					Cortex-M0 指令系统	
					数据传送	
	☆		N/Z	MOV{S}	Rd,Rm	数据传送
1	2	1		MOVS	Rd,#<imm8>	8位立即数 00100 Rd imm8
2	2	1		MOVS	Rd,Rm	Rd=Rm=loreg 00000000 00 Rm Rd
3	2	1		MOV	Rd,Rm	Rd=loreg,Rm=any 01000110 D Rm4 Rd
4	?	3		MOV	PC,Rm	Any to PC (?)
					数据加载	
	☆		不影响标志	LDRB	Rd,[Rn,<Rm|#imm>]	无符号字节,数据加载
	☆		不影响标志	LDRSB	Rd,[Rn,<Rm|#imm>]	有符号字节,数据加载
	☆		不影响标志	LDRH	Rd,[Rn,<Rm|#imm>]	无符号半字,数据加载
	☆		不影响标志	LDRSH	Rd,[Rn,<Rm|#imm>]	有符号半字,数据加载
	☆		不影响标志	LDR	Rd,[Rn,<Rm|#imm>]	字,数据加载
	☆		不影响标志	LDR	Rd,label	相对,数据加载
	☆		不影响标志	LDM	Rn{!},loreglist	多寄存器,数据加载
1	2	2		LDRB	Rd,[Rn,#<+-imm5>]	无符号字节,立即数偏移01111 imm5 Rn Rd
2	2	2		LDRB	Rd,[Rn,Rm]	无符号字节,寄存器偏移0101110 Rm Rn Rd
3	2	2		LDRSB	Rd,[Rn,Rm]	有符号字节,寄存器偏移0101011 Rm Rn Rd
4	2	2		LDRH	Rd,[Rn,#<+-imm5>]	无符号字节,立即数偏移10001 imm5 Rn Rd
5	2	2		LDRH	Rd,[Rn,Rm]	无符号字节,寄存器偏移0101101 Rm Rn Rd
6	2	2		LDRSH	Rd,[Rn,Rm]	有符号字节,寄存器偏移0101111 Rm Rn Rd
7	2	2		LDR	Rd,[Rn,#<+-imm5>]	字,立即数偏移 01101 imm5 Rn Rd
8	2	2		LDR	Rd,[Rn,Rm]	字,寄存器偏移 0101100 Rm Rn Rd
9	2	2		LDR	Rd,<label>	PC相对 01001 Rd imm8
10	2	2		LDR	Rd,[SP,#<imm8>]	SP相对 10011 Rd imm8
11	2	1+N		LDM	Rn!,{<loreglist>}	多寄存器加载 11001 Rn register_list8
12	2	1+N		LDM	Rn,{<loreglist>}	多寄存器加载 11001 Rn register_list8
					数据存储	
	☆		不影响标志	STRB	Rd,[Rn,<Rm|#imm>]	字节,数据存储
	☆		不影响标志	STRH	Rd,[Rn,<Rm|#imm>]	半字,数据存储
	☆		不影响标志	STR	Rd,[Rn,<Rm|#imm>]	字,数据存储
	☆		不影响标志	STM	Rn!,reglist	多寄存器,数据存储
1	2	2		STRB	Rd,[Rn,#<+-imm5>]	字节,立即数偏移 01110 imm5 Rn Rd
2	2	2		STRB	Rd,[Rn,Rm]	字节,寄存器偏移 0101010 Rm Rn Rd
3	2	2		STRH	Rd,[Rn,#<+-imm5>]	半字,立即数偏移 10000 imm5 Rn Rd
4	2	2		STRH	Rd,[Rn,Rm]	半字,寄存器偏移 0101001 Rm Rn Rd
5	2	2		STR	Rd,[Rn,#<+-imm5>]	字,立即数偏移 01100 imm5 Rn Rd
6	2	2		STR	Rd,[Rn,Rm]	字,寄存器偏移 0101 000 Rm Rn Rd
7	2	2		STR	Rd,[SP,#<imm8>]	字,立即数偏移 10010 Rd imm8
8	2	1+N		STM	Rn!,{<loreglist>}	多寄存器 11000 Rn register_list
					进出栈SP操作	
	☆		不影响标志	PUSH	loreglist	进栈
	☆		不影响标志	POP	loreglist	出栈
1	2	1+N		PUSH	{<loreglist>}	1011010M register_list
2	2	1+N		PUSH	{<loreglist>,LR}	1011010M register_list
3	2	1+N		POP	{<loreglist>}	1011110P register_list
4	2	1+N		POP	{<loreglist>,PC}	1011110P register_list
					逻辑操作	
	☆		N/Z	ANDS	{Rd,}Rn,Rm	逻辑与

	☆		N/Z	ORRS	{Rd,}Rn,Rm	逻辑或
	☆		N/Z	EORS	{Rd,}Rn,Rm	逻辑异或
	☆		N/Z	MVNS	Rd,Rm	逻辑非(数据取反传送)
	☆		N/Z	BICS	{Rd,}Rn,Rm	位清零
	☆		N/Z	TST	Rn,Rm	逻辑与测试(影响标志N/Z)
1	2	1		ANDS	Rdn,Rdn,Rm	逻辑与 01000000 00 Rm Rdn
2	2	1		ORRS	Rdn,Rdn,Rm	逻辑或 01000011 00 Rm Rdn
3	2	1		EORS	Rdn,Rdn,Rm	逻辑异或 01000000 01 Rm Rdn
4	2	1		BICS	Rdn,Rdn,Rm	位清零 01000011 10 Rm Rdn
5	2	1		MVNS	Rd,Rm	逻辑非(数据取反传送)01000011 11 Rm Rd
6	2	1		TST	Rn,Rm	逻辑与测试 01000010 00 Rm Rn
					移位操作	
	☆		N/Z/C	LSLS	{Rd,}Rn,<Rm|#imm>	逻辑左移
	☆		N/Z/C	LSRS	{Rd,}Rn,<Rm|#imm>	逻辑右移
	☆		N/Z/C	ASRS	{Rd,}Rn,<Rm|#imm>	算术右移
	☆		N/Z/C	RORS	{Rd,}Rn,Rm	循环右移
1	2	1		LSLS	Rd,Rm,#<imm5>	逻辑左移,立即数移位00000 imm5 Rm Rd
2	2	1		LSLS	Rdn,Rdn,Rm	逻辑左移,寄存器移位01000000 10 Rm Rdn
3	2	1		LSRS	Rd,Rm,#<imm5>	逻辑右移,立即数移位00001 imm5 Rm Rd
4	2	1		LSRS	Rdn,Rdn,Rm	逻辑右移,寄存器移位01000000 11 Rm Rdn
5	2	1		ASRS	Rd,Rm,#<imm5>	算术右移,立即数移位00010 imm5 Rm Rd
6	2	1		ASRS	Rdn,Rdn,Rm	算术右移,寄存器移位01000001 00 Rm Rdn
7	2	1		RORS	Rdn,Rdn,Rm	循环右移,寄存器移位01000001 11 Rm Rdn
					比较操作	
	☆		N/Z/C/V	CMP	Rn,<Rm|#imm>	比较
	☆		N/Z/C/V	CMN	Rn,Rm	负数比较 01000101 N Rm4 Rn
1	2	1		CMP	Rn,Rm	Rn=loreg,Rm=any 01000010 10 Rm Rn
2	2	1		CMP	Rn,#<imm8>	00101 Rn imm8
3	2	1		CMN	Rn,Rm	01000010 11Rm Rn
					加法运算	
	☆		N/Z/C/V	ADD{S}	{Rd,}Rn,<Rm|#imm>	加
	☆		N/Z/C/V	ADCS	{Rd,}Rn,Rm	带借位加
	☆		不影响标志	ADR	Rd,label	小范围地址读取
1	2	1		ADD	Rdn,Rdn,Rm	Rdn=loreg,Rm=any 01000100 DN Rm4 Rdn
2	2	3		ADD	PC,PC,Rm	Any to PC(结果不可预测?)
?	2	1		ADD	Rdm,SP,Rdm	01000100 DM 1101 Rdm
?	2	1		ADD	SP,SP,Rm	01000100 1 Rm4 101(ADD SP SP SP ?)
3	2	1		ADD	SP,SP,#<imm7>	立即数 to SP 10110000 0 imm7
4	2	1		ADD	Rd,SP,#<imm8>	地址 from SP 10101 Rd imm8
5	2	1		ADDS	Rd,Rn,#<imm3>	3位立即数 0001110 imm3 Rn Rd
6	2	1		ADDS	Rd,Rn,Rm	All loreg 0001100 Rm Rn Rd
7	2	1		ADDS	Rdn,Rdn,#<imm8>	8位立即数 00110 Rdn imm8
8	2	1		ADCS	Rdn,Rdn,Rm	带借位加 01000001 01Rm Rdn
9	2	1	不影响标志	ADR	Rd,<label>	地址 from PC 10100 Rd imm8
					减法运算	
	☆		N/Z/C/V	SUB{S}	{Rd,}Rn,<Rm|#imm>	
	☆		N/Z/C/V	SBCS	{Rd,}Rn,Rm	
	☆		N/Z/C/V	RSBS	{Rd,}Rn,#0	
1	2	1		SUB	SP,SP,#<imm7>	立即数 from SP 10110000 1 imm7
2	2	1		SUBS	Rd,Rn,Rm	Rx=loreg 0001101 Rm Rn Rd

3	2	1		SUBS	Rd,Rn,#<imm3>	3位立即数 0001111 imm3 Rn Rd
4	2	1		SUBS	Rdn,Rdn,#<imm8>	8位立即数 00111 Rdn imm8
5	2	1		SBCS	Rdn,Rdn,Rm	带借位减 01000001 10 Rm Rdn
6	2	1		RSBS	Rd,Rn,#0	求补(反向减法) 01000010 01 Rn Rd
					乘法运算	
	☆		N/Z	MULS	Rd,Rn,Rm	乘法,32位结果
1	2	1/32		MULS	Rdm,Rn,Rdm	保存乘法结果低 32位01000011 01 Rn Rdm
					跳转/调用/返回操作	
	☆		不影响标志	B{cc}	label	条件跳转
	☆		不影响标志	BX	Rm	带状态切换跳转
	☆		不影响标志	BL	label	带链接跳转
	☆		不影响标志	BLX	Rm	带链接和状态切换跳转
1	2	1/3	-256	BEQ	<label> +254	Z=1,相等 1101 cond imm8
2	2	1/3		BNE	<label>	Z=0,不等
3	2	1/3		BMI	<label>	N=1,小于0,Minus
4	2	1/3		BPL	<label>	N=0,大于或等于0,Plus
5	2	1/3		BVS	<label>	V=1,溢出
6	2	1/3		BVC	<label>	V=0,没有溢出
7	2	1/3		BCS	BHS <label>	C=1,无符号数大于或等于
8	2	1/3		BCC	BLO <label>	C=0,无符号数小于
9	2	1/3		BHI	<label>	C=1且Z=0,无符号数大于
10	2	1/3		BLS	<label>	C=0或Z=1,无符号数小于或等于
11	2	1/3		BGE	<label>	(N.XOR.V)=0,有符号数大于或等于
12	2	1/3		BLT	<label>	(N.XOR.V)=1,有符号数小于
13	2	1/3		BGT	<label>	(Z.OR.(N.XOR.V))=0,有符号数大于
14	2	1/3	-256	BLE	<label> +254	(Z.OR.(N.XOR.V))=1,有符号数小于或等于
15	2	3	-2048	B	<label> +2046	跳转 11100 imm11
16	2	3		BX	Rm	带状态切换跳转 01000111 0Rm4 000
17	4	4	-16777216	BL	<label> +16777214	带链接跳转 11110S imm10 11J11J2 imm11
18	2	3		BLX	Rm	带链接和状态切换跳转01000111 1Rm4 000
					其他操作	
	☆		不影响标志	UXTB	Rd,Rm	无符号字节扩展
	☆		不影响标志	UXTH	Rd,Rm	无符号半字扩展
	☆		不影响标志	SXTB	Rd,Rm	有符号字节扩展
	☆		不影响标志	SXTH	Rd,Rm	有符号半字扩展
1	2	1		UXTB	Rd,Rm	无符号字节扩展到字 10110010 11 Rm Rd
2	2	1		UXTH	Rd,Rm	无符号半字扩展到字 10110010 10 Rm Rd
3	2	1		SXTB	Rd,Rm	有符号字节扩展到字 10110010 01 Rm Rd
4	2	1		SXTH	Rd,Rm	有符号半字扩展到字 10110010 00 Rm Rd
	☆		不影响标志	REV	Rd,Rm	loreg小端和大端模式字数据互转
	☆		不影响标志	REV16	Rd,Rm	loreg两个小端和大端模式半字数据互转
	☆		不影响标志	REVSH	Rd,Rm	loreg小端和大端模式数据互转,符号半字到字
1	2	1	BA 00 Rm Rd	REV	Rd,Rm	小端和大端模式字数据互转
						uint32_t_REV(uint32_t int value)
2	2	1	BA 01 Rm Rd	REV16	Rd,Rm	两个小端和大端模式半字数据互转
						uint32_t_REV16(uint32_t int value)
3	2	1	BA 11 Rm Rd	REVSH	Rd,Rm	小端和大端模式数据互转,符号半字到字
						uint32_t_REVSH(uint32_t int value)
	☆		不影响标志	SVC	#imm	超级用户调用(状态改变)

	☆		不影响标志	CPSID	i	中断关闭
	☆		不影响标志	CPSIE	i	中断使能
	☆		不影响标志	MRS	Rd,spec_reg	读,特殊状态寄存器-->通用Rd
	☆		N/Z/C/V	MSR	spec_reg,Rn	写,通用Rn-->特殊状态寄存器
	☆		不影响标志	BKPT	#imm	调试断点
1	2	?		SVC	#<imm8>	超级用户调用(状态改变) 11011111 imm8
2	2	1		CPSID	i	中断关闭 10110110 011 imm1 00I0
						void_disable_irq(void)
3	2	1		CPSIE	i	中断使能 10110110 011 imm1 00I0
						void_enable_irq(void)
4	4	4		MRS	Rd,spec_reg	读,特殊状态寄存器-->通用Rd,Rd=any
						uint32_t_get_PRIMASK(void)
						uint32_t_get_CONTROL(void)
						uint32_t_get_MSP(void)
						uint32_t_get_PSP(void)
5	4	4	N/Z/C/V	MSR	spec_reg,Rn	写,通用Rn-->特殊状态寄存器,Rn=any
						void_set_PRIMASK(uint32_t value)
						void_set_CONTROL(uint32_t value)
						void_set_MSP(uint32_t TopOfMainStack)
						void_set_PSP(uint32_t TopOfProcStack)
6	2	?		BKPT	#<imm8>	调试断点 10111110 imm8
	☆		不影响标志	SEV		发送事件
	☆		不影响标志	WFE		等待事件
	☆		不影响标志	WFI		等待中断
	☆		不影响标志	YIELD		操作同NOP
	☆		不影响标志	NOP		
	☆		不影响标志	ISB		屏障,流水线刷新,此指令后所有指令重新取
	☆		不影响标志	DMB		屏障,确保DMB之前数据内存访问完成
	☆		不影响标志	DSB		屏障,此指令后所有指令不执行直到它执行完
1	2	1		SEV	void_SEV(void)	发送事件 10111111 01000000
2	2	2		WFE	void_WFE(void)	等待事件 10111111 00100000
3	2	2		WFI	void_WFI(void)	等待中断 10111111 00110000
4	2	1		YIELD		操作同NOP
5	2	1		NOP	void_NOP(void)	10111111 00000000
6	4	4	指令同步屏蔽	ISB	void_ISB(void)	流水线刷新,此指令后所有指令重新取
7	4	4	数据内存屏蔽	DMB	void_DMB(void)	确保DMB之前数据内存访问完成
8	4	4	数据同步屏蔽	DSB	void_DSB(void)	此指令后所有指令不执行直到它执行完
					Cortex-M0 指令系统	