

致尊敬的顾客

---

## 关于产品目录等资料中的旧公司名称

---

NEC电子公司与株式会社瑞萨科技于2010年4月1日进行业务整合（合并），整合后的新公司暨“瑞萨电子公司”继承两家公司的所有业务。因此，本资料中虽还保留有旧公司名称等标识，但是并不妨碍本资料的有效性，敬请谅解。

瑞萨电子公司网址：<http://www.renesas.com>

2010年4月1日  
瑞萨电子公司

【发行】瑞萨电子公司（<http://www.renesas.com>）

【业务咨询】<http://www.renesas.com/inquiry>

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

# 740 群

瑞萨 8 位单片机

## Cautions

### Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.  
Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

### Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.  
The information described here may contain technical inaccuracies or typographical errors.  
Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.  
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.  
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

## 注意

本文只是参考译文，前页所载英文版“Cautions”具有正式效力。

### 请遵循安全第一进行电路设计

1. 虽然瑞萨科技尽力提高半导体产品的质量和可靠性，但是半导体产品也可能发生故障。半导体的故障可能导致人身伤害、火灾事故以及财产损害。在电路设计时，请充分考虑安全性，采用合适的如冗余设计、利用非易燃材料以及故障或者事故防止等的安全设计方法。

### 关于利用本资料时的注意事项

1. 本资料是为了让用户根据用途选择合适的瑞萨科技产品的参考资料，不转让属于瑞萨科技或者第三者所有的知识产权和其它权利的许可。
2. 对于因使用本资料所记载的产品数据、图、表、程序、算法以及其它应用电路的例子而引起的损害或者对第三者的权力的侵犯，瑞萨科技不承担责任。
3. 本资料所记载的产品数据、图、表、程序、算法以及其它所有信息均为本资料发行时的信息，由于改进产品或者其它原因，本资料记载的信息可能变动，恕不另行通知。在购买本资料所记载的产品时，请预先向瑞萨科技或者经授权的瑞萨科技产品经销商确认最新信息。  
本资料所记载的信息可能存在技术不准确或者印刷错误。因这些错误而引起的损害、责任问题或者其它损失，瑞萨科技不承担责任。  
同时也请通过各种方式注意瑞萨科技公布的信息，包括瑞萨科技半导体网站。  
(<http://www.renesas.com>)
4. 在使用本资料所记载部分或者全部数据、图、表、程序以及算法等信息时，在最终做出有关信息和产品是否适用的判断前，务必对作为整个系统的所有信息进行评价。由于本资料所记载的信息而引起的损害、责任问题或者其它损失，瑞萨科技不承担责任。
5. 瑞萨科技的半导体产品不是为在可能和人命相关的环境下使用的设备或者系统而设计和制造的产品。在研讨将本资料所记载的产品用于运输、交通车辆、医疗、航空宇宙用、原子能控制、海底中继器的设备或者系统等特殊用途时，请与瑞萨科技或者经授权的瑞萨产品经销商联系。
6. 未经瑞萨科技的书面许可，不得翻印或者复制全部或者部分资料的内容。
7. 如果本资料所记载的某产品或者技术内容受日本出口管理限制，必须在得到日本政府的有关部门许可后才能出口，并且不准进口到批准目的地国家以外的国家。  
禁止违反日本和（或者）目的地国家的出口管理法和法规的任何转卖、挪用或者再出口。
8. 如果需要了解本资料所记载的信息或者产品的详细，请与瑞萨科技联系。



---

# 目 录

---

第 1 章 概 要 .....	1
第 2 章 中央处理器 .....	3
2.1 累加器 (A) .....	4
2.2 变址寄存器X (X)、变址寄存器Y (Y) .....	4
2.3 栈指针 (S) .....	4
2.4 程序计数器 (PC) .....	4
2.5 处理器状态寄存器 (PS) .....	5
第 3 章 指令说明 .....	9
3.1 寻址方式 (Addressing mode) .....	9
3.2 指令系统.....	29
3.2.1 数据传送指令 .....	29
3.2.2 运算指令 .....	30
3.2.3 位操作指令 .....	31
3.2.4 标志设定指令 .....	31
3.2.5 跳转、转移和返回指令 .....	32
3.2.6 中断指令 .....	33
3.2.7 特殊指令 .....	33
3.2.8 其它指令 .....	33
3.3 指令的说明.....	34
3.4 中断处理和子程序处理的指令 .....	106
3.4.1 中断处理指令 .....	106
3.4.2 中断控制指令 .....	106
3.4.3 子程序处理指令 .....	106
第 4 章 使用注意事项 .....	109
4.1 输入/输出端口的注意事项.....	109
4.1.1 在待机状态的使用 .....	109
4.1.2 通过位处理指令改写输出数据 .....	109
4.2 未使用管脚处理的注意事项.....	110
4.2.1 未使用管脚的适当处理 .....	110
4.2.2 处理时的注意事项 .....	110
4.3 中断的注意事项.....	111
4.3.1 中断请求位和允许位的设定 .....	111
4.3.2 更改关联寄存器的设定 .....	111
4.3.3 中断请求位的判断 .....	112
4.4 编程的注意事项.....	112
4.4.1 处理器状态寄存器 .....	112
4.4.2 BRK 指令.....	113
4.4.3 10 进制运算.....	114
4.4.4 JMP 指令.....	114
4.4.5 乘除指令.....	114

4.4.6	端口 .....	114
4.4.7	指令的执行时间 .....	114
附录 1	按寻址方式的执行时序 .....	115
附录 2	740 族机器指令一览表 .....	181
附录 3	740 族指令码对应表 .....	187

---

## 第1章 概 要

---

CMOS 8 位单片机 740 族指令的特点概括如下：

- ①高效率的指令群和丰富的寻址方式，能有效使用 ROM 区
- ②能对累加器、存储器、I/O 进行处理的位操作指令和位测试/转移指令
- ③丰富的中断源和处理功能
- ④对于字节单位处理和表参照功能，具有卓绝的变址寻址功能
- ⑤不需软件补正的 10 进制运算功能
- ⑥不经累加器就能进行的存储器和存储器之间、I/O 和 I/O 之间、存储器和 I/O 之间的运算功能



## 第2章 中央处理器

740 族的 CPU（中央处理器）有 6 个寄存器，如图 2.1 所示。

程序计数器（PC）由 16 位构成，但累加器（A）、变址寄存器 X（X）、变址寄存器 Y（Y）、栈指针（S）及处理器状态寄存器（PS）都由 8 位构成。

在硬件复位后，标志位 I 以外的这些寄存器的内容是不定，所以需要程序进行初始化（复位后 I=1）。

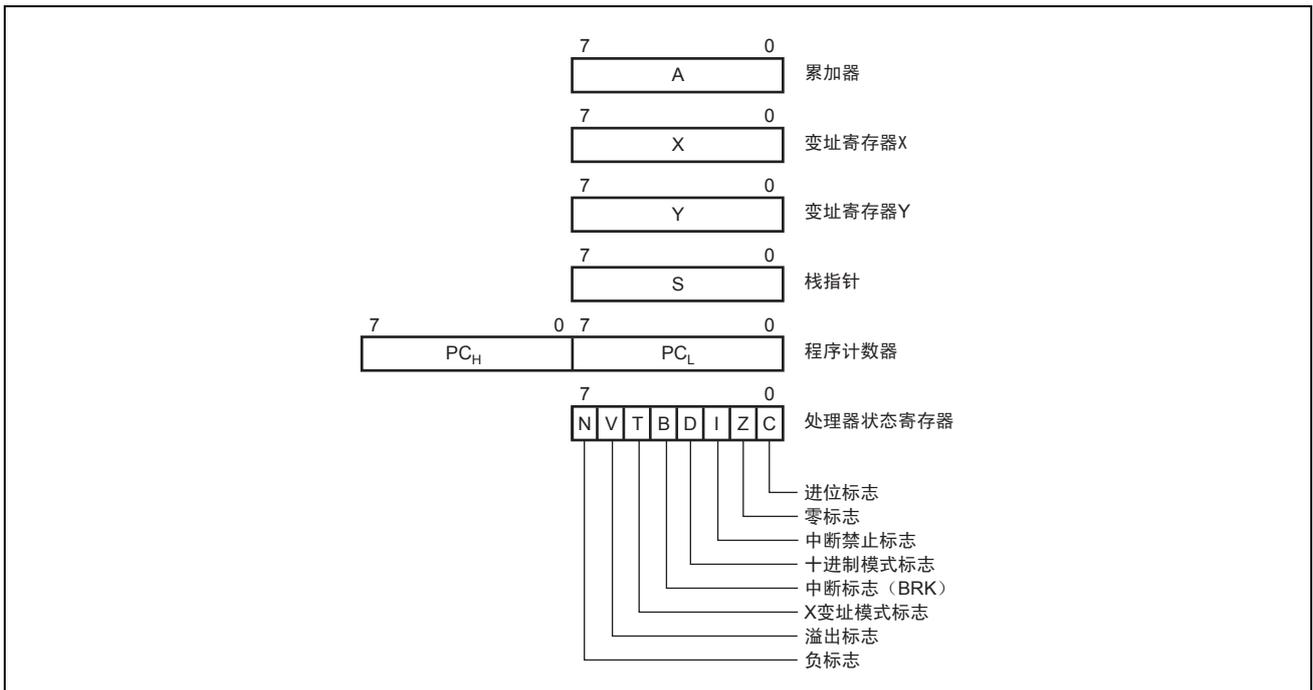


图2.1 740族的寄存器结构图

## 2.1 累加器 (A)

累加器是单片机的核心寄存器，由 8 位构成。它是十分频繁地用于算术运算、数据传送，暂时存储以及条件判断等的通用寄存器。

## 2.2 变址寄存器 X (X)、变址寄存器 Y (Y)

740 族的 CPU 具有 8 位变址寄存器 X 和 8 位变址寄存器 Y。

在使用这些变址寄存器的寻址方式时，存取操作数指定的地址加上变址寄存器内容后的地址。此模式在访问子程序表和访问存储表方面非常有效。

变址寄存器还具有递增、递减、比较以及数据传送等功能，因此能用作简单的累加器。

## 2.3 栈指针 (S)

栈指针 (S) 是 8 位寄存器。

在产生中断和调用子程序时，栈指针指向保存处理结束时的返回地址（程序计数器值）等的存储器地址。

有关中断处理和子程序处理时的栈指针 (S) 的变化，请参照“3.4 有关中断处理和子程序处理的指令”。

## 2.4 程序计数器 (PC)

程序计数器是由 PCH 和 PCL 组成的 16 位计数器，PCH 和 PCL 分别为 8 位。

程序计数器指向下一个保存执行指令的地址。

740 族的 CPU 采用存储程序方式，所以在开始新的运行时需要把指令和相关的数据从存储器传送到 CPU。

通常，程序计数器指向下一个要传送的存储器地址，因此在执行完一条指令后能读取下一条所需的指令。

740 族的程序计数器几乎自动控制。然而，在使用栈指针或直接更改程序计数器内容时，必须防止程序流程与程序计数器内容的不一致。

## 2.5 处理器状态寄存器 (PS)

处理器状态寄存器是 8 位寄存器，由表示内部 CPU 运算状态的 5 个标志和决定运行的 3 个标志构成。除了中断禁止标志为“1”以外，复位后值不定。因此，需要初始化影响程序执行的标志，尤其是对于影响运算的 T 标志和 D 标志，必须初始化。

各标志的说明如下。各标志的置位和清除指令一览表如表 2.1。有关这些标志变化的详细内容，请参照“3.3 指令的说明”或者“附录 2 740 族机器指令一览表”。

### [进位标志 C] \_\_\_\_\_ 位 0

本标志保持运算处理后来自算术逻辑单元的进位或借位。移位指令或循环指令也会改变此位的内容。本标志由 SEC 指令置位，由 CLC 指令清除。

### [零标志 Z] \_\_\_\_\_ 位 1

本标志在运算或数据传送结果为“0”时被置“1”，不为“0”被清零。

### [中断禁止标志 I] \_\_\_\_\_ 位 2

本标志是禁止中断的标志。本标志为“1”时禁止所有中断。在接受中断的同时变为“1”。本标志由 SEI 指令置位，由 CLI 指令清除。

### [十进制模式标志 D] \_\_\_\_\_ 位 3

本标志决定是用二进制还是用十进制进行加减运算。标志为“0”时进行通常的二进制运算，标志为“1”时进行 1 个字节为 2 位的十进制运算。此时自动进行十进制校正。

本标志由 SED 指令置位，由 CLD 指令清除。

只有 ADC 与 SBC 指令进行十进制运算。必须注意：通过这些指令进行十进制运算时 N、V、Z 的 3 个标志无效。

### [中断标志 (BRK) B] \_\_\_\_\_ 位 4

本标志是用于判断是否由 BRK 指令产生中断的标志。由 BRK 指令产生中断时，标志 B 自动置“1”，对于其它中断，标志置“0”并保持在堆栈中。

[X 变址模式标志 T] \_\_\_\_\_ 位 5

在运算时，本标志决定是通过累加器进行运算还是在存储器和存储器之间直接进行运算。标志为“0”时在累加器和存储器之间进行运算，为“1”时在存储器和存储器之间直接进行运算。

本标志由 SET 指令设置，由 CLT 指令清除。

①标志 T=0 时

$$A \leftarrow A * M_2$$

\* : 表示运算

A : 累加器内容

M<sub>2</sub> : 由运算操作的寻址方式指定的存储器内容

②标志 T=1 时

$$M_1 \leftarrow M_1 * M_2$$

\* : 表示运算

M<sub>1</sub> : 直接由变址寄存器 X 指定的存储器内容

M<sub>2</sub> : 由运算操作的寻址方式指定的存储器内容

## [溢出标志 V] \_\_\_\_\_ 位6

带符号运算结果出现溢出时，本标志置“1”。当加减运算结果超过+127（7F<sub>16</sub>）或| -128（80<sub>16</sub>）|时发生溢出。

溢出标志由CLV指令清除，没有置位指令。

另外，在执行了BIT指令时，BIT指令执行对象的存储器的位6存入此标志。

在加减运算结果小于等于上述数值时或者在不同符号的值相加时不发生溢出。

## [负标志 N] \_\_\_\_\_ 位7

本标志被设置与数据操作或者运算处理的结果的符号位（位7）相一致的内容。在执行了BIT指令时，被BIT指令执行的存储器的位7存入此标志。通过本标志，除了能判断运算结果是正还是负，还能进行简单的位测试。

表2.1 处理器状态寄存器各标志的置位和清除指令

	C标志	Z标志	I标志	D标志	B标志	T标志	V标志	N标志
置“1”指令	SEC	—	SEI	SED	—	SET	—	—
清“0”指令	CLC	—	CLI	CLD	—	CLT	CLV	—



## 第3章 指令说明

### 3.1 寻址方式 (Addressing mode)

740 族有 19 种寻址方式和很强的存储器存取能力。

在通过指令进行运算时，为了从存储器取出需要的数据或者将运算结果存入存储器，必须预先指定该存储器的地址。另外，在程序执行中要跳转到某个地址时也需要指定转移地址。此存储器地址的指定方法称为寻址。

寻址所需的数据和寄存器叙述如下。如图 3.1 所示，740 族的指令可以根据程序存储器中的字节数对 740 族指令分为 3 种：1 字节指令、2 字节指令和 3 字节指令。无论那种指令，第一字节称为操作码 (Operation Code)，构成指令的基础；第二或第三字节称为操作数 (Operand)，是影响寻址的部分。变址寄存器 X 与 Y 的内容也影响寻址。

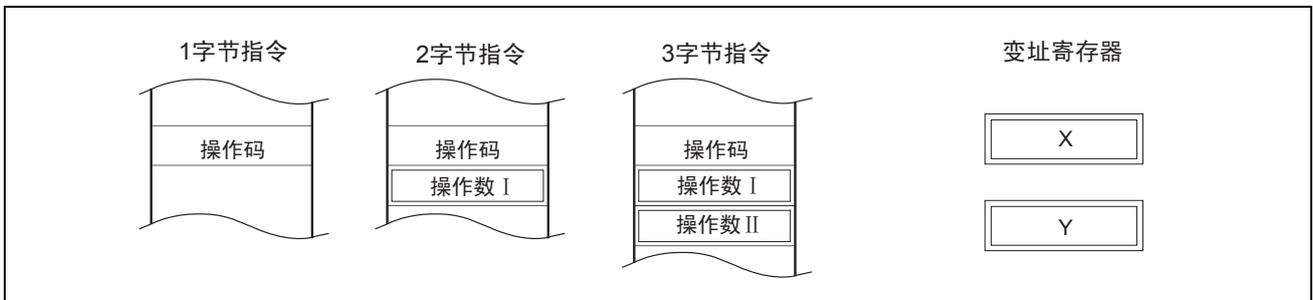


图3.1 指令的字节构成

无论有多少种寻址方式，从“指定特定的存储单元”的角度来看，进行的工作完全相同。所不同的是用操作数还是用变址寄存器的内容或者用两者的组合，指定存储器或者跳转目标。以此 3 种方法为基础，通过位操作指令、跳转指令和算术指令的组合，增大了选择范围，提高了 740 族的操作性能。由于在 1 字节指令时指定累加器或寄存器，所以不存在指定存储器的“操作数”。

## 立即(Immediate)

名称：立即寻址方式

功能：将操作数的内容指定为运算数据。

指令：ADC、AND、CMP、CPX、CPY、EOR、LDA、LDX、LDY、ORA、SBC

例：助记符

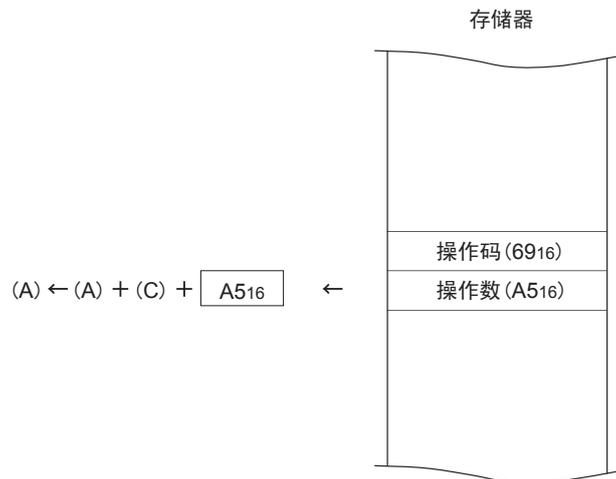
ADC # \$A5

↑

此符号 (#) 表示“为立即寻址方式”

机器码：

69<sub>16</sub> A5<sub>16</sub>



## 累加器(Accumulator)

名称：累加器 寻址方式

功能：直接将累加器的内容指定为运算数据。

指令：ASL、DEC、INC、LSR、ROL、ROR

例：助记符

ROL A

机器码：

2A16



## 零页(Zero page)

名称：零页 寻址方式

功能：将操作数指向的零页区内（地址  $00_{16} \sim FF_{16}$ ）的存储器内容指定为运算数据。

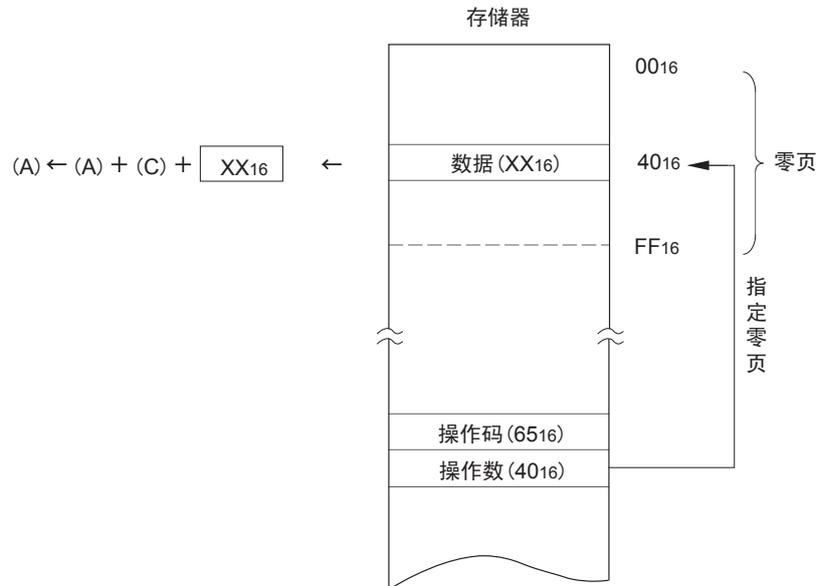
指令：ADC、AND、ASL、BIT、CMP、COM、CPX、CPY、DEC、EOR、INC、LDA、LDM、LDX、LDY、LSR、ORA、ROL、ROR、RRF、SBC、STA、STX、STY、TST

例：助记符

ADC \$40

机器码：

65<sub>16</sub> 40<sub>16</sub>



## 零页 X(Zero page X)

名称：零页 X 寻址方式

功能：将操作数内容和变址寄存器 X 内容的相加值所指向的零页区内的存储器内容指定为运算数据。

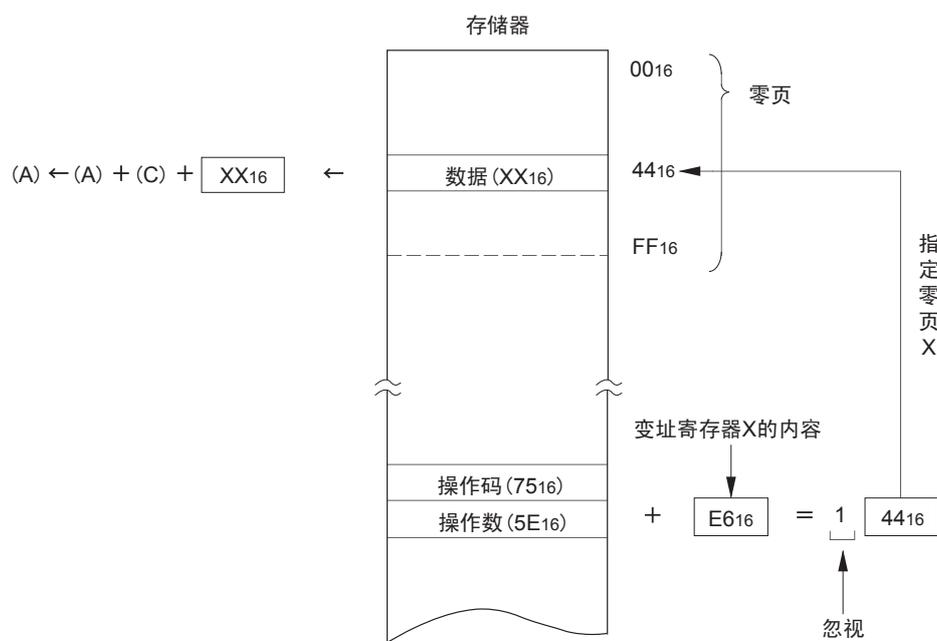
指令：ADC、AND、ASL、CMP、DEC、DIV、EOR、INC、LDA、LDY、LSR、MUL、ORA、ROL、ROR、SBC、STA、STY

例：助记符

ADC \$5E, X

机器码：

75<sub>16</sub> 5E<sub>16</sub>



## 零页 Y (Zero page Y)

名称：零页 Y 寻址方式

功能：将操作数内容和变址寄存器 Y 内容的相加值所指向的零页区内的存储器内容指定为运算数据。

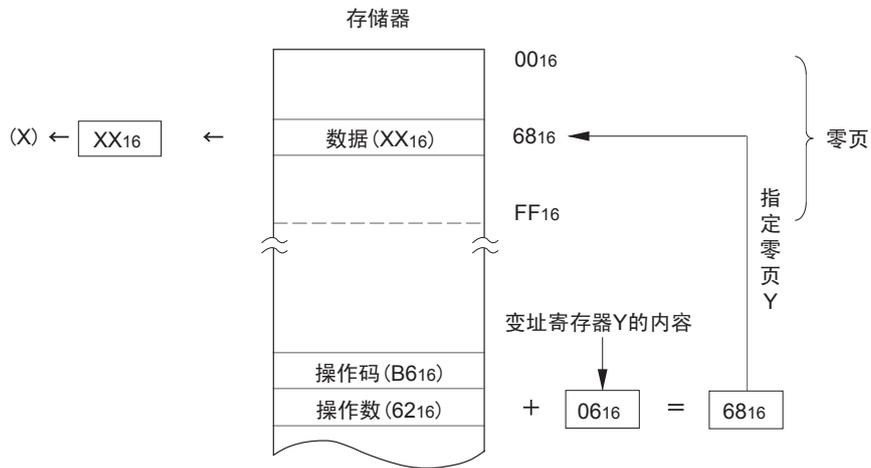
指令：LDX、STX

例：助记符

LDX \$ 62, Y

机器码：

B6<sub>16</sub> 62<sub>16</sub>



## 绝对(Absolute)

名称：绝对寻址方式

功能：将第一操作数和第二操作数指向的存储器内容指定为运算数据。

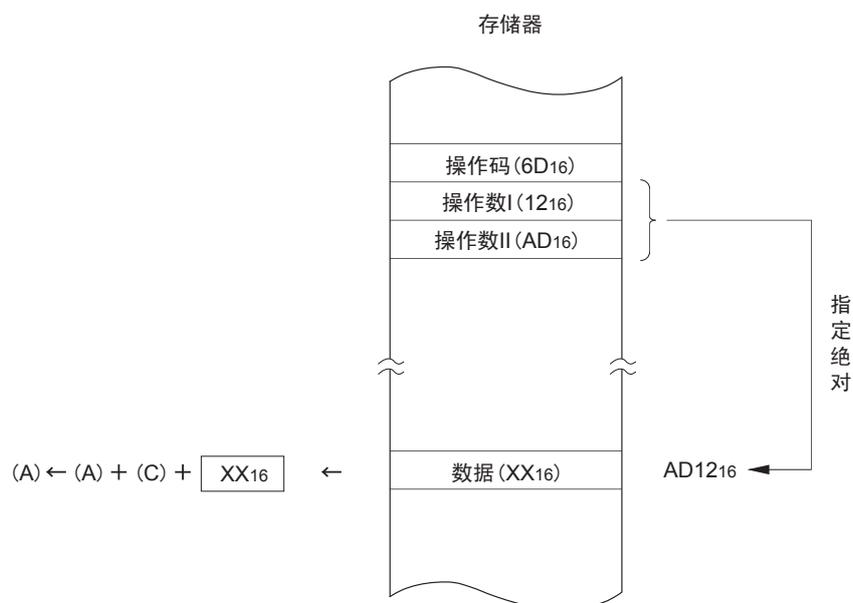
指令：ADC、AND、ASL、BIT、CMP、CPX、CPY、DEC、EOR、INC、JMP、JSR、LDA、LDX、LDY、LSR、ORA、ROL、ROR、SBC、STA、STX、STY

例：助记符

ADC \$AD12

机器码：

6D<sub>16</sub> 12<sub>16</sub> AD<sub>16</sub>





## 绝对 Y(Absolute Y)

名称：绝对 Y 寻址方式

功能：将第一操作数和第二操作数表示的数值和变址寄存器 Y 内容的相加值所指向的存储器内容指定为运算数据。

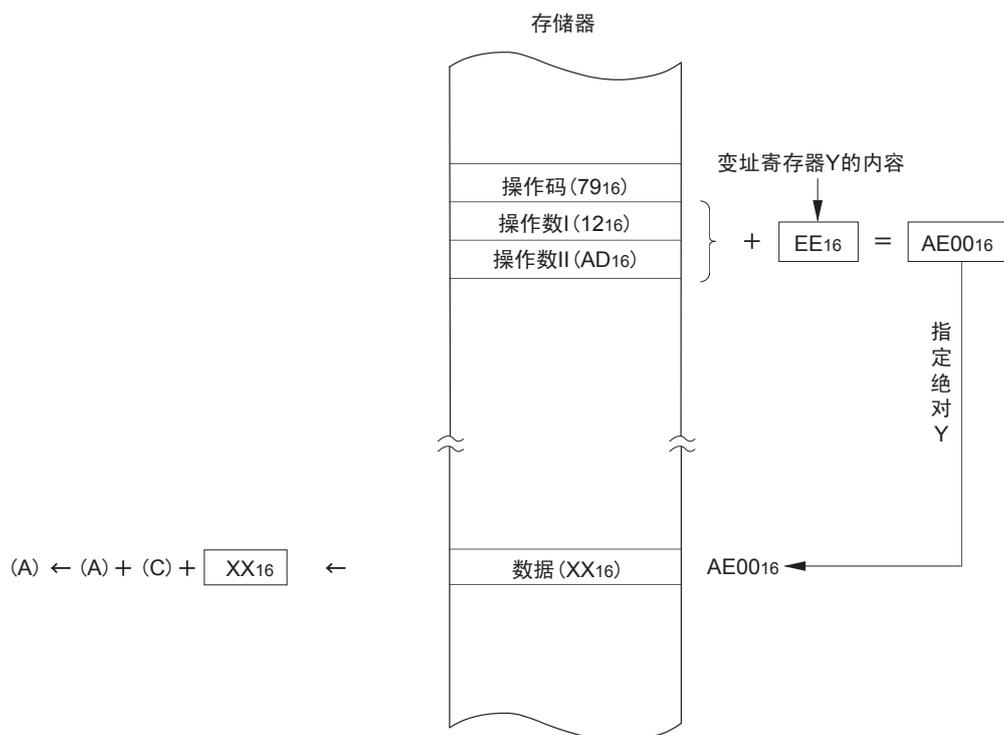
指令：ADC、AND、CMP、EOR、LDA、LDX、ORA、SBC、STA

例：助记符

ADC \$AD12,Y

机器码：

79<sub>16</sub> 12<sub>16</sub> AD<sub>16</sub>





## 相对(Relative)

名称：相对寻址方式

功能：跳转到程序计数器内容和操作数内容的相加值所指向的地址。

指令：BCC、BCS、BEQ、BMI、BNE、BPL、BRA、BVC、BVS

例：助记符

BCC \*-12

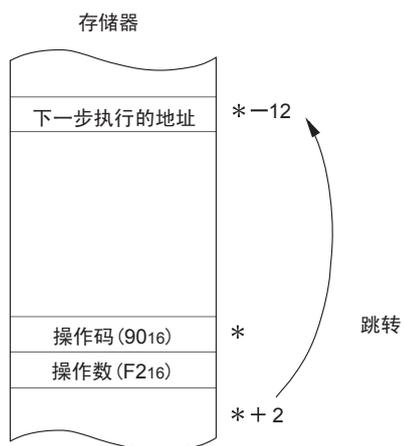
↓

十进制数

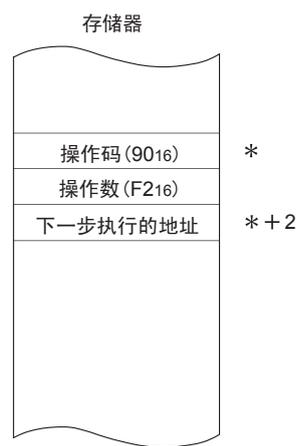
机器码：

90<sub>16</sub> F2<sub>16</sub>

在进位标志(C)已清除时，跳转到地址\*-12



在进位标志(C)已置位时，前进到地址\*+2





## 间接 Y(Indirect Y)

名称：间接 Y 寻址方式

功能：将操作数指定的零页区内连续 2 个字节的存储器内容和变址寄存器 Y 内容的相加值作为地址，将该地址的存储器内容指定为操作数据。

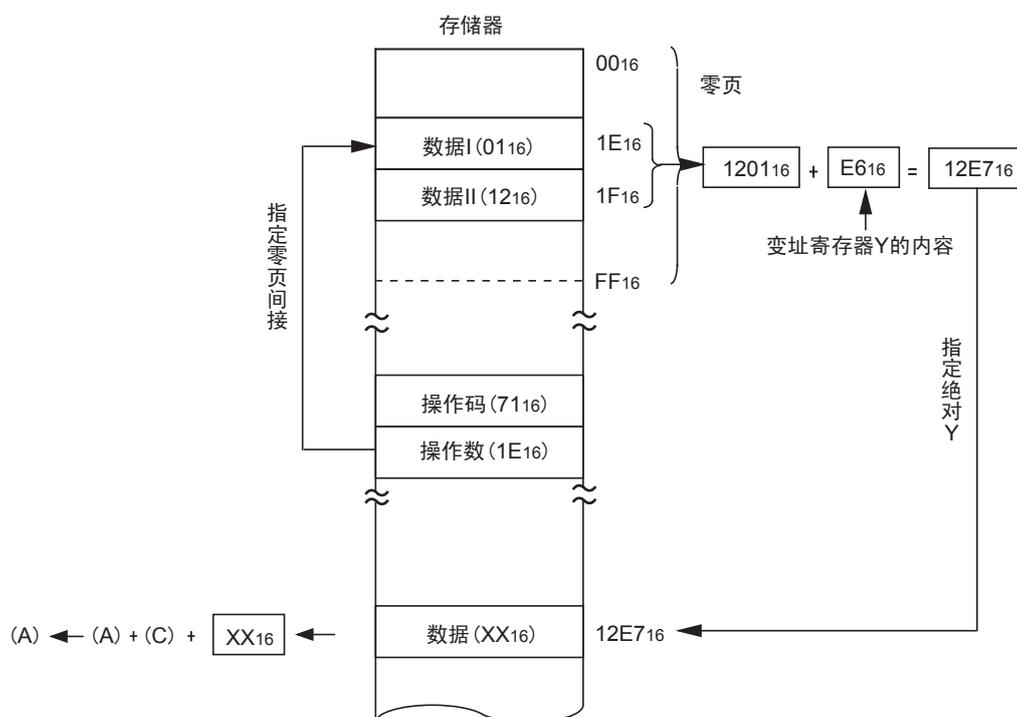
指令：ADC、AND、CMP、EOR、LDA、ORA、SBC、STA

例：助记符

ADC (\$ 1E), Y

机器码：

71<sub>16</sub> 1E<sub>16</sub>



此例中，假设预先保存了数据 I 的“01<sub>16</sub>”和数据 II 的“12<sub>16</sub>”。

## 间接(Indirect)

名称：间接 绝对 寻址方式

功能：跳转到第一操作数和第二操作数指定的连续 2 个字节的存储器内容所指向的地址。

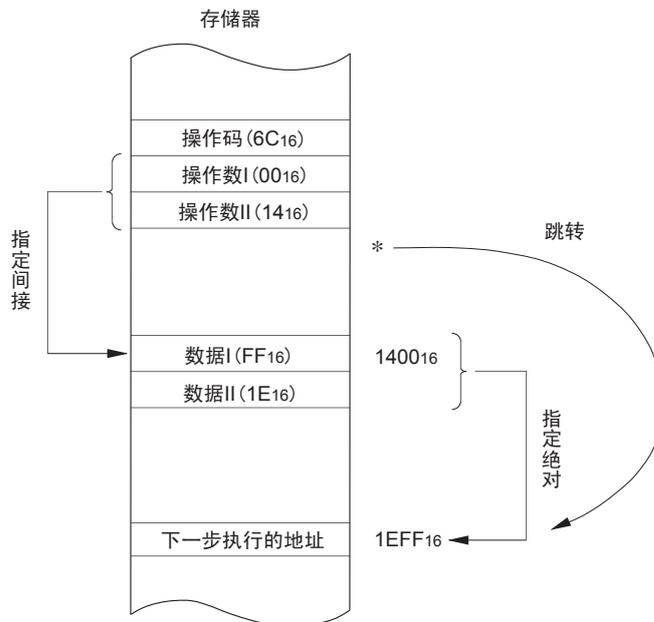
指令：JMP

例：助记符

JMP (\$ 1400)

机器码：

6C<sub>16</sub> 00<sub>16</sub> 14<sub>16</sub>



此例中，假设预先保存了数据I的“FF<sub>16</sub>”和数据II的“1E<sub>16</sub>”。

注．作为间接地址，不能指定页的结束地址（XXFF<sub>16</sub>地址）。也就是说，不能执行 JMP(\$XXFF)指令，必须注意。

## 零页 间接(Zero page Indirect)

名称：零页 间接 绝对 寻址方式

功能：跳转到操作数指定的零页区内连续 2 个字节的存储器内容所指向的地址。

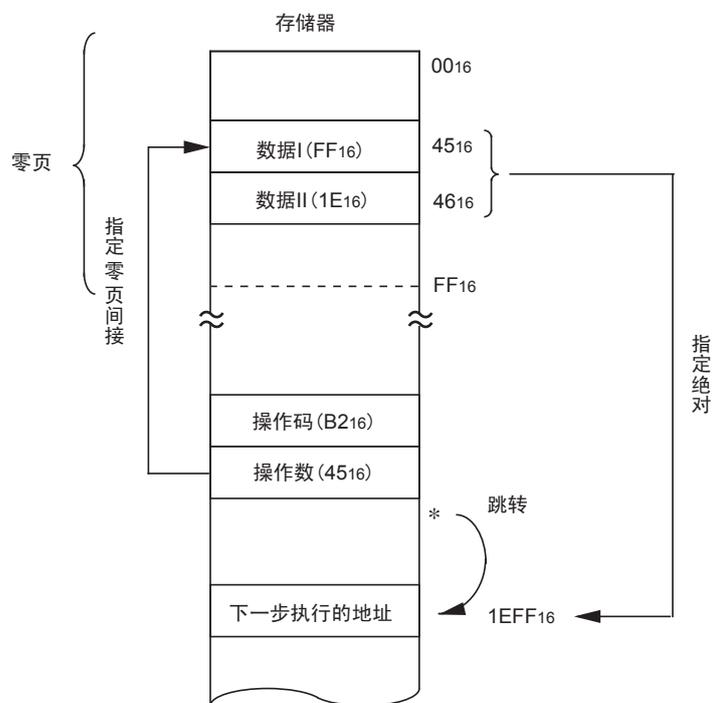
指令：JMP、JSR

例：助记符

JMP (\$45)

机器码：

B2<sub>16</sub> 45<sub>16</sub>



此例中，假设预先保存了数据I的“FF<sub>16</sub>”和数据II的“1E<sub>16</sub>”。

## 专用页(Special page)

名称：专用页 寻址方式

功能：跳转到低 8 位地址为操作数、高 8 位地址为 FF16 的特殊区内的地址。

指令：JSR

例：助记符

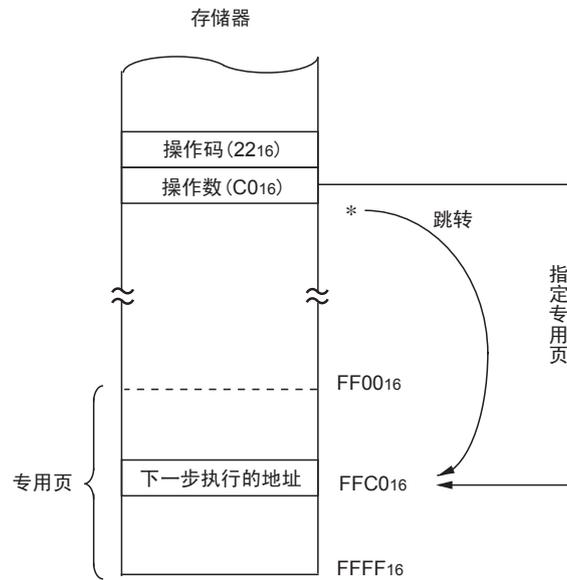
JSR \ \$FFC0

↓

此符号 (\) 表示“为特殊页模式”

机器码：

2216 C016



## 零页 位(Zero page bit)

名 称 : 零页 位 寻址方式

功 能 : 用操作码的高3位指定第一操作数的内容指向的零页区的存储器的特定位。

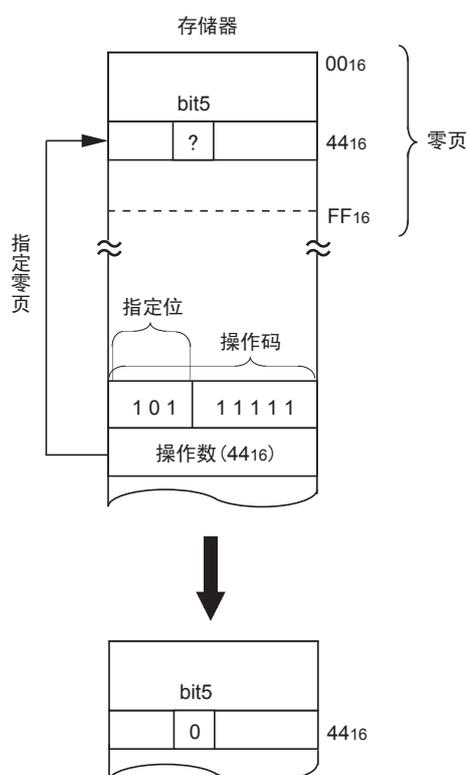
指 令 : CLB、SEB

例 : 助记符

CLB 5,\$44

机器码:

BF16 4416



## 累加器 位(Accumulator bit)

名 称 : 累加器 位 寻址方式

功 能 : 用操作码高 3 位指定累加器中的位。

指 令 : CLB、SEB

例 : 助记符  
CLB 5, A

机器码:  
BB16



## 累加器 位 相对(Accumulator bit relative)

名 称 : 累加器 位 相对 寻址方式

功 能 : 用操作码的高3位指定累加器的位, 根据该位的状态, 跳转到程序计数器内容和操作数内容的相加值所指向的地址。

指 令 : BBC、BBS

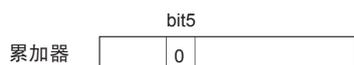
例 : 助记符

BBC 5,A,\*-12

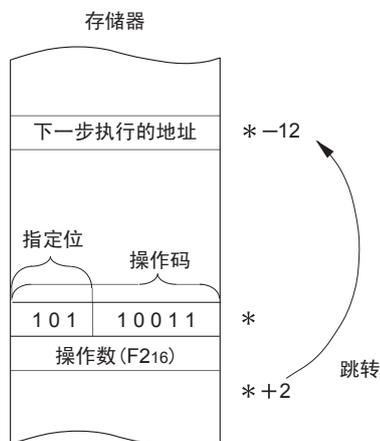
机器码:

B316 F216

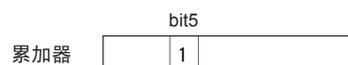
在累加器的位5已清除时



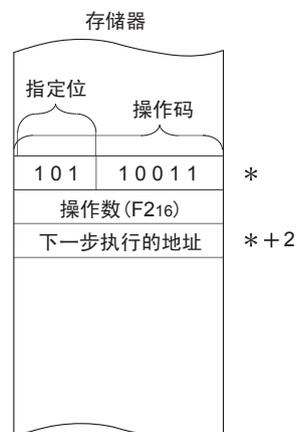
跳转到地址\*-12



在累加器的位5已置位时



前进到地址\*+2



## 零页 位 相对(Zero page bit relative)

名 称 : 零页 位 相对 寻址方式

功 能 : 用操作码的高3位指定零页区的存储器的特定位置 (该零页区由第一操作数的内容指定), 根据该位的状态, 跳转到程序计数器内容和第二操作数内容的相加值所指向的地址。

指 令 : BBC、BBS

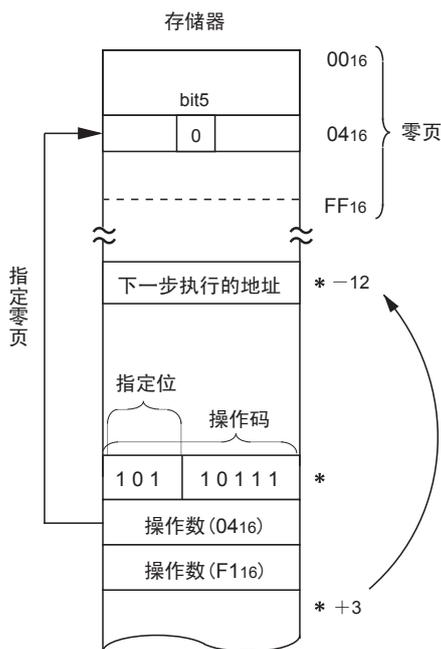
例 : 助记符

BBC 5,\$04,\*-12

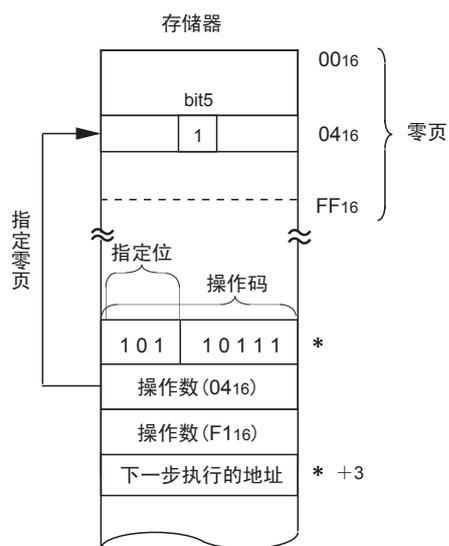
机器码:

B7<sub>16</sub> 04<sub>16</sub> F1<sub>16</sub>

在地址04<sub>16</sub>的位5已清除时, 跳转到地址\*-12



在地址04<sub>16</sub>的位5已置位时, 前进到地址\*+3



## 3.2 指令系统

740 族有 71 种类的指令。这些指令的详细说明叙述在 3.3 中。

请注意：根据单片机的产品种类，有一些不能使用的指令。

### 3.2.1 数据传送指令

这些指令在寄存器之间、寄存器与存储器之间以及存储器之间传送数据。

数据传送指令如下：

	指令	内 容
装 入	LDA	将存储器的内容装入累加器或者变址寄存器X指定的存储器
	LDM	将立即数装入存储器
	LDX	将存储器的内容装入变址寄存器X
	LDY	将存储器的内容装入变址寄存器Y
存 储	STA	将累加器的内容存入存储器
	STX	将变址寄存器X的内容存入存储器
	STY	将变址寄存器Y的内容存入存储器
传 送	TAX	将累加器的内容传送到变址寄存器X
	TXA	将变址寄存器X的内容传送到累加器
	TAY	将累加器的内容传送到变址寄存器Y
	TYA	将变址寄存器Y的内容传送到累加器
	TSX	将栈指针的内容传送到变址寄存器X
	TXS	将变址寄存器X的内容传送到栈指针
栈 操 作	PHA	将累加器的内容压入堆栈
	PHP	将处理器状态压入堆栈
	PLA	从堆栈恢复累加器的内容
	PLP	从堆栈恢复处理器状态

## 3.2.2 运算指令

这些指令对寄存器或者存储器中的数据进行加减运算、逻辑运算、比较、循环以及移位。

运算指令如下：

	指令	内 容
加 减 运 算	ADC	将存储器的内容和C标志加到累加器或者变址寄存器X指定的存储器
	SBC	累加器的内容或者变址寄存器X指定的存储器内容减去存储器的内容和C标志的补码
	INC	存储器或者累加器的内容加1
	DEC	存储器或者累加器的内容减1
	INX	变址寄存器X的内容加1
	DEX	变址寄存器X的内容减1
	INY	变址寄存器Y的内容加1
	DEY	变址寄存器Y的内容减1
乘 除 运 算	MUL (注)	累加器的内容乘以零页X寻址方式指定的存储器内容，结果的高8位存入堆栈、低8位存入累加器
	DIV (注)	以零页X寻址方式指定的存储器内容和其下一个地址的存储器内容为字数据，除以累加器的内容，商存入累加器、余数的1的补码存入堆栈
逻 辑 运 算	AND	存储器的内容与累加器或者变址寄存器X指定的存储器内容进行逻辑与运算
	ORA	存储器的内容与累加器或者变址寄存器X指定的存储器内容进行逻辑或运算
	EOR	存储器的内容与累加器或者变址寄存器X指定的存储器内容进行逻辑异或运算
	COM	将存储器内容的1的补码存入存储器
	BIT	存储器的内容与累加器的内容进行逻辑与运算（不保存结果）
	TST	测试寄存器的内容是否为“0”
比 较	CMP	存储器的内容与累加器的内容或者变址寄存器X指定的存储器内容进行比较
	CPX	比较存储器和变址寄存器X的内容
	CPY	比较存储器和变址寄存器Y的内容
移 位 和 循 环	ASL	存储器或者累加器的内容左移一位
	LSR	存储器或者累加器的内容右移一位
	ROL	存储器或者累加器的内容带进位左循环一位
	ROR	存储器或者累加器的内容带进位右循环一位
	RRF	存储器的内容右循环四位

注：部分产品不能使用乘除运算指令。

### 3.2.3 位操作指令

这些指令对累加器或存储器指定的位置“1”或者置“0”。

位操作指令如下：

	指令	内 容
位 操 作	CLB	累加器或者存储器指定的位清“0”
	SEB	累加器或者存储器指定的位置“1”

### 3.2.4 标志设定指令

这些指令对 C 标志、D 标志、I 标志、T 标志和 V 标志置“0”或者置“1”。

标志设定指令如下：

	指令	内 容
标 志 设 定	CLC	C标志清“0”
	SEC	C标志置“1”
	CLD	D标志清“0”
	SED	D标志置“1”
	CLI	I标志清“0”
	SEI	I标志置“1”
	CLT	T标志清“0”
	SET	T标志置“1”
	CLV	V标志清“0”

## 3.2.5 跳转、转移和返回指令

通过这些指令改变程序顺序。

跳转、转移和返回指令如下：

	指令	内 容
跳 转	JMP BRA JSR	跳转到新地址 跳转到新地址 保存返回地址，跳转到新地址
转 移	BBC BBS BCC BCS BNE BEQ BPL BMI BVC BVS	在累加器或存储器指定的位为“0”时转移 在累加器或存储器指定的位为“1”时转移 在C标志为“0”时转移 在C标志为“1”时转移 在Z标志为“0”时转移 在Z标志为“1”时转移 在N标志为“0”时转移 在N标志为“1”时转移 在V标志为“0”时转移 在V标志为“1”时转移
返 回	RTI RTS	从中断返回 从子程序返回

### 3.2.6 中断指令

该指令产生软件中断。

	指令	内 容
中断	BRK	执行软件中断

### 3.2.7 特殊指令

这些指令控制振荡和内部时钟。

	指令	内 容
特殊	WIT	停止内部时钟
	STP	停止振荡器的振荡

### 3.2.8 其它指令

	指令	内 容
其它	NOP	只进行程序计数器+1

### 3.3 指令的说明

本节详细说明 740 族的各指令，指令的助记符作为标题，按字母顺序排列，原则上以 1 页 1 条指令的形式进行记述。

对各指令，记述了该指令的操作、说明以及状态标志的变化。另外，对该指令使用的寻址方式，还记载了汇编程序的记述格式、机器码、字节数以及周期数的一览表。

说明时使用的符号及其内容如下所示：

符号	内 容	符号	内 容
A	累加器	hh	地址高位字节 0~255的数据
Ai	累加器的位i	ll	地址低位字节 0~255的数据
PC	程序计数器	zz	零页地址 0~255的数据
PCL	程序计数器的低位字节	nn	0~255的数据
PCH	程序计数器的高位字节	i	0~7的数据
PS	处理器状态寄存器	*	程序计数器的内容
S	栈指针	△	制表符或者空格
X、Y	变址寄存器	#	立即方式
M	存储器	\	专用页模式
Mi	存储器的位i	\$	16进制记数
C	进位标志	+	加法
Z	零标志	-	减法
I	中断禁止标志	×	乘法
D	10进制模式标志	/	除法
B	中断标志	∧	逻辑与
T	X变址模式标志	∨	逻辑或
V	溢出标志	∇	逻辑异或
N	负标志	( )	寄存器、存储器等的內容
Rel	相对地址	←	数据传送
BADRS	中断地址		

## ADC

## ADd with Carry

## ADC

操作: (T)=0时 (A) ← (A)+(M)+(C)  
(T)=1时 (M(X)) ← (M(X)+(M)+(C)

说明: 当T标志是0时, 将A、M和C标志的内容相加, 结果保存到A和C标志。  
当T标志是1时, 将M(X)、M和C标志的内容相加, 结果保存到M(X)和C标志。此时, 虽然A的内容不变, 但是状态标志发生变化。M(X)是X指向的地址的存储器内容。

状态标志: N : 当运算结果的位7是1时, 为1; 否则为0。  
V : 如果运算结果超过+127或者|-128|, 就变为1; 否则为0。  
T : 不变。  
B : 不变。  
D : 不变。  
I : 不变。  
Z : 当运算结果是0时, 为1; 否则为0。  
C : 当2进制加法的结果超过255时或者10进制加法的结果超过99时, 为1; 否则为0。

寻址方式	记述格式	机器码	字节数	周期数
立即	△ADC△ # \$nn	6916, nn16	2	2
零页	△ADC△ \$zz	6516, zz16	2	3
零页 X	△ADC△ \$zz, X	7516, zz16	2	4
绝对	△ADC△ \$hhll	6D16, ll16, hh16	3	4
绝对 X	△ADC△ \$hhll, X	7D16, ll16, hh16	3	5
绝对 Y	△ADC△ \$hhll, Y	7916, ll16, hh16	3	5
(间接 X)	△ADC△ (\$zz, X)	6116, zz16	2	6
(间接 Y)	△ADC△ (\$zz, Y)	7116, zz16	2	6

注 1. 当 T 标志是 1 时, 周期数+3。

- 在 10 进制运算方式 (D 标志=1) 中执行 ADC 指令时, 必须在 ADC 指令后间隔一条以上的指令, 然后执行 SEC、CLC、CLD 指令。另外, 在 10 进制运算方式中, N、V、Z 标志无效。

# AND

## logical AND

# AND

操作: (T)=0时 (A) ← (A)^(M)  
(T)=1时 (M(X)) ← (M(X))^(M)

说明: 当T标志是0时, 对A和M内容的各位进行逻辑与, 结果保存到A。  
当T标志是1时, 对M(X)和M内容的各位进行逻辑与, 结果保存到M(X)。此时, 虽然A的内容不变, 但是状态标志发生变化。M(X)是X指向的地址的存储器内容。

状态标志: N : 当运算结果的位7是1时, 为1; 否则为0。  
V : 不变。  
T : 不变。  
B : 不变。  
D : 不变。  
I : 不变。  
Z : 当运算结果是0时, 为1; 否则为0。  
C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
立即	△AND△ #\$nn	29 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	△AND△ \$zz	25 <sub>16</sub> , zz <sub>16</sub>	2	3
零页 X	△AND△ \$zz, X	35 <sub>16</sub> , zz <sub>16</sub>	2	4
绝对	△AND△ \$hhll	2D <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4
绝对 X	△AND△ \$hhll, X	3D <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
绝对 Y	△AND△ \$hhll, Y	39 <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
(间接 X)	△AND△ (\$zz, X)	21 <sub>16</sub> , zz <sub>16</sub>	2	6
(间接 Y)	△AND△ (\$zz), Y	31 <sub>16</sub> , zz <sub>16</sub>	2	6

注. 当T标志是1时, 周期数+3。



# BBC

## Branch on Bit Clear

# BBC

操作: (Mi)或者(Ai)=0时  $(PC) \leftarrow (PC) + n + Rel$   
 (Mi)或者(Ai)=1时  $(PC) \leftarrow (PC) + n$   
 其中, 当寻址方式是零页时,  $n=3$ ; 是累加器时,  $n=2$ 。

说明: 测试被指定的M或者A的位i。  
 如果该位是0, 就转移到指定地址。以相对地址指示转移目标地址。  
 如果该位是1, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
累加器位相对	$\Delta BBC \Delta i, A, \$hhll$	$(20i + 13)_{16}, rr_{16}$	2	4
零页位相对	$\Delta BBC \Delta i, \$zz, \$hhll$	$(20i + 17)_{16}, zz_{16}, rr_{16}$	3	5

注 1.  $rr_{16} = \$hhll - (* + n)$ 。 $rr_{16}$  取  $-128 \sim +127$  范围的值。

- 进行转移时, 周期数+2。
- 要在更改中断请求位的内容后执行 BBC 指令时, 必须在更改后执行 1 条以上的指令以后执行。

# BBS

## Branch on Bit Set

# BBS

操作: (Mi)或者(Ai)=1时  $(PC) \leftarrow (PC)+n+Rel$   
 (Mi)或者(Ai)=0时  $(PC) \leftarrow (PC)+n$   
 其中, 当寻址方式是零页时,  $n=3$ ; 是累加器时,  $n=2$ 。

说明: 测试被指定的M或者A的位i。  
 如果该位是1, 就转移到指定地址。以相对地址指示转移目标地址。  
 如果该位是0, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
累加器位相对	$\Delta BBS \Delta i, A, \$hhll$	$(20i+3)_{16}, rr_{16}$	2	4
零页位相对	$\Delta BBS \Delta i, \$zz, \$hhll$	$(20i+7)_{16}, zz_{16}, rr_{16}$	3	5

注 1.  $rr_{16} = \$hhll - (*+n)$ 。 $rr_{16}$  取  $-128 \sim +127$  范围的值。

2. 进行转移时, 周期数+2。

3. 要在更改中断请求位的内容后执行 BBS 指令时, 必须在更改后执行 1 条以上的指令后执行。

# BCC

## Branch on Carry Clear

# BCC

操作: (C)=0时 (PC)  $\leftarrow$  (PC)+2+Rel  
(C)=1时 (PC)  $\leftarrow$  (PC)+2

说明: 当C标志是0时, 转移到指定地址。以相对地址指示转移目标地址。  
如果C标志是1, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
相对	$\Delta$ BCC $\Delta$ \$hhll	9016, rr16	2	2

注 1.  $rr16 = \$hhll - (*+2)$ 。rr16 取  $-128 \sim +127$  范围的值。

2. 进行转移时, 周期数+2。

# BCS

## Branch on Carry Set

# BCS

操作: (C)=1时 (PC)  $\leftarrow$  (PC)+2+Rel  
(C)=0时 (PC)  $\leftarrow$  (PC)+2

说明: 当C标志是1时, 转移到指定地址。以相对地址指示转移目标地址。  
如果C标志是0, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
相对	$\triangle$ BCS $\triangle$ \$hhll	B016, rr16	2	2

注 1. rr16=\$hhll-(\*)+2)。rr16 取-128~+127 范围的值。

2. 进行转移时, 周期数+2。

# BEQ

## Branch on EQual

# BEQ

操作: (Z)=1时 (PC)  $\leftarrow$  (PC)+2+Rel  
(Z)=0时 (PC)  $\leftarrow$  (PC)+2

说明: 当Z标志是1时, 转移到指定地址。以相对地址指示转移目标地址。  
如果Z标志是0, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
相对	$\triangle$ BEQ $\triangle$ \$hhll	F016, rr16	2	2

注 1.  $rr16 = \$hhll - (*+2)$ 。rr16 取  $-128 \sim +127$  范围的值。

2. 进行转移时, 周期数+2。

# BIT

test BIT in memory with accumulator

# BIT

操 作: (A)^(M)

说 明: 对A和M内容的各位进行逻辑与, 但是不保存结果。N标志、V标志、Z标志的内容发生变化, 而A和M的内容不变。

状态标志: N : 当M的位7是1时, 为1; 否则为0。  
 V : 当M的位6是1时, 为1; 否则为0。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 当逻辑与的结果是0时, 为1; 否则为0。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页	$\Delta$ BIT $\Delta$ \$zz	24 <sub>16</sub> , zz <sub>16</sub>	2	3
绝对	$\Delta$ BIT $\Delta$ \$hhl	2C <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4

**BMI**

## Branch on result Minus

**BMI**

操作: (N)=1时 (PC)  $\leftarrow$  (PC)+2+Rel  
(N)=0时 (PC)  $\leftarrow$  (PC)+2

说明: 当N标志是1时, 转移到指定地址。以相对地址指示转移目标地址。  
如果N标志是0, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
相对	$\triangle$ BMI $\triangle$ \$hhll	3016, rr16	2	2

注 1.  $rr16 = \$hhll - (*+2)$ 。rr16 取  $-128 \sim +127$  范围的值。

2. 进行转移时, 周期数+2。

# BNE

## Branch on Not Equal

# BNE

操作: (Z)=0时 (PC) ← (PC)+2+Rel  
(Z)=1时 (PC) ← (PC)+2

说明: 当Z标志是0时, 转移到指定地址。以相对地址指示转移目标地址。  
如果Z标志是1, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
相对	$\Delta$ BNE $\Delta$ \$hhll	D016, rr16	2	2

注 1. rr16=\$hhll-(\*)+2)。rr16 取-128~+127 范围的值。

2. 进行转移时, 周期数+2。

**BPL**

## Branch on result PLus

**BPL**

操作: (N)=0时  $(PC) \leftarrow (PC)+2+Rel$   
 (N)=1时  $(PC) \leftarrow (PC)+2$

说明: 当N标志是0时, 转移到指定地址。以相对地址指示转移目标地址。  
 如果N标志是1, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
相对	$\Delta BPL \Delta \$hhll$	1016, rr16	2	2

注 1.  $rr16 = \$hhll - (*+2)$ 。rr16 取  $-128 \sim +127$  范围的值。

2. 进行转移时, 周期数+2。

# BRA

## BRanch Always

# BRA

操 作:  $(PC) \leftarrow (PC) + 2 + Rel$

说 明: 转移到指定地址。以相对地址指示转移目标地址。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
相对	$\Delta BRA \Delta \$hhll$	8016, rr16	2	4

注.  $rr16 = \$hhll - (* + 2)$ 。rr16 取  $-128 \sim +127$  范围的值。

# BRK

force BReak

# BRK

操作: (B) ← 1  
(PC) ← (PC)+2  
(M(S)) ← (PCH)  
(S) ← (S)-1  
(M(S)) ← (PCL)  
(S) ← (S)-1  
(M(S)) ← (PS)  
(S) ← (S)-1  
(I) ← 1  
(PC) ← BADRS

说明: 当执行BRK指令时, CPU将当前的PC内容保存到堆栈, 通过中断向量将指定的地址(BADRS)保存到PC。

状态标志: N : 不变。  
V : 不变。  
T : 不变。  
B : 为1。  
D : 不变。  
I : 为1。  
Z : 不变。  
C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ BRK $\Delta$	00 <sub>16</sub>	1	7

- 注 1. 通过执行 BRK 指令, 被保存的 PC 值为 BRK 指令地址+2。因此, 从 BRK 程序返回时, 不执行 BRK 指令的下一个字节。
2. 一部分产品的 BRK 指令和其它的中断源有相同的中断向量。此时, 虽然都转移到中断向量指定的地址, 但是通过测试堆栈保存的处理器状态寄存器的 B 标志, 能判断是否执行了 BRK 指令。

# BVC

## Branch on oVerflow Clear

# BVC

操 作: (V)=0时 (PC)  $\leftarrow$  (PC)+2+Rel  
(V)=1时 (PC)  $\leftarrow$  (PC)+2

说 明: 当V标志是0时, 转移到指定地址。以相对地址指示转移目标地址。  
如果V标志是1, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
相对	$\triangle$ BVC $\triangle$ \$hhll	5016, rr16	2	2

注 1. rr16=\$hhll-( $\ast$ +2)。rr16取-128~+127范围的值。

2. 进行转移时, 周期数+2。

# BVS

## Branch on oVerflow Set

# BVS

操作: (V)=1时 (PC)  $\leftarrow$  (PC)+2+Rel  
(V)=0时 (PC)  $\leftarrow$  (PC)+2

说明: 当V标志是1时, 转移到指定地址。以相对地址指示转移目标地址。  
如果V标志是0, 就继续执行。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
相对	$\triangle$ BVS $\triangle$ \$hhll	7016, rr16	2	2

注 1.  $rr16 = \$hhll - (*+2)$ 。rr16 取  $-128 \sim +127$  范围的值。

2. 进行转移时, 周期数+2。

# CLB

## CLear Bit

# CLB

操 作:  $(A_i) \leftarrow 0$  或者  $(M_i) \leftarrow 0$

说 明: 被指定的A或者M的位i清0。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
累加器 位	$\triangle CLB \triangle i, A$	$(20i + 1B)_{16}$	1	2
零页 位	$\triangle CLB \triangle i, \$zz$	$(20i + 1F)_{16}, zz_{16}$	2	5

# CLC

CLear Carry flag

# CLC

操 作: (C) ← 0

说 明: C标志清0。

状态标志: N : 不变。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 不变。  
 C : 为0。

寻址方式	记述格式	机器码	字节数	周期数
隐含	△CLC	1816	1	2

# CLD

## CLear Decimal mode

# CLD

操 作: (D) ← 0

说 明: D标志清0。

状态标志: N : 不变。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 为0。  
 I : 不变。  
 Z : 不变。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	△CLD	D816	1	2

# CLI

CLear Interrupt disable status

# CLI

操 作: (I) ← 0

说 明: I标志清0。

状态标志:

- N : 不变。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 为0。
- Z : 不变。
- C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	△CLI	5816	1	2

# CLT

CLear Transfer flag

# CLT

操作:  $(T) \leftarrow 0$ 

说明: T标志清0。

状态标志:

- N : 不变。
- V : 不变。
- T : 为0。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 不变。
- C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ CLT	1216	1	2

# CLV

CLear oVerflow flag

# CLV

操 作: (V) ← 0

说 明: V标志清0。

状态标志:

- N : 不变。
- V : 为0。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 不变。
- C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	△CLV	B816	1	2

# CMP

## CoMPare

# CMP

操作: (T)=0时 (A)-(M)  
(T)=1时 (M(X))-(M)

说明: 当T标志是0时, A的内容减去M的内容, 不保存结果。此时, A和M的内容不变。  
当T标志是1时, M(X)的内容减去M的内容, 不保存结果。此时, M(X)、M和A的内容不变。  
M(X)是X指向的地址的存储器内容。

状态标志: N : 当减法结果的位7是1时, 为1; 否则为0。  
V : 不变。  
T : 不变。  
B : 不变。  
D : 不变。  
I : 不变。  
Z : 当减法结果是0时, 为1; 否则为0。  
C : 当减法结果大于等于0时, 为1; 否则为0。

寻址方式	记述格式	机器码	字节数	周期数
立即	$\Delta$ CMP $\Delta$ # $nn$	C9 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	$\Delta$ CMP $\Delta$ \$zz	C5 <sub>16</sub> , zz <sub>16</sub>	2	3
零页 X	$\Delta$ CMP $\Delta$ \$zz, X	D5 <sub>16</sub> , zz <sub>16</sub>	2	4
绝对	$\Delta$ CMP $\Delta$ \$hhll	CD <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4
绝对 X	$\Delta$ CMP $\Delta$ \$hhll, X	DD <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
绝对 Y	$\Delta$ CMP $\Delta$ \$hhll, Y	D9 <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
(间接 X)	$\Delta$ CMP $\Delta$ (\$zz, X)	C1 <sub>16</sub> , zz <sub>16</sub>	2	6
(间接 Y)	$\Delta$ CMP $\Delta$ (\$zz), Y	D1 <sub>16</sub> , zz <sub>16</sub>	2	6

注. 当T标志是1时, 周期数+1。

# COM

## COMplement

# COM

操作:  $(M) \leftarrow (\overline{M})$

说明: 将M内容的1的补码保存到M。

状态标志: N : 当执行结果的M的位7是1时, 为1; 否则为0。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 当执行结果是0时, 为1; 否则为0。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页	$\triangle\text{COM}\triangle \$zz$	4416, ZZ16	2	5

# CPX

ComPare memory and index register X

# CPX

操 作: (X)−(M)

说 明: X的内容减去M的内容, 不保存结果。此时, X和M的内容不变。

状态标志: N : 当减法结果的位7是1时, 为1; 否则为0。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 当减法结果是0时, 为1; 否则为0。  
 C : 当减法结果大于等于0时, 为1; 否则为0。

寻址方式	记述格式	机器码	字节数	周期数
立即	△CPX△ # $nn$	E0 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	△CPX△ \$zz	E4 <sub>16</sub> , zz <sub>16</sub>	2	3
绝对	△CPX△ \$hhll	EC <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4

# CPY

ComPare memory and index register Y

# CPY

操作: (Y)-(M)

说明: Y的内容减去M的内容, 不保存结果。此时, Y和M的内容不变。

状态标志:

- N : 当减法结果的位7是1时, 为1; 否则为0。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 当减法结果是0时, 为1; 否则为0。
- C : 当减法结果大于等于0时, 为1; 否则为0。

寻址方式	记述格式	机器码	字节数	周期数
立即	$\Delta$ CPY $\Delta$ # $\$nn$	C0 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	$\Delta$ CPY $\Delta$ $\$zz$	C4 <sub>16</sub> , zz <sub>16</sub>	2	3
绝对	$\Delta$ CPY $\Delta$ $\$hhll$	CC <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4

# DEC

## DECrement by one

# DEC

操作:  $(A) \leftarrow (A) - 1$  或者  $(M) \leftarrow (M) - 1$

说明: A或者M的内容减1。

状态标志: N : 当执行结果的位7是1时, 为1; 否则为0。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 当执行结果是0时, 为1; 否则为0。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
累加器	$\Delta DEC \Delta A$	1A <sub>16</sub>	1	2
零页	$\Delta DEC \Delta \$zz$	C6 <sub>16</sub> , zz <sub>16</sub>	2	5
零页 X	$\Delta DEC \Delta \$zz, X$	D6 <sub>16</sub> , zz <sub>16</sub>	2	6
绝对	$\Delta DEC \Delta \$hhll$	CE <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	6
绝对 X	$\Delta DEC \Delta \$hhll, X$	DE <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	7

**DEX**

DEcrement index register X by one

**DEX**操 作:  $(X) \leftarrow (X) - 1$ 

说 明: X的内容减1。

状态标志: N : 当执行结果的位7是1时, 为1; 否则为0。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 当执行结果是0时, 为1; 否则为0。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ DEX	CA16	1	2

# DEY

DEcrement index register Y by one

# DEY

操作:  $(Y) \leftarrow (Y) - 1$ 

说明: Y的内容减1。

状态标志:

- N : 当执行结果的位7是1时, 为1; 否则为0。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 当执行结果是0时, 为1; 否则为0。
- C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ DEY	8816	1	2

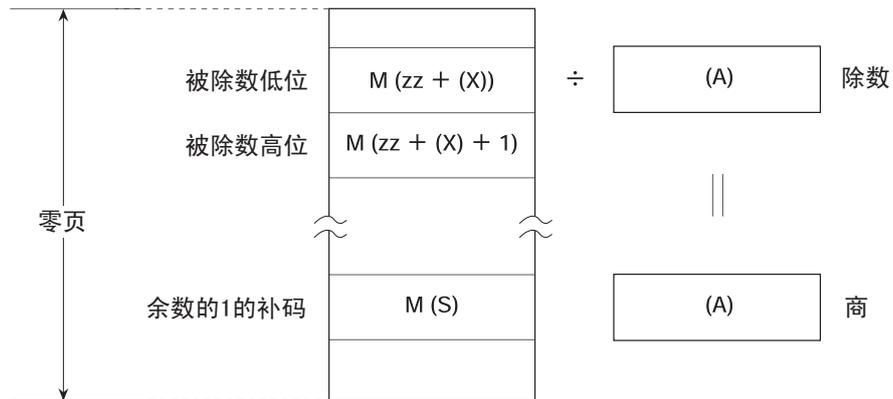
# DIV

## DIVide memory by accumulator

# DIV

操作: (A) ← (M(zz+(X)+1), M(zz+(X))) / (A)  
 M(S) ← 余数的1的补码  
 (S) ← (S)-1

说明: 高8位为M(zz+(X)+1)、低8位为M(zz+(X))的内容的16位数据除以累加器的内容。商保存到累加器, 余数的1的补码保存到堆栈。



状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页 X	$\triangle DIV \triangle \$zz, X$	E216, zz16	2	16

注 1. 因为不检测商的溢出和被 0 除, 所以执行此指令前, 必须在程序中进行处理。

另外, 必须注意: 如果执行此指令, 累加器和栈指针的内容就发生变化。

- 一部分产品不能使用此指令。
- 不受 T 标志和 D 标志的影响。

# EOR

## Exclusive OR memory with accumulator

# EOR

操作: (T)=0时 (A) ← (A) ∨ (M)  
(T)=1时 (M(X)) ← (M(X)) ∨ (M)

说明: 当T标志是0时, 对A和M内容的各位进行逻辑异或, 结果保存到A。  
当T标志是1时, 对M(X)和M内容的各位进行逻辑异或, 结果保存到M(X)。此时, 虽然A的内容不变, 但是状态标志发生变化。M(X)是X指向的地址的存储器内容。

状态标志: N : 当运算结果的位7是1时, 为1; 否则为0。  
V : 不变。  
T : 不变。  
B : 不变。  
D : 不变。  
I : 不变。  
Z : 当运算结果是0时, 为1; 否则为0。  
C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
立即	△EOR△ # \$nn	49 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	△EOR△ \$zz	45 <sub>16</sub> , zz <sub>16</sub>	2	3
零页 X	△EOR△ \$zz, X	55 <sub>16</sub> , zz <sub>16</sub>	2	4
绝对	△EOR△ \$hhll	4D <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4
绝对 X	△EOR△ \$hhll, X	5D <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
绝对 Y	△EOR△ \$hhll, Y	59 <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
(间接 X)	△EOR△ (\$zz, X)	41 <sub>16</sub> , zz <sub>16</sub>	2	6
(间接 Y)	△EOR△ (\$zz), Y	51 <sub>16</sub> , zz <sub>16</sub>	2	6

注. 当T标志是1时, 周期数+3。

# INC

## INCrement by one

# INC

操作:  $(A) \leftarrow (A)+1$  或者  $(M) \leftarrow (M)+1$

说明: A或者M的内容加1。

状态标志: N : 当执行结果的位7是1时, 为1; 否则为0。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 当执行结果是0时, 为1; 否则为0。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
累加器	$\Delta\text{INC}\Delta A$	3A <sub>16</sub>	1	2
零页	$\Delta\text{INC}\Delta \$zz$	E6 <sub>16</sub> , ZZ <sub>16</sub>	2	5
零页 X	$\Delta\text{INC}\Delta \$zz, X$	F6 <sub>16</sub> , ZZ <sub>16</sub>	2	6
绝对	$\Delta\text{INC}\Delta \$hhll$	EE <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	6
绝对 X	$\Delta\text{INC}\Delta \$hhll, X$	FE <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	7

**INX**

INcrement index register X by one

**INX**操 作:  $(X) \leftarrow (X)+1$ 

说 明: X的内容加1。

状态标志: N : 当执行结果的位7是1时, 为1; 否则为0。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 当执行结果是0时, 为1; 否则为0。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ INX	E8 <sub>16</sub>	1	2

# INY

INcrement index register Y by one

# INY

操作:  $(Y) \leftarrow (Y)+1$ 

说明: Y的内容加1。

状态标志:

- N : 当执行结果的位7是1时, 为1; 否则为0。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 当执行结果是0时, 为1; 否则为0。
- C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ INY	C816	1	2

# JMP

## JuMP

# JMP

操作： 当寻址方式是  
绝对时：  
(PC) ← hhl1

间接时：  
(PCL) ← (hhl1)  
(PCH) ← (hhl1+1)

零页间接时：  
(PCL) ← (zz)  
(PCH) ← (zz+1)

说明： 跳转到绝对、间接和零页间接的各寻址方式所示的地址。

状态标志： 不变。

寻址方式	记述格式	机器码	字节数	周期数
绝对	$\Delta\text{JMP}\Delta \$\text{hhl}$	4C <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	3
间接	$\Delta\text{JMP}\Delta (\text{\$hhl})$	6C <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
零页 间接	$\Delta\text{JMP}\Delta (\text{\$zz})$	B2 <sub>16</sub> , zz <sub>16</sub>	2	4

注. 作为间接地址, 不能指定页的结束地址 (XXFF<sub>16</sub> 地址)。也就是说, 不能执行 JMP(\$XXFF)指令, 必须注意。

# JSR

## Jump to SubRoutine

# JSR

操作: (M(S)) ← (PCH)  
 (S) ← (S) - 1  
 (M(S)) ← (PCL)  
 (S) ← (S) - 1

进行上述操作后, 当寻址方式是绝对时:

(PC) ← hhll

专用页时:

(PCL) ← ll

(PCH) ← FF16

零页间接时:

(PCL) ← (zz)

(PCH) ← (zz + 1)

说明: 在PC的内容保存到堆栈后, 跳转到绝对、专用页和零页间接的各寻址方式所示的地址。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
绝对	$\Delta$ JSR $\Delta$ \$hhll	20 <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	6
专用页	$\Delta$ JSR $\Delta$ \ \$hhll (注)	22 <sub>16</sub> , ll <sub>16</sub>	2	5
零页 间接	$\Delta$ JSR $\Delta$ (\$zz)	02 <sub>16</sub> , zz <sub>16</sub>	2	7

注. ‘\’ 是表示专用页的关键字 (在 ASCII 码中为 5C)。另外, 在专用页时, hh<sub>16</sub> 必须为 FF<sub>16</sub>。

# LDA

## Load Accumulator with memory

# LDA

操作: (T)=0时 (A) ← (M)  
(T)=1时 (M(X)) ← (M)

说明: 当T标志是0时, M的内容保存到A。  
当T标志是1时, M的内容保存到M(X)。此时, 虽然A的内容不变, 但是状态标志发生变化。  
M(X)是X指向的地址的存储器内容。

状态标志: N : 当执行结果的位7是1时, 为1; 否则为0。  
V : 不变。  
T : 不变。  
B : 不变。  
D : 不变。  
I : 不变。  
Z : 当执行结果是0时, 为1; 否则为0。  
C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
立即	$\Delta\text{LDA}\Delta \#\$\text{nn}$	A9 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	$\Delta\text{LDA}\Delta \$\text{zz}$	A5 <sub>16</sub> , zz <sub>16</sub>	2	3
零页 X	$\Delta\text{LDA}\Delta \$\text{zz}, \text{X}$	B5 <sub>16</sub> , zz <sub>16</sub>	2	4
绝对	$\Delta\text{LDA}\Delta \$\text{hhll}$	AD <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4
绝对 X	$\Delta\text{LDA}\Delta \$\text{hhll}, \text{X}$	BD <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
绝对 Y	$\Delta\text{LDA}\Delta \$\text{hhll}, \text{Y}$	B9 <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
(间接 X)	$\Delta\text{LDA}\Delta (\$\text{zz}, \text{X})$	A1 <sub>16</sub> , zz <sub>16</sub>	2	6
(间接 Y)	$\Delta\text{LDA}\Delta (\$\text{zz}, \text{Y})$	B1 <sub>16</sub> , zz <sub>16</sub>	2	6

注. 当T标志是1时, 周期数+2。

# LDM

Load immediate Data to Memory

# LDM

操作:  $(M) \leftarrow nn$ 

说明: 将立即数装入M。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页	$\triangle LDM \triangle \# \$nn, \$zz$	3C16, nn16, zz16	3	4

# LDX

Load index register X from memory

# LDX

操作:  $(X) \leftarrow (M)$ 

说明: 将M的内容装入X。

状态标志:

- N : 当执行结果的位7是1时, 为1; 否则为0。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 当执行结果是0时, 为1; 否则为0。
- C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
立即	$\triangle LDX \triangle \# \$nn$	A2 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	$\triangle LDX \triangle \$zz$	A6 <sub>16</sub> , zz <sub>16</sub>	2	3
零页 Y	$\triangle LDX \triangle \$zz, Y$	B6 <sub>16</sub> , zz <sub>16</sub>	2	4
绝对	$\triangle LDX \triangle \$hhll$	AE <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4
绝对 Y	$\triangle LDX \triangle \$hhll, Y$	BE <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5

# LDY

Load index register Y from memory

# LDY

操作:  $(Y) \leftarrow (M)$ 

说明: 将M的内容装入Y。

状态标志:

- N : 当执行结果的位7是1时, 为1; 否则为0。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 当执行结果是0时, 为1; 否则为0。
- C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
立即	$\triangle LDY \triangle \# \$nn$	A0 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	$\triangle LDY \triangle \$zz$	A4 <sub>16</sub> , zz <sub>16</sub>	2	3
零页 X	$\triangle LDY \triangle \$zz, X$	B4 <sub>16</sub> , zz <sub>16</sub>	2	4
绝对	$\triangle LDY \triangle \$hhll$	AC <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4
绝对 X	$\triangle LDY \triangle \$hhll, X$	BC <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5

# LSR

## Logical Shift Right

# LSR

操作: 0 → 

b7								b0
----	--	--	--	--	--	--	--	----

 → 

C
---

说明: A或者M的全部位右移1位。此时, A或者M的位7为0。另外, C标志存放A或者M的位0的内容。

状态标志: N : 为0。  
V : 不变。  
T : 不变。  
B : 不变。  
D : 不变。  
I : 不变。  
Z : 当执行结果是0时, 为1; 否则为0。  
C : 当执行前A或者M的位0是1时, 为1; 否则为0。

寻址方式	记述格式	机器码	字节数	周期数
累加器	$\Delta$ LSR $\Delta$ A	4A <sub>16</sub>	1	2
零页	$\Delta$ LSR $\Delta$ \$zz	46 <sub>16</sub> , ZZ <sub>16</sub>	2	5
零页 X	$\Delta$ LSR $\Delta$ \$zz, X	56 <sub>16</sub> , ZZ <sub>16</sub>	2	6
绝对	$\Delta$ LSR $\Delta$ \$hhll	4E <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	6
绝对 X	$\Delta$ LSR $\Delta$ \$hhll, X	5E <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	7

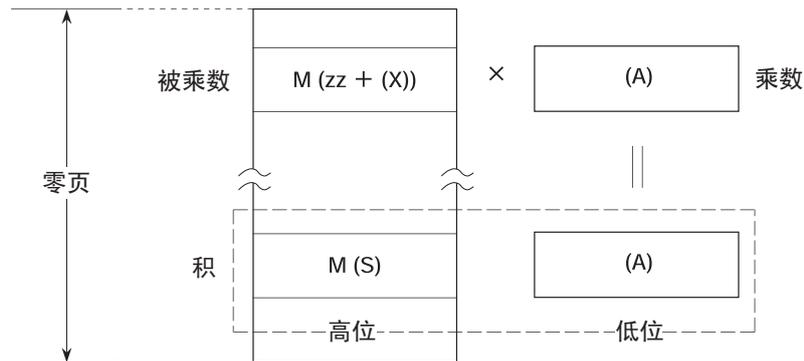
# MUL

## MULTIPLY accumulator and memory

# MUL

操作： $M(S) \cdot (A) \leftarrow (A) \times M(zz + (X))$   
 $(S) \leftarrow (S) - 1$

说明：累加器的内容乘零页X寻址方式指定的存储器内容。结果的积的高8位保存到堆栈，低8位保存到累加器。



状态标志： 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页 X	$\Delta \text{MUL} \Delta \$zz, X$	6216, ZZ16	2	15

- 注 1. 请注意：执行此指令后，累加器和栈指针的内容将发生变化。  
 2. 一部分产品不能使用此指令。

# NOP

No OPeration

# NOP

操 作:  $(PC) \leftarrow (PC)+1$

说 明: 只PC加1, 不进行其它任何操作。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ NOP	EA16	1	2

## ORA

## OR memory with Accumulator

## ORA

操作: (T)=0时 (A) ← (A)∨(M)  
(T)=1时 (M(X)) ← (M(X))∨(M)

说明: 当T标志是0时, 对A和M内容的各位进行逻辑或, 结果保存到A。  
当T标志是1时, 对M(X)和M内容的各位进行逻辑或, 结果保存到M(X)。此时, 虽然A的内容不变, 但是状态标志发生变化。M(X)是X指向的地址的存储器内容。

状态标志: N : 当运算结果的位7是1时, 为1; 否则为0。  
V : 不变。  
T : 不变。  
B : 不变。  
D : 不变。  
I : 不变。  
Z : 当运算结果是0时, 为1; 否则为0。  
C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
立即	△ORA△ # \$nn	09 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	△ORA△ \$zz	05 <sub>16</sub> , zz <sub>16</sub>	2	3
零页 X	△ORA△ \$zz, X	15 <sub>16</sub> , zz <sub>16</sub>	2	4
绝对	△ORA△ \$hhll	0D <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4
绝对 X	△ORA△ \$hhll, X	1D <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
绝对 Y	△ORA△ \$hhll, Y	19 <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
(间接 X)	△ORA△ (\$zz, X)	01 <sub>16</sub> , zz <sub>16</sub>	2	6
(间接 Y)	△ORA△ (\$zz, Y)	11 <sub>16</sub> , zz <sub>16</sub>	2	6

注. 当T标志是1时, 周期数+3。

# PHA

## Push Accumulator on stack

# PHA

操作:  $(M(S)) \leftarrow (A)$   
 $(S) \leftarrow (S) - 1$

说明: 将A的内容保存到S指向的地址的存储器, 然后S的内容减1。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ PHA	4816	1	3

**PHP**

Push Processor status on stack

**PHP**

操 作:  $(M(S)) \leftarrow (PS)$   
 $(S) \leftarrow (S) - 1$

说 明: 将PS的内容保存到S指向的地址的存储器, 然后S的内容减1。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ PHP	0816	1	3

**PLA**

PuLI Accumulator from stack

**PLA**

操作:  $(S) \leftarrow (S)+1$   
 $(A) \leftarrow (M(S))$

说明: S的内容加1,然后将S指向的地址的存储器内容保存到A。

状态标志: N : 当执行结果的位7是1时, 为1; 否则为0。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 当执行结果是0时, 为1; 否则为0。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ PLA	6816	1	4

# PLP

## PuLI Processor status from stack

# PLP

操作: (S) ← (S)+1  
(PS) ← (M(S))

说明: S的内容加1, 然后将S指向的地址的存储器内容保存到PS。

状态标志: 变为保存在堆栈的值。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ PLP	2816	1	4

注. 在执行 PLP 指令后, 必须插入 NOP 指令。

# ROL

Rotate One bit Left

# ROL



说明: 包括C标志, 将A或者M的内容左循环1位。C标志的内容保存到A或者M的位0, A或者M的位7的内容保存到C标志。

状态标志:

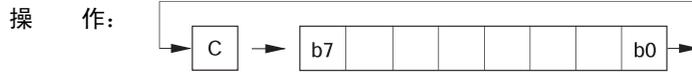
- N : 当执行前位6是1时, 为1; 否则为0。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 当执行结果是0时, 为1; 否则为0。
- C : 当执行前位7是1时, 为1; 否则为0。

寻址方式	记述格式	机器码	字节数	周期数
累加器	$\Delta$ ROL $\Delta$ A	2A <sub>16</sub>	1	2
零页	$\Delta$ ROL $\Delta$ \$zz	26 <sub>16</sub> , zz <sub>16</sub>	2	5
零页 X	$\Delta$ ROL $\Delta$ \$zz, X	36 <sub>16</sub> , zz <sub>16</sub>	2	6
绝对	$\Delta$ ROL $\Delta$ \$hhll	2E <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	6
绝对 X	$\Delta$ ROL $\Delta$ \$hhll, X	3E <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	7

# ROR

Rotate One bit Right

# ROR



说明: 包括C标志, 将A或者M的内容右循环1位。C标志的内容保存到A或者M的位0的内容保存到C标志。。

状态标志:

- N : 当执行前C标志的内容是1时, 为1; 否则为0。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 当执行结果是0时, 为1; 否则为0。
- C : 当执行前位0是1时, 为1; 否则为0。

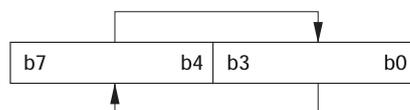
寻址方式	记述格式	机器码	字节数	周期数
累加器	$\Delta$ ROR $\Delta$ A	6A <sub>16</sub>	1	2
零页	$\Delta$ ROR $\Delta$ \$zz	66 <sub>16</sub> , zz <sub>16</sub>	2	5
零页 X	$\Delta$ ROR $\Delta$ \$zz, X	76 <sub>16</sub> , zz <sub>16</sub>	2	6
绝对	$\Delta$ ROR $\Delta$ \$hhll	6E <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	6
绝对 X	$\Delta$ ROR $\Delta$ \$hhll, X	7E <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	7

# RRF

## Rotate Right of Four bits

# RRF

操作:



说明: M的内容右循环4位。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页	$\triangle RRF \triangle \$zz$	8216, ZZ16	2	8

# RTI

## ReTurn from Interrupt

# RTI

操作: (S) ← (S)+1  
 (PS) ← (M(S))  
 (S) ← (S)+1  
 (PCL) ← (M(S))  
 (S) ← (S)+1  
 (PCH) ← (M(S))

说明: 恢复在接受中断时保存到堆栈的状态标志和PC的内容, 恢复到接受中断前的状态。  
 M(S)是S指向的地址的存储器内容。

状态标志: 变为保存在堆栈的值。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ RTI	40 <sub>16</sub>	1	6

# RTS

## ReTurn from Subroutine

# RTS

操作: (S) ← (S)+1  
 (PCL) ← (M(S))  
 (S) ← (S)+1  
 (PCH) ← (M(S))  
 (PC) ← (PC)+1

说明: 将向子程序跳转时保存到堆栈的内容存入PC, 然后PC加1。此时, PC指向JSR的下一条指令。  
 M(S)是S指向的地址的存储器内容。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ RTS	60 <sub>16</sub>	1	6

# SBC

## SuBtract with Carry

# SBC

操作: (T)=0时 (A) ← (A)−(M)−( $\bar{C}$ )  
(T)=1时 (M(X)) ← (M(X))−(M)−( $\bar{C}$ )

说明: 当T标志是0时, A的内容减去M的内容及C标志内容的补码, 结果保存到A和C标志。  
当T标志是1时, M(X)的内容减去M的内容及C标志内容的补码, 结果保存到M(X)和C标志。  
此时, 虽然A的内容不变, 但是状态标志发生变化。M(X)是X指向的地址的存储器内容。

状态标志: N : 当运算结果的位7是1时, 为1; 否则为0。  
V : 当运算结果超过+127或者|−128|时, 为1; 否则为0。  
T : 不变。  
B : 不变。  
D : 不变。  
I : 不变。  
Z : 当运算结果是0时, 为1; 否则为0。  
C : 当结果大于等于0时, 为1; 否则为0, 表示借位。

寻址方式	记述格式	机器码	字节数	周期数
立即	$\Delta$ SBC $\Delta$ # $\$nn$	E9 <sub>16</sub> , nn <sub>16</sub>	2	2
零页	$\Delta$ SBC $\Delta$ $\$zz$	E5 <sub>16</sub> , zz <sub>16</sub>	2	3
零页 X	$\Delta$ SBC $\Delta$ $\$zz, X$	F5 <sub>16</sub> , zz <sub>16</sub>	2	4
绝对	$\Delta$ SBC $\Delta$ $\$hhll$	ED <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	4
绝对 X	$\Delta$ SBC $\Delta$ $\$hhll, X$	FD <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
绝对 Y	$\Delta$ SBC $\Delta$ $\$hhll, Y$	F9 <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
(间接 X)	$\Delta$ SBC $\Delta$ ( $\$zz, X$ )	E1 <sub>16</sub> , zz <sub>16</sub>	2	6
(间接 Y)	$\Delta$ SBC $\Delta$ ( $\$zz, Y$ )	F1 <sub>16</sub> , zz <sub>16</sub>	2	6

注 1. 当 T 标志是 1 时, 周期数+3。

- 在 10 进制运算方式 (D 标志=1) 中执行 SBC 指令时, 必须在 SBC 指令后间隔一条以上的指令, 然后执行 SEC、CLC、CLD 指令。另外, 在 10 进制运算方式中, N、V、Z 标志无效。

# SEB

## SEt Bit

# SEB

操 作:  $(A_i) \leftarrow 1$  或者  $(M_i) \leftarrow 1$

说 明: 被指定的A或者M的位置1。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
累加器 位	$\triangle\text{SEB}\triangle i, A$	$(20i+B)_{16}$	1	2
零页 位	$\triangle\text{SEB}\triangle i, \$zz$	$(20i+F)_{16}, zz_{16}$	2	5

**SEC**

SEt Carry flag

**SEC**

操 作: (C) ← 1

说 明: C标志的内容置1。

状态标志: N : 不变。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 不变。  
 C : 为1。

寻址方式	记述格式	机器码	字节数	周期数
隐含	△SEC	3816	1	2

# SED

## SEt Decimal mode

# SED

操 作: (D) ← 1

说 明: D标志的内容置1。

状态标志: N : 不变。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 为1。  
 I : 不变。  
 Z : 不变。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	△SED	F816	1	2

**SEI**

## SEt Interrupt disable status

**SEI**

操 作: (I) ← 1

说 明: I标志的内容置1。

状态标志: N : 不变。  
 V : 不变。  
 T : 不变。  
 B : 不变。  
 D : 不变。  
 I : 为1。  
 Z : 不变。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	△SEI	7816	1	2

# SET

## SEt Transfer flag

# SET

操 作:  $(T) \leftarrow 1$

说 明: T标志的内容置1。

状态标志: N : 不变。  
 V : 不变。  
 T : 为1。  
 B : 不变。  
 D : 不变。  
 I : 不变。  
 Z : 不变。  
 C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ SET	3216	1	2

# STA

STore Accumulator in memory

# STA

操作:  $(M) \leftarrow (A)$ 

说明: A的内容保存到M。此时, A不变。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页	$\Delta STA \Delta \$zz$	85 <sub>16</sub> , zz <sub>16</sub>	2	4
零页 X	$\Delta STA \Delta \$zz, X$	95 <sub>16</sub> , zz <sub>16</sub>	2	5
绝对	$\Delta STA \Delta \$hhll$	8D <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5
绝对 X	$\Delta STA \Delta \$hhll, X$	9D <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	6
绝对 Y	$\Delta STA \Delta \$hhll, Y$	99 <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	6
(间接 X)	$\Delta STA \Delta (\$zz, X)$	81 <sub>16</sub> , zz <sub>16</sub>	2	7
(间接 Y)	$\Delta STA \Delta (\$zz), Y$	91 <sub>16</sub> , zz <sub>16</sub>	2	7

# STP

## SToP

# STP

操 作： 停止振荡器的振荡。

说 明： 振荡控制触发器因执行STP指令而被复位，所以振荡停止。为了重新启动，需要产生中断或者进行复位。

状态标志： 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ STP	4216	1	2

注. 对于一部分产品，STP 指令无效。如果在 STP 指令无效的情况下执行 STP 指令，其效果就和执行 NOP 指令一样（周期数是 2）。

# STX

STore index register X in memory

# STX

操作:  $(M) \leftarrow (X)$ 

说明: X的内容保存到M。此时, X不变。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页	$\Delta\text{STX}\Delta \$zz$	86 <sub>16</sub> , zz <sub>16</sub>	2	4
零页 Y	$\Delta\text{STX}\Delta \$zz, Y$	96 <sub>16</sub> , zz <sub>16</sub>	2	5
绝对	$\Delta\text{STX}\Delta \$hhll$	8E <sub>16</sub> , ll <sub>16</sub> , hh <sub>16</sub>	3	5

# STY

STore index register Y in memory

# STY

操作:  $(M) \leftarrow (Y)$ 

说明: Y的内容保存到M。此时, Y不变。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页	$\Delta\text{STY}\Delta \$zz$	8416, ZZ16	2	4
零页 X	$\Delta\text{STY}\Delta \$zz, X$	9416, ZZ16	2	5
绝对	$\Delta\text{STY}\Delta \$hhll$	8C16, ll16, hh16	3	5

**TAX**

Transfer Accumulator to index register X

**TAX**操作:  $(X) \leftarrow (A)$ 

说明: 将A的内容传送给X。此时, A不变。

状态标志:

- N : 当传送结果的位7是1时, 为1; 否则为0。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 当传送结果是0时, 为1; 否则为0。
- C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ TAX	AA16	1	2

# TAY

Transfer Accumulator to index register Y

# TAY

操作:  $(Y) \leftarrow (A)$

说明: 将A的内容传送给Y。此时, A不变。

状态标志: N : 当传送结果的位7是1时, 为1; 否则为0。

V : 不变。

T : 不变。

B : 不变。

D : 不变。

I : 不变。

Z : 当传送结果是0时, 为1; 否则为0。

C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ TAY	A816	1	2

**TST**

TeST for negative or zero

**TST**

操 作: M=0?

说 明: 测试M的内容是否为0, 改变N标志和Z标志。

状态标志:

N : 当M的位7是1时, 为1; 否则为0。

V : 不变。

T : 不变。

B : 不变。

D : 不变。

I : 不变。

Z : 当M的内容是0时, 为1; 否则为0。

C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
零页	$\Delta$ TST $\Delta$ \$zz	6416, ZZ16	2	3

# TSX

Transfer Stack pointer to index register X

# TSX

操作:  $(X) \leftarrow (S)$

说明: 将S的内容传送给X。

状态标志: N : 当执行结果的位7是1时, 为1; 否则为0。

V : 不变。

T : 不变。

B : 不变。

D : 不变。

I : 不变。

Z : 当执行结果是0时, 为1; 否则为0。

C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ TSX	BA16	1	2

**TXA**

Transfer index register X to Accumulator

**TXA**

操作: (A) ← (X)

说明: 将X的内容传送给A。

状态标志: N : 当执行结果的位7是1时, 为1; 否则为0。

V : 不变。

T : 不变。

B : 不变。

D : 不变。

I : 不变。

Z : 当执行结果是0时, 为1; 否则为0。

C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	△TXA	8A16	1	2

**TXS**

Transfer index register X to Stack pointer

**TXS**操 作:  $(S) \leftarrow (X)$ 

说 明: 将X的内容传送给S。

状态标志: 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ TXS	9A16	1	2

**TYA**

Transfer index register Y to Accumulator

**TYA**

操作: (A) ← (Y)

说明: 将Y的内容传送给A。

状态标志:

- N : 当执行结果的位7是1时, 为1; 否则为0。
- V : 不变。
- T : 不变。
- B : 不变。
- D : 不变。
- I : 不变。
- Z : 当执行结果是0时, 为1; 否则为0。
- C : 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	△TYA	9816	1	2

**WIT**

Wait

**WIT**

操 作： 停止内部时钟。

说 明： 虽然停止内部时钟，但是振荡电路的振荡不停止。为了重新启动，需要产生中断或者进行复位。

状态标志： 不变。

寻址方式	记述格式	机器码	字节数	周期数
隐含	$\Delta$ WIT	C216	1	2

## 3.4 中断处理和子程序处理的指令

### 3.4.1 中断处理指令

因为当接受中断时，处理器状态寄存器的内容就被保存到栈指针指向的存储器，所以不需要执行 PHP 指令。

如果需要保存累加器的内容，就在中断程序中执行 PHA 指令（在操作累加器的指令前执行）。如果中断程序中执行了 PHA 等栈操作指令，就必须在同一中断程序中执行与该指令对应的 PLA 等指令。

从中断程序返回时必须执行 RTI 指令。

### 3.4.2 中断控制指令

可通过中断禁止标志(I)、对应各中断源的中断允许位和中断请求位(BRK 指令的软件中断除外)来控制中断。

#### (1) 中断禁止

禁止中断有两种方法。一种是通过 SEI 指令将中断禁止标志 I 置“1”；另一种是通过 LDM、CLB 等指令(也能使用其它各种指令)将中断允许位清“0”。

#### (2) 中断允许

在允许中断时，通过 LDM、SEB 等指令将中断允许位置“1”，并且通过 CLI 指令将中断禁止标志 I 清“0”。

#### (3) 中断请求的解除

发生中断时，对应该中断源的中断请求位自动置“1”，在接受中断的同时中断请求位自动清“0”，所以在中断程序中不需要清除中断请求位。

即使在中断禁止的状态下发生中断，中断请求位也被置“1”。此时如果置为中断允许状态（中断禁止标志 I=0 并且中断允许位=1），中断就立即被接受。但是，如果不想接受该中断，就在置为中断允许状态前通过 LDM、CLB 等指令将中断请求位清“0”。此时，必须注意以下情况：

- 在中断禁止标志 I 是“0”时，如果通过 LDM 等指令同时将中断请求位和中断允许位清“0”，实际上因在中断请求位被清除前先进入中断允许状态而接受了中断。  
在这种情况下，必须通过 CLB 等指令，首先将中断请求位清“0”，然后解除中断禁止状态。

#### (4) 中断程序中的中断控制

当接受中断后进入中断程序时，中断禁止标志 I 自动置“1”，禁止多重中断。允许多重中断时，必须在中断程序中使用 CLI 指令将标志 I 清“0”。

### 3.4.3 子程序处理指令

通常使用 JSR 指令进行子程序的跳转。当执行此指令时，当前程序计数器的值按 PCH、PCL 的顺序自动保存到堆栈，栈指针也只移动该部分。但是，与中断处理时不同，因为在子程序调用时处理器状态寄存器不自动保存，所以如果需要保存处理器状态寄存器，就必须执行 PHP 指令。即使执行 JSR 指令，处理器状态寄存器的内容也不发生变化，所以在 JSR 指令之前或者之后（子程序的起始位置）都能通过 PHP 指令保存处理器状态寄存器的内容。但是，如果在子程序中执行这样的栈操作指令，就必须在从该子程序返回前（在子程序中）进行相反的栈操作。

要从子程序返回时执行 RTS 指令。当执行此指令时，通过 JSR 指令保存的返回地址自动恢复到程序计数器。此时也与中断处理时不同，不恢复处理器状态寄存器的内容。如果在子程序中通过 PHP 指令、PHA 指令保存了处理器状态寄存器和累加器，就必须在从子程序返回前（在子程序中）进行 PLP 指令、PLA 指令等相反的栈操

作。

中断和子程序处理时的堆栈保存和恢复操作如图 3.2 所示，累加器和处理器状态寄存器的保存指令和恢复指令如表 3.1 所示。

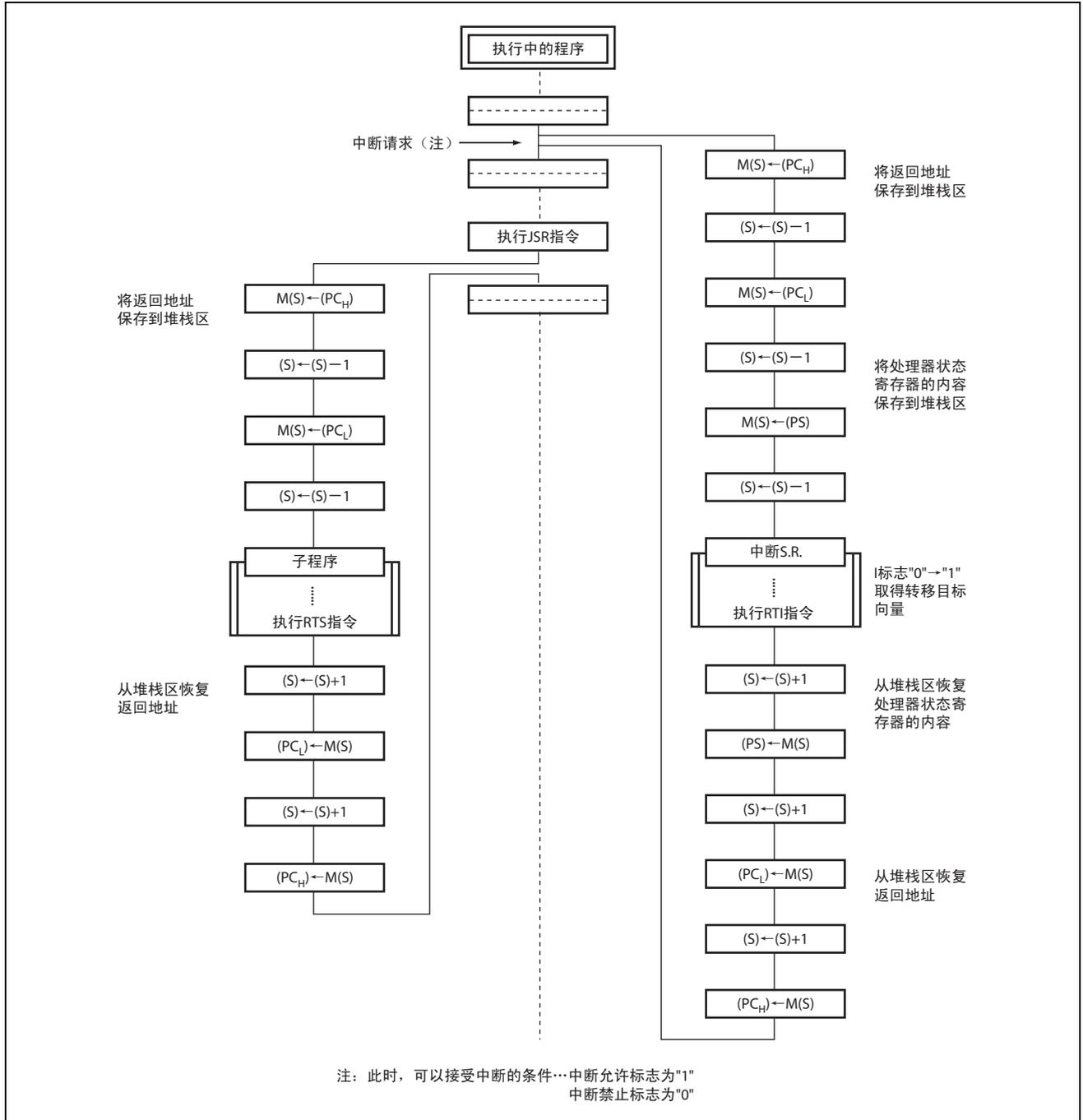


图3.2 堆栈保存和恢复操作

表3.1 累加器和处理器状态寄存器的保存指令和恢复指令

	保存到堆栈的指令	从堆栈恢复的指令
累加器	PHA	PLA
处理器状态寄存器	PHP	PLP

---

## 第4章 使用注意事项

---

在此只说明 740 族共同的注意事项，请同时参照所用产品的注意事项。

### 4.1 输入/输出端口的注意事项

#### 4.1.1 在待机状态的使用

在待机状态下\*<sup>1</sup>使用时，请不要将输入/输出端口的输入电平置为不稳定状态，尤其要注意 N 沟道漏极开路的输入/输出端口。另外，即使在将 N 沟道漏极开路的输入/输出端口设定为输出端口时，也同样必须注意。

此时，必须通过电阻上拉（连接 Vcc）或者下拉（连接 Vss）端口。

在决定电阻值时，请注意以下 2 点：

- 外接电路
- 通常运行时输出电平的波动

#### ●理由

通过方向寄存器设定为输入端口时，因为晶体管为 OFF 状态，所以端口为高阻抗状态。此时，如果将输入电平置为不稳定状态，由于输入到单片机内部的输入缓冲器的电位不稳定，所以就有可能产生电源电流的流动。

另外，在 N 沟道漏极开路的输入/输出端口的端口锁存器内容是“1”时，即使通过方向寄存器设定为输出端口，也会发生与输入端口同样的现象。

\*<sup>1</sup> 待机状态：执行 STP 指令的停止模式  
执行 WIT 指令的等待模式

#### 4.1.2 通过位处理指令改写输出数据

在使用位处理指令\*改写输入/输出端口的端口锁存器时，未指定的位的值有可能发生变化。

#### ●理由

位处理指令是读-改-写形式的指令，以字节为单位进行读和写操作。因此，如果对输入/输出端口的端口锁存器的某位执行此指令，就将对该端口锁存器的全部位进行以下的处理：

- 被设定为输入的位：  
CPU 读管脚的值，在位处理后写入此位。
- 被设定为输出的位：  
CPU 读端口锁存器的位的值，在位处理后写入此位。但要注意以下几点：
  - 即使将被设定为输出的端口改变为输入端口，输出数据也被保存在端口锁存器。
  - 对于被设定为输入的端口锁存器的位，即使在位处理指令未指定时，如果管脚和端口锁存器的内容不同，位的值也发生变化。

\* 位处理指令：SEB 指令、CLB 指令

## 4.2 未使用管脚处理的注意事项

### 4.2.1 未使用管脚的适当处理

#### ①输出专用管脚

必须开路。

#### ②输入专用管脚

各管脚必须通过 1~10kΩ 的电阻连接到 V<sub>CC</sub> 或者 V<sub>SS</sub>。另外，电压电平对运行模式有影响的管脚（CNV<sub>SS</sub> 管脚、INT 管脚等）必须根据模式连接 V<sub>CC</sub> 或者 V<sub>SS</sub>。

#### ③输入/输出端口

在设定为输入模式时，必须通过 1~10kΩ 电阻连接到 V<sub>CC</sub> 或者 V<sub>SS</sub>；在设定为输出模式时，必须在“L”或者“H”输出状态置为开路。

- 在设定为输出模式且开路时，复位后初始状态的输入模式一直保持到由程序将端口转换为输出模式为止。为此，管脚的电压电平不稳定，在端口为输入模式期间，电源电流有可能增加。有关对系统的影响，用户必须进行充分的系统评价。
- 考虑到由噪声或者噪声引起的失控等使方向寄存器发生变化的情况，可定期地通过程序重新设定方向寄存器来提高程序的可靠性。

#### ④不使用 A/D 转换器时的 A/D 转换电源管脚 AV<sub>SS</sub>

当不使用 A/D 转换器时，必须对 A/D 转换的电源管脚 AV<sub>SS</sub>、AV<sub>CC</sub> 进行如下的处理：

- AV<sub>SS</sub>：连接 V<sub>SS</sub>
- AV<sub>CC</sub>：连接 V<sub>CC</sub>

### 4.2.2 处理时的注意事项

#### ①输入端口和输入/输出端口

在输入模式的情况下不能开路。

理由：

- 根据前级电路，电源电流有可能增加。
- 与上述的适当处理的②③相比，容易受噪声的影响。

#### ②输入/输出端口

如果设定为输入模式，不要直接连接 V<sub>CC</sub> 或者 V<sub>SS</sub>。

理由：

在由失控、噪声等引起方向寄存器变为输出模式时有可能短路。

#### ③输入/输出端口

如果设定为输入模式，不要将多个端口一起通过电阻连接到 V<sub>CC</sub> 或者 V<sub>SS</sub>。

理由：

在由失控、噪声引起方向寄存器变为输出模式时，有可能在端口之间短路。

- 在处理未使用管脚时，必须以最短的布线（20mm 以内）处理。

## 4.3 中断的注意事项

### 4.3.1 中断请求位和允许位的设定

必须按照以下的顺序用各自的指令设定中断请求位和中断允许位：

- ①中断请求位清“0”（没有中断请求）。
- ②中断允许位置“1”（允许中断）。

#### ●理由

如果用1条指令进行上述的设定，因为在中断请求位变为“0”之前中断允许位变为“1”（允许中断），所以就会执行不需要的中断处理程序。

### 4.3.2 更改关联寄存器的设定

在设定外部中断的有效边沿或者在切换多个中断源共用同一中断向量的中断源时，如果不需要发生与这些设定同步的中断，就必须按以下的步骤设定：

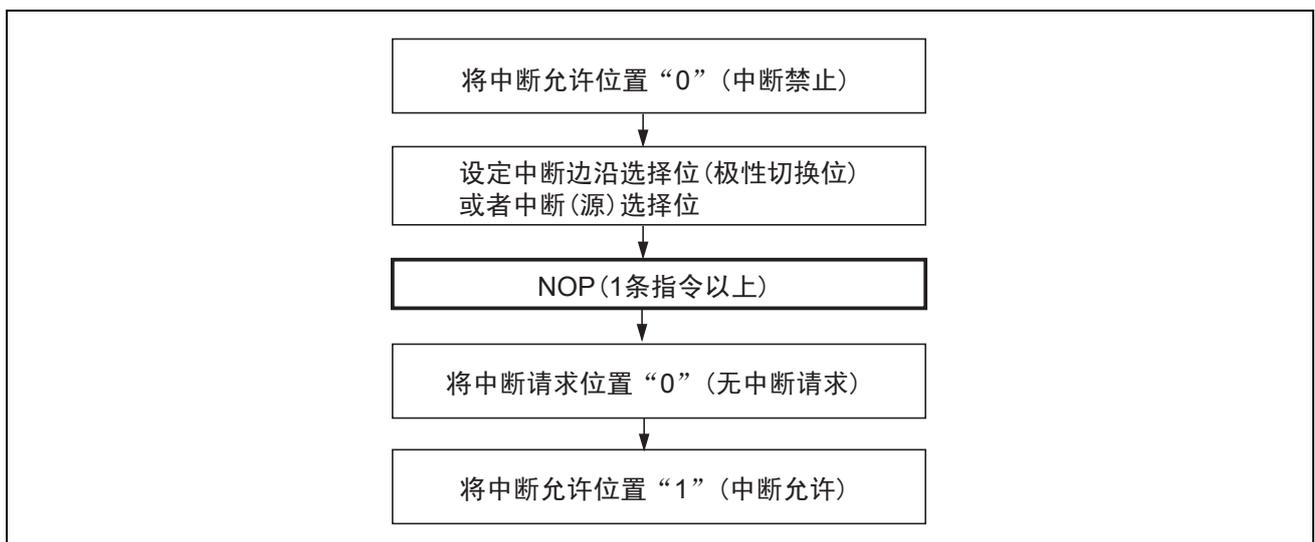


图4.1 关联寄存器的设定更改步骤

#### ●理由

在以下的情况下，中断请求位有可能变为“1”：

- 当设定外部中断的有效边沿时
- 当切换多个中断源共用同一中断向量的中断源时

### 4.3.3 中断请求位的判断

在将中断请求寄存器的中断请求位清“0”后，立即对此中断请求位执行 BBC 指令或者 BBS 指令时，必须在执行 BBC 指令或者 BBS 指令前执行 1 条指令。

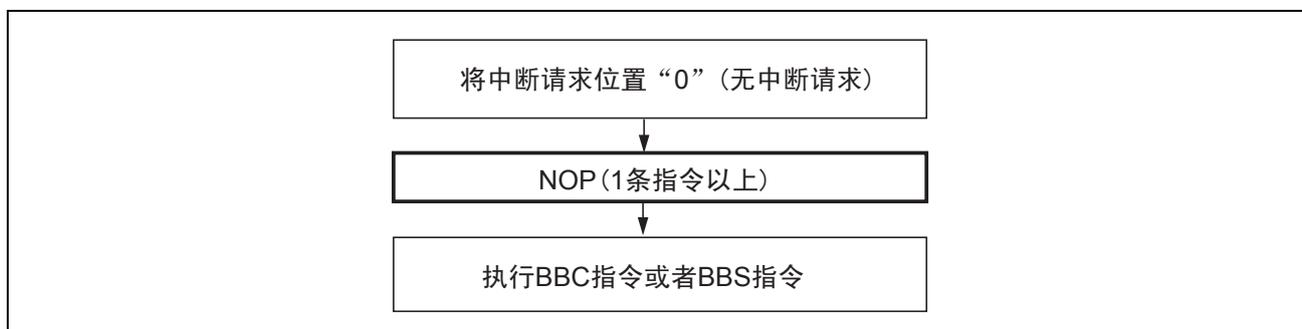


图4.2 中断请求位的判断步骤

●理由

如果在中断请求寄存器的中断请求位清“0”后立即执行 BBC 指令或者 BBS 指令，就判断变为“0”之前的中断请求位的值。

## 4.4 编程的注意事项

### 4.4.1 处理器状态寄存器

#### (1) 处理器状态寄存器的初始化

需要预先对程序的执行有影响的处理器状态寄存器（PS）的标志进行初始化。尤其是标志 T 和标志 D 影响到运算，所以必须初始化。

●理由

处理器状态寄存器（PS）除了标志 I 是“1”以外，复位后的值不定。

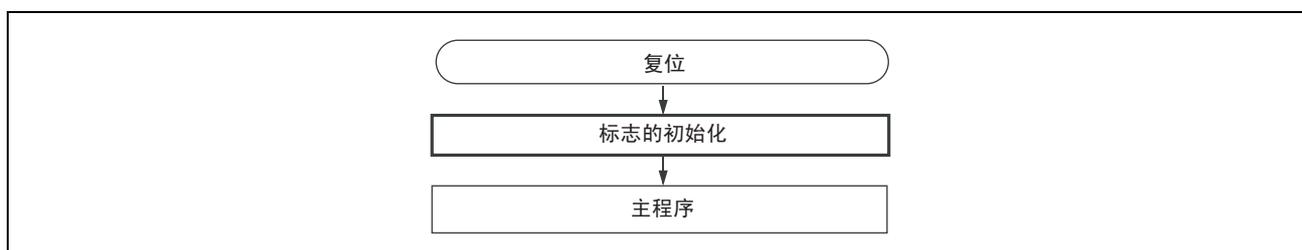


图4.3 处理器状态寄存器标志的初始化

### (2) 处理器状态寄存器的参照方法

当要参照处理器状态寄存器 (PS) 的内容时, 在执行一次 PHP 指令后读(S)+1 的内容, 并且在必要时执行 PLP 指令恢复被保存的 PS。

在执行 PLP 指令后, 必须插入 NOP 指令。

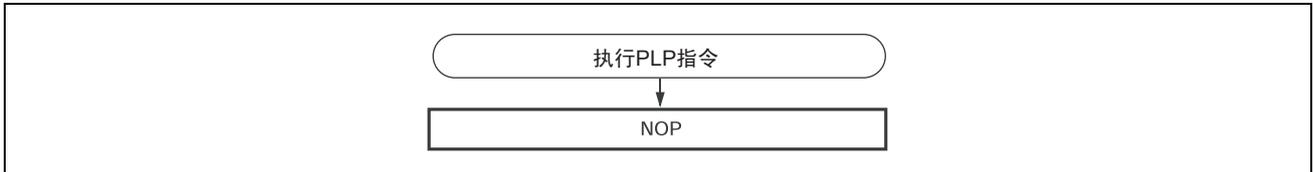


图4.4 执行PLP指令时的步骤

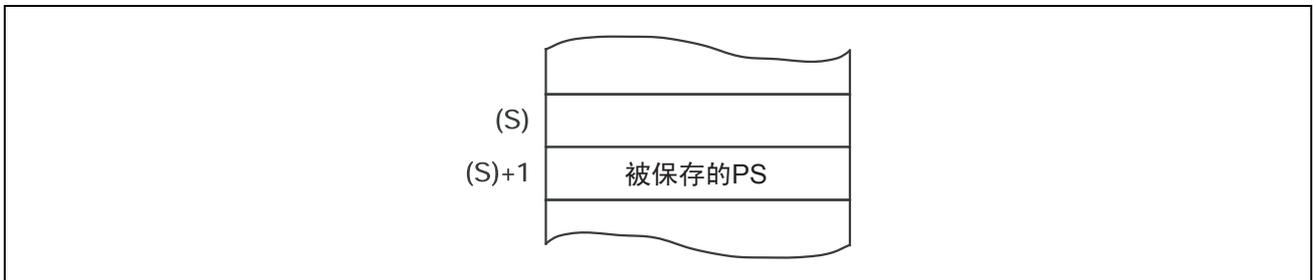


图4.5 执行PHP指令后的堆栈存储器的内容

## 4.4.2 BRK 指令

### (1) 中断源的识别方法

当 BRK 指令和其它中断源是相同的中断向量时, 能根据被保存的标志 B 的内容判断中断发生源是 BRK 指令的中断还是优先级最低的中断, 但是必须在中断程序中进行。

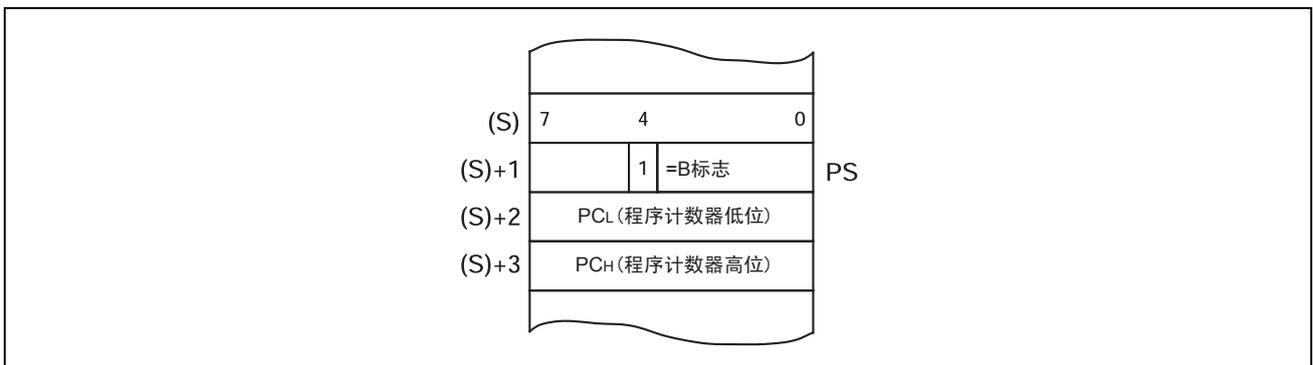


图4.6 中断处理程序中的堆栈存储器的内容

### (2) 中断优先级

如果在下述2种状态下执行 **BRK** 指令,就从该中断源中优先级最高的中断源的中断向量地址开始执行中断:

- 中断请求位和中断允许位都为“1”
- 将标志 I 置“1”后禁止中断

## 4.4.3 10 进制运算

### (1) 10 进制运算指令

在进行 10 进制运算时,通过 **SED** 指令将 10 进制模式标志 **D** 置“1”,然后执行 **ADC** 指令或者 **SBC** 指令。此时,必须在 **ADC** 指令或者 **SBC** 指令后的一条指令之后执行 **SEC** 指令、**CLC** 指令或者 **CLD** 指令。

### (2) 10 进制运算的状态标志

在 10 进制模式 (**D** 标志=1) 时执行 **ADC**、**SBC** 指令后,状态标志中的 3 个 **N**、**V**、**Z** 标志无效。

另外,**C** (进位) 标志在运算结果产生进位时被置“1”,产生借位时被清“0”,所以 **C** (进位) 标志能用作判断运算结果的进位或借位的标志。必须在运算前进行 **C** 标志的初始化。

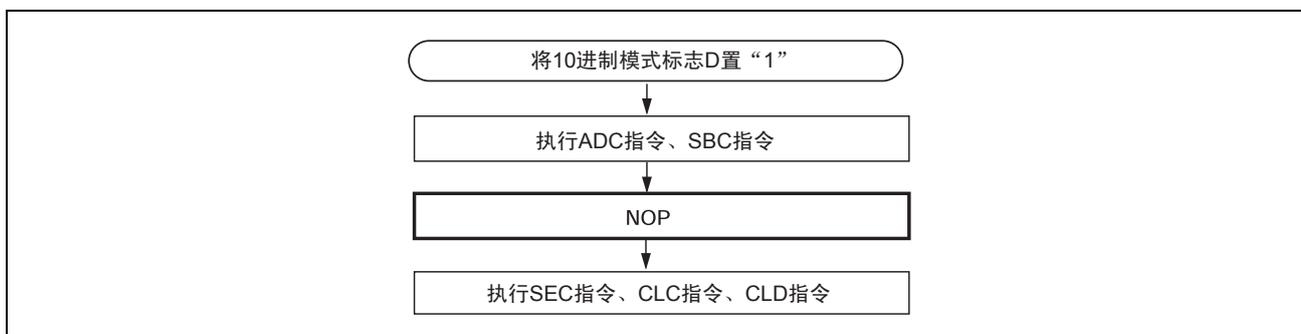


图4.7 10进制运算的状态标志

## 4.4.4 JMP 指令

在使用 **JMP** 指令 (间接寻址方式) 时,不能将低 8 位为“**FF<sub>16</sub>**”的地址指定为操作数。

## 4.4.5 乘除指令

- **MUL**、**DIV** 指令不受标志 **T** 和标志 **D** 的影响。
- 在执行乘除指令时,处理器状态寄存器的内容不变。

## 4.4.6 端口

不能读端口方向寄存器的值。即,不能使用 **LDA** 指令、标志 **T** 为“1”的存储器运算指令、将方向寄存器的值作为变址值的寻址方式、**BBC** 和 **BBS** 等位操作指令。另外,也不能使用 **CLB**、**SEB** 等位操作指令和 **ROR** 运算等方向寄存器的读-改-写指令。必须使用 **LDM** 指令、**STA** 指令等设定方向寄存器。

## 4.4.7 指令的执行时间

指令的执行时间能通过机器指令一览表中记载的周期数乘内部时钟  $\phi$  的周期得到。

## 附录 1 按寻址方式的执行时序

740 族将时钟  $\phi$  作为基准时钟运行。

在每次取指令时，只输出一个时钟  $\phi$  周期的 SYNC 信号。在输出 SYNC 信号的同时，输出各指令的起始地址(PC)。在 PC 输出期间，后半个时钟  $\phi$  周期为取操作码期间。

CPU 通过对操作码的解码，在认识构成该指令的字节数的同时决定以后的操作。

以下按寻址方式表示各指令的操作转移状态。

必要的字节数和周期数分别表示在各指令所需的字节数和时钟  $\phi$  的周期数。

图中的  $\phi$ 、SYNC、 $R/\bar{W}$  ( $\bar{RD}$ 、 $\bar{WR}$ )、ADDR (ADDRH、ADDRL)、DATA 的各个信号表示内部总线的状态。这些信号不能在单芯片模式中直接确认，但是可以在微处理器模式中确认。

这些信号的组合因产品而不同，在各产品的有效信号如下所示：

在各产品的有效信号

产品	$\phi$	SYNC	$R/\bar{W}$	$\bar{RD}$	$\bar{WR}$	ADDR	DATA	ADDRH	ADDRL/DATA
M507XX M509XX M374XX (M37451除外)	○	○	○			○	○		
M38XXX M375XX M372XX M371XX	○	○		○	○	○	○		
M37451	○	○	○	○(注)	○(注)	○	○		
M50734	○	○		○	○			○	○

注. 只限 80 管脚版。

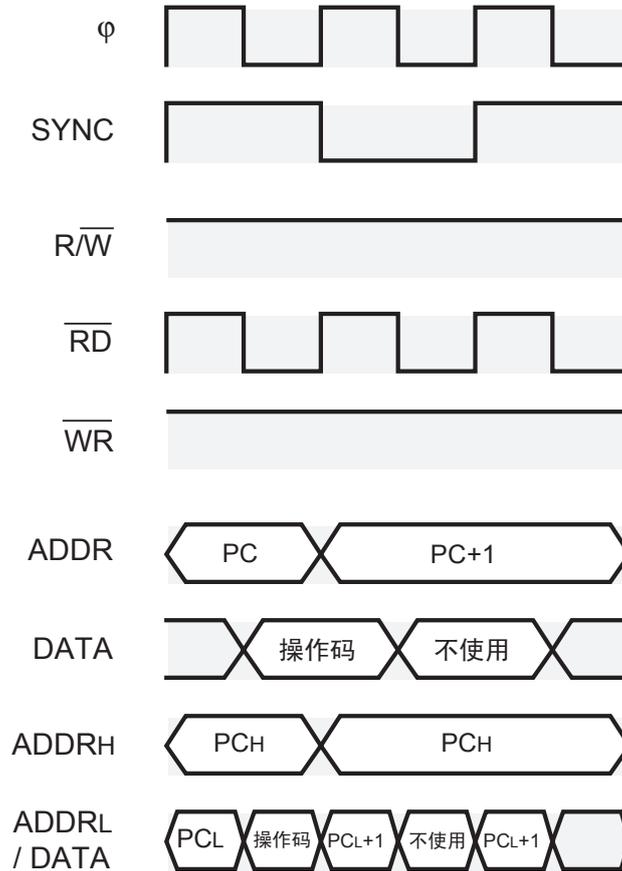
隐含 (Implied)

指 令:  $\Delta$ CLC       $\Delta$ SEC  
 $\Delta$ CLD       $\Delta$ SED  
 $\Delta$ CLI       $\Delta$ SEI  
 $\Delta$ CLT       $\Delta$ SET  
 $\Delta$ CLV       $\Delta$ TAX  
 $\Delta$ DEX       $\Delta$ TAY  
 $\Delta$ DEY       $\Delta$ TSX  
 $\Delta$ INX       $\Delta$ TXA  
 $\Delta$ INY       $\Delta$ TXS  
 $\Delta$ NOP       $\Delta$ TYA

必要字节数: 1

必要周期数: 2

时 序:



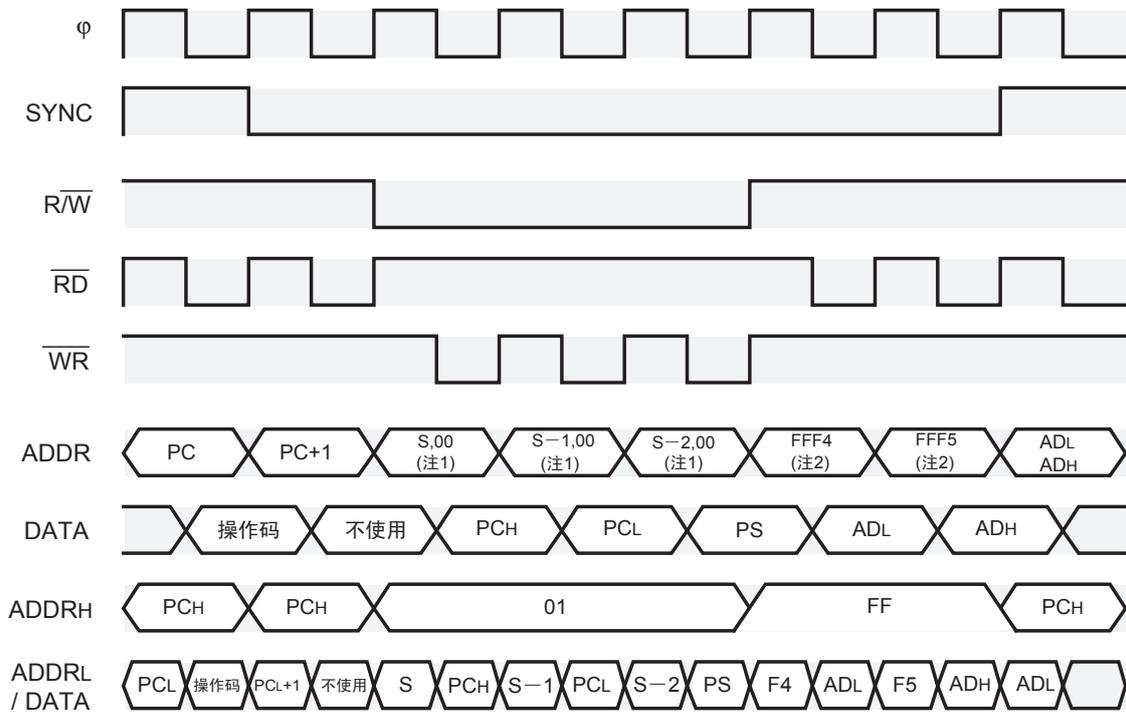
## 隐含 (Implied)

指 令:  $\Delta$ BRK

必要字节数: 1

必要周期数: 7

时 序:



注 1. 根据产品, 为“01”或者 SPS 标志的内容。

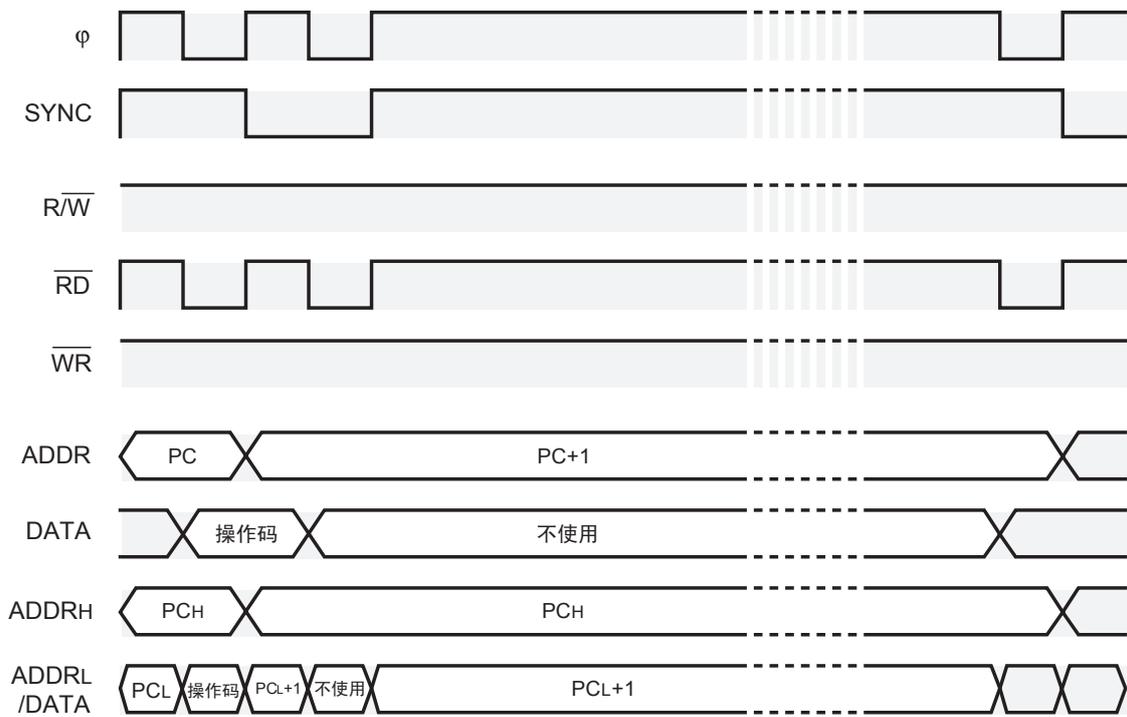
注 2. 地址有可能根据产品而不同。

隐含 (Implied)

指令:  $\Delta$ STP  
 $\Delta$ WIT

必要字节数: 1

时序:



通过外部中断，从待机状态返回。  
 通过内/外部中断，从等待状态返回。

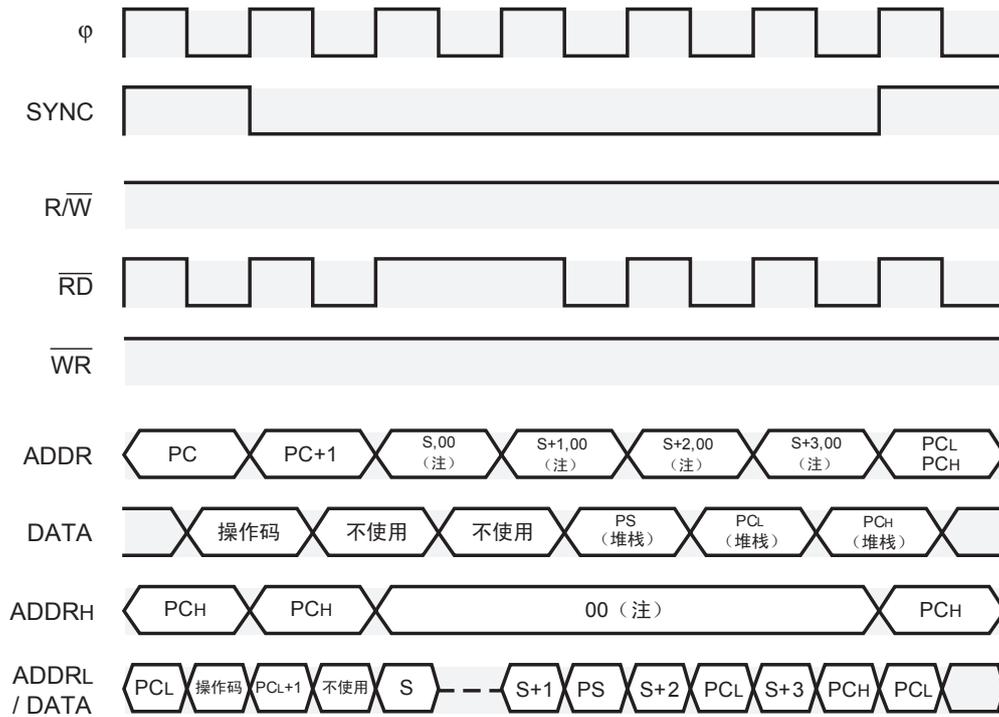
## 隐含 (Implied)

指 令:  $\Delta RTI$ 

必要字节数: 1

必要周期数: 6

时 序:



注. 根据产品, 为“01”或者 SPS 标志的内容。

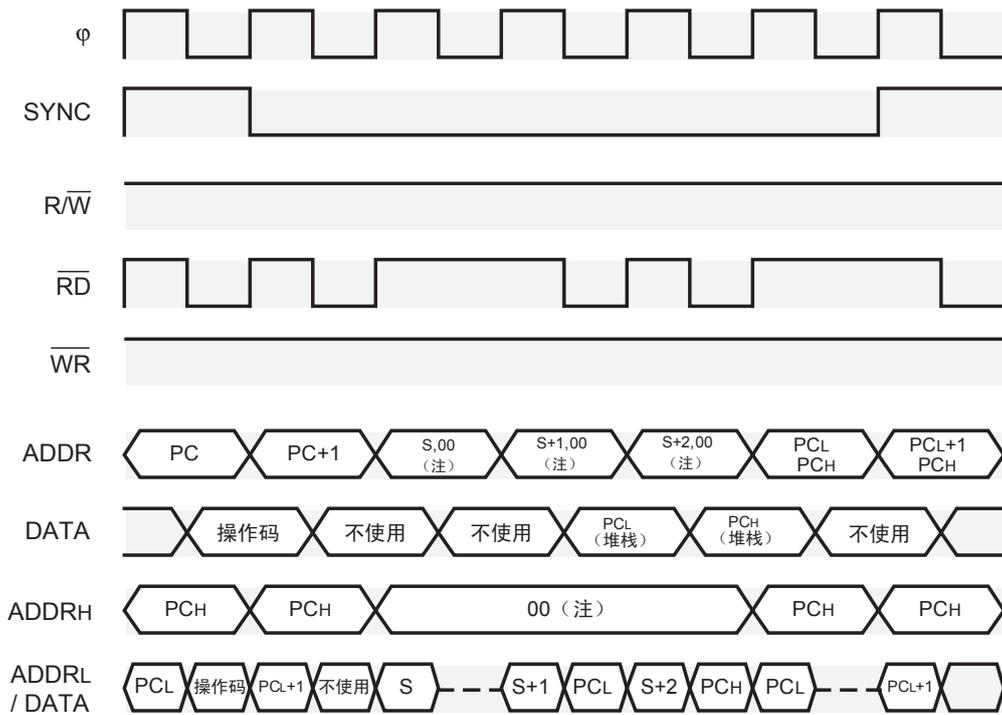
隐含 (Implied)

指 令:  $\Delta$ RTS

必要字节数: 1

必要周期数: 6

时 序:



注. 根据产品, 为“01”或者 SPS 标志的内容。

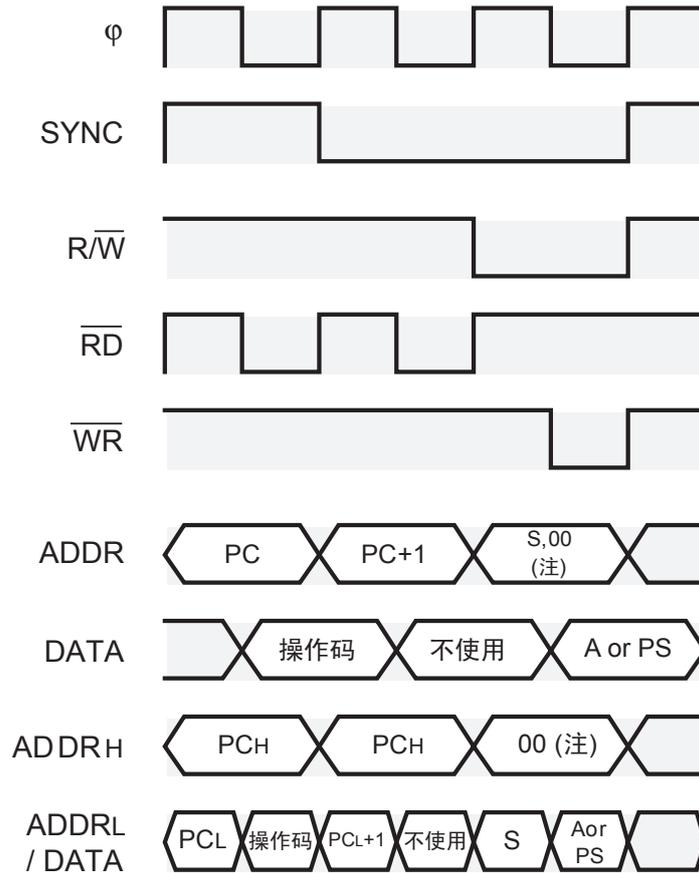
## 隐含 (Implied)

指 令:  $\Delta$ PHA  
 $\Delta$ PHP

必要字节数: 1

必要周期数: 3

时 序:



注. 根据产品, 为“01”或者 SPS 标志的内容。

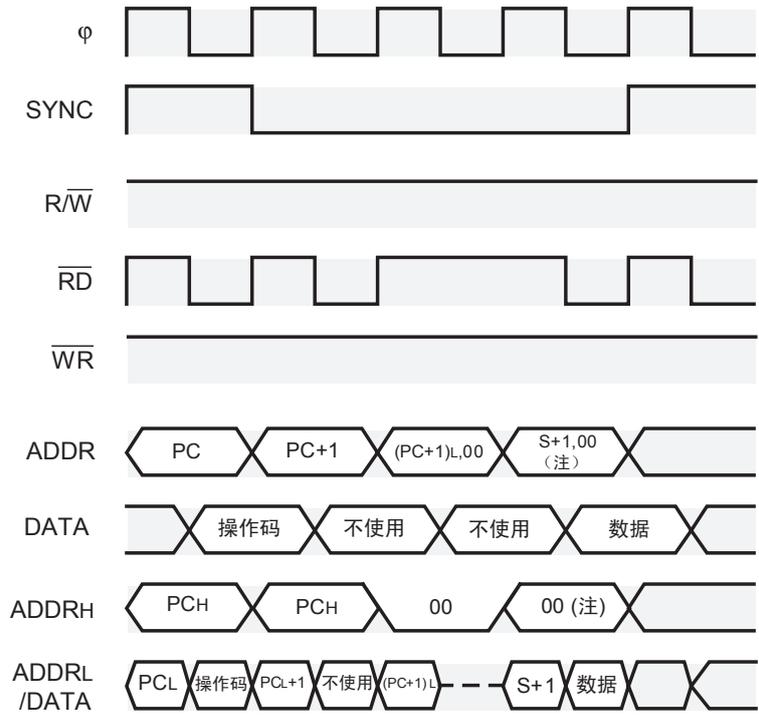
隐含 (Implied)

指令:  $\Delta$ PLA  
 $\Delta$ PLP

必要字节数: 1

必要周期数: 4

时序:



注. 根据产品, 为“01”或者 SPS 标志的内容。

[T=0]

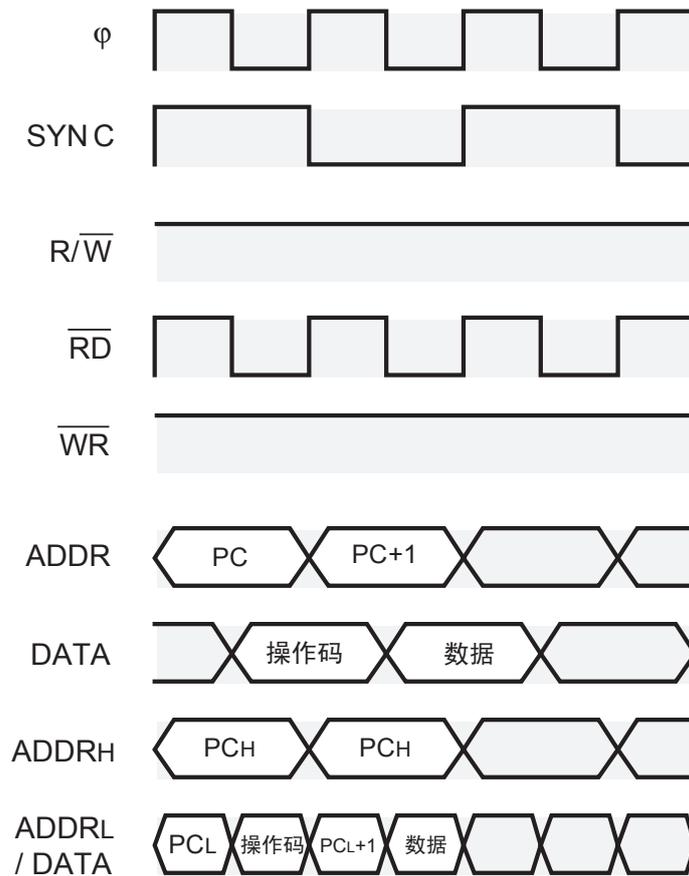
## 立即 (Immediate)

指 令:  $\Delta\text{ADC}\Delta\# \$nn$  (T=0)  
 $\Delta\text{AND}\Delta\# \$nn$  (T=0)  
 $\Delta\text{CMP}\Delta\# \$nn$  (T=0)  
 $\Delta\text{CPX}\Delta\# \$nn$   
 $\Delta\text{CPY}\Delta\# \$nn$   
 $\Delta\text{EOR}\Delta\# \$nn$  (T=0)  
 $\Delta\text{LDA}\Delta\# \$nn$  (T=0)  
 $\Delta\text{LDX}\Delta\# \$nn$   
 $\Delta\text{LDY}\Delta\# \$nn$   
 $\Delta\text{ORA}\Delta\# \$nn$  (T=0)  
 $\Delta\text{SBC}\Delta\# \$nn$  (T=0)

必要字节数: 2

必要周期数: 2

时 序:



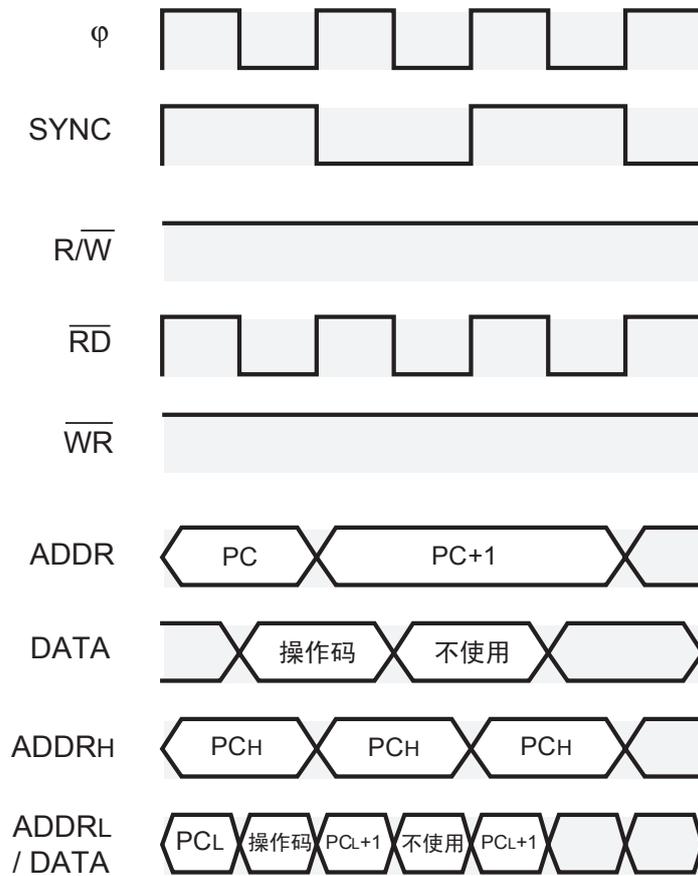
累加器 (Accumulator)

指 令:  $\triangle ASL \triangle A$   
 $\triangle DEC \triangle A$   
 $\triangle INC \triangle A$   
 $\triangle LSR \triangle A$   
 $\triangle ROL \triangle A$   
 $\triangle ROR \triangle A$

必要字节数: 1

必要周期数: 2

时 序:



## 累加器 位 相对 (Accumulator bit relative)

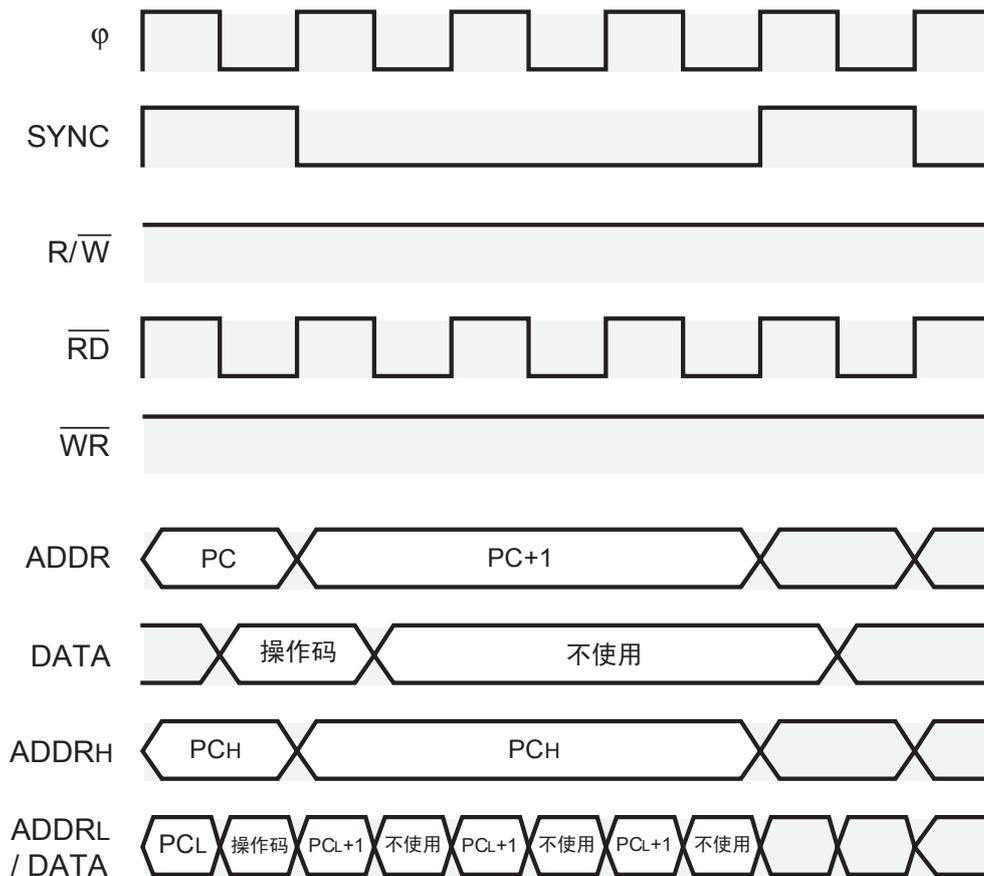
指 令:  $\Delta\text{BBC}\Delta i, A, \$\text{hhl}$   
 $\Delta\text{BBS}\Delta i, A, \$\text{hhl}$

必要字节数: 2

(1) 不转移时

必要周期数: 4

时 序:



累加器 位 相对 (Accumulator bit relative)

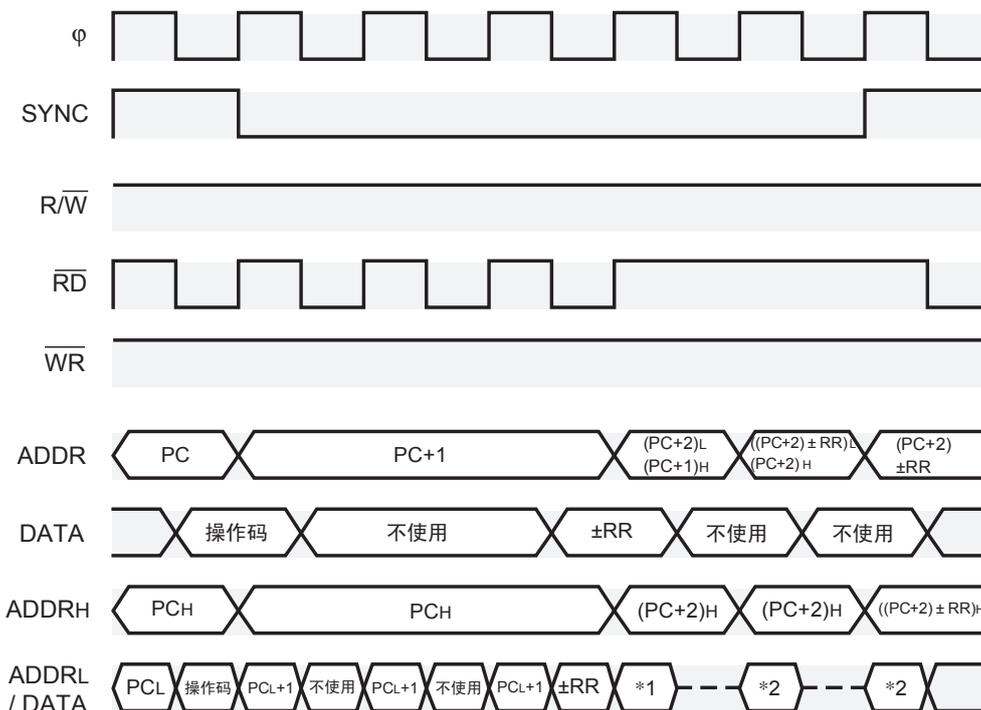
指 令:  $\Delta BBC \Delta i, A, \$hhll$   
 $\Delta BBS \Delta i, A, \$hhll$

必要字节数: 2

(2) 转移时

必要周期数: 6

时 序:



RR: 偏移量的值

\*1 :  $(PC+1)_L$

\*2 :  $((PC+2) \pm RR)_L$

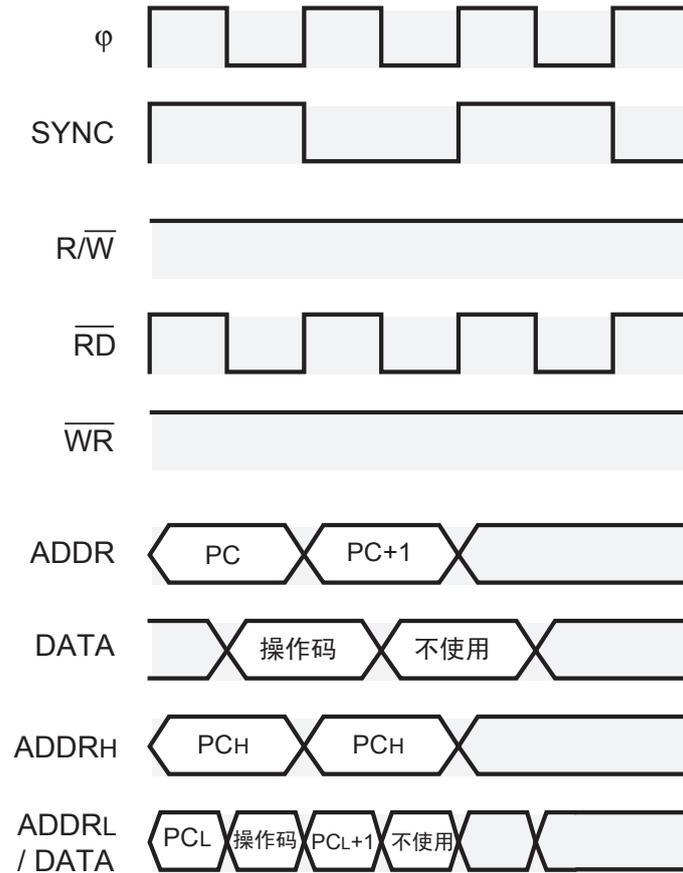
## 累加器 位 (Accumulator bit)

指 令:  $\Delta\text{CLB}\Delta i, A$   
 $\Delta\text{SEB}\Delta i, A$

必要字节数: 1

必要周期数: 2

时 序:



位 相对 (Bit relative)

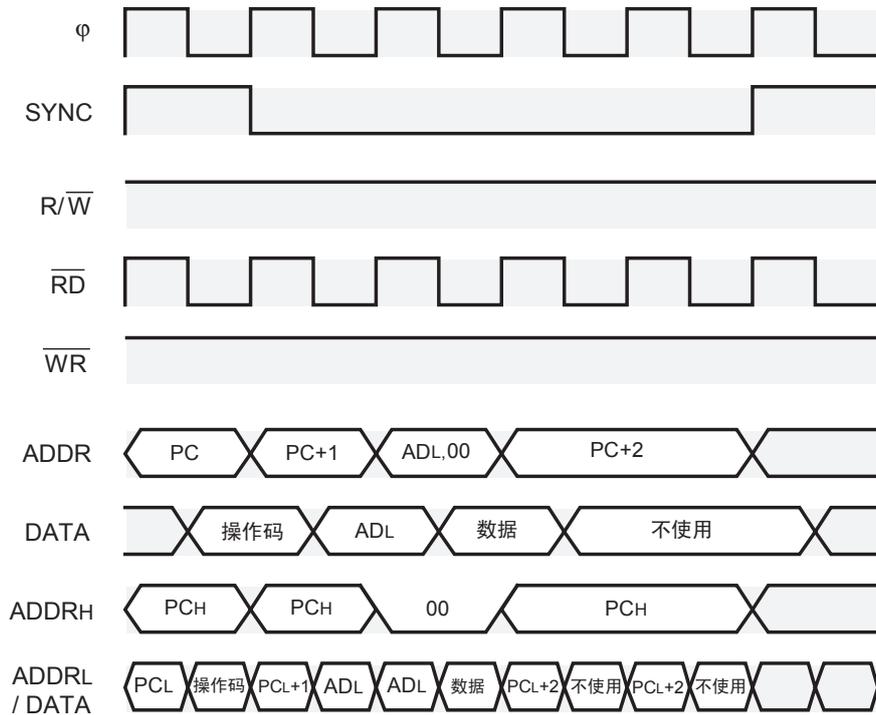
指 令:  $\Delta\text{BBC}\Delta i, \$zz, \$hhl$   
 $\Delta\text{BBS}\Delta i, \$zz, \$hhl$

必要字节数: 3

(1) 不转移时

必要周期数: 5

时 序:



## 位 相对 (Bit relative)

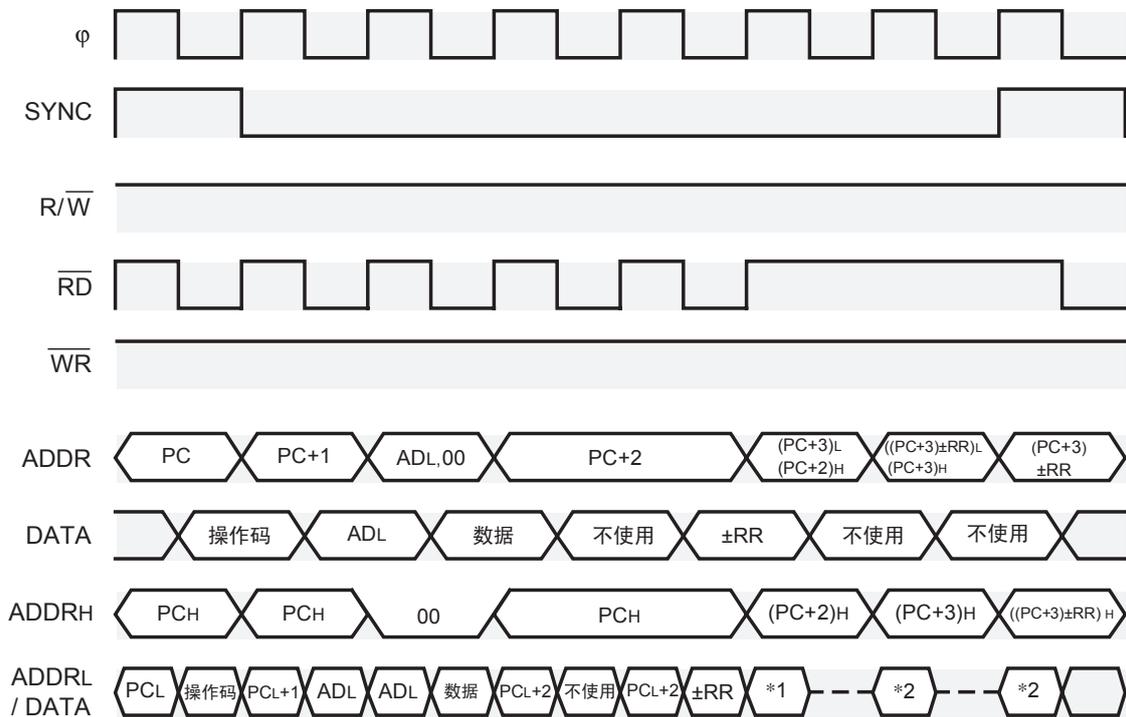
指 令:  $\Delta\text{BBC}\Delta i, \$zz, \$hhl$   
 $\Delta\text{BBS}\Delta i, \$zz, \$hhl$

必要字节数: 3

(2) 转移时

必要周期数: 7

时 序:



RR: 偏移量的值

\*1 : (PC+3)L

\*2 : ((PC+3)±RR)L

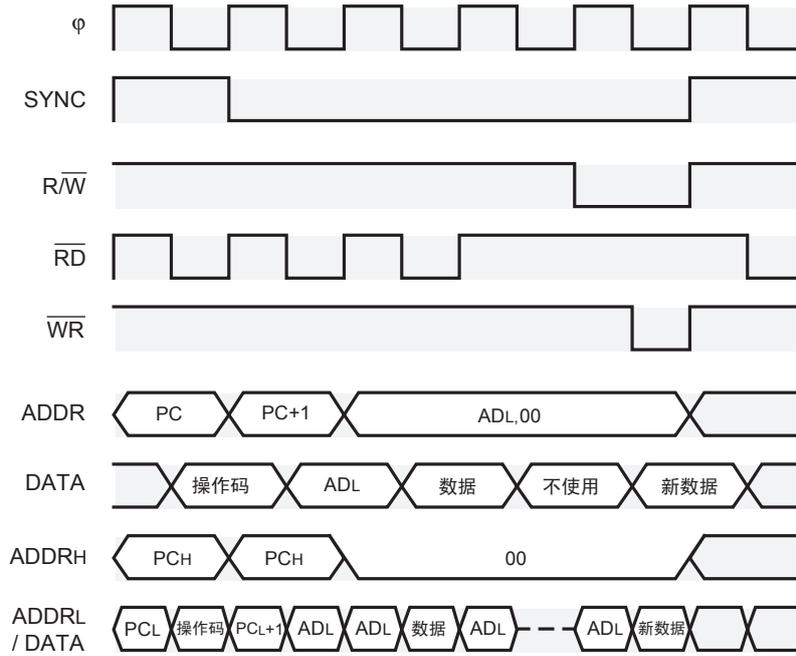
零页位 (Zero page bit)

指令:  $\Delta\text{CLB}\Delta i, \$zz$   
 $\Delta\text{SEB}\Delta i, \$zz$

必要字节数: 2

必要周期数: 5

时序:



[T=0]

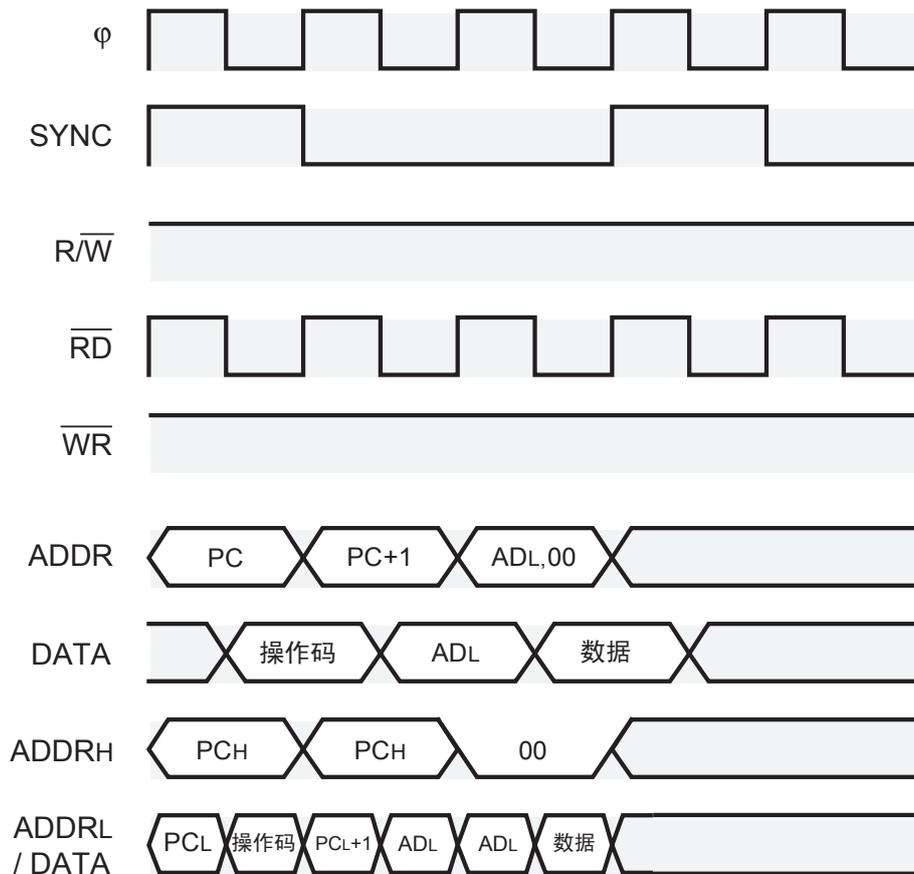
## 零页 (Zero page)

指令:  $\Delta$ ADC  $\Delta$  \$zz (T=0)  
 $\Delta$ AND  $\Delta$  \$zz (T=0)  
 $\Delta$ BIT  $\Delta$  \$zz  
 $\Delta$ CMP  $\Delta$  \$zz (T=0)  
 $\Delta$ CPX  $\Delta$  \$zz  
 $\Delta$ CPY  $\Delta$  \$zz  
 $\Delta$ EOR  $\Delta$  \$zz (T=0)  
 $\Delta$ LDA  $\Delta$  \$zz (T=0)  
 $\Delta$ LDX  $\Delta$  \$zz  
 $\Delta$ LDY  $\Delta$  \$zz  
 $\Delta$ ORA  $\Delta$  \$zz (T=0)  
 $\Delta$ SBC  $\Delta$  \$zz (T=0)  
 $\Delta$ TST  $\Delta$  \$zz

必要字节数: 2

必要周期数: 3

时序:



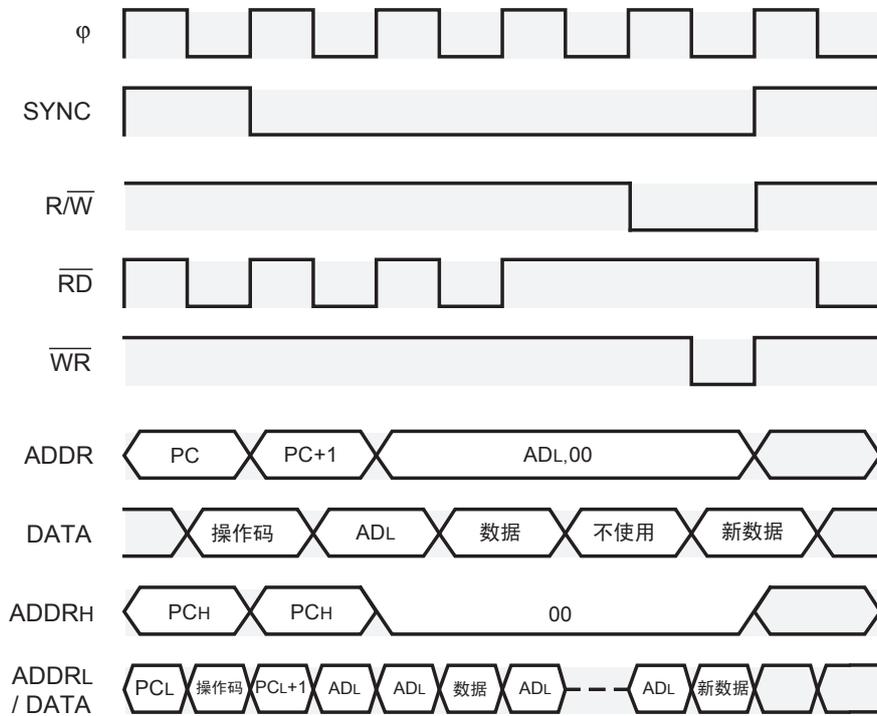
零页 (Zero page)

指 令:  $\Delta ASL \Delta \$ zz$   
 $\Delta COM \Delta \$ zz$   
 $\Delta DEC \Delta \$ zz$   
 $\Delta INC \Delta \$ zz$   
 $\Delta LSR \Delta \$ zz$   
 $\Delta ROL \Delta \$ zz$   
 $\Delta ROR \Delta \$ zz$

必要字节数: 2

必要周期数: 5

时 序:



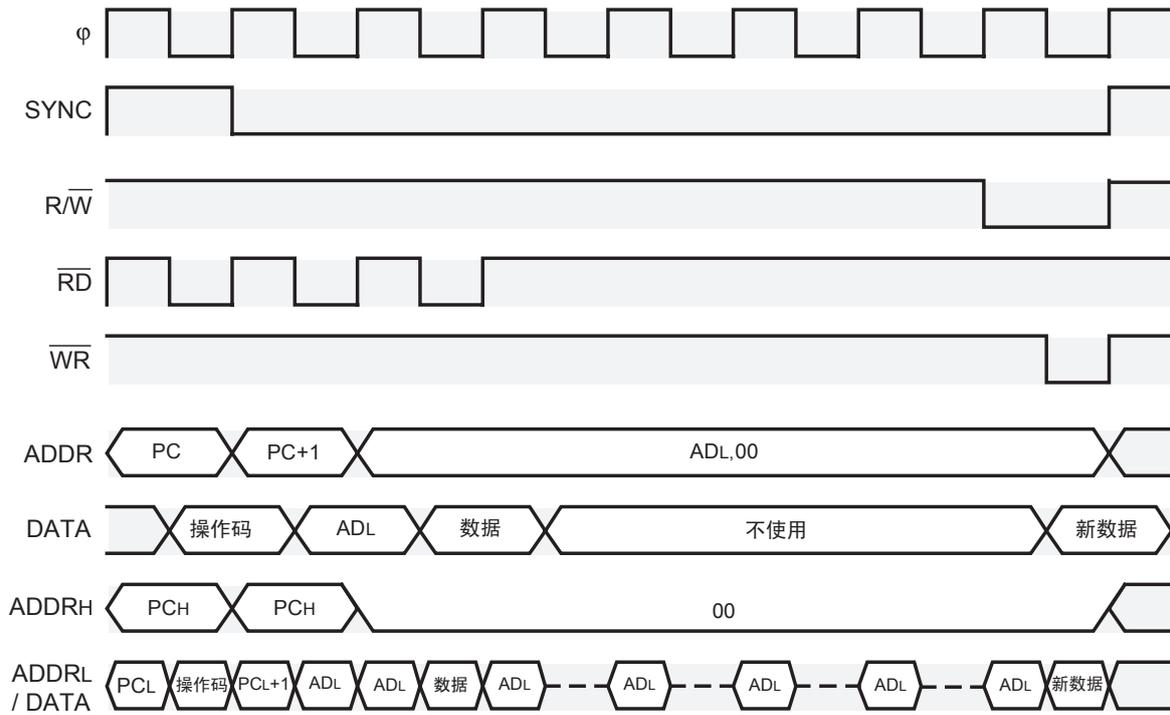
## 零页 (Zero page)

指 令:  $\Delta RRF \Delta \$ zz$ 

必要字节数: 2

必要周期数: 8

时 序:



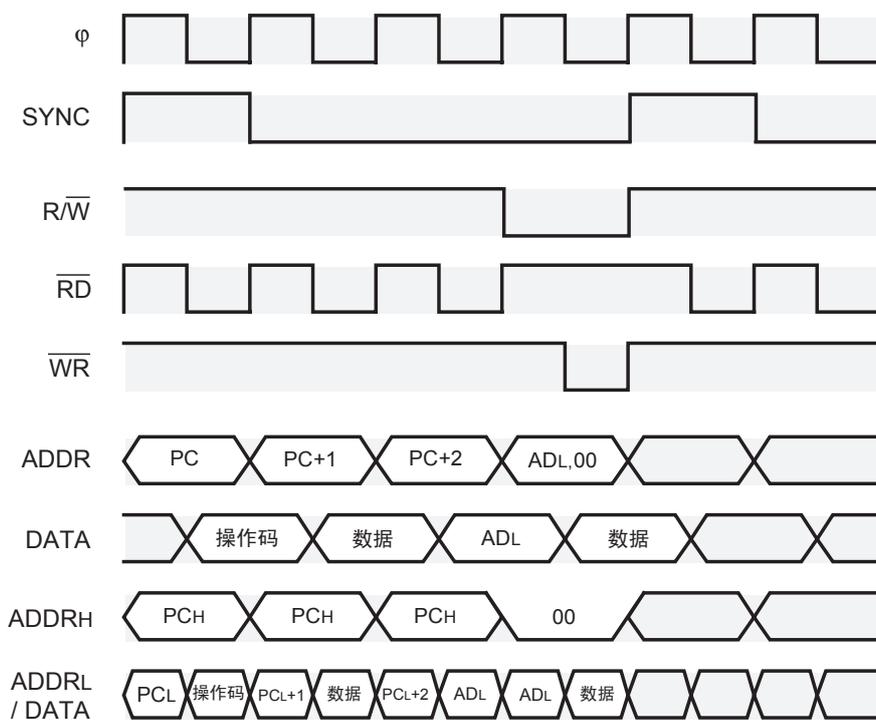
零页 (Zero page)

指令:  $\Delta\text{LDM}\Delta\# \$nn, \$zz$

必要字节数: 3

必要周期数: 4

时序:



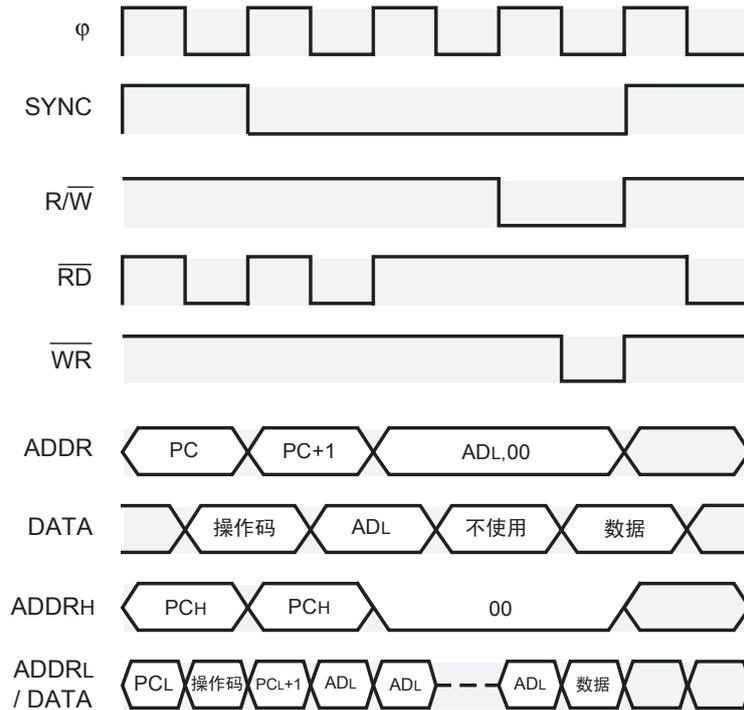
## 零页 (Zero page)

指 令:  $\Delta STA \Delta \$zz$   
 $\Delta STX \Delta \$zz$   
 $\Delta STY \Delta \$zz$

必要字节数: 2

必要周期数: 4

时 序:



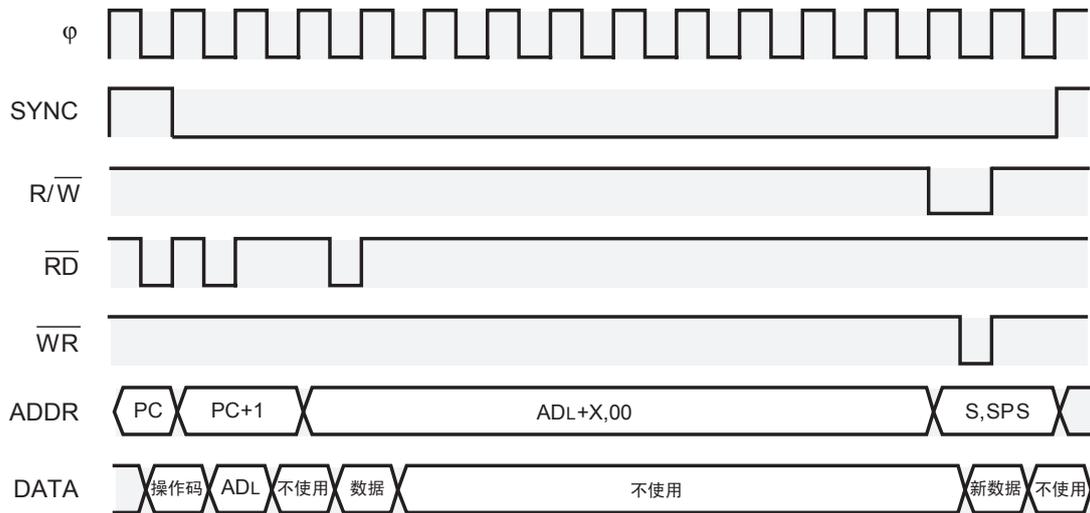
零页 X (Zero page X)

指令:  $\Delta\text{MUL}\Delta \$zz, X$

必要字节数: 2

必要周期数: 15

时序:



SPS: 通过 CPU 模式寄存器的堆栈页选择位选择的页

注. 部分产品不能使用此指令。

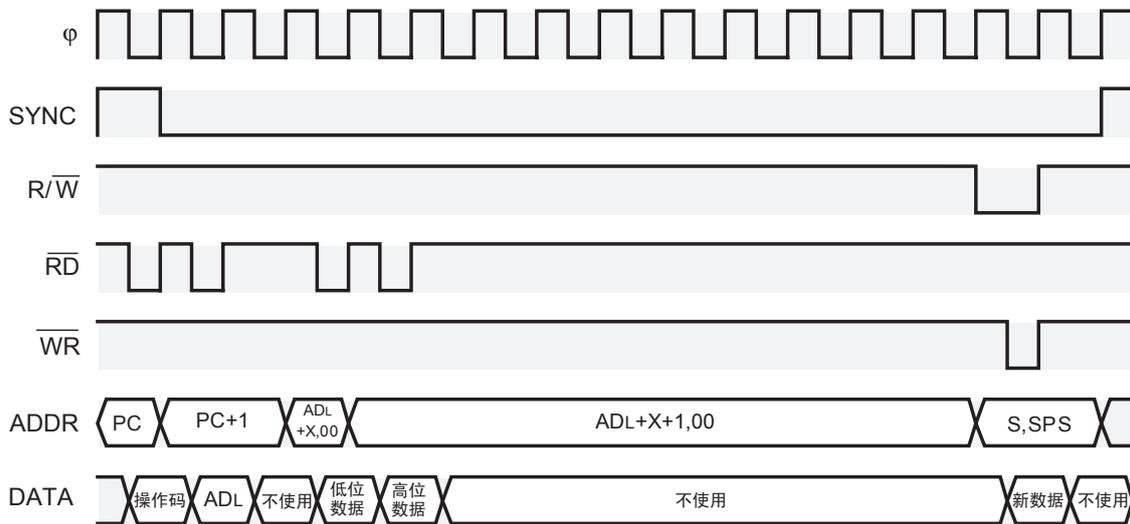
## 零页 X (Zero page X)

指令:  $\Delta\text{DIV}\Delta \$zz, X$ 

必要字节数: 2

必要周期数: 16

时序:



SPS: 通过 CPU 模式寄存器的堆栈页选择位选择的页

注. 部分产品不能使用此指令。

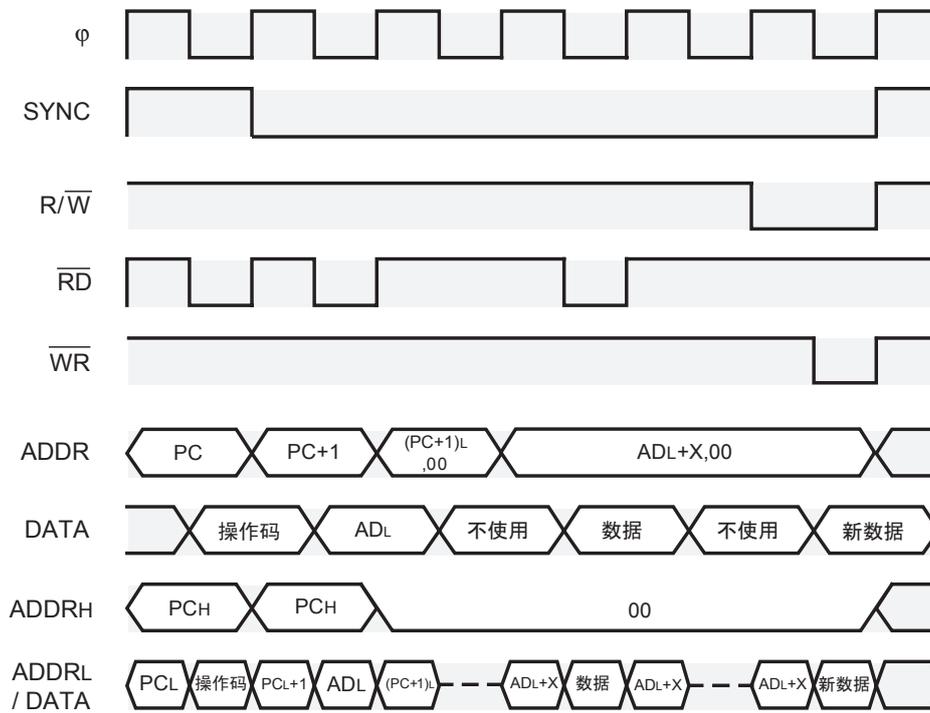
零页 X (Zero page X)

指令:  $\triangle ASL \triangle \$zz, X$   
 $\triangle DEC \triangle \$zz, X$   
 $\triangle INC \triangle \$zz, X$   
 $\triangle LSR \triangle \$zz, X$   
 $\triangle ROL \triangle \$zz, X$   
 $\triangle ROR \triangle \$zz, X$

必要字节数: 2

必要周期数: 6

时序:



[T=0]

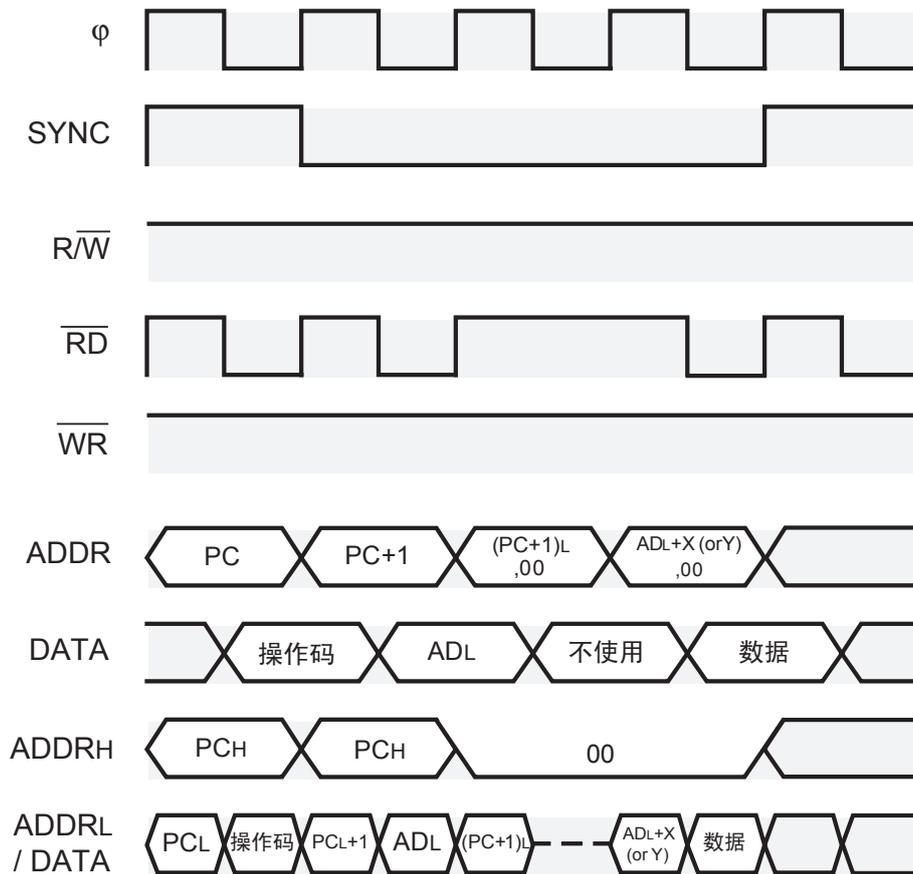
零页 X (Zero page X)  
零页 Y (Zero page Y)

指令:  $\Delta\text{ADC}\Delta \$zz, X$  (T=0)  
 $\Delta\text{AND}\Delta \$zz, X$  (T=0)  
 $\Delta\text{CMP}\Delta \$zz, X$  (T=0)  
 $\Delta\text{EOR}\Delta \$zz, X$  (T=0)  
 $\Delta\text{LDA}\Delta \$zz, X$  (T=0)  
 $\Delta\text{LDX}\Delta \$zz, Y$   
 $\Delta\text{LDY}\Delta \$zz, X$   
 $\Delta\text{ORA}\Delta \$zz, X$  (T=0)  
 $\Delta\text{SBC}\Delta \$zz, X$  (T=0)

必要字节数: 2

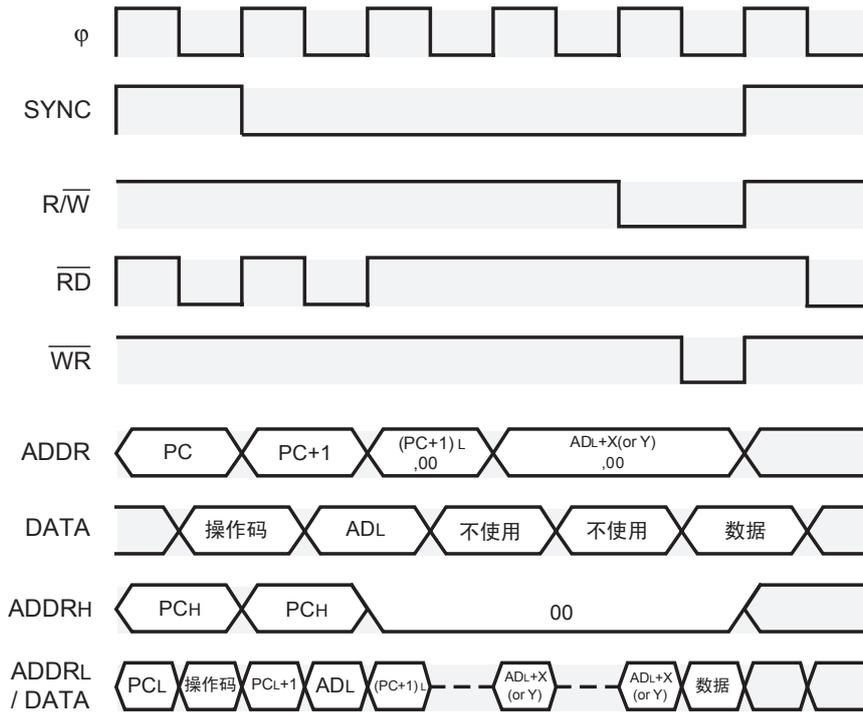
必要周期数: 4

时序:



零页 X (Zero page X)  
零页 Y (Zero page Y)

指令:  $\Delta STA \Delta \$zz, X$   
 $\Delta STX \Delta \$zz, Y$   
 $\Delta STY \Delta \$zz, X$   
 必要字节数: 2  
 必要周期数: 5  
 时序:



[T=0]

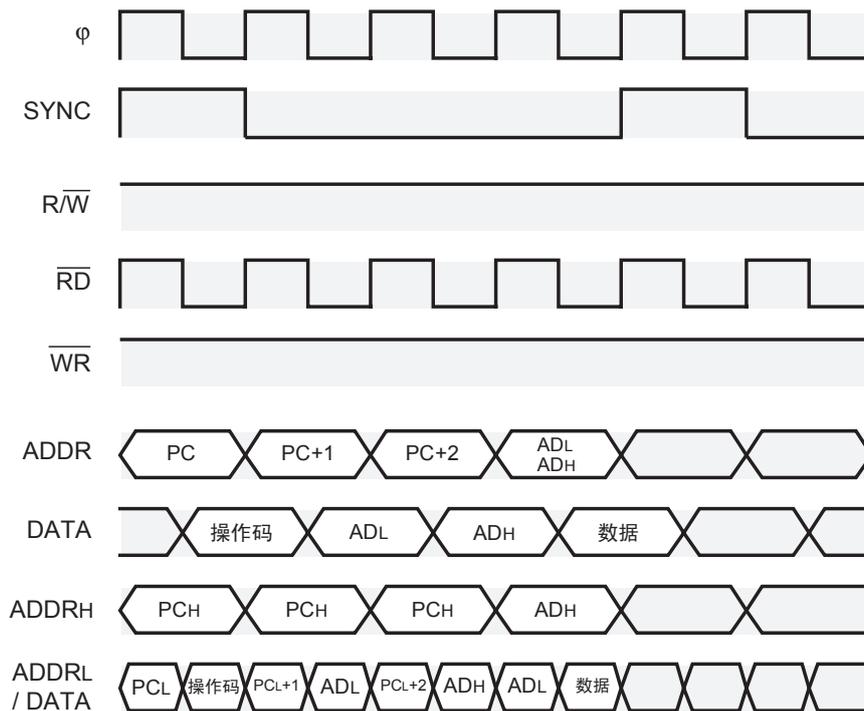
## 绝对 (Absolute)

指 令:  $\triangle\text{ADC}\triangle \$\text{hhl}$  (T=0)  
 $\triangle\text{AND}\triangle \$\text{hhl}$  (T=0)  
 $\triangle\text{BIT}\triangle \$\text{hhl}$   
 $\triangle\text{CMP}\triangle \$\text{hhl}$  (T=0)  
 $\triangle\text{CPX}\triangle \$\text{hhl}$   
 $\triangle\text{CPY}\triangle \$\text{hhl}$   
 $\triangle\text{EOR}\triangle \$\text{hhl}$  (T=0)  
 $\triangle\text{LDA}\triangle \$\text{hhl}$  (T=0)  
 $\triangle\text{LDX}\triangle \$\text{hhl}$   
 $\triangle\text{LDY}\triangle \$\text{hhl}$   
 $\triangle\text{ORA}\triangle \$\text{hhl}$  (T=0)  
 $\triangle\text{SBC}\triangle \$\text{hhl}$  (T=0)

必要字节数: 3

必要周期数: 4

时 序:



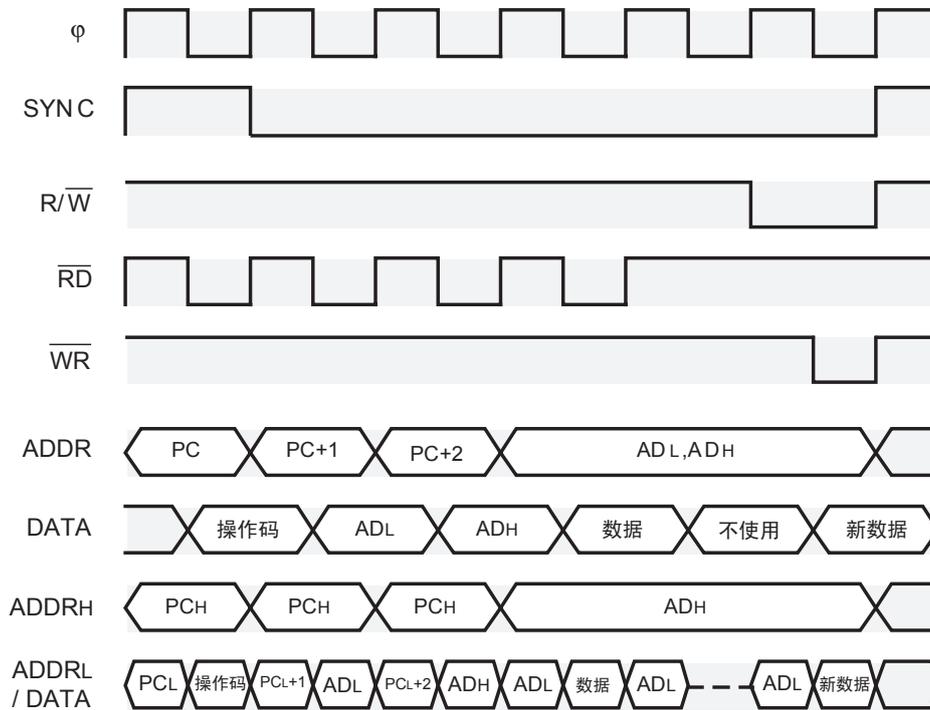
## 绝对 (Absolute)

指 令:  $\Delta$ ASL  $\Delta$  \$ hhl  
 $\Delta$ DEC  $\Delta$  \$ hhl  
 $\Delta$ INC  $\Delta$  \$ hhl  
 $\Delta$ LSR  $\Delta$  \$ hhl  
 $\Delta$ ROL  $\Delta$  \$ hhl  
 $\Delta$ ROR  $\Delta$  \$ hhl

必要字节数: 3

必要周期数: 6

时 序:



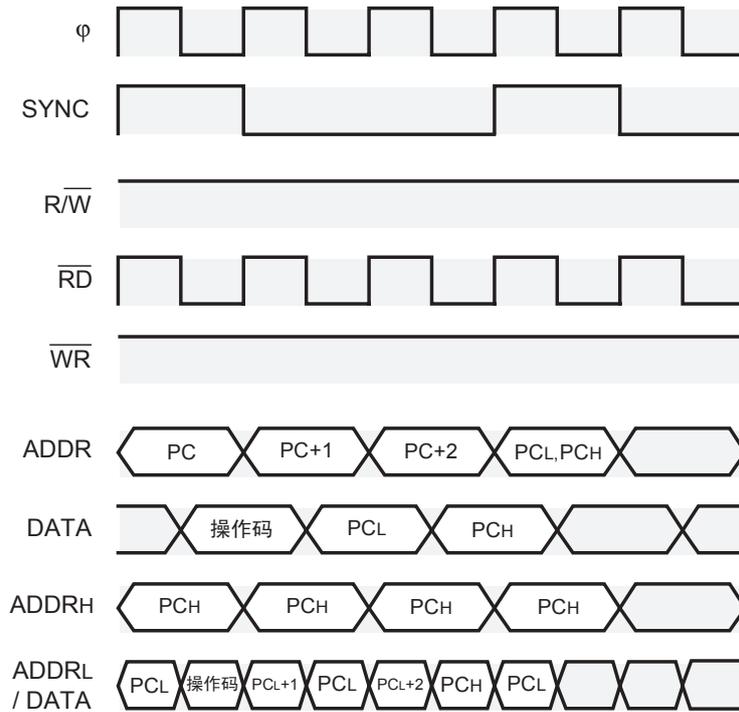
## 绝对 (Absolute)

指令:  $\Delta\text{JMP}\Delta \$\text{hhll}$ 

必要字节数: 3

必要周期数: 3

时序:



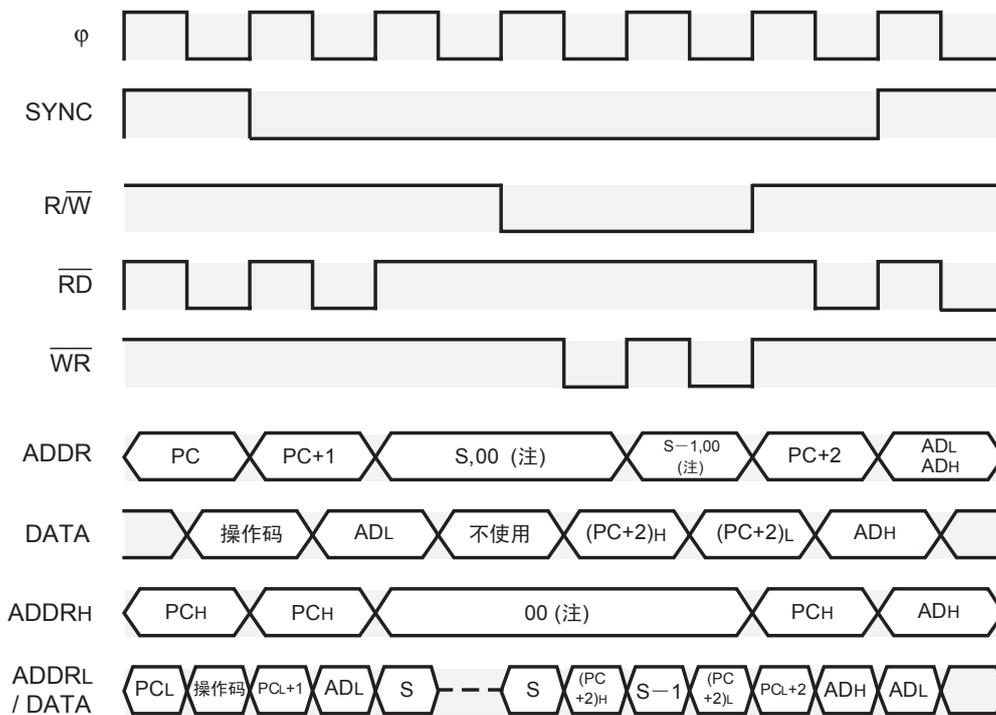
绝对 (Absolute)

指令:  $\Delta$ JSR $\Delta$ \$ hhl

必要字节数: 3

必要周期数: 6

时序:



注. 根据产品, 为“01”或者 SPS 标志的内容。

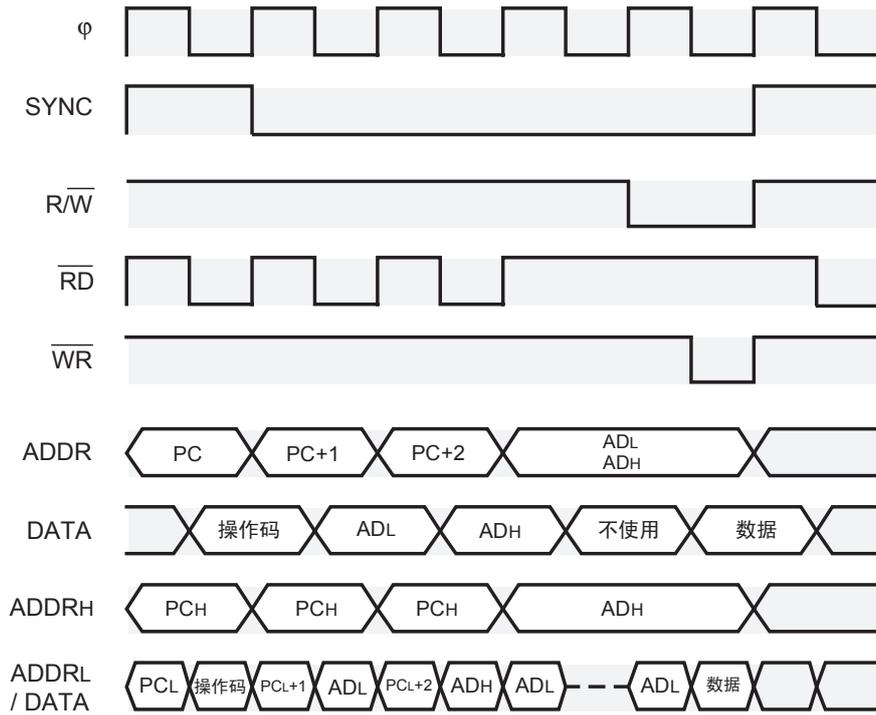
## 绝对 (Absolute)

指 令:  $\Delta$ STA $\Delta$  \$ hhl  
 $\Delta$ STX $\Delta$  \$ hhl  
 $\Delta$ STY $\Delta$  \$ hhl

必要字节数: 3

必要周期数: 5

时 序:



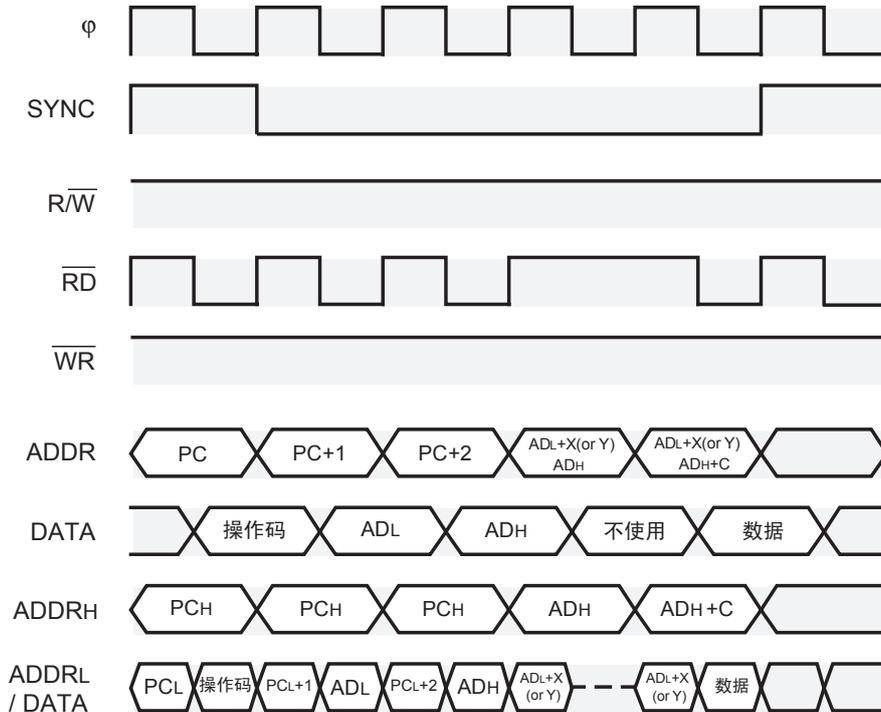
[T=0] 绝对 X (Absolute)  
绝对 Y (Absolute)

指 令:  $\Delta$ ADC  $\Delta$  \$ hhl, X 或者 Y (T=0)  
 $\Delta$ AND  $\Delta$  \$ hhl, X 或者 Y (T=0)  
 $\Delta$ CMP  $\Delta$  \$ hhl, X 或者 Y (T=0)  
 $\Delta$ EOR  $\Delta$  \$ hhl, X 或者 Y (T=0)  
 $\Delta$ LDA  $\Delta$  \$ hhl, X 或者 Y (T=0)  
 $\Delta$ LDX  $\Delta$  \$ hhl, Y  
 $\Delta$ LDY  $\Delta$  \$ hhl, X  
 $\Delta$ ORA  $\Delta$  \$ hhl, X 或者 Y (T=0)  
 $\Delta$ SBC  $\Delta$  \$ hhl, X 或者 Y (T=0)

必要字节数: 3

必要周期数: 5

时 序:



C: ADL+X (或者 Y) 的进位

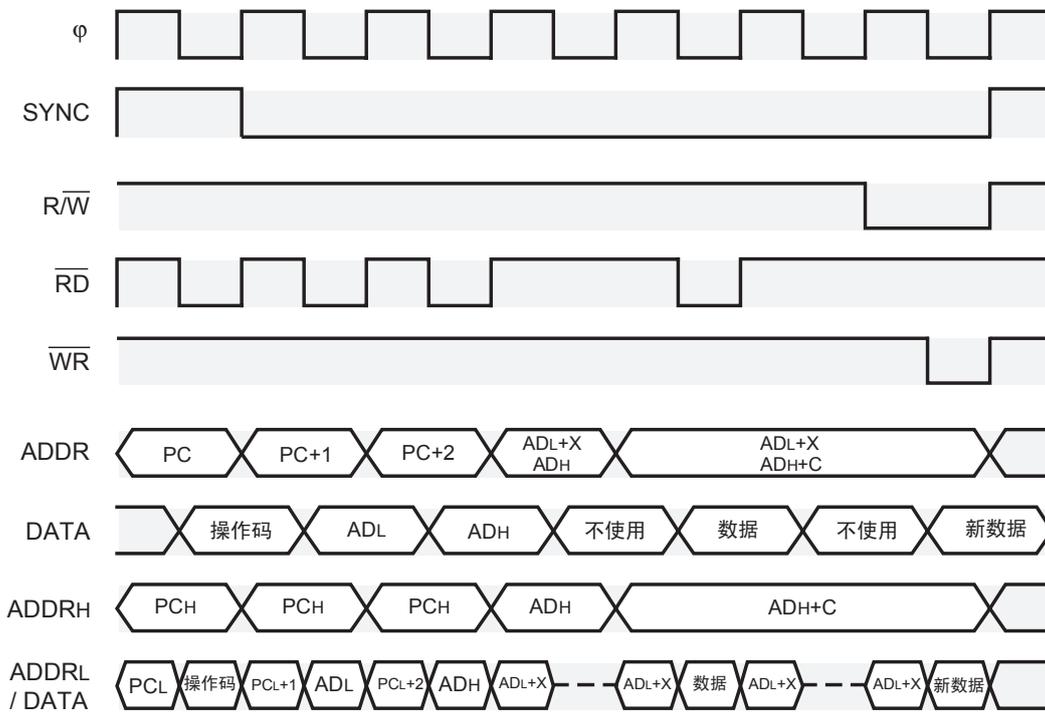
## 绝对 X (Absolute)

指令:  $\triangle ASL \triangle \$ hhl, X$   
 $\triangle DEC \triangle \$ hhl, X$   
 $\triangle INC \triangle \$ hhl, X$   
 $\triangle LSR \triangle \$ hhl, X$   
 $\triangle ROL \triangle \$ hhl, X$   
 $\triangle ROR \triangle \$ hhl, X$

必要字节数: 3

必要周期数: 7

时序:



C: ADL+X 的进位

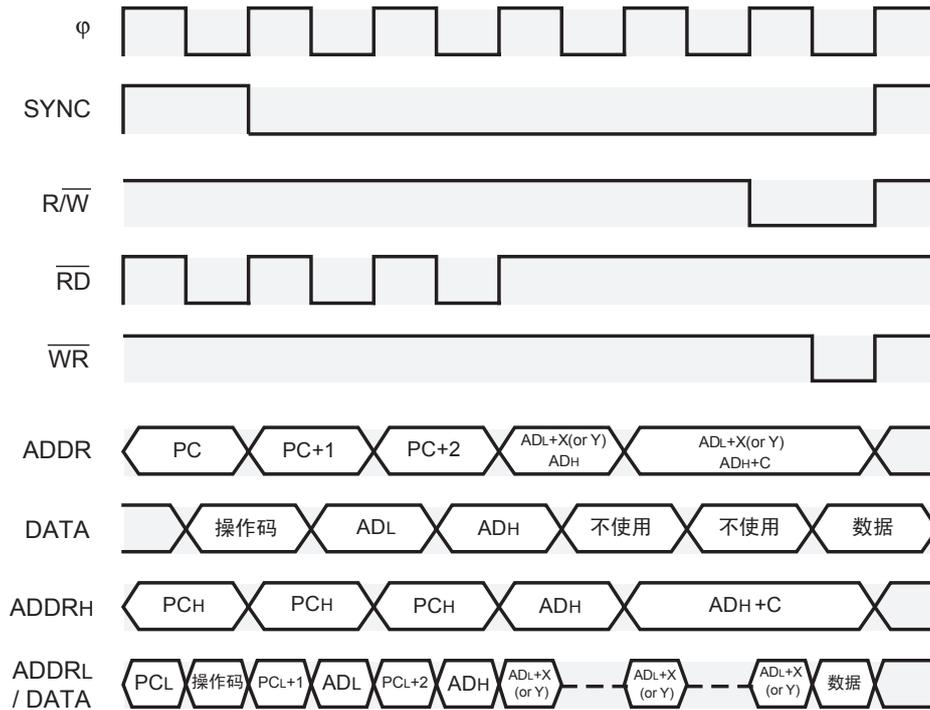
绝对 X (Absolute)  
绝对 Y (Absolute)

指令:  $\Delta STA \Delta \$ hhl, X$  或者  $Y$

必要字节数: 3

必要周期数: 6

时序:



C: ADL+X (或者 Y) 的进位

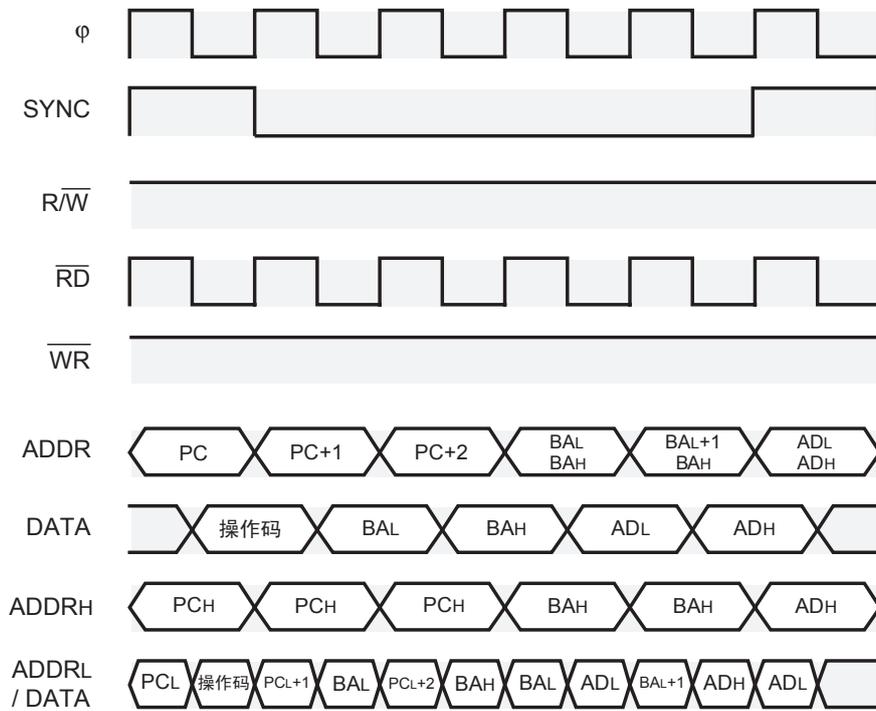
## 间接 (Indirect)

指令:  $\Delta\text{JMP}\Delta (\$hhll)$ 

必要字节数: 3

必要周期数: 5

时序:



BA: 基址

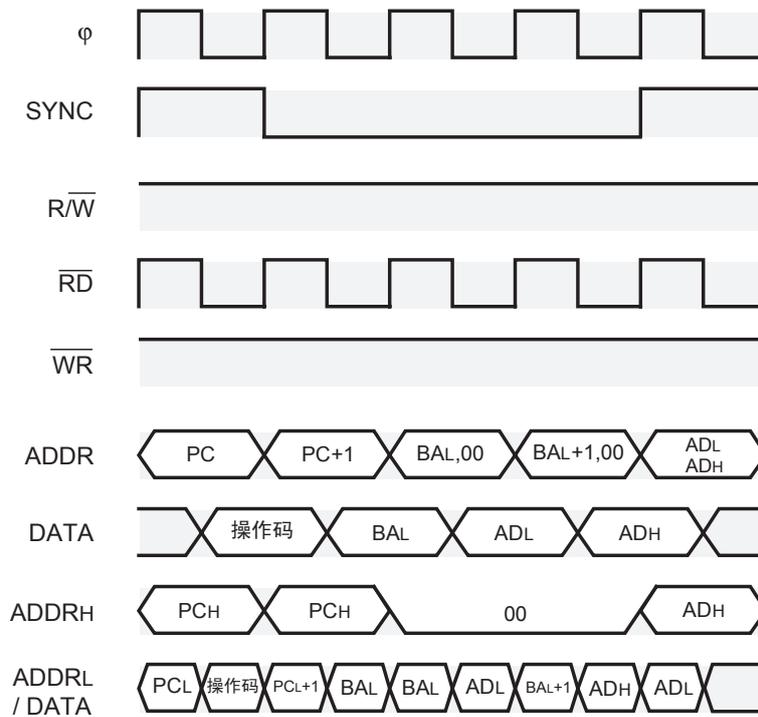
## 零页 间接 (Zero page Indirect)

指 令:  $\Delta\text{JMP}\Delta (\$zz)$

必要字节数: 2

必要周期数: 4

时 序:



BA: 基址

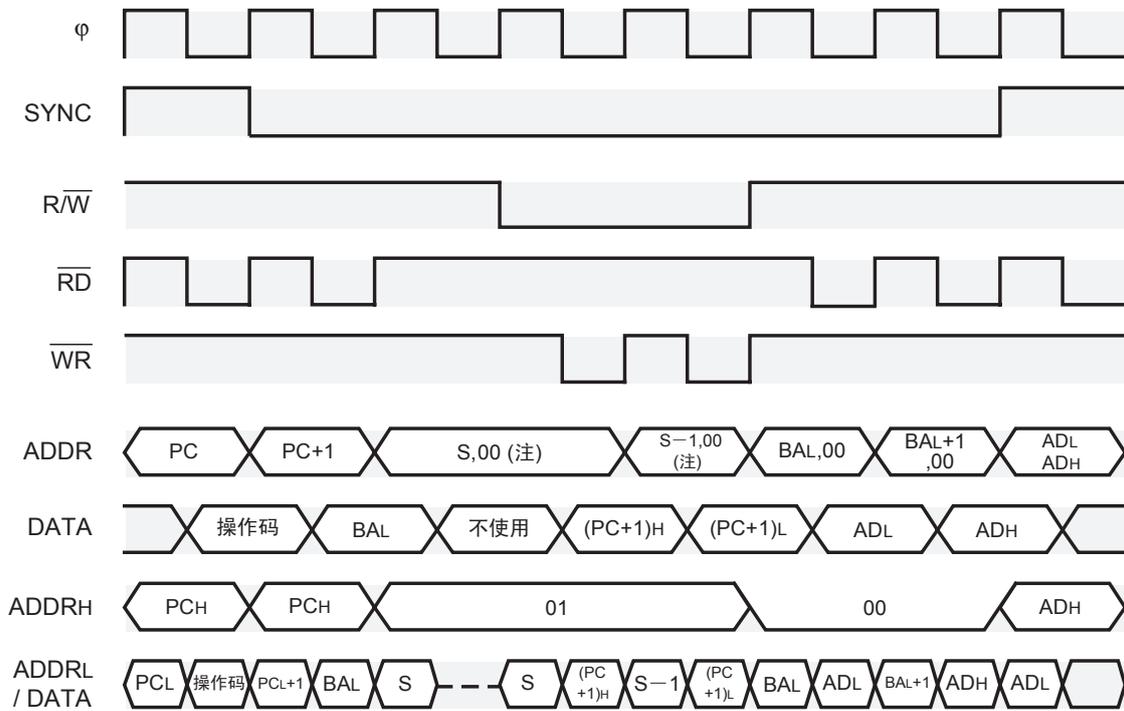
## 零页 间接 (Zero page Indirect)

指 令:  $\Delta$ JSR $\Delta$  (\$zz)

必要字节数: 2

必要周期数: 7

时 序:



BA: 基址

注. 根据产品, 为“01”或者 SPS 标志的内容。

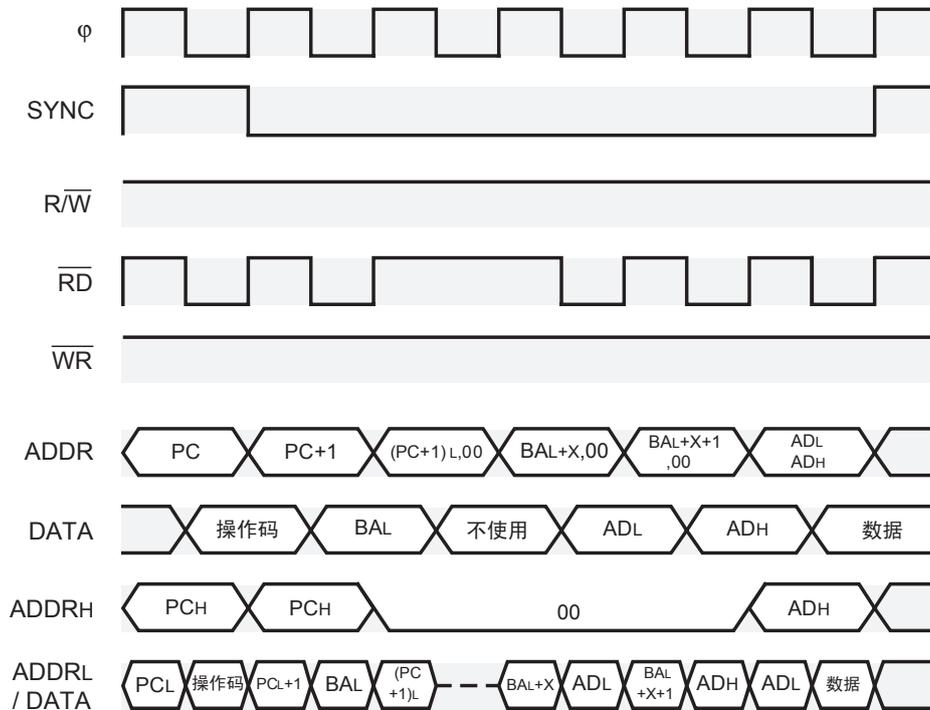
[T=0] 间接 X (Indirect X)

- 指 令:  $\Delta\text{ADC}\Delta(\$zz, X)$  (T=0)  
 $\Delta\text{AND}\Delta(\$zz, X)$  (T=0)  
 $\Delta\text{CMP}\Delta(\$zz, X)$  (T=0)  
 $\Delta\text{EOR}\Delta(\$zz, X)$  (T=0)  
 $\Delta\text{LDA}\Delta(\$zz, X)$  (T=0)  
 $\Delta\text{ORA}\Delta(\$zz, X)$  (T=0)  
 $\Delta\text{SBC}\Delta(\$zz, X)$  (T=0)

必要字节数: 2

必要周期数: 6

时 序:



BA: 基址

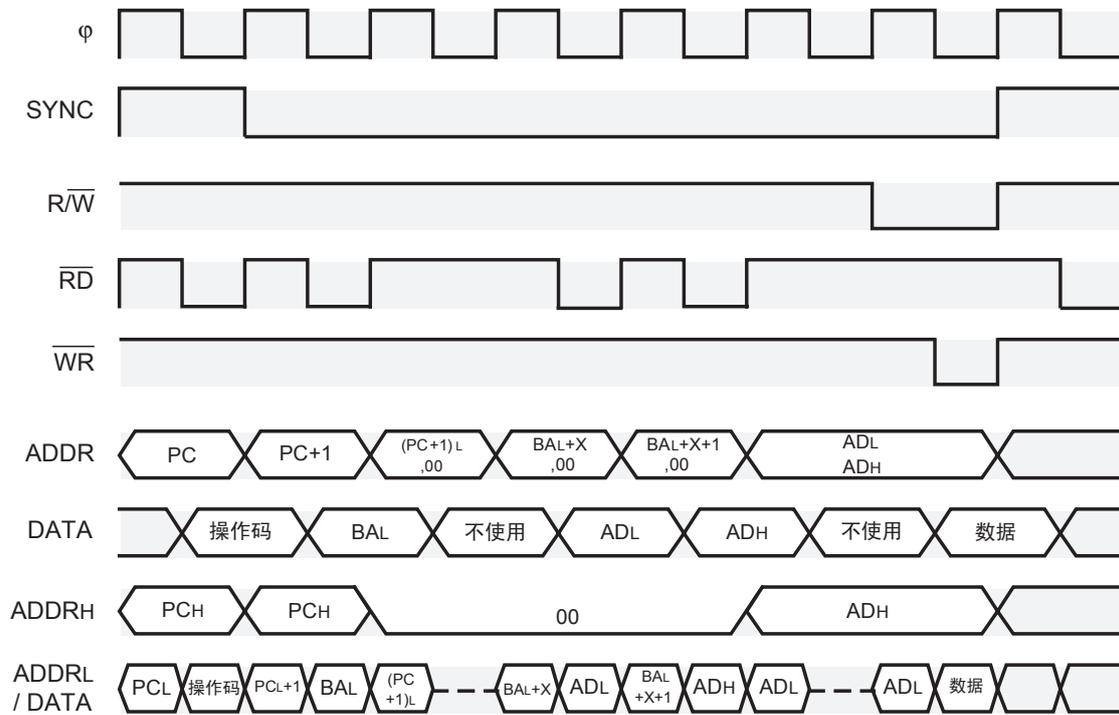
## 间接 X (Indirect X)

指令:  $\Delta STA\Delta(\$zz, X)$ 

必要字节数: 2

必要周期数: 7

时序:



BA: 基址

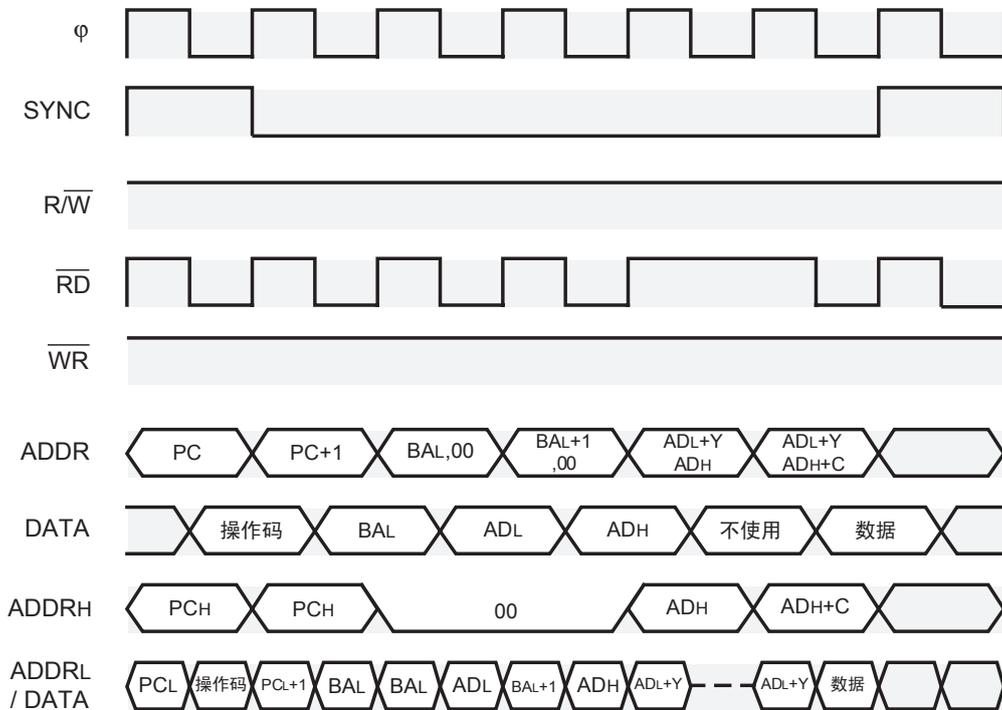
[T=0] 间接 Y (Indirect Y)

指令:  $\Delta ADC \Delta (\$zz), Y$  (T=0)  
 $\Delta AND \Delta (\$zz), Y$  (T=0)  
 $\Delta CMP \Delta (\$zz), Y$  (T=0)  
 $\Delta EOR \Delta (\$zz), Y$  (T=0)  
 $\Delta LDA \Delta (\$zz), Y$  (T=0)  
 $\Delta ORA \Delta (\$zz), Y$  (T=0)  
 $\Delta SBC \Delta (\$zz), Y$  (T=0)

必要字节数: 2

必要周期数: 6

时序:



BA: 基址

C: ADL+Y 的进位

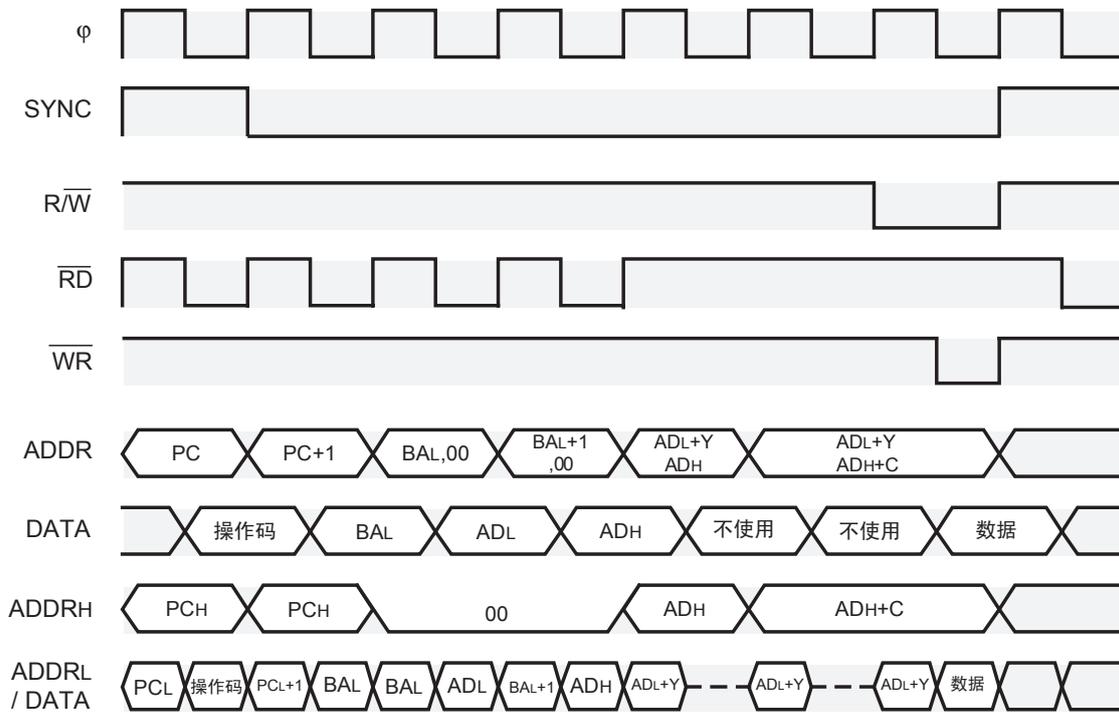
## 间接 Y (Indirect Y)

指令:  $\Delta STA(\$ zz), Y$ 

必要字节数: 2

必要周期数: 7

时序:



BA: 基址

C: ADL+Y 的进位

相对 (Relative)

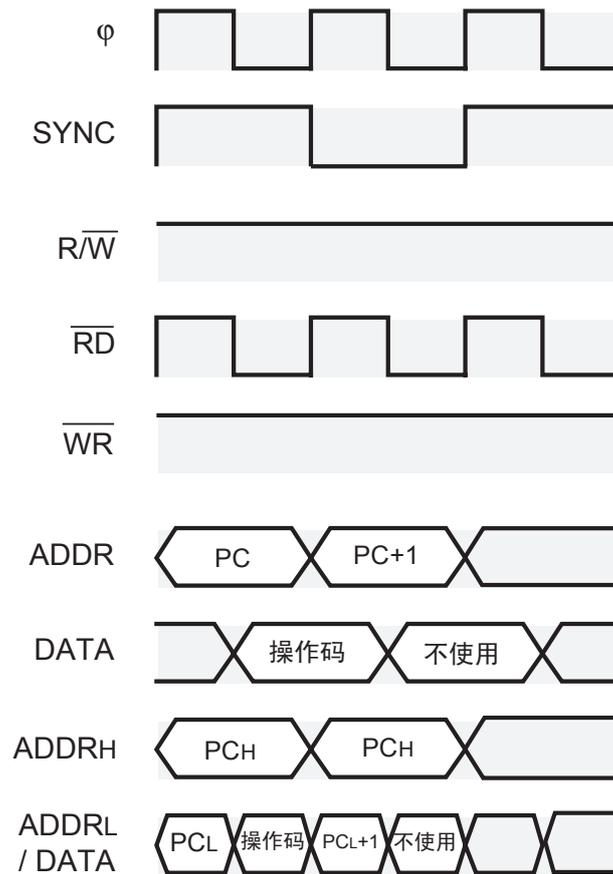
指令:  $\Delta BCC \Delta \$ hhl$   
 $\Delta BCS \Delta \$ hhl$   
 $\Delta BEQ \Delta \$ hhl$   
 $\Delta BMI \Delta \$ hhl$   
 $\Delta BNE \Delta \$ hhl$   
 $\Delta BPL \Delta \$ hhl$   
 $\Delta BVC \Delta \$ hhl$   
 $\Delta BVS \Delta \$ hhl$

必要字节数: 2

(1) 不转移时

必要周期数: 2

时序:



## 相对 (Relative)

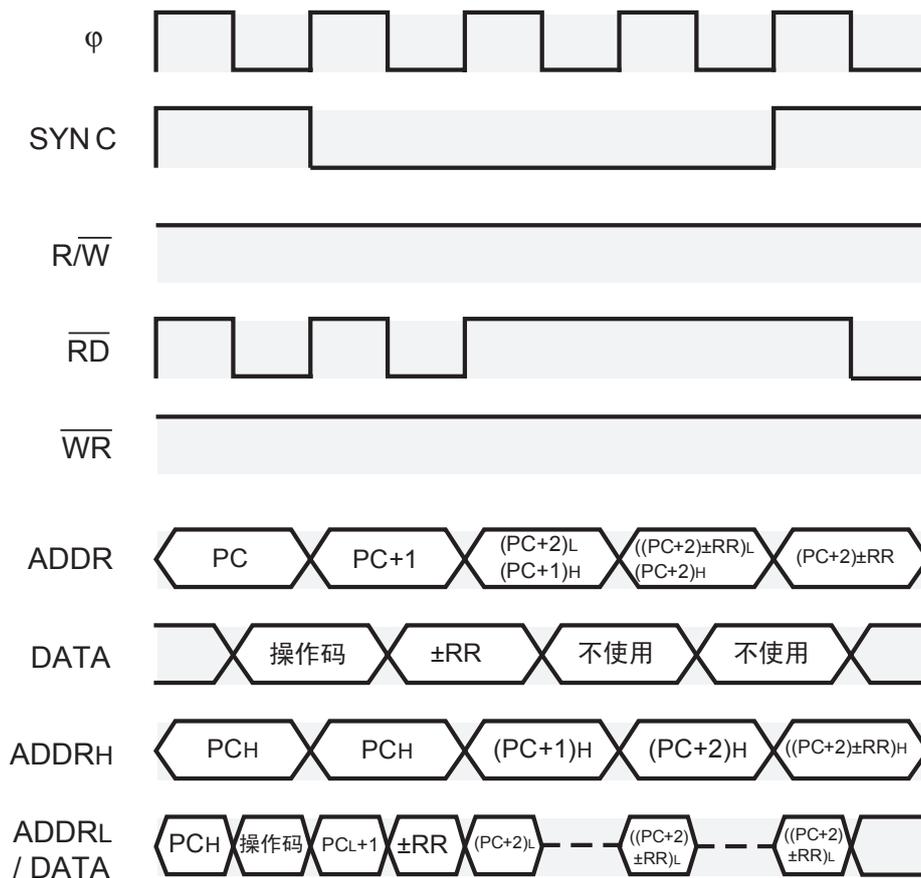
指令:  $\Delta BCC \Delta \$ hhl$   
 $\Delta BCS \Delta \$ hhl$   
 $\Delta BEQ \Delta \$ hhl$   
 $\Delta BMI \Delta \$ hhl$   
 $\Delta BNE \Delta \$ hhl$   
 $\Delta BPL \Delta \$ hhl$   
 $\Delta BVC \Delta \$ hhl$   
 $\Delta BVS \Delta \$ hhl$

必要字节数: 2

(2) 转移时

必要周期数: 4

时序:



RR: 偏移的值

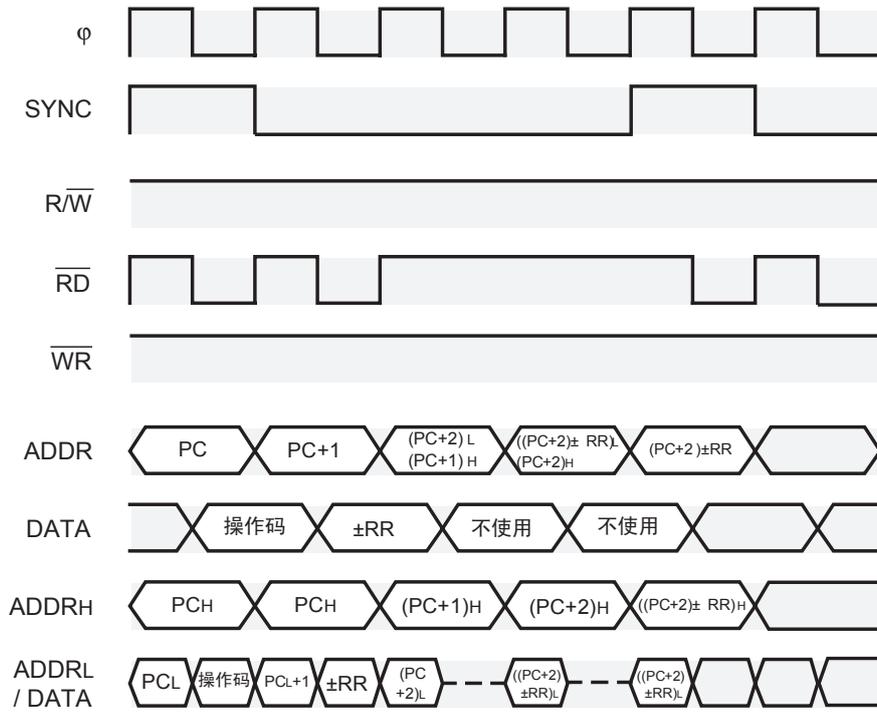
相对 (Relative)

指令:  $\Delta$ BRA $\Delta$  \$hhll

必要字节数: 2

必要周期数: 4

时序:



RR: 偏移的值

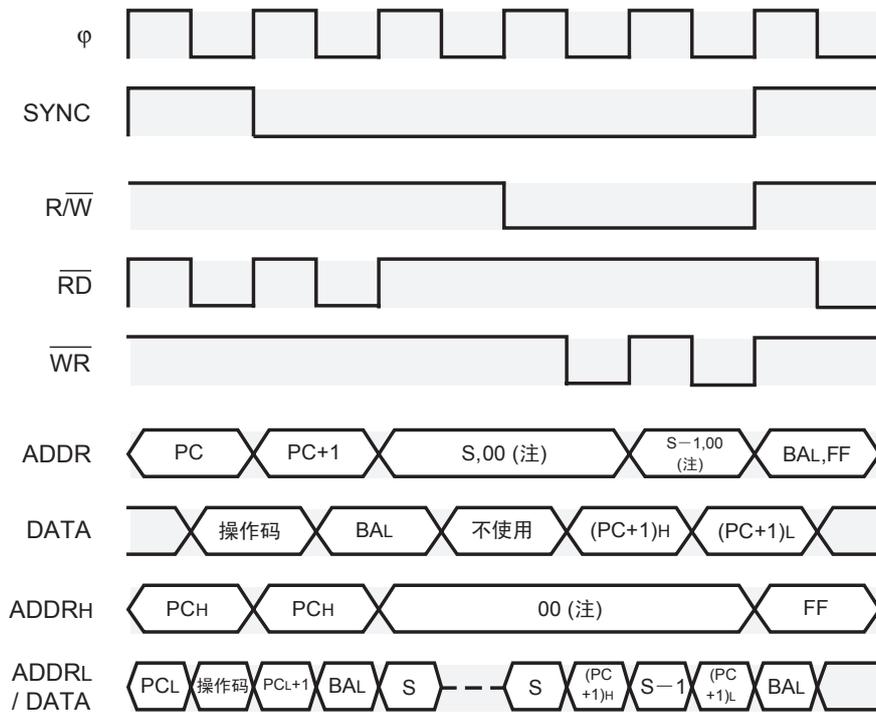
## 专用页 (Special page)

指 令:  $\Delta$ JSR $\Delta$ \\$hhll

必要字节数: 2

必要周期数: 5

时 序:



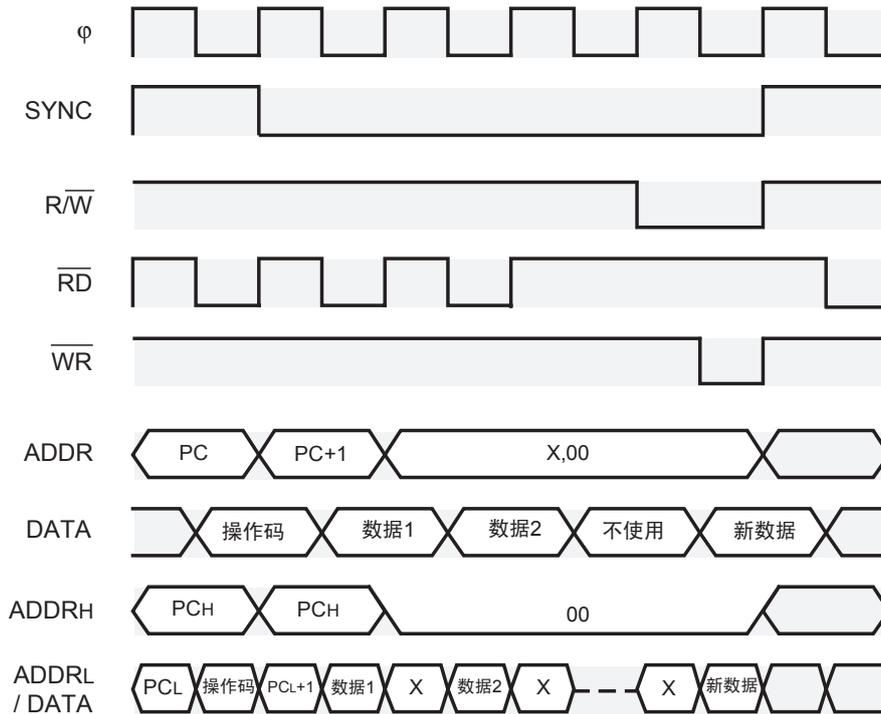
BA: 基址

注. 根据产品, 为“01”或者 SPS 标志的内容。

[T=1] 立即 (Immediate)

指令:  $\Delta\text{ADC}\Delta\# \$nn$  (T=1)  
 $\Delta\text{AND}\Delta\# \$nn$  (T=1)  
 $\Delta\text{EOR}\Delta\# \$nn$  (T=1)  
 $\Delta\text{ORA}\Delta\# \$nn$  (T=1)  
 $\Delta\text{SBC}\Delta\# \$nn$  (T=1)

必要字节数: 2  
 必要周期数: 5  
 时序:



[T=1]

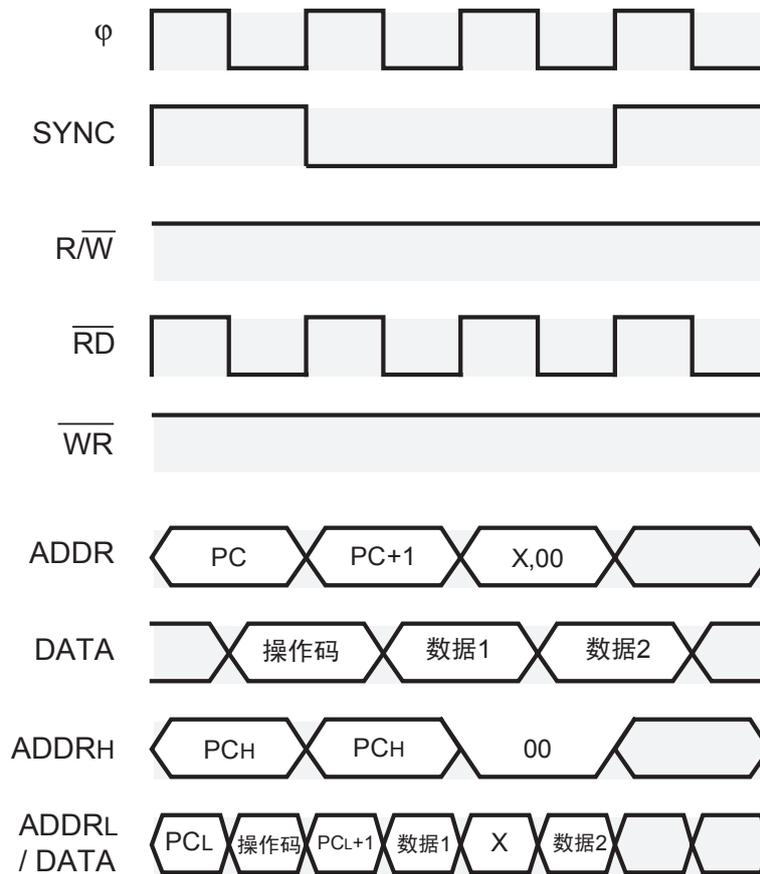
## 立即 (Immediate)

指令:  $\Delta\text{CMP}\Delta\#\ \$nn$  (T=1)

必要字节数: 2

必要周期数: 3

时序:



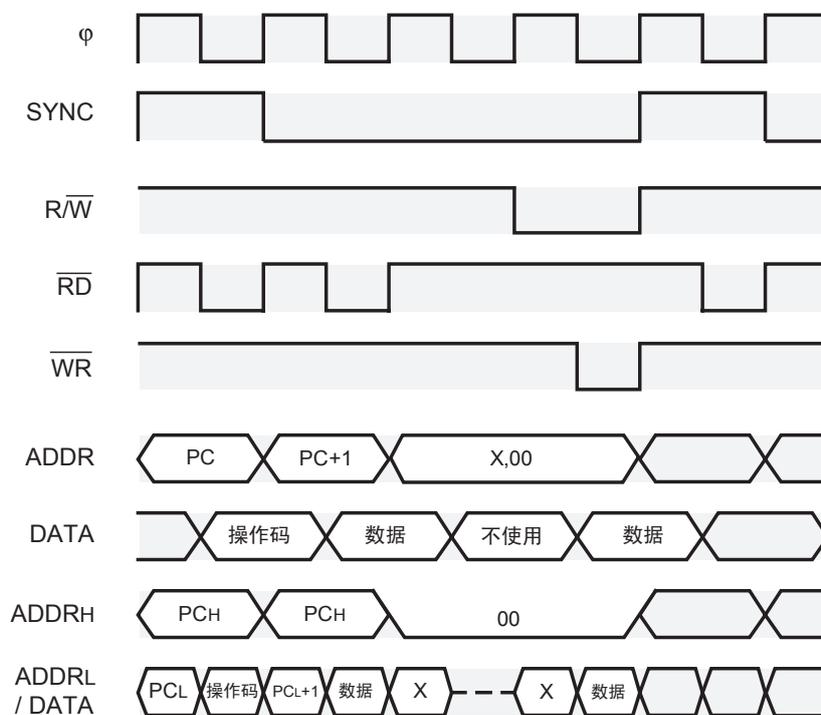
[T=1] 立即 (Immediate)

指令:  $\Delta LDA \Delta \# \$nn$  (T=1)

必要字节数: 2

必要周期数: 4

时序:



[T=1]

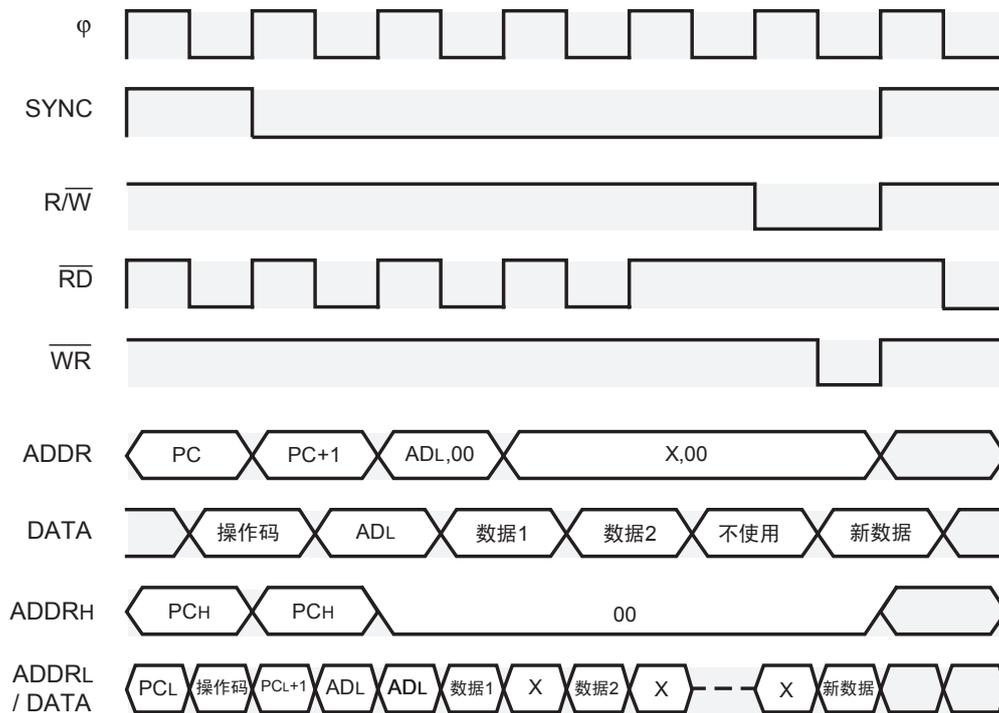
## 零页 (Zero page)

指 令:  $\triangle\text{ADC}\triangle \$zz$  (T=1)  
 $\triangle\text{AND}\triangle \$zz$  (T=1)  
 $\triangle\text{EOR}\triangle \$zz$  (T=1)  
 $\triangle\text{ORA}\triangle \$zz$  (T=1)  
 $\triangle\text{SBC}\triangle \$zz$  (T=1)

必要字节数: 2

必要周期数: 6

时 序:



[T=1]

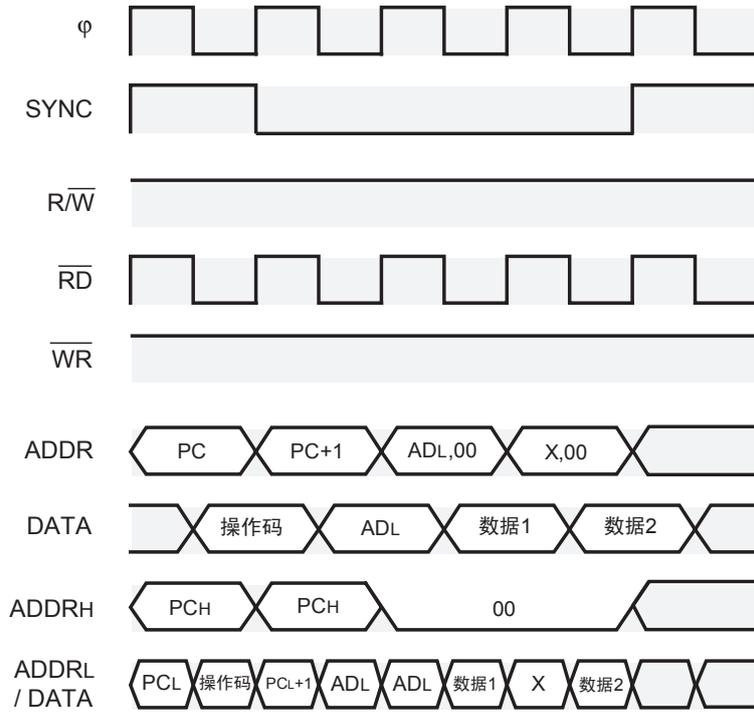
零页 (Zero page)

指令:  $\Delta\text{CMP}\Delta\$zz$  (T=1)

必要字节数: 2

必要周期数: 4

时序:



[T=1]

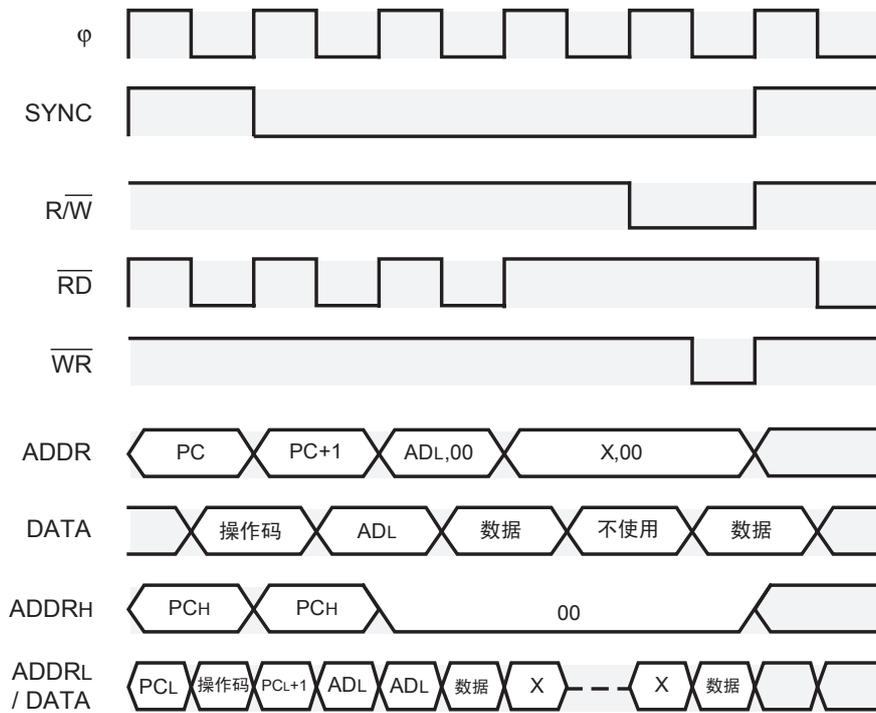
## 零页 (Zero page)

指令:  $\Delta\text{LDA}\Delta\text{\$zz}$  (T=1)

必要字节数: 2

必要周期数: 5

时序:



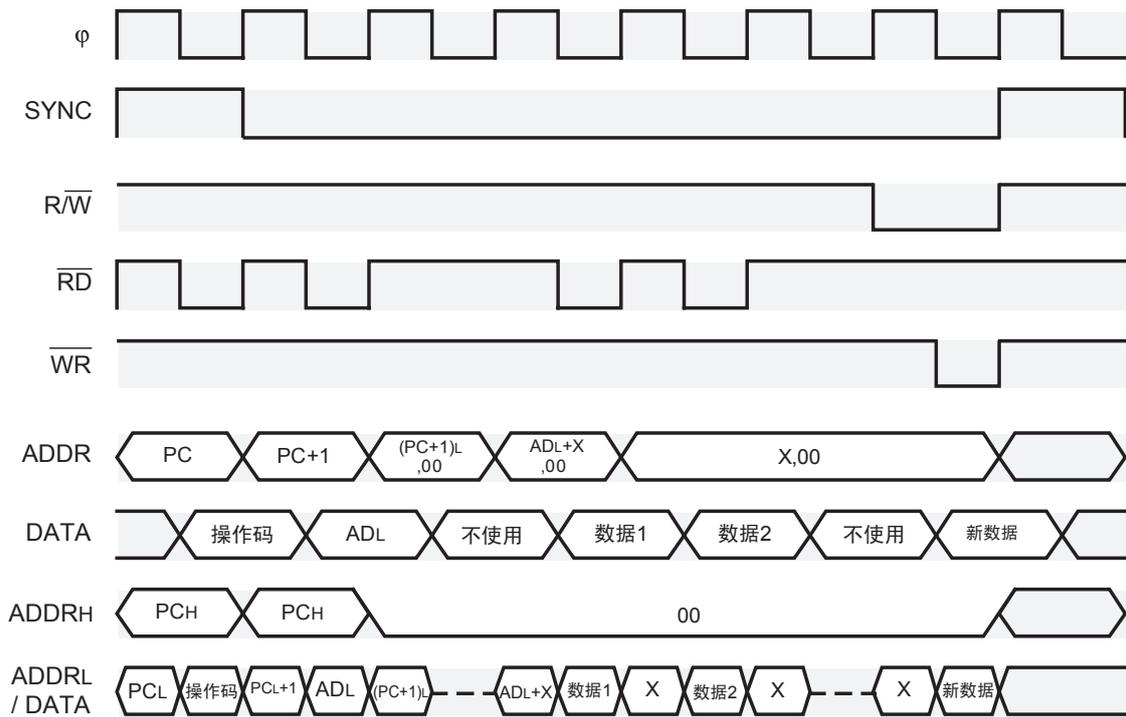
[T=1] 零页 X (Zero page X)

- 指令:  $\Delta\text{ADC}\Delta \$zz, X$  (T=1)  
 $\Delta\text{AND}\Delta \$zz, X$  (T=1)  
 $\Delta\text{EOR}\Delta \$zz, X$  (T=1)  
 $\Delta\text{ORA}\Delta \$zz, X$  (T=1)  
 $\Delta\text{SBC}\Delta \$zz, X$  (T=1)

必要字节数: 2

必要周期数: 7

时序:



[T=1]

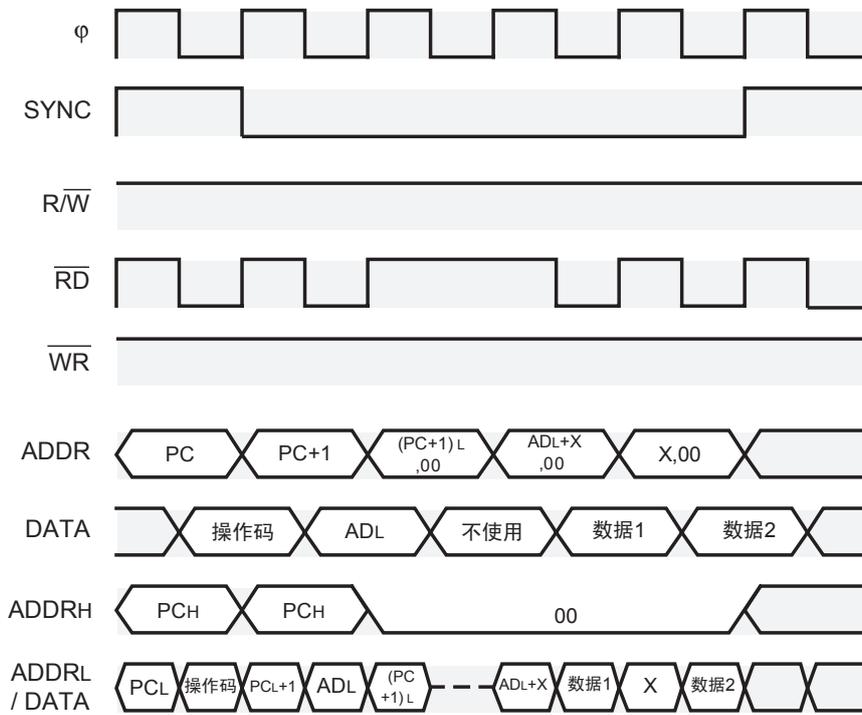
## 零页 X (Zero page X)

指令:  $\Delta\text{CMP}\Delta\$zz, X$  (T=1)

必要字节数: 2

必要周期数: 5

时序:



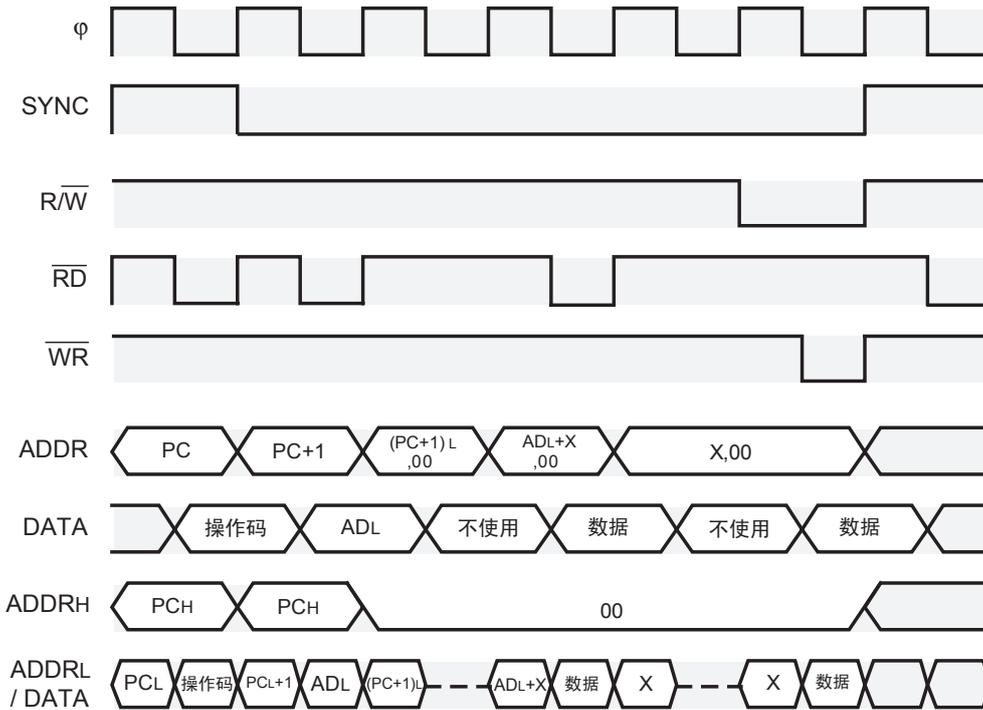
[T=1] 零页 X (Zero page X)

指令:  $\Delta LDA \Delta \$zz, X$  (T=1)

必要字节数: 2

必要周期数: 6

时序:



[T=1]

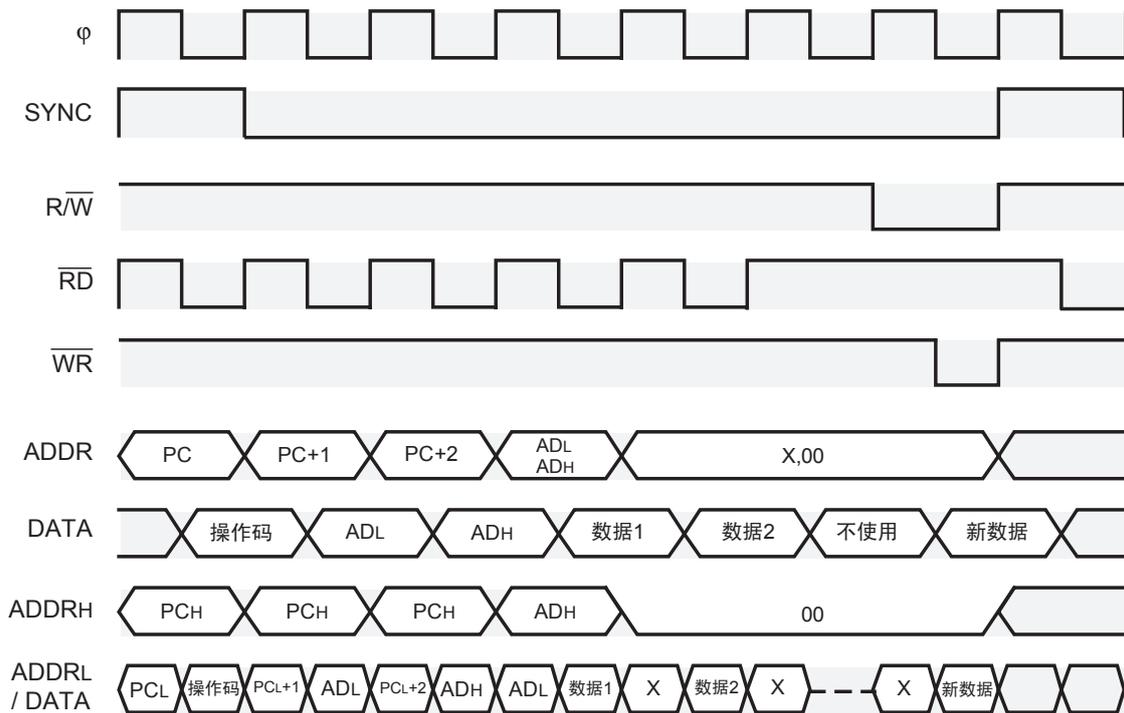
## 绝对 (Absolute)

指 令:  $\triangle\text{ADC}\triangle \$\text{hhll}$  (T=1)  
 $\triangle\text{AND}\triangle \$\text{hhll}$  (T=1)  
 $\triangle\text{EOR}\triangle \$\text{hhll}$  (T=1)  
 $\triangle\text{ORA}\triangle \$\text{hhll}$  (T=1)  
 $\triangle\text{SBC}\triangle \$\text{hhll}$  (T=1)

必要字节数: 3

必要周期数: 7

时 序:



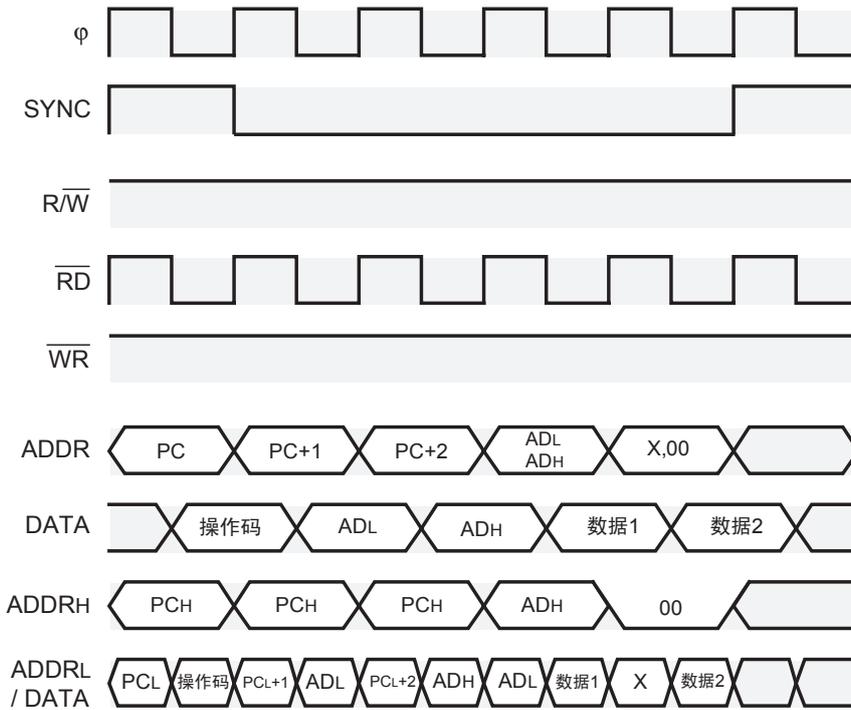
[T=1] 绝对 (Absolute)

指令:  $\Delta\text{CMP}\Delta \$\text{hhll}$  (T=1)

必要字节数: 3

必要周期数: 5

时序:



[T=1]

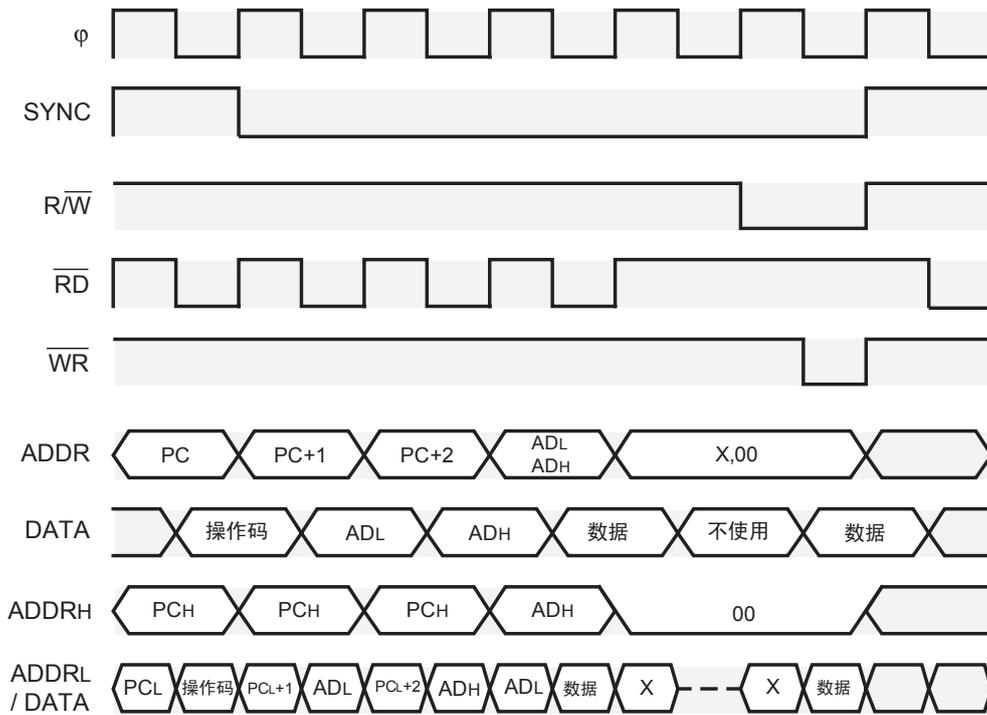
## 绝对 (Absolute)

指令:  $\Delta\text{LDA}\Delta \$\text{hhll}$  (T=1)

必要字节数: 3

必要周期数: 6

时序:



绝对 X (Absolute X)  
绝对 Y (Absolute Y)

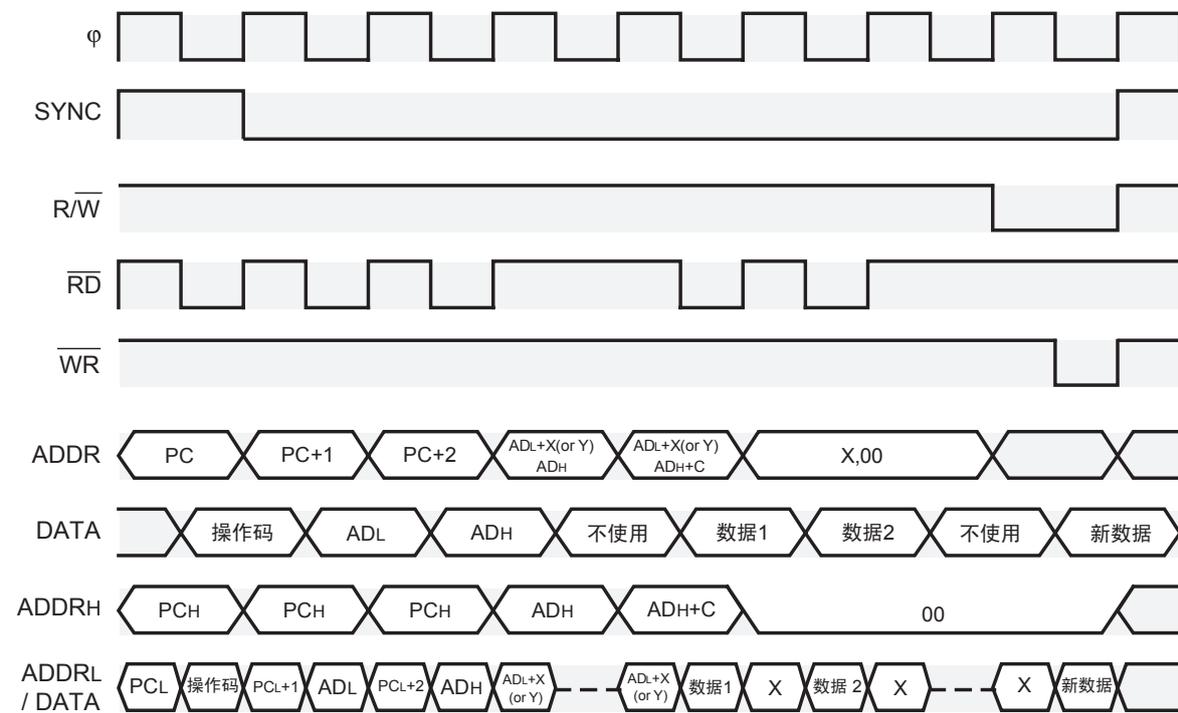
[T=1]

- 指令:  $\Delta\text{ADC}\Delta \$\text{hhl}, \text{X}$  或者  $\text{Y}$  (T=1)  
 $\Delta\text{AND}\Delta \$\text{hhl}, \text{X}$  或者  $\text{Y}$  (T=1)  
 $\Delta\text{EOR}\Delta \$\text{hhl}, \text{X}$  或者  $\text{Y}$  (T=1)  
 $\Delta\text{ORA}\Delta \$\text{hhl}, \text{X}$  或者  $\text{Y}$  (T=1)  
 $\Delta\text{SBC}\Delta \$\text{hhl}, \text{X}$  或者  $\text{Y}$  (T=1)

必要字节数: 3

必要周期数: 8

时序:



C: ADL+X (或者 Y) 的进位

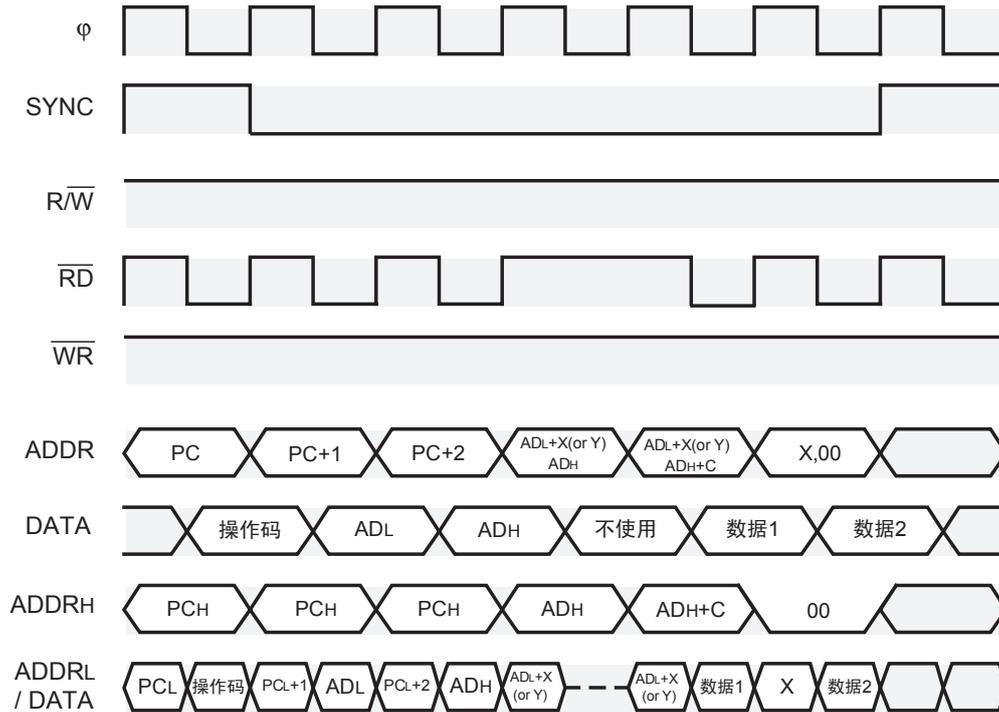
[T=1]

绝对 X (Absolute X)  
绝对 Y (Absolute Y)指令:  $\Delta\text{CMP}\Delta\$hhll, X$  或者  $Y$  (T=1)

必要字节数: 3

必要周期数: 6

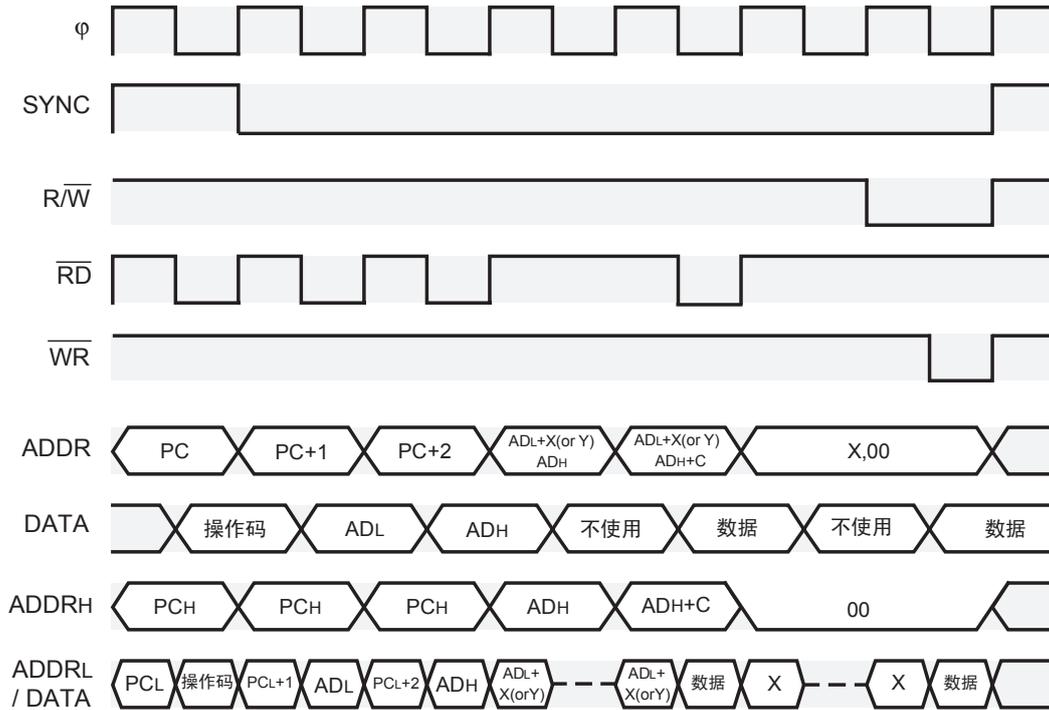
时序:



C: ADL+X (或者 Y) 的进位

[T=1] 绝对 X (Absolute X)  
绝对 Y (Absolute Y)

指令:  $\Delta LDA \Delta \$ hhl, X$  或者  $Y$  (T=1)  
 必要字节数: 3  
 必要周期数: 7  
 时序:



C: ADL+X (或者 Y) 的进位

[T=1]

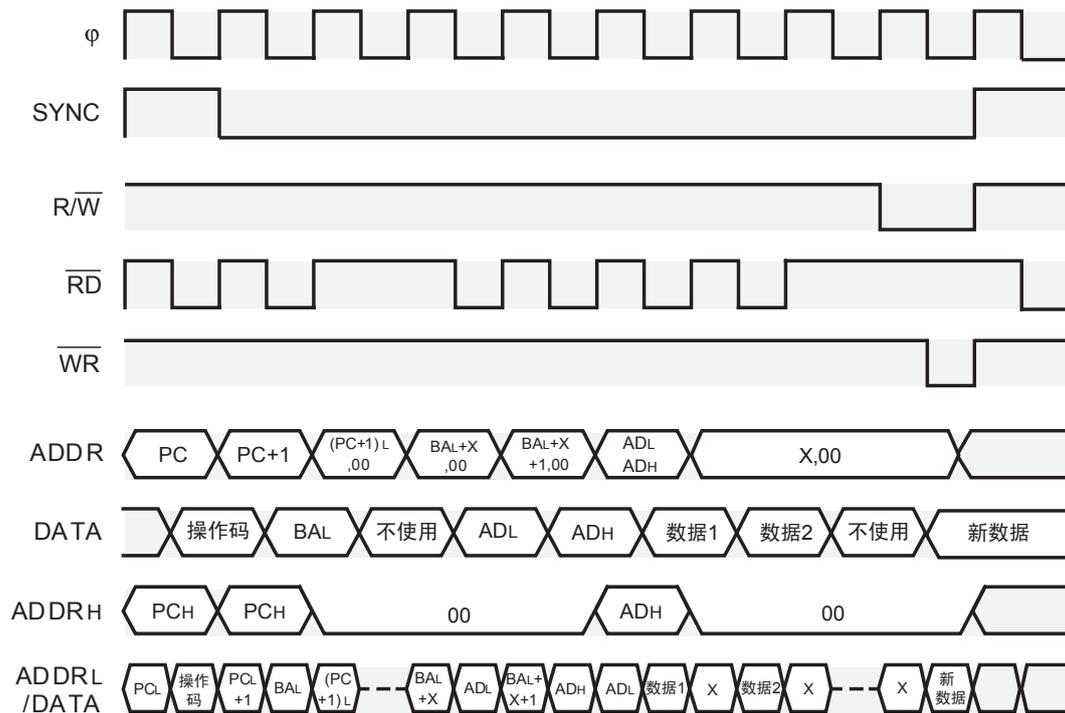
## 间接 X (Indirect X)

指令:  $\Delta\text{ADC}(\$zz, X)$  (T=1)  
 $\Delta\text{AND}(\$zz, X)$  (T=1)  
 $\Delta\text{EOR}(\$zz, X)$  (T=1)  
 $\Delta\text{ORA}(\$zz, X)$  (T=1)  
 $\Delta\text{SBC}(\$zz, X)$  (T=1)

必要字节数: 2

必要周期数: 9

时序:



BA: 基址

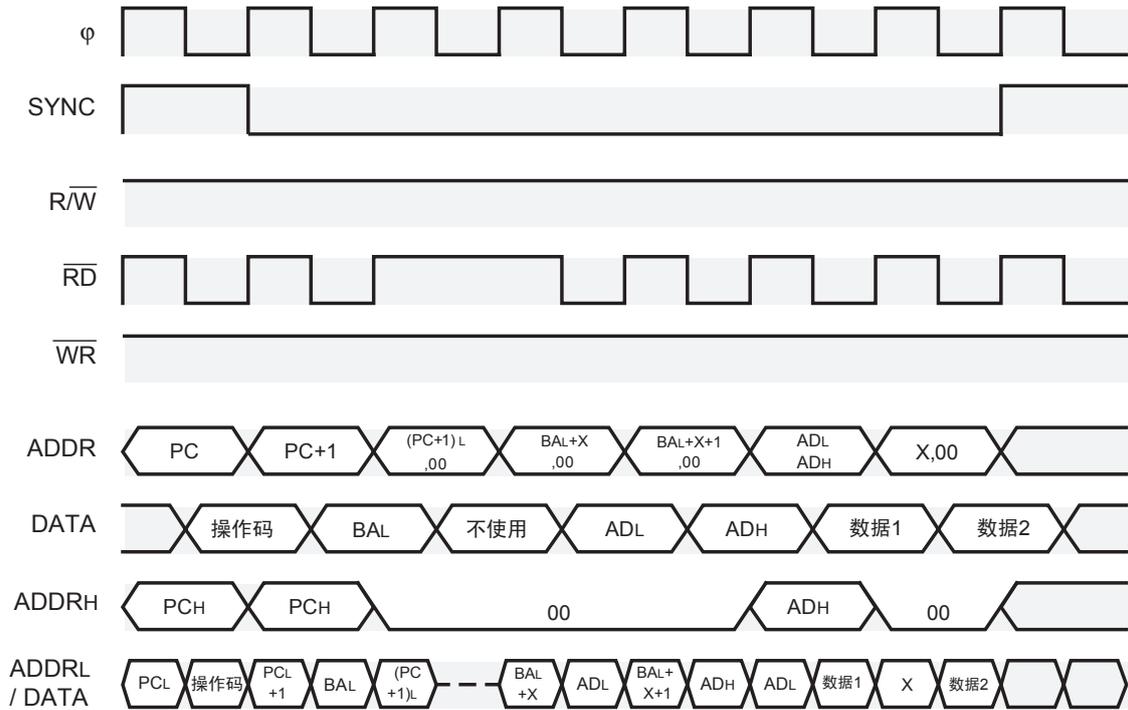
[T=1] 间接 X (Indirect X)

指令:  $\Delta\text{CMP}(\$zz, X)$  (T=1)

必要字节数: 2

必要周期数: 7

时序:



BA: 基址

[T=1]

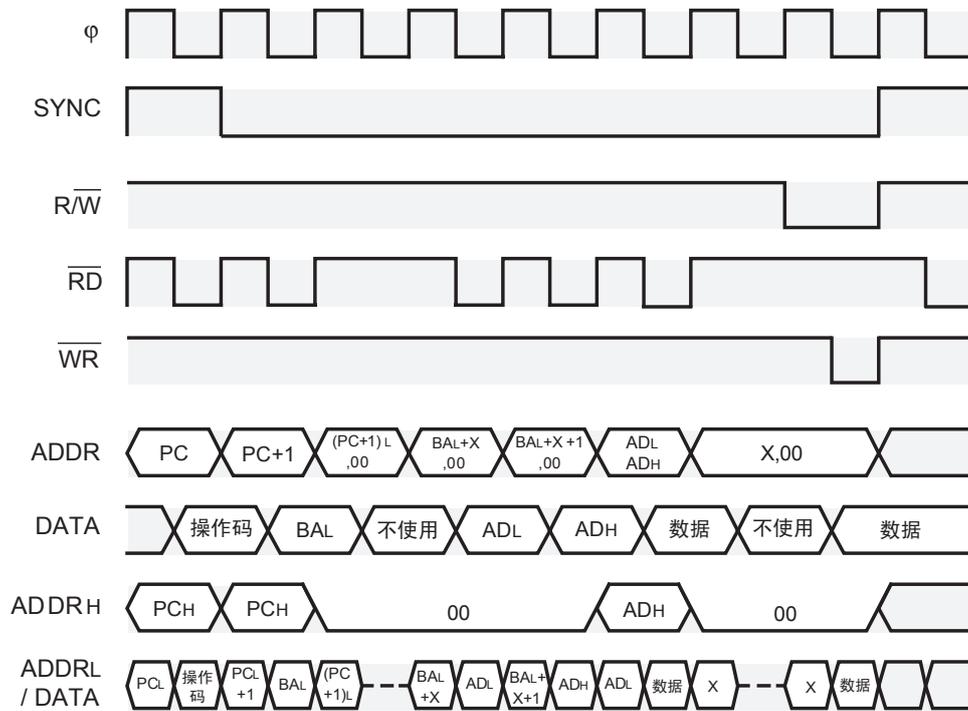
## 间接 X (Indirect X)

指令:  $\Delta\text{LDA}\Delta(\$zz, X)$  (T=1)

必要字节数: 2

必要周期数: 8

时序:



BA: 基址

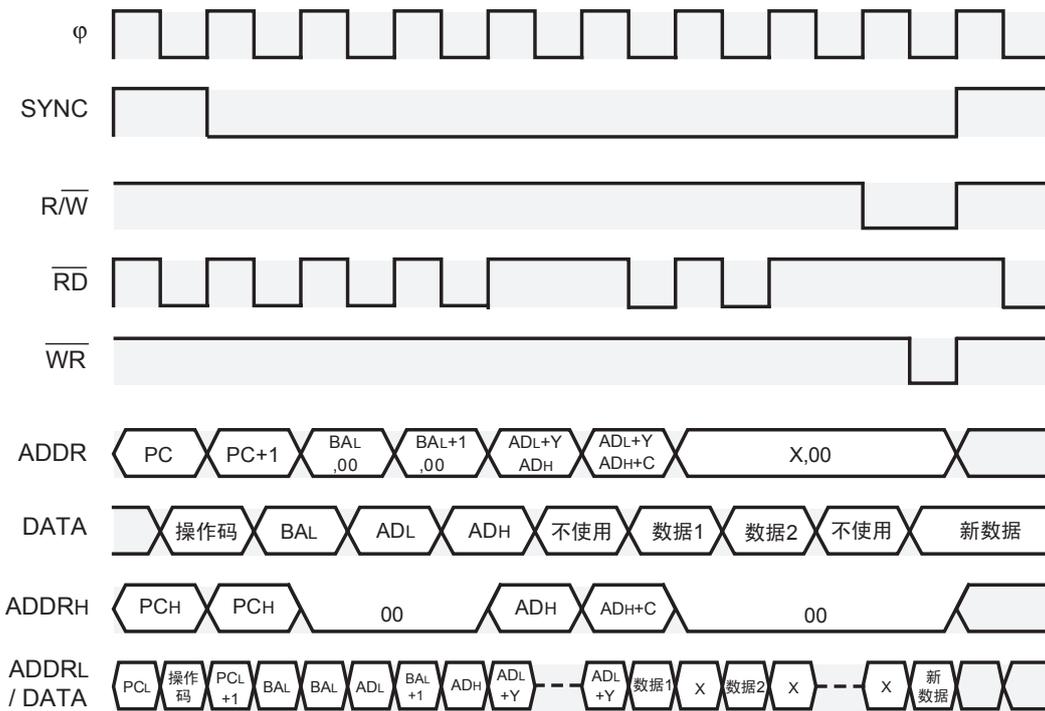
[T=1] 间接 Y (Indirect Y)

指令:  $\Delta\text{ADC}(\$zz), Y$  (T=1)  
 $\Delta\text{AND}(\$zz), Y$  (T=1)  
 $\Delta\text{EOR}(\$zz), Y$  (T=1)  
 $\Delta\text{ORA}(\$zz), Y$  (T=1)  
 $\Delta\text{SBC}(\$zz), Y$  (T=1)

必要字节数: 2

必要周期数: 9

时序:



BA: 基址

C: ADL+Y 的进位

[T=1]

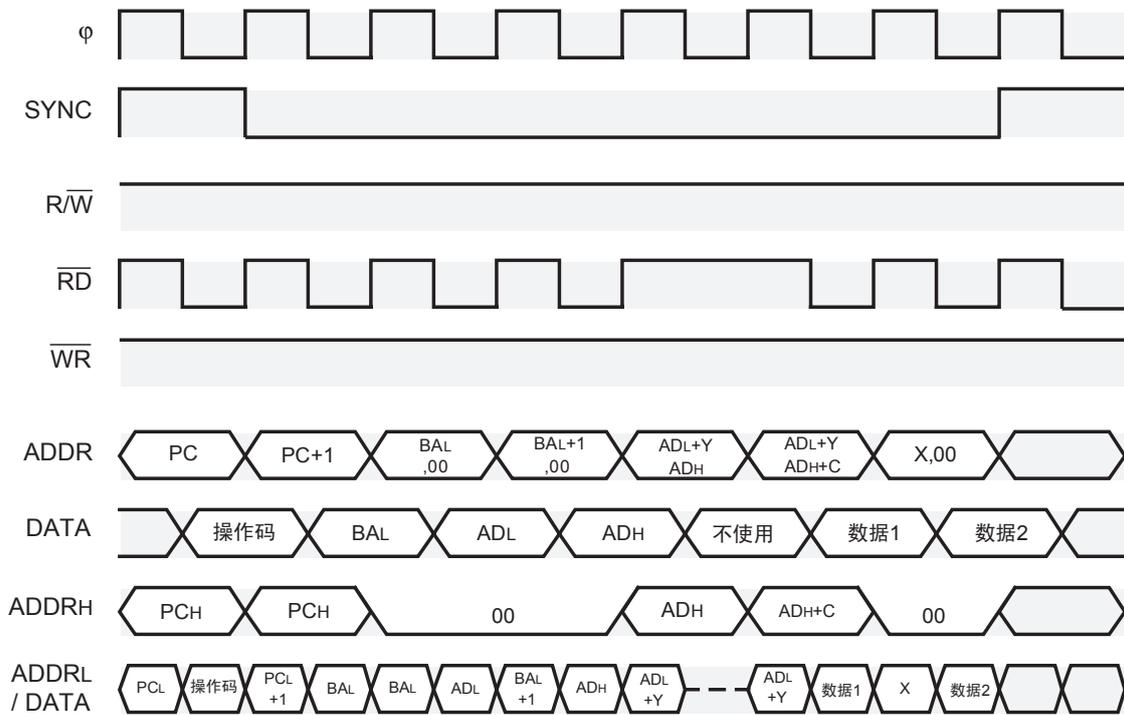
## 间接 Y (Indirect Y)

指令:  $\Delta\text{CMP}\Delta(\$zz), Y$  (T=1)

必要字节数: 2

必要周期数: 7

时序:



BA: 基址

C: ADL+Y 的进位

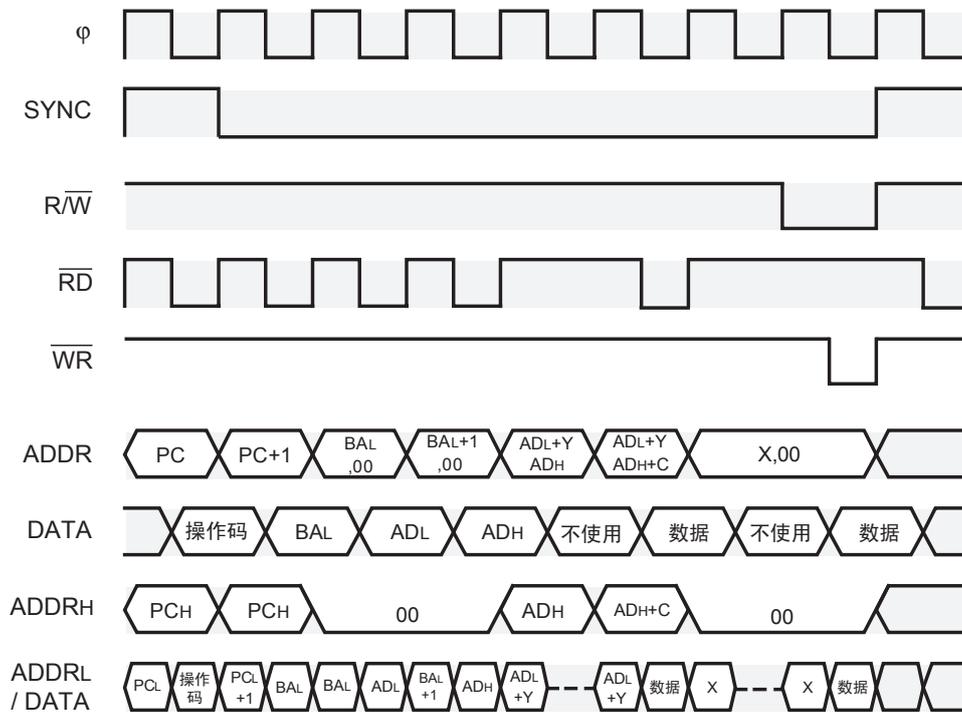
[T=1] 间接 Y (Indirect Y)

指令:  $\Delta LDA(\Delta(\$zz), Y) (T=1)$

必要字节数: 2

必要周期数: 8

时序:



BA: 基址

C: ADL+Y 的进位

# 附录 2 740 族机器指令一览表

项目 分类	指令符号	功 能	标 志	指 令 码			字节数	周期数	注	
			N V T B D I Z C	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	16进制码				
数 据 传 送 指 令	LDA #Snn	(A)← nn	○×××××○×	1 0 1 0	<B <sub>2</sub> >	1 0 0 1	A9	2	2	2
	LDA \$zz	(A)←(M) 其中 M=(zz)	○×××××○×	1 0 1 0	<B <sub>2</sub> >	0 1 0 1	A5	2	3	2
	LDA \$zz,X	(A)←(M) 其中 M=(zz+(X))	○×××××○×	1 0 1 1	<B <sub>2</sub> >	0 1 0 1	B5	2	4	2
	LDA \$hhll	(A)←(M) 其中 M=(hhll)	○×××××○×	1 0 1 0	<B <sub>2</sub> >	1 1 0 1	AD	3	4	2
	LDA \$hhll,X	(A)←(M) 其中 M=(hhll+(X))	○×××××○×	1 0 1 1	<B <sub>2</sub> >	1 1 0 1	BD	3	5	2
	LDA \$hhll,Y	(A)←(M) 其中 M=(hhll+(Y))	○×××××○×	1 0 1 1	<B <sub>2</sub> >	1 0 0 1	B9	3	5	2
	LDA (\$zz,X)	(A)←(M) 其中 M=((zz+(X)+1)(zz+X))	○×××××○×	1 0 1 0	<B <sub>2</sub> >	0 0 0 1	A1	2	6	2
	LDA (\$zz,Y)	(A)←(M) 其中 M=((zz+1)(zz)+(Y))	○×××××○×	1 0 1 1	<B <sub>2</sub> >	0 0 0 1	B1	2	6	2
	LDX #Snn	(X)← nn	○×××××○×	1 0 1 0	<B <sub>2</sub> >	0 0 1 0	A2	2	2	
	LDX \$zz	(X)←(M) 其中 M=(zz)	○×××××○×	1 0 1 0	<B <sub>2</sub> >	0 1 1 0	A6	2	3	
	LDX \$zz,Y	(X)←(M) 其中 M=(zz+(Y))	○×××××○×	1 0 1 1	<B <sub>2</sub> >	0 1 1 0	B6	2	4	
	LDX \$hhll	(X)←(M) 其中 M=(hhll)	○×××××○×	1 0 1 0	<B <sub>2</sub> >	1 1 1 0	AE	3	4	
	LDX \$hhll,Y	(X)←(M) 其中 M=(hhll+(Y))	○×××××○×	1 0 1 1	<B <sub>2</sub> >	1 1 1 0	BE	3	5	
	LDY #Snn	(Y)← nn	○×××××○×	1 0 1 0	<B <sub>2</sub> >	0 0 0 0	A0	2	2	
	LDY \$zz	(Y)←(M) 其中 M=(zz)	○×××××○×	1 0 1 0	<B <sub>2</sub> >	0 1 0 0	A4	2	3	
	LDY \$zz,X	(Y)←(M) 其中 M=(zz+(Y))	○×××××○×	1 0 1 1	<B <sub>2</sub> >	0 1 0 0	B4	2	4	
	LDY \$hhll	(Y)←(M) 其中 M=(hhll)	○×××××○×	1 0 1 0	<B <sub>2</sub> >	1 1 0 0	AC	3	4	
	LDY \$hhll,X	(Y)←(M) 其中 M=(hhll+(X))	○×××××○×	1 0 1 1	<B <sub>2</sub> >	1 1 0 0	BC	3	5	
	LDM #Snn,\$zz	(M)← nn 其中 M=(zz)	××××××××	0 0 1 1	<B <sub>2</sub> >	1 1 0 0	3C	3	4	
	STA \$zz	(M)←(A) 其中 M=(zz)	××××××××	1 0 0 0	<B <sub>2</sub> >	0 1 0 1	85	2	4	
	STA \$zz,X	(M)←(A) 其中 M=(zz+(X))	××××××××	1 0 0 1	<B <sub>2</sub> >	0 1 0 1	95	2	5	
	STA \$hhll	(M)←(A) 其中 M=(hhll)	××××××××	1 0 0 0	<B <sub>2</sub> >	1 1 0 1	8D	3	5	
	STA \$hhll,X	(M)←(A) 其中 M=(hhll+(X))	××××××××	1 0 0 1	<B <sub>2</sub> >	1 1 0 1	9D	3	6	
	STA \$hhll,Y	(M)←(A) 其中 M=(hhll+(Y))	××××××××	1 0 0 1	<B <sub>2</sub> >	1 0 0 1	99	3	6	
	STA (\$zz,X)	(M)←(A) 其中 M=((zz+(X)+1)(zz+X))	××××××××	1 0 0 0	<B <sub>2</sub> >	0 0 0 1	81	2	7	
	STA (\$zz,Y)	(M)←(A) 其中 M=((zz+1)(zz)+(Y))	××××××××	1 0 0 1	<B <sub>2</sub> >	0 0 0 1	91	2	7	
	STX \$zz	(M)←(X) 其中 M=(zz)	××××××××	1 0 0 0	<B <sub>2</sub> >	0 1 1 0	86	2	4	
	STX \$zz,Y	(M)←(X) 其中 M=(zz+(Y))	××××××××	1 0 0 1	<B <sub>2</sub> >	0 1 1 0	96	2	5	
STX \$hhll	(M)←(X) 其中 M=(hhll)	××××××××	1 0 0 0	<B <sub>2</sub> >	1 1 1 0	8E	3	5		
STY \$zz	(M)←(Y) 其中 M=(zz)	××××××××	1 0 0 0	<B <sub>2</sub> >	0 1 0 0	84	2	4		
STY \$zz,X	(M)←(Y) 其中 M=(zz+(X))	××××××××	1 0 0 1	<B <sub>2</sub> >	0 1 0 0	94	2	5		
STY \$hhll	(M)←(Y) 其中 M=(hhll)	××××××××	1 0 0 0	<B <sub>2</sub> >	1 1 0 0	8C	3	5		
传送指令	TAX	(X)←(A)	○×××××○×	1 0 1 0	1 0 1 0	AA	1	2		
	TXA	(A)←(X)	○×××××○×	1 0 0 0	1 0 1 0	8A	1	2		
	TAY	(Y)←(A)	○×××××○×	1 0 1 0	1 0 0 0	A8	1	2		
	TYA	(A)←(Y)	○×××××○×	1 0 0 1	1 0 0 0	98	1	2		
	TSX	(X)←(S)	○×××××○×	1 0 1 1	1 0 1 0	BA	1	2		
	TXS	(S)←(X)	××××××××	1 0 0 1	1 0 1 0	9A	1	2		
栈操作指令	PHA	(M(S))←(A),(S)←(S)-1	××××××××	0 1 0 0	1 0 0 0	48	1	3		
	PHP	(M(S))←(PS),(S)←(S)-1	××××××××	0 0 0 0	1 0 0 0	08	1	3		
	PLA	(S)←(S)+1,(A)←(M(S))	○×××××○×	0 1 1 0	1 0 0 0	68	1	4		
	PLP	(S)←(S)+1,(PS)←(M(S))	(保存到堆栈的值)	0 0 1 0	1 0 0 0	28	1	4		

附录 2 740 族机器指令一览表

项目 分类	指令符号	功 能	标 志	指 令 码			字节数	周期数	注		
			N V T B D I Z C	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	16进制码					
运 算 指 令	ADC # $\$nn$	(A) ← (A) + nn + (C)	00XXXXX00	0 1 1 0	<B <sub>2</sub> >	1 0 0 1	69	2	2	1	
	ADC \$zz	(A) ← (A) + (M) + (C) 其中 M=(zz)	00XXXXX00	0 1 1 0	<B <sub>2</sub> >	0 1 0 1	65	2	3	1	
	ADC \$zz,X	(A) ← (A) + (M) + (C) 其中 M=(zz+(X))	00XXXXX00	0 1 1 1	<B <sub>2</sub> >	0 1 0 1	75	2	4	1	
	ADC \$hhll	(A) ← (A) + (M) + (C) 其中 M=(hhll)	00XXXXX00	0 1 1 0	<B <sub>2</sub> >	1 1 0 1	6D	3	4	1	
	ADC \$hhll,X	(A) ← (A) + (M) + (C) 其中 M=(hhll+(X))	00XXXXX00	0 1 1 1	<B <sub>2</sub> >	1 1 0 1	7D	3	5	1	
	ADC \$hhll,Y	(A) ← (A) + (M) + (C) 其中 M=(hhll+(Y))	00XXXXX00	0 1 1 1	<B <sub>2</sub> >	1 0 0 1	79	3	5	1	
	ADC (\$zz,X)	(A) ← (A) + (M) + (C) 其中 M=((zz+(X)+1)(zz+X))	00XXXXX00	0 1 1 0	<B <sub>2</sub> >	0 0 0 1	61	2	6	1	
	ADC (\$zz,Y)	(A) ← (A) + (M) + (C) 其中 M=((zz+1)(zz)+(Y))	00XXXXX00	0 1 1 1	<B <sub>2</sub> >	0 0 0 1	71	2	6	1	
	SBC # $\$nn$	(A) ← (A) - nn - $\bar{C}$	00XXXXX00	1 1 1 0	<B <sub>2</sub> >	1 0 0 1	E9	2	2	1	
	SBC \$zz	(A) ← (A) - (M) - $\bar{C}$ 其中 M=(zz)	00XXXXX00	1 1 1 0	<B <sub>2</sub> >	0 1 0 1	E5	2	3	1	
	SBC \$zz,X	(A) ← (A) - (M) - $\bar{C}$ 其中 M=(zz+(X))	00XXXXX00	1 1 1 1	<B <sub>2</sub> >	0 1 0 1	F5	2	4	1	
	SBC \$hhll	(A) ← (A) - (M) - $\bar{C}$ 其中 M=(hhll)	00XXXXX00	1 1 1 0	<B <sub>2</sub> >	1 1 0 1	ED	3	4	1	
	SBC \$hhll,X	(A) ← (A) - (M) - $\bar{C}$ 其中 M=(hhll+(X))	00XXXXX00	1 1 1 1	<B <sub>2</sub> >	1 1 0 1	FD	3	5	1	
	SBC \$hhll,Y	(A) ← (A) - (M) - $\bar{C}$ 其中 M=(hhll+(Y))	00XXXXX00	1 1 1 1	<B <sub>2</sub> >	1 0 0 1	F9	3	5	1	
	SBC (\$zz,X)	(A) ← (A) - (M) - $\bar{C}$ 其中 M=((zz+(X)+1)(zz+X))	00XXXXX00	1 1 1 0	<B <sub>2</sub> >	0 0 0 1	E1	2	6	1	
	SBC (\$zz,Y)	(A) ← (A) - (M) - $\bar{C}$ 其中 M=((zz+1)(zz)+(Y))	00XXXXX00	1 1 1 1	<B <sub>2</sub> >	0 0 0 1	F1	2	6	1	
	INC A	(A) ← (A) + 1	0XXXXXX0X	0 0 1 1		1 0 1 0	3A	1	2		
	INC \$zz	(M) ← (M) + 1 其中 M=(zz)	0XXXXXX0X	1 1 1 0	<B <sub>2</sub> >	0 1 1 0	E6	2	5		
	INC \$zz,X	(M) ← (M) + 1 其中 M=(zz+(X))	0XXXXXX0X	1 1 1 1	<B <sub>2</sub> >	0 1 1 0	F6	2	6		
	INC \$hhll	(M) ← (M) + 1 其中 M=(hhll)	0XXXXXX0X	1 1 1 0	<B <sub>2</sub> >	1 1 1 0	EE	3	6		
	INC \$hhll,X	(M) ← (M) + 1 其中 M=(hhll+(X))	0XXXXXX0X	1 1 1 1	<B <sub>2</sub> >	1 1 1 0	FE	3	7		
	DEC A	(A) ← (A) - 1	0XXXXXX0X	0 0 0 1		1 0 1 0	1A	1	2		
	DEC \$zz	(M) ← (M) - 1 其中 M=(zz)	0XXXXXX0X	1 1 0 0	<B <sub>2</sub> >	0 1 1 0	C6	2	5		
	DEC \$zz,X	(M) ← (M) - 1 其中 M=(zz+(X))	0XXXXXX0X	1 1 0 1	<B <sub>2</sub> >	0 1 1 0	D6	2	6		
	DEC \$hhll	(M) ← (M) - 1 其中 M=(hhll)	0XXXXXX0X	1 1 0 0	<B <sub>2</sub> >	1 1 1 0	CE	3	6		
	DEC \$hhll,X	(M) ← (M) - 1 其中 M=(hhll+(X))	0XXXXXX0X	1 1 0 1	<B <sub>2</sub> >	1 1 1 0	DE	3	7		
	乘 除 运 算 指 令	INX	(X) ← (X) + 1	0XXXXXX0X	1 1 1 0		1 0 0 0	E8	1	2	
	DEX	(X) ← (X) - 1	0XXXXXX0X	1 1 0 0		1 0 1 0	CA	1	2		
INY	(Y) ← (Y) + 1	0XXXXXX0X	1 1 0 0		1 0 0 0	C8	1	2			
DEY	(Y) ← (Y) - 1	0XXXXXX0X	1 0 0 1		1 0 0 0	88	1	2			
MUL \$zz,X	M(S), (A) ← (A) × M(zz+(X)) (S) ← (S) - 1	XXXXXXXXXX	0 1 1 0		0 0 1 0	62	2	15	6		
DIV \$zz,X	(A) ← (M(zz+(X)+1), M(zz+(X))) ÷ A M(S) ← 余数的1的补码 (S) ← (S) - 1	XXXXXXXXXX	1 1 1 0		0 0 1 0	E2	2	16	6		

项目 分类	指令符号	功 能	标 志	指 令 码		字节数	周期数	注		
			N V T B D I Z C	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	16进制码					
运 算 指 令	AND # $\$nn$	$(A) \leftarrow (A) \wedge nn$	0×××××0××	0 0 1 0	$\langle B_2 \rangle$ 1 0 0 1	29	2	2	1	
	AND $\$zz$	$(A) \leftarrow (A) \wedge (M)$ 其中 $M=(zz)$	0×××××0××	0 0 1 0	$\langle B_2 \rangle$ 0 1 0 1	25	2	3	1	
	AND $\$zz,X$	$(A) \leftarrow (A) \wedge (M)$ 其中 $M=(zz+(X))$	0×××××0××	0 0 1 1	$\langle B_2 \rangle$ 0 1 0 1	35	2	4	1	
	AND $\$hhll$	$(A) \leftarrow (A) \wedge (M)$ 其中 $M=(hhll)$	0×××××0××	0 0 1 0	$\langle B_2 \rangle$ 1 1 0 1	2D	3	4	1	
	AND $\$hhll,X$	$(A) \leftarrow (A) \wedge (M)$ 其中 $M=(hhll+(X))$	0×××××0××	0 0 1 1	$\langle B_2 \rangle$ 1 1 0 1	3D	3	5	1	
	AND $\$hhll,Y$	$(A) \leftarrow (A) \wedge (M)$ 其中 $M=(hhll+(Y))$	0×××××0××	0 0 1 1	$\langle B_2 \rangle$ 1 0 0 1	39	3	5	1	
	AND ( $\$zz,X$ )	$(A) \leftarrow (A) \wedge (M)$ 其中 $M=((zz+(X)+1)(zz+X))$	0×××××0××	0 0 1 0	$\langle B_2 \rangle$ 0 0 0 1	21	2	6	1	
	AND ( $\$zz,Y$ )	$(A) \leftarrow (A) \wedge (M)$ 其中 $M=((zz+1)(zz)+(Y))$	0×××××0××	0 0 1 1	$\langle B_2 \rangle$ 0 0 0 1	31	2	6	1	
	ORA # $\$nn$	$(A) \leftarrow (A) \vee nn$	0×××××0××	0 0 0 0	$\langle B_2 \rangle$ 1 0 0 1	09	2	2	1	
	ORA $\$zz$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(zz)$	0×××××0××	0 0 0 0	$\langle B_2 \rangle$ 0 1 0 1	05	2	3	1	
	ORA $\$zz,X$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(zz+(X))$	0×××××0××	0 0 0 1	$\langle B_2 \rangle$ 0 1 0 1	15	2	4	1	
	ORA $\$hhll$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(hhll)$	0×××××0××	0 0 0 0	$\langle B_2 \rangle$ 1 1 0 1	0D	3	4	1	
	ORA $\$hhll,X$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(hhll+(X))$	0×××××0××	0 0 0 1	$\langle B_2 \rangle$ 1 1 0 1	1D	3	5	1	
	ORA $\$hhll,Y$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(hhll+(Y))$	0×××××0××	0 0 0 1	$\langle B_2 \rangle$ 1 0 0 1	19	3	5	1	
	ORA ( $\$zz,X$ )	$(A) \leftarrow (A) \vee (M)$ 其中 $M=((zz+(X)+1)(zz+X))$	0×××××0××	0 0 0 0	$\langle B_2 \rangle$ 0 0 0 1	01	2	6	1	
	ORA ( $\$zz,Y$ )	$(A) \leftarrow (A) \vee (M)$ 其中 $M=((zz+1)(zz)+(Y))$	0×××××0××	0 0 0 1	$\langle B_2 \rangle$ 0 0 0 1	11	2	6	1	
	EOR # $\$nn$	$(A) \leftarrow (A) \vee nn$	0×××××0××	0 1 0 0	$\langle B_2 \rangle$ 1 0 0 1	49	2	2	1	
	EOR $\$zz$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(zz)$	0×××××0××	0 1 0 0	$\langle B_2 \rangle$ 0 1 0 1	45	2	3	1	
	EOR $\$zz,X$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(zz+(X))$	0×××××0××	0 1 0 1	$\langle B_2 \rangle$ 0 1 0 1	55	2	4	1	
	EOR $\$hhll$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(hhll)$	0×××××0××	0 1 0 0	$\langle B_2 \rangle$ 1 1 0 1	4D	3	4	1	
	EOR $\$hhll,X$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(hhll+(X))$	0×××××0××	0 1 0 1	$\langle B_2 \rangle$ 1 1 0 1	5D	3	5	1	
	EOR $\$hhll,Y$	$(A) \leftarrow (A) \vee (M)$ 其中 $M=(hhll+(Y))$	0×××××0××	0 1 0 1	$\langle B_2 \rangle$ 1 0 0 1	59	3	5	1	
	EOR ( $\$zz,X$ )	$(A) \leftarrow (A) \vee (M)$ 其中 $M=((zz+(X)+1)(zz+X))$	0×××××0××	0 1 0 0	$\langle B_2 \rangle$ 0 0 0 1	41	2	6	1	
	EOR ( $\$zz,Y$ )	$(A) \leftarrow (A) \vee (M)$ 其中 $M=((zz+1)(zz)+(Y))$	0×××××0××	0 1 0 1	$\langle B_2 \rangle$ 0 0 0 1	51	2	6	1	
	COM $\$zz$	$(M) \leftarrow (\bar{M})$ 其中 $M=(zz)$	0×××××0××	0 1 0 0	$\langle B_2 \rangle$ 0 1 0 0	44	2	5		
	BIT $\$zz$	$(A) \wedge (M)$ 其中 $M=(zz)$	$M_7M_6 \times \times \times \times 0 \times \times$	0 0 1 0	$\langle B_2 \rangle$ 0 1 0 0	24	2	3		
	BIT $\$hhll$	$(A) \wedge (M)$ 其中 $M=(hhll)$	$M_7M_6 \times \times \times \times 0 \times \times$	0 0 1 0	$\langle B_2 \rangle$ 1 1 0 0	2C	3	4		
	TST $\$zz$	$(M)=0?$ 其中 $M=(zz)$	0×××××0××	0 1 1 0	$\langle B_2 \rangle$ 0 1 0 0	64	2	3		
	比 较 指 令	CMP # $\$nn$	$(A)-nn$	0×××××0××	1 1 0 0	$\langle B_2 \rangle$ 1 0 0 1	C9	2	2	3
		CMP $\$zz$	$(A)-(M)$ 其中 $M=(zz)$	0×××××0××	1 1 0 0	$\langle B_2 \rangle$ 0 1 0 1	C5	2	3	3
		CMP $\$zz,X$	$(A)-(M)$ 其中 $M=(zz+(X))$	0×××××0××	1 1 0 1	$\langle B_2 \rangle$ 0 1 0 1	D5	2	4	3
		CMP $\$hhll$	$(A)-(M)$ 其中 $M=(hhll)$	0×××××0××	1 1 0 0	$\langle B_2 \rangle$ 1 1 0 1	CD	3	4	3
CMP $\$hhll,X$		$(A)-(M)$ 其中 $M=(hhll+(X))$	0×××××0××	1 1 0 1	$\langle B_2 \rangle$ 1 1 0 1	DD	3	5	3	
CMP $\$hhll,Y$		$(A)-(M)$ 其中 $M=(hhll+(Y))$	0×××××0××	1 1 0 1	$\langle B_2 \rangle$ 1 0 0 1	D9	3	5	3	
CMP ( $\$zz,X$ )		$(A)-(M)$ 其中 $M=((zz+(X)+1)(zz+X))$	0×××××0××	1 1 0 0	$\langle B_2 \rangle$ 0 0 0 1	C1	2	6	3	
CMP ( $\$zz,Y$ )		$(A)-(M)$ 其中 $M=((zz+1)(zz)+(Y))$	0×××××0××	1 1 0 1	$\langle B_2 \rangle$ 0 0 0 1	D1	2	6	3	
CPX # $\$nn$		$(X)-nn$	0×××××0××	1 1 1 0	$\langle B_2 \rangle$ 0 0 0 0	E0	2	2		
CPX $\$zz$		$(X)-(M)$ 其中 $M=(zz)$	0×××××0××	1 1 1 0	$\langle B_2 \rangle$ 0 1 0 0	E4	2	3		
CPX $\$hhll$		$(X)-(M)$ 其中 $M=(hhll)$	0×××××0××	1 1 1 0	$\langle B_2 \rangle$ 1 1 0 0	EC	3	4		
CPY # $\$nn$		$(Y)-nn$	0×××××0××	1 1 0 0	$\langle B_2 \rangle$ 0 0 0 0	C0	2	2		
CPY $\$zz$	$(Y)-(M)$ 其中 $M=(zz)$	0×××××0××	1 1 0 0	$\langle B_2 \rangle$ 0 1 0 0	C4	2	3			
CPY $\$hhll$	$(Y)-(M)$ 其中 $M=(hhll)$	0×××××0××	1 1 0 0	$\langle B_2 \rangle$ 1 1 0 0	CC	3	4			



项目 分类	指令符号	功 能	标 志							指 令 码			字节数	周期数	注							
			N	V	T	B	D	I	Z	C	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>				16进制码						
转 跳 指 令	BRA \$hhll	(PC)←(PC)+2+Rel	×	×	×	×	×	×	×	1	0	0	0	0	0	0	0	80	2	4		
	JMP \$hhll	(PC)←hhll	×	×	×	×	×	×	×	0	1	0	0	1	1	0	0	4C	3	3		
	JMP (\$hll)	(PC <sub>L</sub> )←(hll),(PC <sub>H</sub> )←(hll+1)	×	×	×	×	×	×	×	0	1	1	0	1	1	0	0	6C	3	5		
	JMP (\$zz)	(PC <sub>L</sub> )←(zz),(PC <sub>H</sub> )←(zz+1)	×	×	×	×	×	×	×	1	0	1	1	0	0	1	0	B2	2	4		
	JSR \$hhll	按(M(S))←(PC <sub>H</sub> ),(S)←(S)-1,(M(S))←(PC <sub>L</sub> ), (S)←(S)-1顺序进行, 然后(PC)←hhll	×	×	×	×	×	×	×	0	0	1	0	0	0	0	0	20	3	6		
	JSR (\$zz)	按(M(S))←(PC <sub>H</sub> ),(S)←(S)-1,(M(S))←(PC <sub>L</sub> ), (S)←(S)-1顺序进行, 然后(PC <sub>L</sub> )←zz,(PC <sub>H</sub> )←(zz+1)	×	×	×	×	×	×	×	0	0	0	0	0	0	1	0	02	2	7		
JSR \ \$hhll	按(M(S))←(PC <sub>H</sub> ),(S)←(S)-1,(M(S))←(PC <sub>L</sub> ), (S)←(S)-1顺序进行, 然后(PC <sub>L</sub> )←ll(地址低位字节), (PC <sub>H</sub> )←FF	×	×	×	×	×	×	×	0	0	1	0	0	0	1	0	22	2	5			
移 回 指 令	BBC i,A,\$hhll	在(Ai)=0时 (PC)←(PC)+2+Rel 其中 i=0~7 在(Ai)=1时 (PC)←(PC)+2	×	×	×	×	×	×	×	i	i	i	1	0	0	1	1	(1+2i)×10+3	2	4	4	
	BBC i,\$zz,\$hhll	在(Mi)=0时 (PC)←(PC)+3+Rel 其中 i=0~7 在(Mi)=1时 (PC)←(PC)+3	×	×	×	×	×	×	×	i	i	i	1	0	1	1	1	(1+2i)×10+7	3	5	4	
	BBS i,A,\$hhll	在(Ai)=1时 (PC)←(PC)+2+Rel 其中 i=0~7 在(Ai)=0时 (PC)←(PC)+2	×	×	×	×	×	×	×	i	i	i	0	0	0	1	1	2i×10+3	2	4	4	
	BBS i,\$zz,\$hhll	在(Mi)=1时 (PC)←(PC)+3+Rel 其中 i=0~7 在(Mi)=0时 (PC)←(PC)+3	×	×	×	×	×	×	×	i	i	i	0	0	1	1	1	2i×10+7	3	5	4	
	BCC \$hhll	在(C)=0时 (PC)←(PC)+2+Rel 在(C)=1时 (PC)←(PC)+2	×	×	×	×	×	×	×	1	0	0	1	0	0	0	0	90	2	2	4	
	BCS \$hhll	在(C)=1时 (PC)←(PC)+2+Rel 在(C)=0时 (PC)←(PC)+2	×	×	×	×	×	×	×	1	0	1	1	0	0	0	0	B0	2	2	4	
	BNE \$hhll	在(Z)=0时 (PC)←(PC)+2+Rel 在(Z)=1时 (PC)←(PC)+2	×	×	×	×	×	×	×	1	1	0	1	0	0	0	0	D0	2	2	4	
	BEQ \$hhll	在(Z)=1时 (PC)←(PC)+2+Rel 在(Z)=0时 (PC)←(PC)+2	×	×	×	×	×	×	×	1	1	1	1	0	0	0	0	F0	2	2	4	
	BPL \$hhll	在(N)=0时 (PC)←(PC)+2+Rel 在(N)=1时 (PC)←(PC)+2	×	×	×	×	×	×	×	0	0	0	1	0	0	0	0	10	2	2	4	
	BMI \$hhll	在(N)=1时 (PC)←(PC)+2+Rel 在(N)=0时 (PC)←(PC)+2	×	×	×	×	×	×	×	0	0	1	1	0	0	0	0	30	2	2	4	
	BVC \$hhll	在(V)=0时 (PC)←(PC)+2+Rel 在(V)=1时 (PC)←(PC)+2	×	×	×	×	×	×	×	0	1	0	1	0	0	0	0	50	2	2	4	
	BVS \$hhll	在(V)=1时 (PC)←(PC)+2+Rel 在(V)=0时 (PC)←(PC)+2	×	×	×	×	×	×	×	0	1	1	1	0	0	0	0	70	2	2	4	
	RTI	(S)←(S)+1,(PS)←(M(S)),(S)←(S)+1, (PC <sub>L</sub> )←(M(S)),(S)←(S)+1,(PC <sub>H</sub> )←(M(S))	(保存到堆栈的值)	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	40	1	6	
	RTS	(S)←(S)+1,(PC <sub>L</sub> )←(M(S)),(S)←(S)+1, (PC <sub>H</sub> )←(M(S)),(PC)←(PC)+1	×	×	×	×	×	×	×	0	1	1	0	0	0	0	0	60	1	6		
中 断	BRK	(B)←1,(PC)←(PC)+2,(M(S))←(PC <sub>H</sub> ),(S)←(S)-1, (M(S))←(PC <sub>L</sub> ),(S)←(S)-1,(M(S))←(PS), (S)←(S)-1,(I)←1,(PC)←BADRS	×	×	×	1	1	1	×	×	0	0	0	0	0	0	0	0	00	1	7	
其它	NOP	(PC)←(PC)+1	×	×	×	×	×	×	×	1	1	1	0	1	0	1	0	EA	1	2		
特殊	WIT	停止内部时钟	×	×	×	×	×	×	×	1	1	0	0	0	0	1	0	C2	1	2		
	STP	停止振荡	×	×	×	×	×	×	×	0	1	0	0	0	0	1	0	42	1	2	5	

符 号	内 容	符 号	内 容
A	累加器	hh	地址高位字节 0~255的数据
Ai	累加器的位i	ll	地址低位字节 0~255的数据
X	变址寄存器X	zz	零页地址 0~255的数据
Y	变址寄存器Y	nn	0~255的数据
M	存储器	i	0~7的数据
Mi	存储器的位i	iii	0~7的数据
PS	处理器状态寄存器	<B <sub>2</sub> >	指令的第2字节
S	栈指针	<B <sub>3</sub> >	指令的第3字节
PC	程序计数器	Rel	相对地址
PC <sub>L</sub>	程序计数器的低位字节	BADRS	中断地址
PC <sub>H</sub>	程序计数器的高位字节	←	表示移动数据的方向
N	负标志	0	表示寄存器、存储器等的內容
V	溢出标志	+	加法
T	X变址模式标志	-	减法
B	中断标志	×	乘法
D	10进制模式标志	÷	除法
I	中断禁止标志	∨	逻辑或
Z	零标志	∧	逻辑与
C	进位标志	∨	逻辑异或
#	立即模式	—	求反
\$	16进制记数	×	在指令执行后, 标志不变
¥	专用页	○	在指令执行后, 标志变化

注 1. 表中记载的功能为(T)=0时的功能。在(T)=1时, 用(M(X))替代(A), 周期数+3。  
 2. 同上, 但是, 周期数+2。  
 3. 同上, 但是, 周期数+1。  
 4. 在发生转移时, 周期数+2。  
 5. 根据产品, 能使STP指令无效, 当STP指令无效时, 和2次连续执行NOP指令相同的运行, 周期数为2周期。  
 6. 部分产品不能使用乘法和除法指令。





## 附录 3 740 族指令码对应表

D7-D4 \ D3-D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND,X	JSR ZP,IND	BBS 0,A	—	ORA ZP	ASL ZP	BBS 0,ZP	PHP	ORA IMM	ASL A	SEB 0,A	—	ORA ABS	ASL ABS	SEB 0,ZP
0001	1	BPL	ORA IND,Y	CLT	BBC 0,A	—	ORA ZP,X	ASL ZP,X	BBC 0,ZP	CLC	ORA ABS,Y	DEC A	CLB 0,A	—	ORA ABS,X	ASL ABS,X	CLB 0,ZP
0010	2	JSR ABS	AND IND,X	JSR SP	BBS 1,A	BIT ZP	AND ZP	ROL ZP	BBS 1,ZP	PLP	AND IMM	ROL A	SEB 1,A	BIT ABS	AND ABS	ROL ABS	SEB 1,ZP
0011	3	BMI	AND IND,Y	SET	BBC 1,A	—	AND ZP,X	ROL ZP,X	BBC 1,ZP	SEC	AND ABS,Y	INC A	CLB 1,A	LDM ZP	AND ABS,X	ROL ABS,X	CLB 1,ZP
0100	4	RTI	EOR IND,X	STP (注)	BBS 2,A	COM ZP	EOR ZP	LSR ZP	BBS 2,ZP	PHA	EOR IMM	LSR A	SEB 2,A	JMP ABS	EOR ABS	LSR ABS	SEB 2,ZP
0101	5	BVC	EOR IND,Y	—	BBC 2,A	—	EOR ZP,X	LSR ZP,X	BBC 2,ZP	CLI	EOR ABS,Y	—	CLB 2,A	—	EOR ABS,X	LSR ABS,X	CLB 2,ZP
0110	6	RTS	ADC IND,X	MUL ZP,X (注)	BBS 3,A	TST ZP	ADC ZP	ROR ZP	BBS 3,ZP	PLA	ADC IMM	ROR A	SEB 3,A	JMP IND	ADC ABS	ROR ABS	SEB 3,ZP
0111	7	BVS	ADC IND,Y	—	BBC 3,A	—	ADC ZP,X	ROR ZP,X	BBC 3,ZP	SEI	ADC ABS,Y	—	CLB 3,A	—	ADC ABS,X	ROR ABS,X	CLB 3,ZP
1000	8	BRA	STA IND,X	RRF ZP	BBS 4,A	STY ZP	STA ZP	STX ZP	BBS 4,ZP	DEY	—	TXA	SEB 4,A	STY ABS	STA ABS	STX ABS	SEB 4,ZP
1001	9	BCC	STA IND,Y	—	BBC 4,A	STY ZP,X	STA ZP,X	STX ZP,Y	BBC 4,ZP	TYA	STA ABS,Y	TXS	CLB 4,A	—	STA ABS,X	—	CLB 4,ZP
1010	A	LDY IMM	LDA IND,X	LDX IMM	BBS 5,A	LDY ZP	LDA ZP	LDX ZP	BBS 5,ZP	TAY	LDA IMM	TAX	SEB 5,A	LDY ABS	LDA ABS	LDX ABS	SEB 5,ZP
1011	B	BCS	LDA IND,Y	JMP ZP,IND	BBC 5,A	LDY ZP,X	LDA ZP,X	LDX ZP,Y	BBC 5,ZP	CLV	LDA ABS,Y	TSX	CLB 5,A	LDY ABS,X	LDA ABS,X	LDX ABS,Y	CLB 5,ZP
1100	C	CPY IMM	CMP IND,X	WIT	BBS 6,A	CPY ZP	CMP ZP	DEC ZP	BBS 6,ZP	INY	CMP IMM	DEX	SEB 6,A	CPY ABS	CMP ABS	DEC ABS	SEB 6,ZP
1101	D	BNE	CMP IND,Y	—	BBC 6,A	—	CMP ZP,X	DEC ZP,X	BBC 6,ZP	CLD	CMP ABS,Y	—	CLB 6,A	—	CMP ABS,X	DEC ABS,X	CLB 6,ZP
1110	E	CPX IMM	SBC IND,X	DIV ZP,X (注)	BBS 7,A	CPX ZP	SBC ZP	INC ZP	BBS 7,ZP	INX	SBC IMM	NOP	SEB 7,A	CPX ABS	SBC ABS	INC ABS	SEB 7,ZP
1111	F	BEQ	SBC IND,Y	—	BBC 7,A	—	SBC ZP,X	INC ZP,X	BBC 7,ZP	SED	SBC ABS,Y	—	CLB 7,A	—	SBC ABS,X	INC ABS,X	CLB 7,ZP

注：有无这些指令根据产品而不同。

- 3字节指令
- 2字节指令
- 1字节指令

时钟控制指令和乘法指令取决于产品，所以请参照各产品的功能说明。



修订记录

740 族 软件手册

Rev.	发行日	修订内容	
		页	修订要点
1.00	2006.03.21	—	初版发行



---

瑞萨16位单片机  
软件手册  
740族

Publication Date: 1st Edition, March 21, 2006  
Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.  
Edited by: Customer Support Department  
Global Strategic Communication Div.  
Renesas Solutions Corp.

---

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

---



**RENESAS SALES OFFICES**

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

**Renesas Technology America, Inc.**  
450 Holger Way, San Jose, CA 95134-1368, U.S.A  
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

**Renesas Technology Europe Limited**  
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

**Renesas Technology (Shanghai) Co., Ltd.**  
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120  
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

**Renesas Technology Hong Kong Ltd.**  
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong  
Tel: <852> 2265-6688, Fax: <852> 2730-6071

**Renesas Technology Taiwan Co., Ltd.**  
10th Floor, No.99, Fushing North Road, Taipei, Taiwan  
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

**Renesas Technology Singapore Pte. Ltd.**  
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: <65> 6213-0200, Fax: <65> 6278-8001

**Renesas Technology Korea Co., Ltd.**  
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea  
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

**Renesas Technology Malaysia Sdn. Bhd**  
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: <603> 7955-9390, Fax: <603> 7955-9510





740群

**RENESAS**

瑞萨电子株式会社

RCJ09B0013-0100

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [16-bit Microcontrollers - MCU category](#):*

*Click to view products by [Renesas manufacturer](#):*

Other Similar products are found below :

[MB90F036APMC-GSE1](#) [MB90F342CASPMC-GSE1](#) [MB90F345CESPMC-GE1](#) [MB90F349CAPFR-GSE1](#) [MB90F428GCPFR-GSE1](#)  
[MB90F462APFM-GE1](#) [MB90F462APMC-G-SNE1](#) [MB90F497GPF-GE1](#) [MB90F546GSPFR-GE1](#) [MB90F947APFR-GS-SPE1](#)  
[MB96F683RBPMC-GSAE1](#) [R5F11BGEAFB#30](#) [DF3026XBL25V](#) [S912ZVFP64F1VLL](#) [R4F24268NVRFQV](#) [R5F107DEGSP#X0](#)  
[R5F11B7EANA#U0](#) [R5F21172DSP#U0](#) [MB90092PF-G-BNDE1](#) [MB90F335APMC1-G-SPE1](#) [MB90F342CASPF-R-GS-N2E1](#)  
[MB90F345CAPFR-GSE1](#) [MB90F543GPF-GE1](#) [MB90F546GSPF-GE1](#) [MB90F568PMCR-GE1](#) [MB90F594APFR-GE1](#) [MB90F882ASPMC-](#)  
[GE1](#) [MB96F387RSBPMC-GSE2](#) [MB96F387RSBPMC-GS-N2E2](#) [MB96F395RSAPMC-GSE2](#) [MB96F623RBPMC1-GSE1](#)  
[MB96F646RBPMC-GSE1](#) [XE167F96F66LACFXUMA1](#) [MB96F696RBPMC-GSAE1](#) [MB96F018RBPMC-GSE1](#) [MB90F962SPMCR-GE1](#)  
[MB90F867ASPFR-GE1](#) [MB90F543GPF-G-FLE1](#) [MB90F345CESPF-GE1](#) [M30290FCHP#U3A](#) [R5F104AEASP#V0](#) [R5F100BCANA#U0](#)  
[R5F100ACASP#V0](#) [R5F117BCGFP#30](#) [M30626FJPGP#U5C](#) [R5F11AGGANB#20](#) [R5F11B7CANA#U0](#) [MB90F362TESPMCR-GN9E1](#)  
[MB91F248PFV-GE1](#) [LC88F58B0AU-SQFPH](#)