



AP180 Application Note for Instruction Set

***AN-01
April 2004***

Application Notice Title

AP180 Instruction SET

Revision History

Rev.	History	Issue Date	Remark
1.0	Initial issue	April 27, 2004	Preliminary

PRELIMINARY

**AP180 INSTRUCTION SET**

The instruction set of the AP180 is fully compatible with the industry standard 8051. Standard 8051 development tools can be used to develop software for the AP180. All AP180 instructions are the binary and functional equivalent of 8051 counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is much faster than that of the standard 8051 since AP180's machine cycles is with 1 system-clock length.

The AP180 provides a function with decrement for the data pointer by the instruction "DB A5H".

Table1. AP180 INSTRUCTION SET SUMMARY

Mnemonic	Description	Bytes	Clock Cycles
ARITHMETIC OPERATIONS			
ADD A,Rn	Add register to A	1	1
ADD A,direct	Add direct byte to A	2	2
ADD A,@Ri	Add indirect RAM to A	1	1
ADD A,#data	Add immediate to A	2	2
ADDC A,Rn	Add register to A with carry	1	1
ADDC A,direct	Add direct byte to A with carry	2	2
ADDC A,@Ri	Add indirect RAM to A with carry	1	1
ADDC A,#data	Add immediate to A with carry	2	2
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,direct	Subtract direct byte from A with borrow	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	1	1
SUBB A,#data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	1
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	1
INC DPTR	Increment DPTR	1	1
DB A5H	* Decrement DPTR	1	1
MUL AB	* Multiply A and B; MAC = MAC+A*B	1	2



AP180 Application Note for Instruction Set

AN-01
April 2004

Mnemonic	Description	Bytes	Clock Cycles
DIV AB	Divide A by B	1	11
DA A	Decimal Adjust A	1	1
LOGICAL OPERATIONS			
ANL A,Rn	AND Register to A	1	1
ANL A,direct	AND direct byte to A	2	2
ANL A,@Ri	AND indirect RAM to A	1	1
ANL A,#data	AND immediate to A	2	2
ANL direct,A	AND A to direct byte	2	2
ANL direct,#data	AND immediate to direct byte	3	3
ORL A,Rn	OR Register to A	1	1
ORL A,direct	OR direct byte to A	2	2
ORL A,@Ri	OR indirect RAM to A	1	1
ORL A,#data	OR immediate to A	2	2
ORL direct,A	OR A to direct byte	2	2
ORL direct,#data	OR immediate to direct byte	3	3
XRL A,Rn	Exclusive-OR Register to A	1	1
XRL A,direct	Exclusive-OR direct byte to A	2	2
XRL A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL A,#data	Exclusive-OR immediate to A	2	2
XRL direct,A	Exclusive-OR A to direct byte	2	2
XRL direct,#data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
SWAP A	Swap nibbles of A	1	1
DATA TRANSFER			
MOV A,Rn	Move register to A	1	1
MOV A,direct	Move direct byte to A	2	2



AP180 Application Note for Instruction Set

AN-01
April 2004

Mnemonic	Description	Bytes	Clock Cycles
MOV A,@Ri	Move indirect RAM to A	1	1
MOV A,#data	Move immediate to A	2	2
MOV Rn,A	Move A to register	1	1
MOV Rn,direct	Move direct byte to register	2	2
MOV Rn,#data	Move immediate to register	2	2
MOV direct,A	Move A to direct byte	2	2
MOV direct,Rn	Move register to direct byte	2	2
MOV direct,direct	Move direct byte to direct	3	3
MOV direct,@Ri	Move indirect RAM to direct byte	2	2
MOV direct,#data	Move immediate to direct byte	3	3
MOV @Ri,A	Move A to indirect RAM	1	1
MOV @Ri,direct	Move direct to indirect RAM	2	2
MOV @Ri,#data	Move immediate to indirect RAM	2	2
MOV DPTR,#data16	Load data pointer with 16-bit constant	3	3
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A,@A+PC	Move code byte relative PC to A	1	3
MOVX A,@Ri	Move external data (8-bit address) to A	1	1
MOVX @Ri,A	Move A to external data (8-bit address)	1	1
MOVX A,@DPTR	Move external data (16-bit address) to A	1	1
MOVX @DPTR,A	Move A to external data (16-bit address)	1	1
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A,Rn	Exchange register with A	1	2
XCH A,direct	Exchange direct byte with A	2	3
XCH A,@Ri	Exchange indirect RAM with A	1	2
XCHD A,@Ri	Exchange low nibble of indirect RAM with A	1	2
BOOLEAN MANIPULATION			
CLR C	Clear carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement carry	1	1



AP180 Application Note for Instruction Set

AN-01
April 2004

Mnemonic	Description	Bytes	Clock Cycles
CPL bit	Complement direct bit	2	2
ANL C,bit	AND direct bit to carry	2	2
ANL C,/bit	AND complement of direct bit to carry	2	2
ORL C,bit	OR direct bit to carry	2	2
ORL C,/bit	OR complement of direct bit to carry	2	2
MOV C,bit	Move direct bit to carry	2	2
MOV bit,C	Move carry to direct bit	2	2
JC rel	Jump if carry is set jump: 3 cycles	2	3 (2)
JNC rel	Jump if carry is not set no jump: (2 cycles)	2	3 (2)
JB bit,rel	Jump if direct bit is set	3	4 (3)
JNB bit,rel	Jump if direct bit is not set	3	4 (3)
JBC bit,rel	Jump if direct bit is set and clear bit	3	4 (3)
PROGRAM BRANCHING			
ACALL addr11	Absolute subroutine call	2	4
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	4
RETI	Return from interrupt	1	4
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	2
JZ rel	Jump if A equals zero	2	3 (2)
JNZ rel	Jump if A does not equal zero	2	3 (2)
CJNE A,direct,rel	Compare direct byte to A and jump if not equal	3	4 (3)
CJNE A,#data,rel	Compare immediate to A and jump if not equal	3	4 (3)
CJNE Rn,#data,rel	Compare immediate to register and jump if not equal	3	4 (3)
CJNE @Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	5 (4)
DJNZ Rn,rel	Decrement register and jump if not zero	2	4 (3)
DJNZ direct,rel	Decrement direct byte and jump if not zero	3	5 (4)
NOP	No operation	1	1