Opcode	Mnemonic	Notes
1n ll hh	set Rn, hhll	Set Register Rn to hhll
2n	ld Rn	Load R0 from Rn
3n	st Rn	Store Rn to R0
4n	ld @Rn	Load byte from memory[Rn] to R0, upper cleared. Post increments Rn
5n	st @Rn	Store byte in R0 to memory[Rn] Post increments Rn
6n	ldd @Rn	Load word from memory[Rn,Rn+1] to R0 Post increments Rn by 2.
7n	std @Rn	Store R0 to memory[Rn,Rn+1] as a word. Post increments Rn by 2.
8n	pop @Rn	Load byte from memory[Rn] to R0, upper cleared. Pre decrements Rn
9n	stp @Rn	Store byte in R0 to memory[Rn] Pre decrements Rn
An	add Rn	Add Rn to R0, carry into R14 bit 8
Bn	sub Rn	Subtract Rn from R0 using 2's complement , R0 := R0 + ~Rn + 1, carry into R14 bit 8
Cn	popd @Rn	Load R0 with memory[Rn,Rn+1], pre decrementing. The high byte is read first, so this should be the opppsite of STD @Rn]
Dn	cpr Rn	Compare . As subtraction but the result is written to R13 and the carry. *Branch conditions represent R13*
En	inr Rn	Increment Rn.
Fn	dcr Rn	Decrement Rn.
00	rtn	Return to 6502 processor
01	br ea	Branch always (offset as for 65C02)
02	bnc ea	Branch if No Carry (e.g. bit 8 of R14 clear)
03	bc ea	Branch if Carry (e.g. bit 8 of R14 set)
04	bp ea	Branch if Plus (e.g. bit 15 of R13 clear)
05	bm ea	Branch if Minus (e.g. bit 15 of R13 set)
06	bz ea	Branch if Zero (the next 4 all apply to R13
07	bnz ea	Branch if NonZero
08	bm1 ea	Branch if Minus 1
09	bnm1 ea	Branch if Not Minus 1
0A	bk	Break . Dumps the registers

0B	rs	Return from Subroutine
0C	bs ea	Branch to Subroutine
0D	bsl ea.long	As branch but with a 16 bit relative opcode.
0E nr	ext16 Rn,func	Performs extended register function func on Register Rn.
0F nn	ext16 func	Performs extended function func.
Register	**Name**	**Notes**
R0	Accumulator	
R12	Stack ptr	
R13	Compare	The result of CPR goes here.
R14	Status	High order byte of R14 contains prior register, used to check tests. Bit 0 of this byte is the carry.
R15	PC	

Prior Register

Only register operations (10-FF) change the high byte of R14. This is set to the register number (0-F) and used for tests *except* for add, sub and cpr which set it to zero.