CH55X 汇编指令周期表

CH55X汇编指令概述：
非跳转指令的指令周期数与指令字节数相同；
跳转指令含MOVC/RET/CALL通常比字节数多若干个周期；
MOVC指令多4或5个周期（下条指令地址为奇数时多5个）；
RET/RETI指令多3或4个周期（返回地址为奇数时多4个）；
其余指令多2或3个周期（目标地址是奇数时多3个）；
条件跳转指令如果未发生跳转则周期数与指令字节数相同；
以上周期是指当前 CH55X 的系统主频的倒数。

CH559 指令周期表					
类别	指令格式	功能描述	字节	机器周期 1/12 倍数	比普通 51 快倍数
传送类指令(29条)	MOV A, Rn	（Rn）→（A） Rn 中的内容送到累加器 A 中，Rn=R0-R7	1	1	12
	MOV A,data	（data）→（A） 直接单元地址中的内容送到累加器 A	2	2	6
	MOV A,@Ri	（（Ri））→（A） Ri 内容指向的地址单元中的内容送到累加器 A，Ri=R0 或 R1	1	1	12
	MOV A,#data	#data→（A） 立即数送到累加器 A 中。	2	2	6
	MOV Rn,A	（A）→（Rn） 累加器 A 中的内容送到寄存器 Rn 中	1	1	12
	MOV Rn,data	（data）→（Rn）直接寻址单元中的内容送寄存器	2	2	12
	MOV Rn,#data	#data→（Rn）立即数直接送到寄存器 Rn 中。	2	2	12
	MOV data,A	(A)→（data）累加器 A 中的内容送直接寻址单元。	2	2	6
	MOV data,Rn	（Rn）→（data）寄存器中的内容送直接寻址单元	2	2	12
	MOV data2,data1	（data1）→（data2）直接寻址单元中的内容 1 送直接寻址单元 2 。	3	3	8
	MOV data,@Ri	（Rn）→（data）寄存器中的内容送直接寻址单元	2	2	12
	MOV data,#data	#data→（data）立即数送直接寻址单元 。	3	3	8
	MOV @Ri,A	（A）→（（Ri）） 累加器 A 中的内容送到以 Ri 中的内容为地址的 RAM 单元	1	1	12
	MOV @Ri,data	（data）→（（Ri））直接寻址单元内容送到以 Ri 中的内容为地址的 RAM 单元	2	2	12
	MOV @Ri,#data	#data→（（Ri））立即数送到以 Ri 中的内容为地址的 RAM 单元	2	2	6

	MOV DPTR,#data16	#dataH→（DPH），#dataL→（DPL）16 位常数的高 8 位送到 DPH，低 8 位送到 DPL	3	3	8
	MOVX A,@DPTR	（（DPTR））→（A） 数据指针指向片外 RAM 地址中的内容送到累加器 A 中	1	1	24
	MOVX @DPTR,A	（A）→（（DPTR）） 累加器中的内容送到数据指针指向片外 RAM 地址中	1	1	24
	MOVX A,@Ri	（（Ri））→（A） 寄存器 Ri 指向片外 RAM 地址中的内容送到累加器 A 中。	1	1	24
	MOVX @Ri,A	（A）→（（Ri）） 累加器中的内容送到寄存器 Ri 指向片外 RAM 地址中。	1	1	24
	MOVC A,@A+DPTR	（（A））+（DPTR）→（A） 表格地址单元中的内容送到累加器 A 中	1	6	2
	MOVC A,@A+PC	（（PC））+1→（A），（（A））+（PC）→（A） 表格地址单元中的内容送到累加器 A 中	1	6	2
	XCH A,Rn	（A）←→（Rn）累加器与工作寄存器 Rn 中的内容互换	1	1	12
	XCH A,data	（data）→（A）累加器 A 的内容与直接寻址单元的内容交换	2	2	6
	XCH A,@Ri	（A）←→（（Ri））累加器与工作寄存器 Ri 所指的存储单元中的内容互换	1	1	12
	XCHD A,@Ri	（$A_{3\text{-}0}$）←→（（Ri）$_{3\text{-}0}$）累加器与工作寄存器 Ri 所指的存储单元中的内容低半字节互换	1	1	12
	SWAP A	（$A_{3\text{-}0}$）←→（$A_{7\text{-}4}$）累加器中的内容高低半字节互换。	1	1	12
	PUSH data	（SP）+1→（SP），（data）→（SP） 堆栈指针首先加 1，直接寻址单元中的数据送到堆栈指针 SP 所指的单元中	2	2	12
	POP data	（SP）→（data）（SP）-1→（SP）， 堆栈指针 SP 所指的单元数据送到直接寻址单元中，堆栈指针 SP 再进行减 1 操作	2	2	12
算术运算类指令(24条)	ADD A,Rn	（A）+（Rn）→（A） 累加器 A 中的内容与工作寄存器 Rn 中的内容相加，结果存在 A 中。	1	1	12
	ADD A,data	（A）+（data）→（A） 累加器 A 中的内容与直接地址单元中的内容相加，结果存在 A 中。	2	2	6
	ADD A,@Ri	（A）+（（Ri））→（A） 累加器 A 中的内容与工作寄存器 Ri 所指向地址单元中的内容相加，结果存在 A 中。	1	1	12
	ADD A,#data	（A）+#data→（A） 累加器 A 中的内容与立即数#data 相加，结果存在 A 中。	2	2	6
	ADDC A,Rn	（A）+Rn+（C）→（A） 累加器 A 中的内容与工作寄存器 Rn 中的内容、连同进位位相加，结果存在 A 中。	1	1	12
	ADDC A,data	（A）+（data）+（C）→（A） 累加器 A 中的内容与直接地址单元的内容连同进位位相加，	2	2	6

		结果存在 A 中。			
	ADDC A,@Ri	（A）+（（Ri））+（C）→（A） 累加器 A 中的内容与工作寄存器 Ri 指向地址单元中的内容、连同进位位相加，结果存在 A 中。	1	1	12
	ADDC A,#data	（A）+#data +（C）→（A） 累加器 A 中的内容与立即数连同进位位相加，结果存在 A 中。	2	2	6
	INC A	（A）+1→（A） 累加器 A 中的内容加 1，结果存在 A 中。	1	1	12
	INC Rn	（Rn）+1→（Rn）寄存器 Rn 的内容加 1，结果送回原地址单元中。	1	1	12
	INC data	（data）+1→（data） 直接地址单元中的内容加 1，结果送回原地址单元中。	2	2	6
	INC @Ri	（（Ri））+1→（（Ri）） 寄存器的内容指向的地址单元中的内容加 1，结果送回原地址单元中。	1	1	12
	INC DPTR	（DPTR）+1→（DPTR）数据指针的内容加 1，结果送回数据指针中。	1	1	12
	SUBB A,Rn	（A）-（Rn）-（C）→（A） 累加器 A 中的内容与工作寄存器中的内容、连同借位位相减，结果存在 A 中。	1	1	12
	SUBB A,data	（A）-（data）-（C）→（A） 累加器 A 中的内容与直接地址单元中的内容、连同借位位相减，结果存在 A 中。	2	2	6
	SUBB A,@Ri	（A）-（（Ri））-（C）→（A） 累加器 A 中的内容与工作寄存器 Ri 指向的地址单元中的内容、连同借位位相减，结果存在 A 中。	1	1	12
	SUBB A,#data	（A）-#data -（C）→（A） 累加器 A 中的内容与立即数、连同借位位相减，结果存在 A 中。	2	2	6
	DEC A	（A）-1→（A）累加器 A 中的内容减 1，结果送回累加器 A 中。	1	1	12
	DEC Rn	（Rn）-1→（Rn）寄存器 Rn 中的内容减 1，结果送回寄存器 Rn 中。	1	1	12
	DEC data	（data）-1→（data）直接地址单元中的内容减 1，结果送回直接地址单元中。	2	2	6
	DEC @Ri	（（Ri））-1→（（Ri））寄存器 Ri 指向的地址单元中的内容减 1，结果送回原地址单元中。	1	1	12
	MUL AB	（A）×（B）→（A）和（B） 累加器 A 中的内容与寄存器 B 中的内容相乘，结果存在 A、B 中。	1	1	48
	DIV AB	（A）÷（B）→（A）和（B） 累加器 A 中的内容除以寄存器 B 中的内容，所得到的商存在累加器 A，而余数存在寄存器 B 中。	1	1	48
	DA A	在进行 BCD 码运算时，这条指令总是跟在 ADD 或 ADDC 指令之后，其功能是将执行加法运算后存于累加器 A 中的结果进行调整和修正。	1	1	12
逻	ANL A,Rn	累加器 A 的内容和寄存器 Rn 中的内容执行与逻	1	1	12

辑操作指令(24条)		辑操作。结果存在累加器 A 中。			
	ANL A,direct	累加器 A 中的内容和直接地址单元中的内容执行与逻辑操作。结果存在寄存器 A 中。	2	2	6
	ANL A,@Ri	累加器 A 的内容和工作寄存器 Ri 指向的地址单元中的内容执行与逻辑操作。结果存在累加器 A 中。	1	1	12
	ANL A,#data	累加器 A 的内容和立即数执行与逻辑操作。结果存在累加器 A 中。	2	2	6
	ANL data,A	直接地址单元中的内容和累加器 A 的内容执行与逻辑操作。结果存在直接地址单元中。	2	2	6
	ANL data,#data	直接地址单元中的内容和立即数执行与逻辑操作。结果存在直接地址单元中。	3	3	8
	ORL A,Rn	累加器 A 的内容和寄存器 Rn 中的内容执行逻辑或操作。结果存在累加器 A 中。	1	1	12
	ORL A,data	累加器 A 中的内容和直接地址单元中的内容执行逻辑或操作。结果存在寄存器 A 中。	2	2	6
	ORL A,@Ri	累加器 A 的内容和工作寄存器 Ri 指向的地址单元中的内容执行逻辑或操作。结果存在累加器 A 中。	1	1	12
	ORL A,#data	累加器 A 的内容和立即数执行逻辑或操作。结果存在累加器 A 中。	2	2	6
	ORL data,A	直接地址单元中的内容和累加器 A 的内容执行逻辑或操作。结果存在直接地址单元中。	2	2	6
	ORL data,#data	直接地址单元中的内容和立即数执行逻辑或操作。结果存在直接地址单元中。	3	3	8
	XRL A,Rn	累加器 A 的内容和寄存器 Rn 中的内容执行逻辑异或操作。结果存在累加器 A 中。	1	1	12
	XRL A,data	累加器 A 中的内容和直接地址单元中的内容执行逻辑异或操作。结果存在寄存器 A 中。	2	2	6
	XRL A,@Ri	累加器 A 的内容和工作寄存器 Ri 指向的地址单元中的内容执行逻辑异或操作。结果存在累加器 A 中。	1	1	12
	XRL A,#data	累加器 A 的内容和立即数执行逻辑异或操作。结果存在累加器 A 中。	2	2	6
	XRL data,A	直接地址单元中的内容和累加器 A 的内容执行逻辑异或操作。结果存在直接地址单元中。	2	2	6
	XRL data,#data	直接地址单元中的内容和立即数执行逻辑异或操作。结果存在直接地址单元中。	3	3	8
	CLR A	0→（A），累加器中的内容清 0。	1	1	12
	CPL A	累加器中的内容按位取反。	1	1	12
	RL A	累加器 A 中的内容左移一位。	1	1	12
	RR A	累加器 A 中的内容右移一位。	1	1	12
	RLC A	累加器 A 中的内容连同进位位 CY 左移一位。	1	1	12
	RRC A	累加器 A 中的内容连同进位位 CY 右移一位。	1	1	12
控	LJMP	addr16→（PC），给程序计数器赋予新值（16	3	6	2

制转移指令(16条)	addr16	位地址）。			
	AJMP addr11	（PC）+2→（PC），addr11→（PC10-0）程序计数器赋予新值（11 位地址），（PC15-11）不改变。	2	5	4.8
	SJMP rel	（PC）+ 2 + rel→（PC）当前程序计数器先加上 2 再加上偏移量给程序计数器赋予新值。	2	5	4.8
	JMP @A+DPTR	（A）+（DPTR）→（PC），累加器所指向地址单元的值加上数据指针的值给程序计数器赋予新值。	1	5	4.8
	JZ rel	A=0,（PC）+ 2 + rel→（PC）,累加器中的内容为 0，则转移到偏移量所指向的地址，否则程序往下执行。	2	2	12
	JNZ rel	A≠0,（PC）+ 2 + rel→（PC）,累加器中的内容不为 0，则转移到偏移量所指向的地址，否则程序往下执行。	2	2	12
	CJNE A,#data,rel	A≠#data,（PC）+ 3 + rel→（PC）,累加器中的内容不等于立即数，则转移到偏移量所指向的地址，否则程序往下执行。	3	6	4
	CJNE Rn,#data,rel	A≠#data,（PC）+ 3 + rel→（PC）,工作寄存器 Rn 中的内容不等于立即数，则转移到偏移量所指向的地址，否则程序往下执行。	3	6	4
	CJNE @Ri,#data,rel	A≠#data,（PC）+ 3 + rel→（PC）,工作寄存器 Ri 指向地址单元中的内容不等于立即数，则转移到偏移量所指向的地址，否则程序往下执行。	3	6	4
	CJNE A,data,rel	A≠（data），（PC）+ 3 + rel→（PC）,累加器中的内容不等于直接地址单元的内容，则转移到偏移量所指向的地址，否则程序往下执行	3	6	4
	DJNZ Rn,rel	（Rn）-1→（Rn），（Rn)≠0,（PC）+ 2 + rel→（PC）工作寄存器 Rn 减 1 不等于 0，则转移到偏移量所指向的地址，否则程序往下执行	2	4	6
	DJNZ data,rel	（Rn）-1→（Rn），（Rn)≠0,（PC）+ 2 + rel→（PC）直接地址单元中的内容减 1 不等于 0，则转移到偏移量所指向的地址，否则程序往下执行。	3	5	4.8
	LCALL addr16	长调用指令，可在 64kB 空间调用子程序。此时（PC）+ 3→（PC），（SP）+ 1→（SP），（PC_{7-0}）→（SP），（SP）+ 1→（SP），（PC_{15-8}）→（SP），addr16→（PC），即分别从堆栈中弹出调用子程序时压入的返回地址。	3	6	4
	ACALL addr11	绝对调用指令，可在 2kB 空间调用子程序，此时（PC）+ 2→（PC），（SP）+ 1→（SP），（PC_{7-0}）→（SP），（SP）+ 1→（SP），（PC_{15-8}）→（SP），addr11→（PC_{10-0}）	2	5	4.8
	RET	子程序返回指令。此时（SP）→（PC_{15-8}），（SP）	1	5	4.8

		－1→（SP），（SP）→（PC_{7-0}），（SP）－1→（SP）			
	RETI	中断返回指令，除具有 RET 功能外，还具有恢复中断逻辑的功能，需注意的是，RETI 指令不能用 RET 代替。	1		
位操作指令(18条)	MOV C,bit	bit→CY，某位数据送 CY。	2	2	6
	MOV bit,C	CY→bit，CY 数据送某位。	2	2	12
	CLR C	0→CY,清 CY	1	1	12
	CLR bit	0→bit,清某一位。	2	2	6
	SETB C	1→CY,置位 CY	1	1	12
	SETB bit	1→bit,置位某一位	2	2	6
	CPL C	CY 取反	1	1	12
	CPL bit	Bit 取反	2	2	6
	ANL C,bit	(CY)∧(bit)→CY	2	2	12
	ANL C,/bit	（CY）∧(bit 取反）→CY	2	2	12
	ORL C,bit	（CY）∨(bit）→CY	2	2	12
	ORL C,/bit	（CY）∨(bit 取反)CY	2	2	12
	JC rel	(CY)=1 转移，（PC）+2+rel→PC，否则程序往下执行，（PC）+2→PC。	2	2	12
	JNC rel	(CY)=0 转移，（PC）+2+rel→PC，否则程序往下执行，（PC）+2→PC。	2	4	6
	JB bit,rel	位状态为 1 转移，（PC）+3+rel→PC，否则程序往下执行，（PC）+3→PC。	3	6	4
	JNB bit,rel	位状态为 0 转移，（PC）+3+rel→PC，否则程序往下执行，（PC）+3→PC。	3	5	4.8
	JBC bit,rel	位状态为 1 转移，并使该位清“0”，（PC）+3+rel→PC，否则程序往下执行，（PC）+3→PC。	3	6	4
	NOP	这条指令除了使 PC 加 1，消耗一个机器周期外，没有执行任何操作。可用于短时间的延时。	1	1	12

注：标记为深色，代表指令字节数不等于执行该指令需要的机器周期；