

6502 Dead Cycles ¹ Opcode/Addressing Mode	Cycle ^{2,3}	Internal Operation ⁴	External Operation 6502 ¹⁵	External Operation 65C02 ¹⁵	C74-6502 NMOS	C74-6502 CMOS	C74-6502 Base Microinstruction
1. zpg,X/Y	3	BAL + X	PFA ⁶	PBA ⁷	B := *zDP	B := *PBA	DPL := B + X
2. abs,X/Y – read w/ pg. crossing	4*	BAH + 1	PFA	PBA	B := *DP	B := *PBA	DPH := DPH + 1 -- circuitry
3. abs,X/Y – write w/o pg. crossing ⁸	4	BAH + 0	FFA ^{5,8}	FFA ^{5,8}	B := *DP	B := *DP	DPH := B + 0; USE(IC) -- uses "B := *PBA" for 'C02 ... incompatible
4. abs,X/Y – write w/ pg. crossing	4	BAH + 1	PFA	PBA	B := *DP	B := *PBA	DPH := B + 0; USE(IC)
5. (zpg),Y – read w/ pg. crossing	5*	BAH + 1	PFA	PBA	B := *DP	B := *PBA	DPH := DPH + 1 -- circuitry
6. (zpg),Y – write w/o pg.crossing	5	BAH + 0	FFA	PBA	B := *DP	B := *PBA	DPH := DPH + 0; USE(IC) -- UFOs use *DPt
7. (zpg),Y – write w/ pg.crossing	5	BAH + 1	PFA	PBA	B := *DP	B := *PBA	DPH := DPH + 0; USE(IC) -- UFOs use *DPt
8. (zpg,X)	3	BAL + X	PFA	PBA	B := *zDP	B := *PBA	DPL := B + X
9. RMW zpg,X – "Read, Modify, Write" Opcodes ¹¹	3	BAL + X	PFA	PBA	B := *zDP	B := *PBA	DPL := B + X
10. RMW abs,X – 6502 w/o pg. crossing	4	BAH + 0	FFA		B := *DP		DPH := DPH + 0; USE(IC)
11. RMW abs,X – 6502 w/ pg. crossing	4	BAH + 1	PFA		B := *DP		DPH := DPH + 0; USE(IC)
12. RMW abs,X – 65C02 INC/DEC w/o pg. crossing	4	BAH + 0		FFA		B := *DP	DPH := DPH + 0; USE(IC) -- uses "B := *PBA" for 'C02 ... incompatible.
13. RMW abs,X – 65C02 INC/DEC w/ pg. crossing	4	BAH + 1		PBA		B := *PBA	DPH := DPH + 0; USE(IC)
14. RMW abs,X – 65C02 ASL LSR ROL ROR w/ pg. crossing ¹²	4*	BAH + 1		PBA		B := *PBA	DPH := DPH + 1 -- circuitry
15. RMW zpg	4	Modify	FFA (W) ⁹	FFA	B := *PBA	B := *PBA	Can't duplicate (W) for NMOS 6502 ... incompatible
16. RMW abs	5	Modify	FFA (W)	FFA	B := *PBA	B := *PBA	Can't duplicate (W) for NMOS 6502
17. RMW zpg,X	5	Modify	FFA (W)	FFA	B := *PBA	B := *PBA	Can't duplicate (W) for NMOS 6502
18. RMW abs,X	6 ¹²	Modify	FFA (W)	FFA	B := *PBA	B := *PBA	Can't duplicate (W) for NMOS 6502
19. All 1 Byte Opcodes	2	–	PC ¹⁰	PC ¹⁰	B := *PC	B := *PC	B := DPL := *PC -- circuitry
20. JMP (abs)/(abs,X) – 65C02	4	–		PBA	N/A	B := *PBA	NOP (B := *PBA)
21. JSR	3	–	SP	SP	B := *SP	B := *SP	B := *SP
22. RTS, RTI, PLP, PLA, PLY, PLX	3	SP + 1	SP	SP	B := *SP	B := *SP	SP += 1
23. RTS	6	PC + 1	PC ¹⁰	PC ¹⁰	B := *PC	B := *PC	DPL := B := *PC; PC += 1
24. Branch (taken) ¹⁴	3	PCL + BOF ¹³	PC ¹⁰	PC ¹⁰	B := *PC	B := *PC	PCL := PCL + B; EXIT.CC
25. Branch (taken) – w/ pg. crossing ¹⁴	4*	PCH +/- 1	PC ¹⁰	PC ¹⁰	B := *PC	B := *PC	PCH = PCH + signextend(*); USE(IC)
26. Extra BCD Cycle – 65C02	*	BCD		PBA	N/A	B := *PBA	

¹ A Dead Cycle is one in which the CPU is busy with an internal operation and does not make specific use of the external buses.

² Cycle numbers are with reference to Fetch_Opcode as cycle 1.

³ "*" denotes the cycle is added on a page crossing.

⁴ BAL and BAH refer to "Base Address Low" and "Base Address High" respectively.

⁵ FFA refers to a Fully Formed Address, which is the final target address of a given addressing mode.

⁶ PFA refers to a Partially-Formed Address, one which has yet to be adjusted, as follows:

- zpg,X/Y – ("00", BAL) Base address before the index register is added to the low-byte (BAL doesn't yet reflect the index offset)
- abs,X/Y – (BAH, BAL+X/Y) Base address before the low-byte carry is added to the high-byte (BAH doesn't yet reflect the page crossing)
- (zpg),Y – (BAH, BAL+Y) Base address before the low-byte carry is added to the high-byte (BAH doesn't yet reflect the page crossing)
- (zpg,X) – ("00", BAL) Base address before the index register is added to the low-byte (BAL doesn't yet reflect the index offset)

⁷ PBA refers to the Previous Bus Address (i.e., the value on the address bus from the previous cycle). This is the "fix" introduced by the 65C02. Re-reading the PBA is assumed to be a safe action, preferable to generating a "stray" read with an "invalid address", aka a Partially Formed Address (PFA), as the NMOS 6502 does. A PBA is also used on the 65C02 as a "safe" address for the dead cycle in JMP (abs) and the extra cycle in BCD operations.

⁸ abs,X/Y write operations without a page-crossing actually read from the Fully Formed Address before writing to it. The read can be troublesome when accessing I/O devices where reads are destructive. For 65C02 there's a software workaround, which is to ensure that the write to abs,X/Y triggers a page-crossing, which means the address during the dead cycle will be a PBA, not the Fully Formed Address. The only software workaround that works on both NMOS and CMOS 6502 is to avoid abs,X/Y address mode when writing to the read-sensitive device.

⁹ External Operations are Reads unless otherwise noted by "(W)".

¹⁰ PC refers to the address at PC, as follows:

- 1-Byte Opcodes Cycle 2 – Address of next opcode
- RTS Cycle 6 – Return address - 1 (as retrieved from the stack)
- Branch (taken) Cycle 3 – Address of next opcode after Branch
- Branch (taken) w/ pg. crossing Cycle 4 – High-byte unchanged from prior cycle/Low-byte of target address (PCL + BOF¹³)

¹¹ "Read, Write, Modify" Opcodes refers to INC, DEC, ASL, LSR, ROL and ROR.

¹² On the 65C02, the "Modify" operation occurs in cycle 5 for ASL, LSR, ROL and ROR if a page is not crossed.

¹³ BOF refers to the Branch Offset value.

¹⁴ Dead cycles are 5 and 6* for 65C02 BBR and BBS (rather than cycles 3 and 4* for standard branches).

¹⁵ The behaviour shown for each CPU has been verified on the Visual 6502 (www.visual6502.org) and on a WDC 65C02 respectively. See also http://archive.6502.org/books/mcs6500_family_hardware_manual.pdf Appendix A.