

M8Rxxx

指令集说明

Version 1.04

2014年11月



上海磐芯电子有限公司

SHANGHAI MASSES ELECTRONIC Co., Ltd.

本公司保留对产品在可靠性，功能和设计方面的改进作进一步说明的权利。
说明文档的更改，恕不另行通知。

修正记录

版本	日期	描述
Ver1.02	2014-5-15	补全指令说明
Ver1.03	2014-10-8	勘误
Ver1.04	2014-11-24	删除 ADCIA 指令

目录

目录.....	3
1 M8RXXX 指令集简述.....	5
1.1 概述.....	5
1.2 符号说明.....	5
2 M8RXXX 指令集表.....	6
3 M8RXXX 指令说明.....	8
ADDAR R,D.....	8
ADCAR R,D.....	8
SUBAR R,D.....	8
SBCAR R,D.....	8
SUBRA R,D.....	8
SBCRA R,D.....	8
ANDAR R,D.....	9
ORAR R,D.....	9
XORAR R,D.....	9
COMR R,D.....	9
MOVR R,D.....	9
MOVAR R.....	9
CLRR R.....	10
SWAPR R,D.....	10
RLR R,D.....	10
RLRNC R,D.....	10
RRR R,D.....	10
RRRNC R,D.....	10
DECR R,D.....	11
DJZR R,D.....	11
INCR R,D.....	11
JZR R,D.....	11
JNZR R,D.....	11
DJNZR R,D.....	11
JCMPPAR R.....	12
JNCMPAR R.....	12
JGAR R.....	12
JLAR R.....	12
XCHAR R.....	12
JBTS0 R,B.....	12
JBTS1 R,B.....	13
BCLR R,B.....	13
BSET R,B.....	13
ADDIA I.....	13

SUBIA I.....	13
SBCIA I.....	13
SUBAI I.....	14
SBCAI I.....	14
ANDIA I.....	14
ORIA I.....	14
XORIA I.....	14
MOVIA I.....	14
RETIA I.....	14
JCMPAI I.....	14
JNCMPAI I.....	15
RLA.....	15
RLANC.....	15
RRA.....	15
RRANC.....	15
DECA.....	15
DJZA.....	15
INCA.....	15
JZA.....	16
RETIE.....	16
RETURN.....	16
NOP.....	16
RDT.....	16
DAA.....	16
DSA.....	16
PUSH.....	16
POP.....	17
CLRWDT.....	17
CALL I.....	17
GOTO I.....	17

1 M8Rxxx指令集简述

1.1 概述

M8Rxxx系列指令集是一种精简指令集（RISC），指令宽度为16位，由操作码和0~2个操作数组成。指令按照功能可分为5类，即字节操作指令、位操作指令、立即数指令、分支指令、特殊控制指令。

每个指令周期由4个振荡周期组成，除非条件测试结果为真或指令执行改变了程序计数器的值，否则执行所有的指令都只需要一个指令周期。对于上述两种特征情况，指令执行需要两个指令周期。

任何一条指定文件寄存器作为指令一部分的指令都进行读-修改-写操作。读寄存器、修改数据并根据指令或目标标识符“d”存储结果。即使是写寄存器的指令也将先对改寄存器进行读操作。

1.2 符号说明

符号	范围	说明	符号	范围	说明
R/r	0-0x1ff	寄存器地址	C	-	进位标志
A	-	ACC 寄存器	DC	-	半进位标志
B/b	0-7	位地址	Z	-	零标志
I/i	0-0xff	立即数	d	0-1	目的操作数定义
K/k	0-0x1fff	标号	GIE	-	总中断使能位
TOS	-	栈顶	stkp	-	堆栈指针
PC	-	PC 指针			

2 M8Rxxx指令集表

指令集表中，d=1,目的操作数为R；d=0,目的操作数为A

指令类型	助记符	指令说明	周期数	影响标志位	备注
寄存器操作指令	ADDAR R,d	$R+A \rightarrow d$	1	Z,DC,C	
	ADCAR R,d	$R+A+C \rightarrow d$	1	Z,DC,C	
	SUBAR R,d	$A-R \rightarrow d$	1	Z,DC,C	
	SBCAR R,d	$A-R-\overline{C} \rightarrow d$	1	Z,DC,C	
	SUBRA R,d	$R-A \rightarrow d$	1	Z,DC,C	
	SBCRA R,d	$R-A-\overline{C} \rightarrow d$	1	Z,DC,C	
	ANDAR R,d	$R\&A \rightarrow d$	1	Z	
	ORAR R,d	$R A \rightarrow d$	1	Z	
	XORAR R,d	$R^{\wedge}A \rightarrow d$	1	Z	
	COMR R,d	$\overline{R} \rightarrow d$	1	Z	
	MOVR R,d	$R \rightarrow d$	1	Z	
	MOVAR R	$A \rightarrow R$	1	-	
	CLRR R	$0 \rightarrow R$	1	Z	
	SWAPR R,d	R 半字节交换 $\rightarrow d$	1	-	
	RLR R,d	$R[7] \rightarrow C, \{R[6:0],C\} \rightarrow d$	1	C	
	RLRNC R,d	$\{R[6:0],0\} \rightarrow d$	1	-	
	RRR R,d	$R[0] \rightarrow C, \{C,R[7:1]\} \rightarrow d$	1	C	
	RRRNC R,d	$\{0,R[7:1]\} \rightarrow d$	1	-	
	DECR R,d	$R-1 \rightarrow d$	1	Z	
	DJZR R,d	$R-1 \rightarrow d, \text{SKIP if } 0$	1(2)	-	
	INCR R,d	$R+1 \rightarrow d$	1	Z	
	JZR R,d	$R+1 \rightarrow d, \text{SKIP if } 0$	1(2)	-	
	JNZR R,d	$R+1 \rightarrow d, \text{SKIP if } !0$	1(2)	Z,C	(*1)
	DJNZR R,d	$R-1 \rightarrow d, \text{SKIP if } !0$	1(2)	Z,C	(*1)
	JCMPI R	SKIP if A=R	1(2)	Z,C	(*1)
	JNCMP R	SKIP if A≠R	1(2)	Z,C	(*1)
	JGAR R	SKIP if A≥R	1(2)	Z,C	(*1)
	JLAR R	SKIP if A<R	1(2)	Z,C	(*1)
XCHAR R	$A \leftrightarrow R$	1(2)	Z	(*1)	
位操作指令	JBTS0 R,b	SKIP if R[b]=0	1(2)	-	
	JBTS1 R,b	SKIP if R[b]=1	1(2)	-	
	BCLR R,b	$0 \rightarrow R[b]$	1	-	
	BSET R,b	$1 \rightarrow R[b]$	1	-	

指令类型	助记符	指令说明	周期数	影响标志位	备注
立即数操作指令	ADDIA I	$I+A \rightarrow A$	1	Z,DC,C	
	SUBIA I	$I-A \rightarrow A$	1	Z,DC,C	
	SBCIA I	$I-A-\overline{C} \rightarrow A$	1	Z,DC,C	
	SUBAI I	$A-I \rightarrow A$	1	Z,DC,C	
	SBCAI I	$A-I-\overline{C} \rightarrow A$	1	Z,DC,C	
	ANDIA I	$A\&I \rightarrow A$	1	Z	
	ORIA I	$A I \rightarrow A$	1	Z	
	XORIA I	$A\wedge I \rightarrow A$	1	Z	
	MOVIA I	$I \rightarrow A$	1	-	
	RETIA I	Stack \rightarrow PC, $I \rightarrow A$	2	-	
	JCMPAI I	SKIP if A=I	1(2)	Z,C	(*1)
	JNCPAI I	SKIP if A \neq I	1(2)	Z,C	(*1)
特殊操作指令	RLA	$A[7] \rightarrow C, \{A[6:0], C\} \rightarrow A$	1	C	
	RLANC	$\{A[6:0], 0\} \rightarrow A$	1	-	
	RRA	$A[0] \rightarrow C, \{C, A[7:1]\} \rightarrow A$	1	C	
	RRANC	$\{0, A[7:1]\} \rightarrow A$	1	-	
	DECA	$A-1 \rightarrow A$	1	Z	
	DJZA	$A-1 \rightarrow A$, SKIP if 0	1(2)	-	
	INCA	$A+1 \rightarrow A$	1	Z	
	JZA	$A+1 \rightarrow A$, SKIP if 0	1(2)	-	
	RETIE	Stack \rightarrow PC, $1 \rightarrow$ GIE	2	-	
	RETURN	Stack \rightarrow PC	2	-	
	NOP	None Operation	1	-	
	RDT	ROM[$\{fsr1, fsr0\}$] \rightarrow {HBUF, A}	2	-	
	DAA	加法后十进制调整	1	DC, C	
	DSA	减法后十进制调整	1	DC, C	
	PUSH	A, STATUS 压栈	1	-	(*1)(*2)
	POP	A, STATUS 出栈	1	Z, DC, C	(*1)(*2)
	CLRWDT	清除 WDT 寄存器	1	PD, TO	(*1)
分支指令	CALL I	$I \rightarrow$ PC, PC \rightarrow Stack	2	-	
	GOTO I	$I \rightarrow$ PC	2	-	

备注: (*1)此指令 M8R82 不支持
 (*2)此指令 M8R62/621/622 不支持

3 M8Rxxx指令说明

ADDAR R,d

说明: R 与 A 相加

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) + (A) \rightarrow$ (目标寄存器)

受影响的状态位: Z, DC, C

说明: 将寄存器 R 的内容与 A 寄存器相加.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

ADCAR R,d

说明: R 与 A 带进位相加

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) + (A) + (C) \rightarrow$ (目标寄存器)

受影响的状态位: Z, DC, C

说明: 将寄存器 R 的内容与 A 寄存器及进位值相加.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

SUBAR R,d

说明: A 减去 R

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(A) - (R) \rightarrow$ (目标寄存器)

受影响的状态位: Z, DC, C

说明: 按二进制补码方式, 用 A 寄存器内容减去寄存器 R 的内容,

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

SBCAR R,d

说明: A 带借位减 R

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(A) - (R) - (\overline{C}) \rightarrow$ (目标寄存器)

受影响的状态位: Z, DC, C

说明: 按二进制补码方式, 用 A 寄存器内容减去寄存器 R 的内容.

如果 C = 0, 说明前次运算有借位, 则需再减去 1; 否则不需要再减.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

SUBRA R,d

说明: R 减去 A

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) - (A) \rightarrow$ (目标寄存器)

受影响的状态位: Z, DC, C

说明: 按二进制补码方式, 用寄存器 R 内容减去 A 寄存器的内容,

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

SBCRA R,d

说明: R 带借位减 A

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) - (A) - (\overline{C}) \rightarrow$ (目标寄存器)

受影响的状态位: Z, DC, C

说明: 按二进制补码方式, 用寄存器 R 内容减去 A 寄存器的内容.

如果 C = 0, 说明前次运算有借位, 则需再减去 1; 否则不需要再减.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

ANDAR R,d

说明： R 与 A 做逻辑与运算

操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $(R) \text{ AND } (A) \rightarrow$ (目标寄存器)

受影响的状态位： Z

说明： 将寄存器 R 的内容与 A 寄存器进行逻辑与运算。

如果 d 为 0, 结果存入 A 寄存器。

如果 d 为 1, 结果存入寄存器 R。

ORAR R,d

说明： R 与 A 做逻辑或运算

操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $(R) \text{ OR } (A) \rightarrow$ (目标寄存器)

受影响的状态位： Z

说明： 将寄存器 R 的内容与 A 寄存器进行逻辑或运算。

如果 d 为 0, 结果存入 A 寄存器。

如果 d 为 1, 结果存入寄存器 R。

XORAR R,d

说明： R 与 A 做逻辑异或运算

操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $(R) \text{ XOR } (A) \rightarrow$ (目标寄存器)

受影响的状态位： Z

说明： 将寄存器 R 的内容与 A 寄存器进行逻辑异或运算。

如果 d 为 0, 结果存入 A 寄存器。

如果 d 为 1, 结果存入寄存器 R。

COMR R,d

说明： 对 R 取反

操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $\overline{(R)} \rightarrow$ (目标寄存器)

受影响的状态位： Z

说明： 将寄存器 R 的内容取反。

如果 d 为 0, 结果存入 A 寄存器。

如果 d 为 1, 结果存入寄存器 R。

MOVR R,d

说明： 传送 R

操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $(R) \rightarrow$ (目标寄存器)

受影响的状态位： Z

说明： 根据 d 的值将寄存器 R 的内容传送到目标寄存器。

如果 d 为 0, 传送到 A 寄存器。

如果 d 为 1, 传送到寄存器 R 本身。

因为状态标志位 Z 会受影响, 所以可用 d = 1 对寄存器 R 进行测试。

MOVAR R

说明： 将 A 的内容传送到 R

操作数： $0 \leq R \leq 0x1ff$

操作： $(A) \rightarrow (R)$

受影响的状态位： 无

说明： 将 A 寄存器的内容传送到寄存器 R。

CLRR R

说明： 将 R 清零
 操作数： $0 \leq R \leq 0x1ff$

操作： $0x00 \rightarrow (R)$
 $1 \rightarrow Z$

受影响的状态位： Z

说明： 寄存器 R 的内容清零
 Z 位置 1

SWAPR R,d

说明： 将 R 的高半字节和低半字节交换
 操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $(R[3:0]) \rightarrow (\text{目标寄存器}[7:4])$
 $(R[7:4]) \rightarrow (\text{目标寄存器}[3:0])$

受影响的状态位： 无

说明： 寄存器 R 内容的高半字节和低半字节相互交换，
 如果 d 为 0，结果存入 A 寄存器。
 如果 d 为 1，结果存入寄存器 R。

RLR R,d

说明： R 带进位循环左移

操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $(R[7]) \rightarrow (C)$
 $(\{R[6:0],(C)\}) \rightarrow (\text{目标寄存器})$

受影响的状态位： C

说明： 将寄存器 R 的内容连同进位值一起循环左移 1 位
 如果 d 为 0，结果存入 A 寄存器。
 如果 d 为 1，结果存入寄存器 R。

RLRNC R,d

说明： R 不带进位左移

操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $(\{R[6:0],0\}) \rightarrow (\text{目标寄存器})$

受影响的状态位： 无

说明： 将寄存器 R 的内容左移 1 位，最低位补 0
 如果 d 为 0，结果存入 A 寄存器。
 如果 d 为 1，结果存入寄存器 R。

RRR R,d

说明： R 带进位循环右移

操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $(R[0]) \rightarrow (C)$
 $(\{(C), R[7:1]\}) \rightarrow (\text{目标寄存器})$

受影响的状态位： C

说明： 将寄存器 R 的内容连同进位值一起循环右移 1 位
 如果 d 为 0，结果存入 A 寄存器。
 如果 d 为 1，结果存入寄存器 R。

RRRNC R,d

说明： R 不带进位右移

操作数： $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作： $(\{0, R[7:1]\}) \rightarrow (\text{目标寄存器})$

受影响的状态位： 无

说明： 将寄存器 R 的内容右移 1 位，最高位补 0
 如果 d 为 0，结果存入 A 寄存器。
 如果 d 为 1，结果存入寄存器 R。

DECR R,d

说明: R 递减 1

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) - 1 \rightarrow$ (目标寄存器)

受影响的状态位: Z

说明: 将寄存器 R 的内容递减 1.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

JZR R,d

说明: R 递增 1, 为 0 则跳过

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) + 1 \rightarrow$ (目标寄存器)

如果结果=0 则跳过

受影响的状态位: 无

说明: 将寄存器 R 的内容递增 1.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

如果结果不为 0, 则执行下一条指令

如果结果为 0, 则代为执行 NOP 指令

DJZR R,d

说明: R 递减 1, 为 0 则跳过

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) - 1 \rightarrow$ (目标寄存器)

如果结果=0 则跳过

受影响的状态位: 无

说明: 将寄存器 R 的内容递减 1.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

如果结果不为 0, 则执行下一条指令

如果结果为 0, 则代为执行 NOP 指令

JNZR R,d

说明: R 递增 1, 非 0 则跳过

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) + 1 \rightarrow$ (目标寄存器)

如果结果 \neq 0 则跳过

受影响的状态位: Z, C

说明: 将寄存器 R 的内容递增 1.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

如果结果不为 0, 则代为执行 NOP 指令

如果结果为 0, 则执行下一条指令

INCR R,d

说明: R 递增 1

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) + 1 \rightarrow$ (目标寄存器)

受影响的状态位: Z

说明: 将寄存器 R 的内容递增 1.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

DJNZR R,d

说明: R 递减 1, 非 0 则跳过

操作数: $0 \leq R \leq 0x1ff$

$d \in [0,1]$

操作: $(R) - 1 \rightarrow$ (目标寄存器)

如果结果 \neq 0 则跳过

受影响的状态位: Z, C

说明: 将寄存器 R 的内容递减 1.

如果 d 为 0, 结果存入 A 寄存器.

如果 d 为 1, 结果存入寄存器 R.

如果结果不为 0, 则代为执行 NOP 指令

如果结果为 0, 则执行下一条指令

JCMPPAR R

说明： 比较 A 与 R, 相等则跳过
 操作数： $0 \leq R \leq 0x1ff$

操作： (A) Vs (R)
 如果 $A = R$ 则跳过

受影响的状态位： Z, C

说明： 将 A 寄存器的内容与寄存器 R 比较。
 如果 $A = R$, 则代为执行 NOP 指令
 如果 $A \neq R$, 则执行下一条指令

JLAR R

说明： 比较 A 与 R, A 小于 R 则跳过
 操作数： $0 \leq R \leq 0x1ff$

操作： (A) Vs (R)
 如果 $A < R$ 则跳过

受影响的状态位： Z, C

说明： 将 A 寄存器的内容与寄存器 R 比较。
 如果 $A \geq R$, 则执行下一条指令
 如果 $A < R$, 则代为执行 NOP 指令

JNCMPAR R

说明： 比较 A 与 R, 不等则跳过
 操作数： $0 \leq R \leq 0x1ff$

操作： (A) Vs (R)
 如果 $A \neq R$ 则跳过

受影响的状态位： Z, C

说明： 将 A 寄存器的内容与寄存器 R 比较。
 如果 $A = R$, 则执行下一条指令
 如果 $A \neq R$, 则代为执行 NOP 指令

XCHAR R

说明： A 与 R 交换
 操作数： $0 \leq R \leq 0x1ff$

操作： (R) \rightarrow (A)
 (A) \rightarrow (R)

受影响的状态位： Z

说明： 将 A 寄存器的内容与寄存器 R 交换。

JGAR R

说明： 比较 A 与 R, A 大于等于 R 则跳过
 操作数： $0 \leq R \leq 0x1ff$

操作： (A) Vs (R)
 如果 $A \geq R$ 则跳过

受影响的状态位： Z, C

说明： 将 A 寄存器的内容与寄存器 R 比较。
 如果 $A \geq R$, 则代为执行 NOP 指令
 如果 $A < R$, 则执行下一条指令

JBTS0 R,b

说明： 测试 R 的指定位, 为 0 则跳过
 操作数： $0 \leq R \leq 0x1ff$
 $0 \leq b \leq 0x7$

操作： 如果 $(R[b]) = 0$ 则跳过

受影响的状态位： 无

说明： 如果寄存器 R 的位 $b = 1$, 则执行下一条指令
 如果寄存器 R 的位 $b = 0$, 则代为执行 NOP 指令

JBTS1 R,b

说明： 测试 R 的指定位，为 1 则跳过

操作数： $0 \leq R \leq 0x1ff$

$0 \leq b \leq 0x7$

操作： 如果 $(R[b]) = 1$ 则跳过

受影响的状态位： 无

说明： 如果寄存器 R 的位 $b = 1$ ，则代为执行 NOP 指令

如果寄存器 R 的位 $b = 0$ ，则执行下一条指令

BCLR R,b

说明： 将 R 的指定位清零

操作数： $0 \leq R \leq 0x1ff$

$0 \leq b \leq 0x7$

操作： $0 \rightarrow (R[b])$

受影响的状态位： 无

说明： 将寄存器 R 的位 b 清零

BSET R,b

说明： 将 R 的指定位置 1

操作数： $0 \leq R \leq 0x1ff$

$0 \leq b \leq 0x7$

操作： $1 \rightarrow (R[b])$

受影响的状态位： 无

说明： 将寄存器 R 的位 b 置 1

ADDIA I

说明： I 与 A 相加

操作数： $0 \leq I \leq 0xff$

操作： $(I) + (A) \rightarrow (A)$

受影响的状态位： Z, DC, C

说明： 将立即数 I 与 A 寄存器相加。
结果存入 A 寄存器。

SUBIA I

说明： I 减去 A

操作数： $0 \leq I \leq 0xff$

操作： $(I) - (A) \rightarrow (A)$

受影响的状态位： Z, DC, C

说明： 按二进制补码方式，用立即数 I 减去 A 寄存器，结果存入 A 寄存器。

SBCIA I

说明： I 带借位减 A

操作数： $0 \leq I \leq 0xff$

操作： $(I) - (A) - (\overline{C}) \rightarrow (A)$

受影响的状态位： Z, DC, C

说明： 按二进制补码方式，用立即数 I 减去 A 寄存器的内容。

如果 $C = 0$ ，说明前次运算有借位，则需再减去 1；否则不需要再减。

结果存入 A 寄存器。

SUBAI I

说明： A 减去 I

操作数： $0 \leq I \leq 0xff$

操作： $(A) - (I) \rightarrow (A)$

受影响的状态位： Z, DC, C

说明： 按二进制补码方式，用 A 寄存器内容减去立即数 I，结果存入 A 寄存器。

XORIA I

说明： A 与 I 做逻辑异或运算

操作数： $0 \leq I \leq 0xff$

操作： $(A) \text{ XOR } (I) \rightarrow (A)$

受影响的状态位： Z

说明： 将 A 寄存器的内容与立即数 I 进行逻辑异或运算，结果存入 A 寄存器。

SBCAI I

说明： A 带借位减 I

操作数： $0 \leq I \leq 0xff$

操作： $(A) - (I) - (\overline{C}) \rightarrow (A)$

受影响的状态位： Z, DC, C

说明： 按二进制补码方式，用 A 寄存器内容减去立即数 I。
 如果 C = 0，说明前次运算有借位，则需再减去 1；否则不需要再减。
 结果存入 A 寄存器。

MOVIA I

说明： 将 I 传送到 A

操作数： $0 \leq I \leq 0xff$

操作： $(I) \rightarrow (A)$

受影响的状态位： 无

说明： 将立即数 I 传送到 A 寄存器。

ANDIA I

说明： A 与 I 做逻辑与运算

操作数： $0 \leq I \leq 0xff$

操作： $(A) \text{ AND } (I) \rightarrow (A)$

受影响的状态位： Z

说明： 将 A 寄存器的内容与立即数 I 进行逻辑与运算，结果存入 A 寄存器。

RETIA I

说明： 返回并将立即数传送到 A

操作数： $0 \leq I \leq 0xff$

操作： $(I) \rightarrow (A)$
 $\text{Stack} \rightarrow \text{PC}$

受影响的状态位： 无

说明： 将立即数 I 传送到 A 寄存器。
 将栈顶的返回地址值传送到程序计数器。

ORIA I

说明： A 与 I 做逻辑或运算

操作数： $0 \leq I \leq 0xff$

操作： $(A) \text{ OR } (I) \rightarrow (A)$

受影响的状态位： Z

说明： 将 A 寄存器的内容与立即数 I 进行逻辑或运算，结果存入 A 寄存器。

JCMPAI I

说明： 比较 A 与 I，相等则跳过

操作数： $0 \leq I \leq 0xff$

操作： $(A) \text{ Vs } (I)$
 如果 $A = I$ 则跳过

受影响的状态位： Z, C

说明： 将 A 寄存器的内容与立即数 I 比较。
 如果 $A = I$ ，则代为执行 NOP 指令
 如果 $A \neq I$ ，则执行下一条指令

JNCMPAI I

说明： 比较 A 与 I, 不等则跳过

操作数： $0 \leq I \leq 0xff$

操作： (A) Vs (I)
 如果 $A \neq I$ 则跳过

受影响的状态位： Z, C

说明： 将 A 寄存器的内容与立即数 I 比较。
 如果 $A = I$, 则执行下一条指令
 如果 $A \neq I$, 则代为执行 NOP 指令

RLA

说明： A 带进位循环左移

操作数： 无

操作： $(A[7]) \rightarrow (C)$
 $(\{A[6:0], (C)\}) \rightarrow (A)$

受影响的状态位： C

说明： 将 A 寄存器的内容连同进位值一起
 循环左移 1 位, 结果存入 A 寄存器.

RLANC

说明： A 不带进位左移

操作数： 无

操作： $(\{A[6:0], 0\}) \rightarrow (A)$

受影响的状态位： 无

说明： 将寄存器 A 的内容左移 1 位, 最低位
 补 0. 结果存入 A 寄存器.

RRA

说明： A 带进位循环右移

操作数： 无

操作： $(A[0]) \rightarrow (C)$
 $(\{(C), A[7:1]\}) \rightarrow (A)$

受影响的状态位： C

说明： 将 A 寄存器的内容连同进位值一起
 循环右移 1 位. 结果存入 A 寄存器.

RRANC

说明： A 不带进位右移

操作数： 无

操作： $(\{0, A[7:1]\}) \rightarrow (A)$

受影响的状态位： 无

说明： 将 A 寄存器的内容右移 1 位, 最高位
 补 0. 结果存入 A 寄存器.

DECA

说明： A 递减 1

操作数： 无

操作： $(A) - 1 \rightarrow (A)$

受影响的状态位： Z

说明： 将 A 寄存器的内容递减 1.
 结果存入 A 寄存器.

DJZA

说明： A 递减 1, 为 0 则跳过

操作数： 无

操作： $(A) - 1 \rightarrow (A)$
 如果结果=0 则跳过

受影响的状态位： 无

说明： 将 A 寄存器的内容递减 1.
 结果存入 A 寄存器.
 如果结果不为 0, 则执行下一条指令
 如果结果为 0, 则代为执行 NOP 指令

INCA

说明： A 递增 1

操作数： 无

操作： $(A) + 1 \rightarrow (A)$

受影响的状态位： Z

说明： 将 A 寄存器的内容递增 1.
 结果存入 A 寄存器.

JZA

说明： A 递增 1, 为 0 则跳过

操作数： 无

操作： $(A) + 1 \rightarrow (A)$
如果结果=0 则跳过

受影响的状态位： 无

说明： 将 A 寄存器的内容递增 1.
结果存入 A 寄存器。
如果结果不为 0, 则执行下一条指令
如果结果为 0, 则代为执行 NOP 指令

RETIE

说明： 从中断返回

操作数： 无

操作： $1 \rightarrow GIE$
 $Stack \rightarrow PC$

受影响的状态位： 无

说明： 全局中断允许位 GIE 置 1。
将栈顶的返回地址值传送到
程序计数器。

RETURN

说明： 从子程序返回

操作数： 无

操作： $Stack \rightarrow PC$

受影响的状态位： 无

说明： 从子程序返回, 执行出栈动作, 将栈顶
的返回地址值传送到程序计数器。

NOP

说明： 空操作

操作数： 无

操作： 空操作

受影响的状态位： 无

说明： 不执行任何操作。

RDT

说明： 读表

操作数： 无

操作： $ROM[{\{fsr1,fsr0\}}]高 8 位 \rightarrow HBUF$
 $ROM[{\{fsr1,fsr0\}}]低 8 位 \rightarrow A$

受影响的状态位： 无

说明： 进行读表操作, 将 {fsr1,fsr0} 所指向的
ROM Code 高 8 位传送到寄存器 HBUF,
低 8 位传送到 A 寄存器。

DAA

说明： 加法后十进制调整

操作数： 无

操作： (A)加法十进制调整 $\rightarrow (A)$

受影响的状态位： DC, C

说明： 将 A 寄存器的内容进行加法十进制
调整, 结果存入 A 寄存器。

DSA

说明： 减法后十进制调整

操作数： 无

操作： (A)减法十进制调整 $\rightarrow (A)$

受影响的状态位： DC, C

说明： 将 A 寄存器的内容进行减法十进制
调整, 结果存入 A 寄存器。

PUSH

说明： 进栈

操作数： 无

操作： $PC \rightarrow Stack$
保存 A, STATUS

受影响的状态位： 无

说明： 进栈指令, 程序计数器值存入栈顶,
保存 A 寄存器和 STATUS 寄存器值。

POP

说明： 出栈

操作数： 无

操作： Stack → PC
恢复 A, STATUS

受影响的状态位： Z, DC, C

说明： 出栈指令，将栈顶的返回地址值传送到程序计数器，恢复原保存的 A 寄存器和 STATUS 寄存器值。

GOTO I

说明： 无条件跳转

操作数： $0 \leq I \leq 0x3fff$

操作： I → PC

受影响的状态位： 无

说明： 无条件跳转，将立即数 I 作为地址值传送到程序计数器

CLRWDT

说明： 清零看门狗计数器

操作数： 无

操作： 0x00 → WDT
1 → TO
1 → PD

受影响的状态位： TO, PD

说明： 将看门狗计数器清零，状态位 TO 和 PD 置 1。

CALL I

说明： 调用子程序

操作数： $0 \leq I \leq 0x3fff$

操作： PC → Stack
I → PC

受影响的状态位： 无

说明： 调用子程序，将程序计数器存入栈顶，将立即数 I 作为地址值传送到程序计数器