

6502 cheat sheet

ADC arg hex len time **N V Z C**
add with carry

#n	\$69	2	2
ZP	\$65	2	3
ZP,X	\$75	2	4
ABS	\$6D	3	4
ABS,X	\$7D	3	4+
ABS,Y	\$79	3	4+
(ZP,X)	\$61	2	6
(ZP),Y	\$71	2	5+

AND arg hex len time **N Z**
bitwise AND with accumulator

#n	\$29	2	2
ZP	\$25	2	3
ZP,X	\$35	2	4
ABS	\$2D	3	4
ABS,X	\$3D	3	4+
ABS,Y	\$39	3	4+
(ZP,X)	\$21	2	6
(ZP),Y	\$31	2	5+

ASL arg hex len time **N Z C**
arithmetic shift left

A	\$0A	1	2
ZP	\$06	2	5
ZP,X	\$16	2	6
ABS	\$0E	3	6
ABS,X	\$1E	3	7

BIT arg hex len time **N V Z**
test bits

ZP	\$24	2	3
ABS	\$2C	3	4

BRK arg hex len time **B (on stack)**
break

rel	\$00	2	7
-----	------	---	---

CMP arg hex len time **N Z C**
compare accumulator
Z C
A<M 0 0
A=M 1 1
A>M 0 1

#n	\$C9	2	2
ZP	\$C5	2	3
ZP,X	\$D5	2	4
ABS	\$CD	3	4
ABS,X	\$DD	3	4+
ABS,Y	\$D9	3	4+
(ZP,X)	\$C1	2	6
(ZP),Y	\$D1	2	5+

CPX arg hex len time **N Z C**
compare X register

#n	\$E0	2	2
ZP	\$E4	2	3
ABS	\$EC	3	4

CPY arg hex len time **N Z C**
compare Y register

#n	\$C0	2	2
ZP	\$C4	2	3
ABS	\$CC	3	4

DEC arg hex len time **N Z**
decrement memory

ZP	\$C6	2	5
ZP,X	\$D6	2	6
ABS	\$CE	3	6
ABS,X	\$DE	3	7

EOR arg hex len time **N Z**
bitwise exclusive OR

#n	\$49	2	2
ZP	\$45	2	3
ZP,X	\$55	2	4
ABS	\$4D	3	4
ABS,X	\$5D	3	4+
ABS,Y	\$59	3	4+
(ZP,X)	\$41	2	6
(ZP),Y	\$51	2	5+

INC arg hex len time **N Z**
increment memory

ZP	\$E6	2	5
ZP,X	\$F6	2	6
ABS	\$EE	3	6
ABS,X	\$FE	3	7

JMP arg hex len time
jump

ABS	\$4C	3	3
(ABS)	\$6C	3	5

JSR arg hex len time
jump to subroutine

ABS	\$20	3	6
-----	------	---	---

LDA arg hex len time **N Z**
load accumulator

#n	\$A9	2	2
ZP	\$A5	2	3
ZP,X	\$B5	2	4
ABS	\$AD	3	4
ABS,X	\$BD	3	4+
ABS,Y	\$B9	3	4+
(ZP,X)	\$A1	2	6
(ZP),Y	\$B1	2	5+

LDX arg hex len time **N Z**
load X register

#n	\$A2	2	2
ZP	\$A6	2	3
ZP,X	\$B6	2	4
ABS	\$AE	3	4
ABS,Y	\$BE	3	4+

LDY arg hex len time **N Z**
load Y register

#n	\$A0	2	2
ZP	\$A4	2	3
ZP,X	\$B4	2	4
ABS	\$AC	3	4
ABS,X	\$BC	3	4+

LSR arg hex len time **N Z C**
logical shift right

A	\$4A	1	2
ZP	\$46	2	5
ZP,X	\$56	2	6
ABS	\$4E	3	6
ABS,X	\$5E	3	7

NOP arg hex len time
no operation

-	\$EA	1	2
---	------	---	---

ORA arg hex len time **N Z**
bitwise OR with accumulator

#n	\$09	2	2
ZP	\$05	2	3
ZP,X	\$15	2	4
ABS	\$0D	3	4
ABS,X	\$1D	3	4+
ABS,Y	\$19	3	4+
(ZP,X)	\$01	2	6
(ZP),Y	\$11	2	5+

ROL arg hex len time **N Z C**
rotate left

A	\$2A	1	2
ZP	\$26	2	5
ZP,X	\$36	2	6
ABS	\$2E	3	6
ABS,X	\$3E	3	7

ROR arg hex len time **N Z C**
rotate right

A	\$6A	1	2
ZP	\$66	2	5
ZP,X	\$76	2	6
ABS	\$6E	3	6
ABS,X	\$7E	3	7

RTI arg hex len time **All (from stack)**
return from interrupt

-	\$40	1	6
---	------	---	---

RTS arg hex len time
return from subroutine

-	\$60	1	6
---	------	---	---

SBC arg hex len time **N V Z C**
subtract with carry

#n	\$E9	2	2
ZP	\$E5	2	3
ZP,X	\$F5	2	4
ABS	\$ED	3	4
ABS,X	\$FD	3	4+
ABS,Y	\$F9	3	4+
(ZP,X)	\$E1	2	6
(ZP),Y	\$F1	2	5+

STA arg hex len time
store accumulator

ZP	\$85	2	3
ZP,X	\$95	2	4
ABS	\$8D	3	4
ABS,X	\$9D	3	5
ABS,Y	\$99	3	5
(ZP,X)	\$81	2	6
(ZP),Y	\$91	2	6

STX arg hex len time
store X register

ZP	\$86	2	3
ZP,Y	\$96	2	4
ABS	\$8E	3	4

STY arg hex len time
store Y register

ZP	\$84	2	3
ZP,X	\$94	2	4
ABS	\$8C	3	4

Branch Instructions:

BPL arg hex len time
branch on plus

rel	\$10	2	2/3+
-----	------	---	------

BMI arg hex len time
branch on minus

rel	\$30	2	2/3+
-----	------	---	------

BVC arg hex len time
branch on overflow clear

rel	\$50	2	2/3+
-----	------	---	------

BVS arg hex len time
branch on overflow set

rel	\$70	2	2/3+
-----	------	---	------

BCC arg hex len time
branch on carry clear

rel	\$90	2	2/3+
-----	------	---	------

BCS arg hex len time
branch on carry set

rel	\$B0	2	2/3+
-----	------	---	------

BNE arg hex len time
branch on not equal

rel	\$D0	2	2/3+
-----	------	---	------

BEQ arg hex len time
branch on equal

rel	\$F0	2	2/3+
-----	------	---	------

Flag Instructions:

CLC arg hex len time **C**
clear carry

-	\$18	1	2
---	------	---	---

SEC arg hex len time **C**
set carry

-	\$38	1	2
---	------	---	---

CLI arg hex len time **I**
clear interrupt

-	\$58	1	2
---	------	---	---

SEI arg hex len time **I**
set interrupt

-	\$78	1	2
---	------	---	---

CLV arg hex len time **V**
clear overflow

-	\$B8	1	2
---	------	---	---

CLD arg hex len time **D**
clear decimal

-	\$D8	1	2
---	------	---	---

SED arg hex len time **D**
set decimal

-	\$F8	1	2
---	------	---	---

Stack Instructions:

TXS arg hex len time
transfer X to SP

-	\$9A	1	2
---	------	---	---

TSX arg hex len time **N Z**
transfer SP to X

-	\$BA	1	2
---	------	---	---

PHA arg hex len time
push accumulator

-	\$48	1	3
---	------	---	---

PLA arg hex len time **N Z**
pull accumulator

-	\$68	1	4
---	------	---	---

PHP arg hex len time
push processor status

-	\$08	1	3
---	------	---	---

PLP arg **All (from stack)** hex len time
pull processor status

-	\$28	1	4
---	------	---	---

Register Instructions:

TAX arg hex len time **N Z**
transfer A to X

-	\$AA	1	2
---	------	---	---

TXA arg hex len time **N Z**
transfer X to A

-	\$8A	1	2
---	------	---	---

TAY arg hex len time **N Z**
transfer A to Y

-	\$A8	1	2
---	------	---	---

TYA arg hex len time **N Z**
transfer Y to A

-	\$98	1	2
---	------	---	---

DEX arg hex len time **N Z**
decrement X

-	\$CA	1	2
---	------	---	---

INX arg hex len time **N Z**
increment X

-	\$E8	1	2
---	------	---	---

DEY arg hex len time **N Z**
decrement Y

-	\$88	1	2
---	------	---	---

INY arg hex len time **N Z**
increment Y

-	\$C8	1	2
---	------	---	---

6502 Registers:

- SP - Stack Pointer
- PC - Program Counter
- X - Index Register
- Y - Index Register
- A - Accumulator
- F - Flag Register (Processor Status)

N	V	*	B	D	I	Z	C
---	---	---	---	---	---	---	---

Branch Execution Time:

A branch that is not taken requires two machine cycles. Add one if the branch is taken and add another if the branch crosses the page boundary.

Vectors:

- \$FFFA - NMI
- \$FFFC - RESET
- \$FFFE - IRQ