

			Manufact.	Intel	various	STM	PHILIPS	DALLAS/M AXIM	DALLAS/M AXIM	SILABS	AD	GOAL	GOAL	ATMEL	WINBOND	TI
			Type	8051	RD2 (incl. P89C668)	uPSD3354	P89LPC935	DS80C320	DS89C450	C8051F120	ADuC841	VERSA1	VERSA MIX VMX1020	AT89LP405 2	W77E516	MSC1211Y5
			max.freq.	12MHz	20MHz	40MHz	12MHz	33MHz	33MHz	100MHz	20MHz	16MHz	14MHz	20MHz	40MHz	30MHz
			SRAM	no	0.75kB (1kB- 7.75kB)	32kB (program may execute)	0.5kB	no	1kB	8kB	2kB (can contain stack)	1kB	1kB	no	1kB	1kB (program can execute)
			FLASH	4kB	64kB (+8kB)	256kB+32kB	8kB	no	64kB	128kB	62kB	64kB(+2kB OTP)	56kB	4kB	64kB+4kB	32kB (can be mapped data)
			EEPROM	no	(2kB)	no	0.5kB	no	no	no	4kB	no		no	64B	no
			bus	UART	UART, (I2C, SPI)	2xUART (IrDA), SPI, I2C	UART, I2C, SPI	2xUART	2xUART	2xUART,I2C, SPI	UART, I2C or SPI	2xUART, SPI, RS485	2xUART, SPI, I2C, RS485	UART, SPI	2xUART	2xUART, SPI, I2C
			other		PCA, WDT	16 macrocell PLD, freely mappable FLASH/RAM via PLD, JTAG onchip debug, PCA, WDT, BOD, 8MUX 10bit 100ksps ADC, push- pull output (8x)	onchip RC oscillator, 2x 4MUX 8bit ADC, push- pull outputs, 2x comparator, BOD, WDT, RTC, capture/com pare unit	BOD, WDT	BOD, WDT	onchip oscillator + PLL, onchip JTAG debugger, 8MUX 12bit 100ksps ADC, 8MUX 8bit 500ksps ADC, 2x 12bit DAC, MAC, 5x 16bit timers, WDT, BOD, onchip temperature sensor, PCA, push-pull outputs, partially exchangable	8MUX 420kSPS 12bit ADC, on chop reference, 2x12bit DAC, PWM, on chip temperature monitor, WD, power supply monitor	4MUX 12bit ADC BOD, MAC	onchip debugger (UART), 3xcompare/c apture, onchip voltage reference, 4x16-bit PWM, 4MUX 12bit 10kSps ADC, programmab le current source, OPAMP, 2x digipot, 1x switch, BOD, WDT	WDT,BOD,P O R,Push- pull capable IO, comparator, PWM	WDT,WaitSt ate for XRAM	8xMUX 24- bit ADC + PGA, 4x16- bit DAC, onchip reference, WaitState, BOD, WDT, LowVoltage detect, 16- bit PWM, 2 HW breakpoints, boot+debug ROM
			package	DIL40, PLCC44	DIL40, PLCC44 (TQFP44)	TQPF52, TQFP80	TSSOP28, PLCC28, HVQFN28	DIP40, PLCC44, TQFP44	DIP40, PLCC44, TQFP44	TQFP100	PQFP52, CSP56	PQFP44	QFP64	DIL20,SOIC 20, TSSOP20	DIL40,PLCC 44, QFP44	TQFP64
			supply volt.	5V	5V (3V)	5V+3.3V/3.3 V	3.3V	5V	5V	3.3V	3V/5V	5V	5V	2.4-5V	5V	2.7-5V
			flash programmi	parallel	parallel, UART ISP,	parallel, JTAG ISP,	parallel, SPI- like "ICP",	no	UART ISP, IAP	JTAG, IAP	parallel, UART ISP,	I2C (12V)	I2C (12V)	parallel, SPI	parallel, IAP	parallel, UART

SUBB A, @Ri	Subtract data memory from A with borrow	1	96-97h	1	1	1	1	1	2	2	2	1	2	2	1	1
SUBB A, #data	Subtract immediate from A with borrow	2	94h	1	1	1	1	2	2	2	2	2	2	2	2	2
INC A	Increment A	1	04h	1	1	1	1	1	1	1	1	1	1	1	1	1
INC Rn	Increment register	1	08-0Fh	1	1	1	1	1	1	1	1	1	2	1	1	1
INC direct	Increment direct byte	2	05h	1	1	1	1	2	2	2	2	2	3	2	2	2
INC @Ri	Increment data memory	1	06-07h	1	1	1	1	1	2	2	2	1	3	2	1	1
DEC A	Decrement A	1	14h	1	1	1	1	1	1	1	1	1	1	1	1	1
DEC Rn	Decrement register	1	18-1Fh	1	1	1	1	1	1	1	1	1	2	1	1	1
DEC direct	Decrement direct byte	2	15h	1	1	1	1	2	2	2	2	2	3	2	2	2
DEC @Ri	Decrement data memory	1	16-17h	1	1	1	1	1	2	2	2	1	3	2	1	1
INC DPTR	Increment data pointer	1	A3h	2	2	2	2	3	1	1	3	3	1	2	2	3
MUL AB	Multiply A by B	1	A4h	4	4	4	4	5	9	4	9	5	5	2	5	5
DIV AB	Divide A by B	1	84h	4	4	4	4	5	10	8	9	5	5	4	5	5
DA A	Decimal adjust A	1	D4h	1	1	1	1	1	2	1	2	1	1	1	1	1
Logical Instructions																
ANL A, Rn	AND register to A	1	58-5Fh	1	1	1	1	1	1	1	1	1	1	1	1	1
ANL A, direct	AND direct byte to A	2	55h	1	1	1	1	2	2	2	2	2	2	2	2	2
ANL A, @Ri	AND data memory to A	1	56-57h	1	1	1	1	1	2	2	2	1	2	2	1	1

ANL A, #data	AND immediate to A	2	54h	1	1	1	1	2	2	2	2	2	2	2	2	2
ANL direct, A	AND A to direct byte	2	52h	1	1	1	1	2	2	2	2	2	3	2	2	2
ANL direct, #data	AND immediate data to direct byte	3	53h	2	2	2	2	3	3	3	3	3	4	3	3	3
ORL A, Rn	OR register to A	1	48-4Fh	1	1	1	1	1	1	1	1	1	1	1	1	1
ORL A, direct	OR direct byte to A	2	45h	1	1	1	1	2	2	2	2	2	2	2	2	2
ORL A, @Ri	OR data memory to A	1	46-47h	1	1	1	1	1	2	2	2	1	2	2	1	1
ORL A, #data	OR immediate to A	2	44h	1	1	1	1	2	2	2	2	2	2	2	2	2
ORL direct, A	OR A to direct byte	2	42h	1	1	1	1	2	2	2	2	2	3	2	2	2
ORL direct, #data	OR immediate data to direct byte	3	43h	2	2	2	2	3	3	3	3	3	4	3	3	3
XRL A, Rn	Exclusive-OR register to A	1	68-6Fh	1	1	1	1	1	1	1	1	1	1	1	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	65h	1	1	1	1	2	2	2	2	2	2	2	2	2
XRL A, @Ri	Exclusive-OR data memory to A	1	66-67h	1	1	1	1	1	2	2	2	1	2	2	1	1
XRL A, #data	Exclusive-OR immediate to A	2	64h	1	1	1	1	2	2	2	2	2	2	2	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	62h	1	1	1	1	2	2	2	2	2	3	2	2	2
XRL direct, #data	Exclusive-OR immediate to	3	63h	2	2	2	2	3	3	3	3	3	4	3	3	3

	direct byte															
CLR A	Clear A	1	E4h	1	1	1	1	1	1	1	1	1	1	1	1	1
CPL A	Compliment A	1	F4h	1	1	1	1	1	1	1	1	1	1	1	1	1
RL A	Rotate A left	1	23h	1	1	1	1	1	1	1	1	1	1	1	1	1
RLC A	Rotate A left through carry	1	33h	1	1	1	1	1	1	1	1	1	1	1	1	1
RR A	Rotate A right	1	03h	1	1	1	1	1	1	1	1	1	1	1	1	1
RRC A	Rotate A right through carry	1	13h	1	1	1	1	1	1	1	1	1	1	1	1	1
SWAP A	Swap nibbles of A	1	C4h	1	1	1	1	1	1	1	1	1	1	1	1	1
Data Transfer Instructions																
MOV A, Rn	Move register to A	1	E8-EFh	1	1	1	1	1	1	1	1	1	1	1	1	1
MOV A, direct	Move direct byte to A	2	E5h	1	1	1	1	2	2	2	2	2	2	2	2	2
MOV A, @Ri	Move data memory to A	1	E6-E7h	1	1	1	1	1	2	2	2	1	2	2	1	1
MOV A, #data	Move immediate to A	2	74h	1	1	1	1	2	2	2	2	2	2	2	2	2
MOV Rn, A	Move A to register	1	F8-FFh	1	1	1	1	1	1	1	1	1	2	1	1	1
MOV Rn, direct	Move direct byte to register	2	A8-AFh	2	2	2	2	2	2	2	2	2	4	2	2	2
MOV Rn, #data	Move immediate to register	2	78-7Fh	1	1	1	1	2	2	2	2	2	2	2	2	2
MOV direct, A	Move A to direct byte	2	F5h	1	1	1	1	2	2	2	2	2	3	2	2	2
MOV direct, Rn	Move register to direct byte	2	88-8Fh	2	2	2	2	2	2	2	2	2	3	2	2	2
MOV direct, direct	Move direct byte to direct	3	85h	2	2	2	2	3	3	3	3	3	4	3	3	3

	byte															
MOV direct, @Ri	Move data memory to direct byte	2	86-87h	2	2	2	2	2	2	2	2	2	4	2	2	2
MOV direct, #data	Move immediate to direct byte	3	75h	2	2	2	2	3	3	3	3	3	3	3	3	3
MOV @Ri, A	Move A to data memory	1	F6-F7h	1	1	1	1	1	1	2	2	1	3	1	1	1
MOV @Ri, direct	Move direct byte to data memory	2	A6-A7h	2	2	2	2	2	2	2	2	2	5	2	2	2
MOV @Ri, #data	Move immediate to data memory	2	76-77h	1	1	1	1	2	2	2	2	2	3	2	2	2
MOV DPTR, #data16	Move immediate to data pointer	3	90h	2	2	2	2	3	3	3	3	3	3	3	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	93h	2	2	2	2	3	3	3	4	3	3	3	2	3
MOVC A, @A+PC	Move code byte relative PC to A	1	83h	2	2	2	2	3	3	3	4	3	3	3	2	3
MOVX A, @Ri	Move external data (A8) to A	1	E2-E3h	2	2	2	2	2-9	2	3	4	3	3-10	-	2-9	2-9
MOVX A, @DPTR	Move external data (A 16) to A	1	E0h	2	2	2	2	2-9	2	3	4	3	3-10	-	2-9	2-9
MOVX @Ri, A	Move A to external data (A8)	1	F2-F3h	2	2	2	2	2-9	2	3	4	3	4-11	-	2-9	2-9
MOVX @DPTR, A	Move A to external data (A 16)	1	F0h	2	2	2	2	2-9	2	3	4	3	4-11	-	2-9	2-9
PUSH direct	Push direct byte onto stack	2	C0h	2	2	2	2	2	2	2	2	2	4	2	2	2

[illegible]

Instructions																
ACALL addr 11	Absolute call to subroutine	2	11-F1h	2	2	2	2	3	2	3	3	3	6	3	3	3
LCALL addr 16	Long call to subroutine	3	12h	2	2	2	2	4	3	4	4	4	6	4	4	4
RET	Return from subroutine	1	22h	2	2	2	2	4	3	5	4	4	4	4	2	4
RETI	Return from interrupt	1	32h	2	2	2	2	4	3	5	4	4	4	4	2	4
AJMP addr 11	Absolute jump unconditional	2	01-E1h	2	2	2	2	3	2	3	3	3	3	3	3	3
LJMP addr 16	Long jump unconditional	3	02h	2	2	2	2	4	3	4	4	4	4	4	4	4
SJMP rel	Short jump (relative address)	2	80h	2	2	2	2	3	3	3	3	3	3	3	3	3
JC rel	Jump on carry = 1	2	40h	2	2	2	2	3	3	2/3	3	3	3	3	3	3
JNC rel	Jump on carry = 0	2	50h	2	2	2	2	3	3	2/3	3	3	3	3	3	3
JB bit, rel	Jump on direct bit = 1	3	20h	2	2	2	2	4	4	3/4	4	4	4	4	4	4
JNB bit, rel	Jump on direct bit = 0	3	30h	2	2	2	2	4	4	3/4	4	4	4	4	4	4
JBC bit, rel	Jump on direct bit = 1 and clear	3	10h	2	2	2	2	4	4	3/4	4	4	4	4	4	4
JMP @A+DPTR	Jump indirect relative DPTR	1	73h	2	2	2	2	3	3	3	3	3	2	3	2	3
JZ rel	Jump on accumulator = 0	2	60h	2	2	2	2	3	3	2/3	3	3	3	3	3	3
JNZ rel	Jump on accumulator != 0	2	70h	2	2	2	2	3	3	2/3	3	3	3	3	3	3
CJNE A,	Compare A,	3	B5h	2	2	2	2	4	5	3/4	4	4	4	4	4	4

[illegible]

		cycles		49	49	49	49	72	72	55	74	73	94	74	70	72
		clocks		588	294	196	98	288	72	55	74	292	94	74	280	288
		execution time [us]		49,00	14,70	4,90	8,17	8,73	2,18	0,55	3,70	18,25	6,71	3,70	7,00	9,60