Introduction
This application note provides reference information to help determine the number of cycles (latency) the RX600 Series CPU takes to access an I/O register. It summarizes some key points to consider in identifying that latency.

Target Device
RX600 Series

Contents

1. Overview ........................................................................................................................................... 2
2. Number of I/O Register Access Cycles............................................................................................. 2
3. Varying Number of I/O Register Access Cycles Depending on Clock Settings............................ 5
4. Reference Documents....................................................................................................................... 9
1. Overview

- The number of cycles (latency) from the time the RX600 Series CPU executes an I/O register access instruction to the time a value is stored in the I/O register depends on:
  - Number of bus cycles on internal main bus 1
  - Number of cycles for divided-frequency clock synchronization
  - Number of bus cycles on the internal peripheral bus

- This application note describes bus operation which takes place when the RX600 Series CPU accesses an I/O register.

- This application note also describes the operation from the standpoint of an I/O register. Thus, this is different from the CPU pipeline operation.

2. Number of I/O Register Access Cycles

The number of cycles required for access to an I/O register is calculated as follows (*1).

\[
\text{Number of I/O register access cycles} = \text{Number of bus cycles on internal main bus 1} + \text{Number of cycles for frequency-divided clock synchronization} + \text{Number of bus cycles on the internal peripheral bus} \quad (*2)
\]

The number of cycles on the internal peripheral bus differs for each I/O register which is accessed. When a connected peripheral function is accessed via the internal peripheral bus synchronized with the peripheral module clock (PCLK) or external bus clock (BCLK), cycles for frequency-divided clock synchronization are added.

The number of cycles for frequency-divided clock synchronization depends on the frequency ratio between the system clock (ICLK) and PCLK (or BCLK) and on the bus access timing. It is equal to the ICLK cycle count multiplied by a number from 0 to 7.

To find the number of I/O register access cycles for each register, refer to the "List of I/O Register Addresses (in Numerical Order)" in the hardware manual of the relevant product.

Notes:

*1 The number of cycles when the CPU accesses the register without fetching an instruction for external memory or competing with a different bus master (DMAC or DTC) for bus access.

*2 There are six types of internal peripheral buses. Internal peripheral buses to be used and peripheral devices to be connected vary for each RX600 Series product. For the bus specifications, refer to the "Bus Specifications" in the hardware manual of the relevant product.
2.1 Number of Bus Cycles on Internal Main Bus 1

The CPU uses internal main bus 1. The CPU requires one ICLK clock cycle to access an I/O register via internal main bus 1.

2.2 Number of Cycles for Frequency-divided Clock Synchronization

The CPU accesses an I/O register via internal main bus 1. The I/O registers operate with different reference clocks. For example, PCLK is a reference clock for the peripheral functions such as the serial communication interface (SCI) and compare match timer (CMT) while BCLK is a reference clock for the bus (BSC). On the other hand, ICLK is a reference clock for both the data transfer controller (DTC) and interrupt controller (ICU). This reference clock is for internal main bus 1 as well.

ICLK, PCLK and BCLK have different frequencies, and the RX600 Series can select any one of these clocks. If PCLK is configured to have a lower frequency than ICLK (or if BCLK is configured to have a lower frequency than ICLK), cycles for frequency-divided clock synchronization occur. Figure 1 presents an outline of these cycles. This figure assumes that:

- \[ PCLK = \frac{ICLK}{4} \]
- Number of bus cycles on the internal peripheral bus = 1 PCLK

![Diagram of Cycles for Frequency-divided Clock Synchronization](image)

Figure 1 Outline of Cycles for Frequency-divided Clock Synchronization

In case 1 above, the CPU accesses the I/O register at the start of the PCLK cycle. In this case, the number of cycles for frequency-divided clock synchronization is equal to the ICLK cycle count multiplied by 3. In case 4, the CPU accesses the I/O register at the end of the PCLK cycle. In this case, the number of cycles for frequency-divided clock synchronization is equal to the ICLK cycle count multiplied by 0.

Accordingly, the number of cycles for frequency-divided clock synchronization depends on which point in PCLK cycle time I/O register access takes place.

The explanation above is based on the internal peripheral bus which uses PCLK as the reference clock. The same applies to the internal peripheral bus which uses BCLK as the reference clock.
Table 1 shows the relationship between the I/O register access timing and the number of cycles for frequency-divided clock synchronization.

Table 1  Relationship between the I/O Register Access Timing and the Number of Cycles for Frequency-divided Clock Synchronization

<table>
<thead>
<tr>
<th>I/O register access timing</th>
<th>Number of frequency-divided clock synchronization states</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start of PCLK</td>
<td>3*ICLK</td>
</tr>
<tr>
<td>Start of PCLK + 1*ICLK</td>
<td>2*ICLK</td>
</tr>
<tr>
<td>Start of PCLK + 2*ICLK</td>
<td>1*ICLK</td>
</tr>
<tr>
<td>End of PCLK</td>
<td>0*ICLK</td>
</tr>
</tbody>
</table>

Note: This table is based on the assumption that PCLK = ICLK/4 and the number of bus cycles on the internal peripheral bus = 1.

2.3   Number of Internal-Peripheral-Bus Clocks

The number of internal-peripheral-bus clocks refers to the number of access cycles required for each I/O register. This number varies for each peripheral function.

This number is the same as the number of access cycles shown in the "List of I/O Register Addresses (in Numerical Order)" in the hardware manual of the relevant product.
3. Varying Number of I/O Register Access Cycles Depending on Clock Settings

Whether the reference clock for the peripheral bus is PCLK or BCLK, the concept of the I/O register access cycle count is the same. So, this application note describes the number of I/O register access cycles for the peripheral bus which uses the reference clock, PCLK, as an example.

Section 2.2 describes an example of using PCLK = ICLK/4. Below is an explanation concerning the number of I/O register access cycles which depends on the ICLK and PCLK settings.

As the equation in Section 2 shows, the number of I/O register access cycles is the total number of cycles on internal main bus 1, cycles for frequency-divided clock synchronization and cycles on the internal peripheral bus. Because the number of cycles on internal main bus 1 and that of cycles on the internal peripheral bus are fixed, the number of I/O register access cycles depends on the number of cycles for frequency-divided clock synchronization.

The number of cycles for frequency-divided clock synchronization depends on the time the CPU accesses an I/O register and on the frequency ratio between ICLK and PCLK.

Table 2 shows the relationship between the ICLK and PCLK settings and the number of cycles for frequency-divided clock synchronization.

Table 2  Relationship between the ICLK and PCLK Settings and the Number of Cycles for Frequency-divided Clock Synchronization (ICLK ≥ PCLK)

<table>
<thead>
<tr>
<th>ICLK</th>
<th>PCLK</th>
<th>Meaning</th>
<th>Number of Cycles for Frequency-Divided Clock Synchronization</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTAL x 1</td>
<td>EXTAL x 1</td>
<td>PCLK = ICLK</td>
<td>0</td>
</tr>
<tr>
<td>EXTAL x 2</td>
<td>EXTAL x 1</td>
<td>PCLK = ICLK/2</td>
<td>0 to 1*ICLK</td>
</tr>
<tr>
<td>EXTAL x 2</td>
<td>EXTAL x 2</td>
<td>PCLK = ICLK</td>
<td>0</td>
</tr>
<tr>
<td>EXTAL x 4</td>
<td>EXTAL x 1</td>
<td>PCLK = ICLK/4</td>
<td>0 to 3*ICLK</td>
</tr>
<tr>
<td>EXTAL x 4</td>
<td>EXTAL x 2</td>
<td>PCLK = ICLK/2</td>
<td>0 to 1*ICLK</td>
</tr>
<tr>
<td>EXTAL x 4</td>
<td>EXTAL x 4</td>
<td>PCLK = ICLK</td>
<td>0</td>
</tr>
<tr>
<td>EXTAL x 8</td>
<td>EXTAL x 1</td>
<td>PCLK = ICLK/8</td>
<td>0 to 7*ICLK</td>
</tr>
<tr>
<td>EXTAL x 8</td>
<td>EXTAL x 2</td>
<td>PCLK = ICLK/4</td>
<td>0 to 3*ICLK</td>
</tr>
<tr>
<td>EXTAL x 8</td>
<td>EXTAL x 4</td>
<td>PCLK = ICLK/2</td>
<td>0 to 1*ICLK</td>
</tr>
<tr>
<td>EXTAL x 8</td>
<td>EXTAL x 8</td>
<td>PCLK = ICLK</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: If ICLK < PCLK, the number of cycles for frequency-divided clock synchronization is 0*ICLK.

The following describes the number of cycles for frequency-divided clock synchronization when PCLK = ICLK, PCLK = ICLK/2, PCLK = ICLK/4 and PCLK = ICLK/8. The explanation below also applies to the case where BCLK = ICLK, BCLK = ICLK/2, BCLK = ICLK/4 and BCLK = ICLK/8.
3.1 Number of Cycles for Frequency-Divided Clock Synchronization When PCLK = ICLK

Figure 2 shows the number of cycles for frequency-divided clock synchronization when PCLK = ICLK.

This number is equal to the ICLK cycle count multiplied by 0. Cycles for frequency-divided clock synchronization occur when the ICLK and PCLK frequencies are different.

![Figure 2 Number of Cycles for Frequency-divided Clock Synchronization When PCLK = ICLK](image)

3.2 Number of Cycles for Frequency-Divided Clock Synchronization When PCLK = ICLK/2

Figure 3 shows the number of cycles for frequency-divided clock synchronization when PCLK = ICLK/2.

This number is equal to the ICLK cycle count multiplied by 0 or 1.

![Figure 3 Number of Cycles for Frequency-divided Clock Synchronization When PCLK = ICLK/2](image)
3.3 Number of Cycles for Frequency-Divided Clock Synchronization When PCLK = ICLK/4

Figure 4 shows the number of cycles for frequency-divided clock synchronization when PCLK = ICLK/4.

This number is equal to the ICLK cycle count multiplied by a number from 0 to 3.

![Diagram showing number of cycles for frequency-divided clock synchronization](image)

**Figure 4 Number of Cycles for Frequency-divided Clock Synchronization When PCLK = ICLK/4**
3.4 Number of Cycles for Frequency-Divided Clock Synchronization When PCLK = ICLK/8

Figure 5 shows the number of cycles for frequency-divided clock synchronization when PCLK = ICLK/8.

This number is equal to the ICLK cycle count multiplied by a number from 0 to 7.
4. Reference Documents

- Hardware Manuals
  RX610 Group Hardware Manual (REJ09B0460)
  (The most up-to-date versions of the documents are available on the Renesas Electronics Website.)

- User's Manuals
  RX62N Group, RX621 Group User's Manual: Hardware (R01UH0033EJ)
  (The most up-to-date versions of the documents are available on the Renesas Electronics Website.)

Website and Support

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

Inquiries
  http://www.renesas.com/inquiry

All trademarks and registered trademarks are the property of their respective owners.
# Revision Record

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
<th>Page</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Dec 17, 2010</td>
<td>—</td>
<td>—</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>
General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   ← The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   ← The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   ← The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable.
   When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   ← When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal.
   Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products
   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.
   ← The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.