resistance, because if these pins are not initialized, the leak current will be increased.

5 Additional Information

For more Information on Cypress products, visit the following web sites:

http://www.cypress.com/cypress-microcontrollers

Document History

Document Title: AN204959 - F²MC-8FX Family 8-bit Microcontroller Low Power Consumption Strategy

Document Number: 002-04959

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	—	HUAL	12/07/2009	Initial release
			01/08/2009	add detail description
			01/12/2009	add 3.3.4
			01/14/2009	update incorrect expression
*A	5265540	HUAL	05/10/2016	Migrated Spansion Application Note “MCU-AN-500074-E-13” to Cypress format

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com