To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

RENESAS

Power Consumption Optimization in NEC Electronics Microcontrollers

Application Note **August 2003**

Introduction

For embedded microcontroller applications that are battery-powered, it is important to minimize power consumption to maximize battery life. In conventional microcontroller designs, this need dictates the use of a system clock with slower speed. Advanced microcontroller architectures employ low-power modes instead. NEC Electronics collectively refers to these low-power modes as the *standby function*. Several modes within the standby function provide shutdown methods for unused internal circuitry.

NEC Electronics' microcontroller architecture improves low-power operation by providing a special clock prescaling to the CPU and separately to each integrated peripheral device. Users can switch in real time between different clock settings to achieve an optimal ratio of system performance versus power consumption.

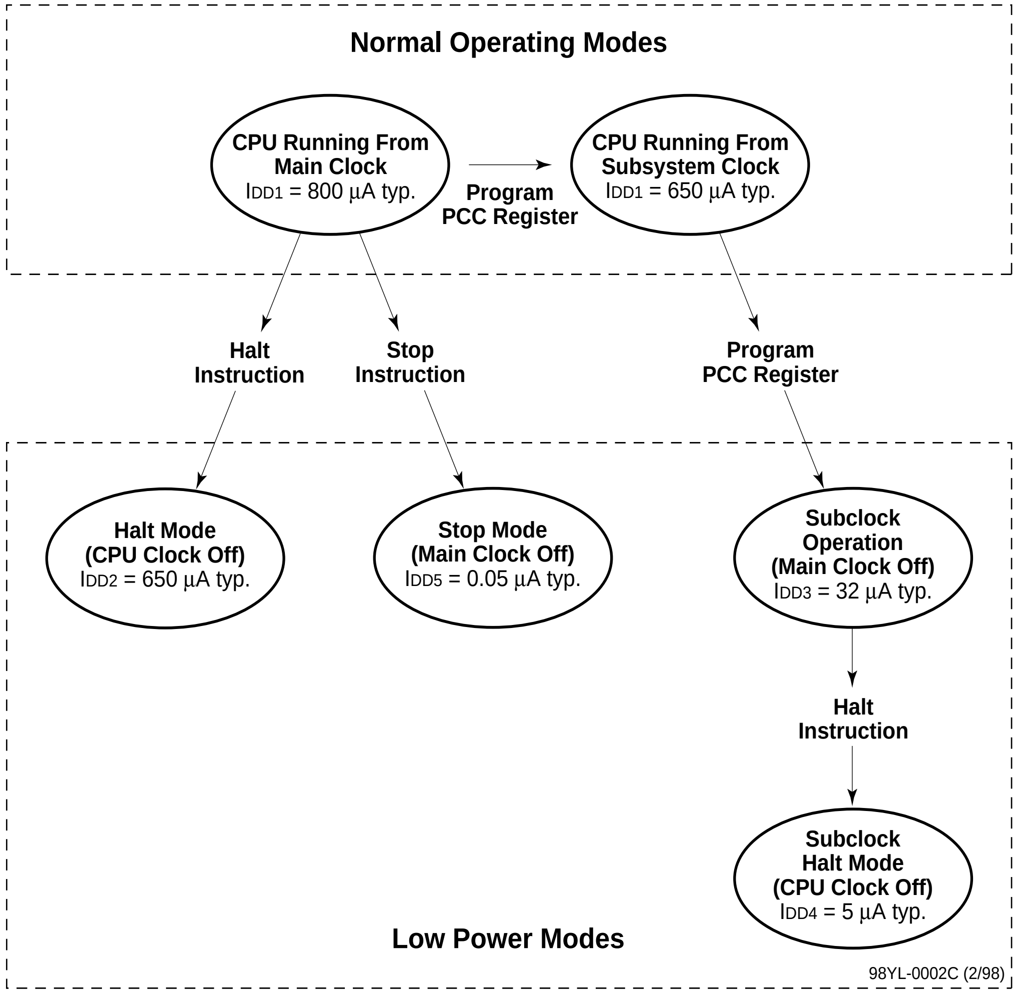
To further reduce power consumption and enhance functionality, NEC Electronics has integrated two independent clocks into its microcontrollers, the primary high-speed main clock and a secondary 32.768-kHz subsystem clock, each with its own set of standby modes. The main clock operates all parts of the microcontroller, while the subsystem clock is dedicated to the CPU, a real-time clock (also called a watch timer), and LCD controller (if present). NEC Electronics provides dual oscillators for applications needing the ability to operate from a low-speed, very low-power-consuming clock source and then switch to a high-speed clock for time-critical functions. Security systems, data acquisition systems, and real-time control systems are good examples of these applications.

The main clock structure has two standby modes: halt and stop. The addition of a subsystem clock offers two complementary modes: subclock operation and subclock halt. This document describes NEC Electronics' dual-clock architecture, illustrated in Table 1 and later in Figure 1, and how to use the clocks to optimize power consumption.

Table 1. Standby Functions in NEC Electronics Microcontrollers

Clock Source	Mode	Description
Main	Halt	CPU clock is OFF and peripheral clocks are ON
	Stop	Main clock is OFF and the rest of the microcontroller, except for the interrupt controller, is OFF
Subsystem	Subclock operation	Main clock is OFF; subsystem oscillator clocks the CPU, real-time clock, and an LCD controller, if present
	Subclock halt	Main clock is OFF and CPU clock halted (OFF); the subsystem clock pulses the real-time clock and an LCD controller, if present

Figure 1. Dual-Clock Implementation



Note: Operating current is shown for the µPD78064 at 3 volts and 5 MHz.

Tracking the Time of Day

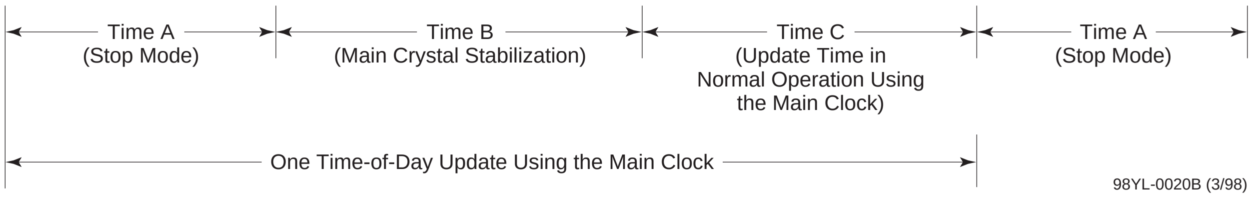
Some battery-powered applications require the microcontroller to keep track of the time of day, a good example of a low-power application. For this purpose, many NEC Electronics microcontrollers contain a watch timer (real-time clock) that updates the time of day every one-half second. The watch timer, driven by a 32.768-kHz subsystem clock, must be ON for the microcontroller to continually track the time of day in real time.

The CPU can use either the main clock or the subsystem clock to update the time of day. This application note explains both options and shows how subsystem clock operation uses less current than main clock operation.

Main Clock Operation

When the time of day is updated by the CPU using the main clock, executing a stop instruction puts the microcontroller into stop mode and turns OFF the main clock. The watch timer uses the subsystem clock and continues to run, and issues an interrupt every 0.5 second, at which time the main crystal is started and an interrupt service routine increases the time count by 0.5 second. After the time is updated, the microcontroller returns to stop mode and waits for the next watch timer interrupt (Figure 2).

Figure 2. ***Time-of-Day Update Using the Main Clock***



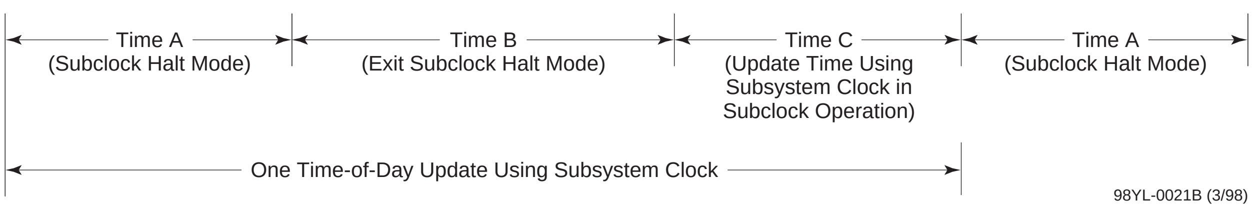
Subsystem Clock Operation

Alternatively, the time can be updated by the CPU using the subsystem clock. In this case, the microcontroller must be put into subclock halt mode using this procedure.

1. Switch to the subsystem clock by setting bit CSS of the Processor Clock Control (PCC) register to logic 1.
2. Verify that the CPU clock source is switched to the subsystem clock and that bit CLS of the PCC register is set to logic 1.
3. Enter subclock operation by shutting off the main oscillator and setting bit MCC of the PCC register to logic 1.
4. Enter subclock halt mode by executing a halt instruction.

When the watch timer generates a 0.5-second interrupt, the microcontroller exits subclock halt mode and enters subclock operation, whereby the subsystem clock is connected to the CPU and the CPU updates the time of day by 0.5 second. During this time, the microcontroller's main clock is OFF. After the time of day is updated, a halt instruction must be executed to disconnect the subsystem clock from the CPU. As shown in Figure 3, the microcontroller waits for the next watch timer interrupt to update the time of day. Note that because the subsystem clock is always oscillating, 'TIME B' is zero (0).

Figure 3. ***Time-of-Day Update Using the Subsystem Clock***



Example Operating Current Calculation

This section explains how to calculate operating current for NEC microcontrollers. The example uses data pertaining to the µPD78064 and is based on these assumptions:

1. It takes 500 CPU clock periods to update the time of day.
2. The main clock has a ceramic resonator with a stabilization time of 3 ms.
3. The oscillator stabilization time select (OSTS) register allows a range of different wait times to be selected to match the crystal resonator.
4. The oscillation stabilization wait time closest to 3 ms, 3.28 ms, is selected.
5. Maximum main clock frequency (fx) for the µPD78064 is 5.00 MHz.
6. The current specified in the data sheet = internal prescaler setting of fxx/16.
7. The CPU cycle period for fxx/16 = 3.2 µs.
8. Therefore, TIME C in Figure 2 is 500 x (3.2 µs) = 1.6 ms.

Table 2. ***µPD78064 Specifications***

		IDD at 3 Volts	
Mode	**Clocks**	**Typical**	**Maximum**
Normal operation	Main and subclocks ON	800 µA	2,400 µA
Halt mode	Main and subclocks ON; clock to CPU OFF	650 µA	1,950 µA
Stop mode	Main clock OFF	0.05 µA	10 µA
Subclock operation	Main clock OFF and subclock to CPU ON	32 µA	64 µA
Subclock halt	Main clock OFF and subclock to CPU OFF	5 µA	15 µA

Table 3 shows this information as it appears on the auto-calculate worksheet, a program especially developed by NEC Electronics America to calculate power consumption for NEC Electronics microcontrollers. The worksheet is available at www.necelam.com/docs/files/power_a1%204.xls.

Table 3. ***Example of Auto-Calculate Worksheet Data Input (µPD78064 at 3 Volts)***

		Input Parameters (mA)		
		Typical	Maximum	
Main clock operation	I_{DD1}	0.80	2.40	
Main clock halt mode	I_{DD2}	0.65	1.95	
Subclock operation	I_{DD3}	0.032	0.064	
Subclock operation at 2 volts	I_{DD3} at 2 volts	0.024	0.048	
Subclock halt mode	I_{DD4}	0.005	0.015	
Subclock halt mode at 2 volts	I_{DD4} at 2 volts	0.0025	0.0125	
Stop mode	I_{DD5}	0.00005	0.010	
CPU clock divided internally; main clock divided by		16	2	<-- Subclock divided by
Default oscillation frequency		5	3.28	<-- Osc. stabilization time
Enter number of CPU clocks here ===>		500		

Main Clock Operating Current

- Standby current = $I_{DD\ Total}$ = ($I_{DD\ TIME\ A}$) + ($I_{DD\ TIME\ B}$) + ($I_{DD\ TIME\ C}$), where halt mode current is noted as 'TIME B.' See Figure 2.

  TIME A + TIME B + TIME C = 500 ms
  TIME A = 500 ms – TIME B – TIME C
  TIME A = 500 ms – 3.28 ms – 1.60 ms = 495.12 ms

  % of TIME A = 495.12 ms / 500 ms x (100) = 99.02%
  % of TIME B = 3.28 ms / 500 ms x (100) = 0.66%
  % of TIME C = 1.60 ms / 500 ms x (100) = 0.32%

- Typical current = $I_{DD\ Typ}$ = ($I_{DD\ Typ\ TIME\ A}$) + ($I_{DD\ Typ\ TIME\ B}$) + ($I_{DD\ Typ\ TIME\ C}$)

  $I_{DD\ Typ}$ = 99.02% /100 x (0.005 µA) + 0.6% / 100 x (0.65 mA) + 0.32% / 100 x (0.8 mA)
  $I_{DD\ Typ}$ = 4.95 µA + 4.26 µA + 2.56 µA = <u>11.77 µA</u>

- Maximum current = $I_{DD\ Max}$ = ($I_{DD\ Max\ TIME\ A}$) + ($I_{DD\ Max\ TIME\ B}$) + ($I_{DD\ Max\ TIME\ C}$)

  $I_{DD\ Max}$ = 99.02% / 100 x (15 µA) + 0.66% / 100 x (1.95 mA) + 0.32% / 100 x (2.4 mA)
  $I_{DD\ Max}$ = 14.85 µA + 12.78 µA + 7.68 µA = <u>35.31 µA</u>

Subsystem Clock Operating Current

The CPU is inactive during halt mode. The interrupt generated by the watch timer activates the CPU, which is immediately connected to the subsystem clock. Since the subsystem crystal is always oscillating, 'TIME B' is equal to zero. See Figure 3. Therefore, the calculation formula is modified as follows:

- $I_{DD\ Total}$ = (I_{DD} TIME A) + ($I_{DD\ TIME\ B}$) + ($I_{DD\ TIME\ C}$) = (I_{DD} TIME A) + 0 + ($I_{DD\ TIME\ C}$) = (I_{DD} TIME A) + ($I_{DD\ TIME\ C}$)

- Cycle time = 2 x (1/32.768 kHz) = 61 µs

  TIME A + TIME C = 500 ms
  TIME C = 500 CPU clocks = 500 x (61 µs) = 31.52 ms
  TIME A = 500 ms – TIME C = 500 – 31.52 = 469.48 ms

  % of TIME A = 469.48 ms / 500 ms x (100) = 93.90%
  % of TIME B = 0%
  % of TIME C = 61 ms / 500 ms x (100) = 6.10%

- Typical current = $I_{DD\ Typ}$ = ($I_{DD\ Typ}$ TIME A) + ($I_{DD\ Typ}$ TIME C)

  $I_{DD\ Typ}$ = 93.90% / 100 x (2.5 µA) + 6.10% / 100 x (5 µA)
  $I_{DD\ Typ}$ = 4.69 µA + 1.95 µA = <u>6.65 µA</u>

- Maximum current = $I_{DD\ Max}$ = ($I_{DD\ Max}$ TIME A) + ($I_{DD\ Max}$ TIME C)

  $I_{DD\ Max}$ = 93.90% / 100 x (12.5 µA) + 6.10% / 100 x (15 µA)
  $I_{DD\ Max}$ = 14.08 µA + 3.91 µA = <u>17.99 µA</u>

Both the typical and maximum operating currents using the subsystem clock are less than those using the main system clock.

Table 4 shows how this information is displayed on the auto-calculate worksheet.

Table 4. Current Consumption Results From the Auto-Calculate Worksheet

		Main System Clock		Subsystem Clock	
		5,000,000 Hz		32,768 kHz	
Subroutine execution time interval	ms	500.00		500.00	
Oscillation stabilization time	ms	4.00		Subclock on	
Min. CPU cycle time (set by PCC)	μs	3.20		61.04	
Clock cycles needed by subroutine	each	500.00		500.00	
Time required by subroutine	ms	1.60		30.52	
Overall sleep time	ms	494.40		469.48	
		Main System Clock		Subsystem Clock	
Sleep time (subhalt mode)	%	98.88		93.90	
Oscillation stabilization time	%	0.80		0.00	
Microcontroller is active	%	0.32		6.10	
Total	%	100.00		100.00	
		Typ.	Max.	Typ.	Max.
Sleep time (subhalt mode)	μA	4.94	14.83	4.69	14.08
Oscillation stabilization time	μA	5.20	15.60	0.00	0.00
Microcontroller active	μA	2.56	7.68	1.95	3.91
Total	μA	12.70	38.11	6.65	17.99

Effect of Crystal Stabilization Time on I_{DD}

The resonator's stabilization time has a profound effect on overall input current consumption. The original calculation example used a resonator stabilization time of 3.28 ms. The calculations shown in Table 5 are based on a stabilization time of 10 ms and stabilization wait time of 13.1 ms, resulting in a substantial increase in power consumption. As demonstrated here, the shorter the oscillation stabilization time, the lower the overall power consumption. The subsystem clock is clearly advantageous over the fastest (and most expensive) crystal resonators.

Table 5. Oscillation Stabilization Times Versus Subsystem Clock at 500 Cycles

Power Consumption	Main Clock		Subclock
	3.28 ms	13.1 ms	
I_{DD} Typical (mA)	11.77	24.45	6.65
I_{DD} Maximum (mA)	35.31	73.36	17.99

Effect of CPU Cycles on I_{DD}

The larger the current ratio of main clock oscillation to subsystem clock operation, the better the power savings. To prove this hypothesis, Table 6 compares power consumption at 250, 500, and 5000 cycles.

Table 6. Power Consumption Rates

Power Consumption	250 Cycles		500 Cycles		5000 Cycles	
	Main	Subclock	Main	Subclock	Main	Subclock
Typical current (mA)	10.50	5.82	11.77	6.65	34.67	21.48
Maximum current (mA)	31.50	16.50	35.31	17.99	104.00	44.91

The subsystem clock clearly uses less power than the main clock. A shorter program also lowers the average input current.

It is important to pay attention to the maximum number of clock cycles required by a program, because the subsystem clock is much slower than the main clock and processing power is reduced when the subsystem clock is used. If the program is too long, it may be impossible to run it using the subsystem clock. This problem could be solved, however, by optimizing the code.

Another potential problem arises when attempting to execute a very long subroutine. Therefore, with some specific microcontrollers, the power savings becomes negligible as the number of CPU cycles increases. Always perform this initial calculation in order to design a power-optimized system.

Note: NEC Electronics onstantly improves power consumption reduction techniques, and newest microcontrollers have more optimized main clock operation.

Auto-Calculate Worksheet

The auto-calculate worksheet in Table 7 demonstrates the results obtained for the μPD78064 applying the same assumptions used in the earlier calculations. The auto-calculate program is available at www.necelam.com/docs/files/power_a1%204.xls. To use the program, you must have Microsoft® Excel™ software installed on your personal computer.

1. Read the instructions on page 1 of worksheet 1.
2. From the **Tools** menu, click **Macro**.
3. Scroll through the list of part numbers and select one.
4. Click **Run**.

The program automatically calculates current consumption and displays typical and maximum specifications, as shown in Table 7.

Table 7. Auto-Worksheet Calculations for the μPD78064 at 3 Volts

			Input Parameters (mA)		
			Typ.	Max.	
Main clock operation	I_{DD1}		0.80	2.40	
Main clock halt mode	I_{DD2}		0.65	1.95	
Subclock operation	I_{DD3}		0.032	0.064	
Subclock operation at 2 volts	I_{DD3} (2V)		0.024	0.048	
Subclock halt mode	I_{DD4}		0.005	0.015	
Subclock halt mode at 2 volts	I_{DD4} (2V)		0.0025	0.0125	
Stop mode	I_{DD5}		0.00005	0.010	
CPU clock is divided internally; main divided by			16	2	<--Subclock divided by
Default oscillation frequency			5	3.28	<--osc. stabilization time
Enter the number of CPU clock cycles here = = =>			500		
		Main Clock (5,000,000 Hz)		Subsystem Clock (32,768 kHz)	
Subroutine execution time interval	ms	500.00		500.00	
Oscillation stabilization time	ms	3.28		Subclock on	
Mininimum CPU cycle time (set by PCC)	μs	3.20		61.04	
Clock cycles required by subroutine	each	500.00		500.00	
Time required by subroutine	ms	1.60		30.52	
Overall sleep time	ms	495.12		469.48	
Sleep time (subhalt mode)	%	99.02		93.90	
Oscillation stabilization time	%	0.66		0.00	
Microcontroller active	%	0.32		6.10	
Total	%	100.00		1000.00	
		Typ.	Max.	Typ.	Max.
Sleep time (subhalt mode)	μA	4.95	14.85	4.69	14.08
Oscillation stabilization time	μA	4.26	12.78	0.00	0.00
Microcontroller active	μA	2.56	7.68	1.95	3.91
Total	μA	11.77	35.31	6.65	17.99
Power savings SUB versus main clock	%			43.52	49.05
Scratch pad. Enter the comparison data below.					
Compared to CPU clock cycles (μA)	250	10.50	31.50	5.82	16.50
Compared to CPU clock cycles (%)	250			44.53	47.63

A given device's default parameters are entered into the worksheet when you select that device. You will need to vary the number of instructions to find the optimum current. Also, the program makes assumptions about voltage, CPU clock division rate, oscillator frequency, and number of CPU cycles. You can only modify the number of CPU cycles for a specific application in order to generate the best power consumption model.

Power consumption information for each part in the matrix was checked at the time of this document's publication. To verify accuracy, please refer to the latest published data sheet for your specific device. For any device not shown in the matrix, please request the latest data from NEC Electronics and manually enter those numbers in the Input Parameters box. The worksheet program will calculate the results, but without saving the data in the password-protected data base.

NEC

For literature, call **1-800-366-9782** 7 a.m. to 6 p.m. Pacific time
or fax your request to **1-800-729-9288**
or visit our web site at **www.necelam.com**

NEC Electronics America, Inc.

CORPORATE HEADQUARTERS
2880 Scott Boulevard
Santa Clara, CA 95050-2554

TEL 408-588-6000

U13274EU3V0AN00