ADC Add Memory to Accumulator with Carry

A + M + C -> A, C

N	Z	C	I	D	V
+	+	+	-	-	+

addressing	assembler	opc	bytes	cyles
immidiate	ADC #oper	69	2	2
zeropage	ADC oper	65	2	3
zeropage,X	ADC oper,X	75	2	4
absolute	ADC oper	6D	3	4
absolute,X	ADC oper,X	7D	3	4*
absolute,Y	ADC oper,Y	79	3	4*
(indirect,X)	ADC (oper,X)	61	2	6
(indirect),Y	ADC (oper),Y	71	2	5*

AND AND Memory with Accumulator

A AND M -> A

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
immidiate	AND #oper	29	2	2
zeropage	AND oper	25	2	3
zeropage,X	AND oper,X	35	2	4
absolute	AND oper	2D	3	4
absolute,X	AND oper,X	3D	3	4*
absolute,Y	AND oper,Y	39	3	4*
(indirect,X)	AND (oper,X)	21	2	6
(indirect),Y	AND (oper),Y	31	2	5*

ASL Shift Left One Bit (Memory or Accumulator)

C <- [76543210] <- 0

N	Z	C	I	D	V
+	+	+	-	-	-

addressing	assembler	opc	bytes	cyles
accumulator	ASL A	0A	1	2
zeropage	ASL oper	06	2	5
zeropage,X	ASL oper,X	16	2	6
absolute	ASL oper	0E	3	6
absolute,X	ASL oper,X	1E	3	7

BCC Branch on Carry Clear

branch on C = 0

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
relative	BCC oper	90	2	2**

BCS Branch on Carry Set

branch on C = 1

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
relative	BCS oper	B0	2	2**

BEQ Branch on Result Zero

branch on Z = 1

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
relative	BEQ oper	F0	2	2**

BIT Test Bits in Memory with Accumulator

bits 7 and 6 of operand are transfered to
bit 7 and 6 of SR (N,V);
the zeroflag is set to the result of operand
AND accumulator.

A AND M, M7 -> N, M6 -> V

N	Z	C	I	D	V
M7	+	-	-	-	M6

addressing	assembler	opc	bytes	cyles
zeropage	BIT oper	24	2	3
absolute	BIT oper	2C	3	4

BMI Branch on Result Minus

branch on N = 1

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
relative	BMI oper	30	2	2**

BNE Branch on Result not Zero

branch on Z = 0

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
relative	BNE oper	D0	2	2**

BPL Branch on Result Plus

branch on N = 0

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
relative	BPL oper	10	2	2**

BRK Force Break

interrupt,
push PC+2, push SR

N	Z	C	I	D	V
-	-	-	1	-	-

addressing	assembler	opc	bytes	cyles
implied	BRK	00	1	7

BVC Branch on Overflow Clear

branch on V = 0

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
relative	BVC oper	50	2	2**

BVS Branch on Overflow Set

branch on V = 1

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
relative	BVC oper	70	2	2**

CLC Clear Carry Flag

0 -> C

N	Z	C	I	D	V
-	-	0	-	-	-

addressing	assembler	opc	bytes	cyles
implied	CLC	18	1	2

CLD Clear Decimal Mode

0 -> D

N	Z	C	I	D	V
-	-	-	-	0	-

addressing	assembler	opc	bytes	cyles
implied	CLD	D8	1	2

CLI Clear Interrupt Disable Bit

```
     0 -> I                           N Z C I D V
                                      - - - 0 - -

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     implied       CLI           58    1     2
```

CLV Clear Overflow Flag

```
     0 -> V                           N Z C I D V
                                      - - - - - 0

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     implied       CLV           B8    1     2
```

CMP Compare Memory with Accumulator

```
     A - M                              N Z C I D V
                                      + + + - - -

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     immidiate     CMP #oper     C9    2     2
     zeropage      CMP oper      C5    2     3
     zeropage,X    CMP oper,X    D5    2     4
     absolute      CMP oper      CD    3     4
     absolute,X    CMP oper,X    DD    3     4*
     absolute,Y    CMP oper,Y    D9    3     4*
     (indirect,X)  CMP (oper,X)  C1    2     6
     (indirect),Y  CMP (oper),Y  D1    2     5*
```

CPX Compare Memory and Index X

```
     X - M                            N Z C I D V
                                      + + + - - -

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     immidiate     CPX #oper     E0    2     2
     zeropage      CPX oper      E4    2     3
     absolute      CPX oper      EC    3     4
```

CPY Compare Memory and Index Y

```
     Y - M                            N Z C I D V
                                      + + + - - -

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     immidiate     CPY #oper     C0    2     2
     zeropage      CPY oper      C4    2     3
     absolute      CPY oper      CC    3     4
```

DEC Decrement Memory by One

M - 1 -> M

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
zeropage	DEC oper	C6	2	5
zeropage,X	DEC oper,X	D6	2	6
absolute	DEC oper	CE	3	3
absolute,X	DEC oper,X	DE	3	7

DEX Decrement Index X by One

X - 1 -> X

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	DEC	CA	1	2

DEY Decrement Index Y by One

Y - 1 -> Y

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	DEC	88	1	2

EOR Exclusive-OR Memory with Accumulator

A EOR M -> A

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
immidiate	EOR #oper	49	2	2
zeropage	EOR oper	45	2	3
zeropage,X	EOR oper,X	55	2	4
absolute	EOR oper	4D	3	4
absolute,X	EOR oper,X	5D	3	4*
absolute,Y	EOR oper,Y	59	3	4*
(indirect,X)	EOR (oper,X)	41	2	6
(indirect),Y	EOR (oper),Y	51	2	5*

INC Increment Memory by One

M + 1 -> M

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
zeropage	INC oper	E6	2	5
zeropage,X	INC oper,X	F6	2	6
absolute	INC oper	EE	3	6
absolute,X	INC oper,X	FE	3	7

INX Increment Index X by One

X + 1 -> X

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	INX	E8	1	2

INY Increment Index Y by One

Y + 1 -> Y

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	INY	C8	1	2

JMP Jump to New Location

(PC+1) -> PCL
(PC+2) -> PCH

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
absolute	JMP oper	4C	3	3
indirect	JMP (oper)	6C	3	5

JSR Jump to New Location Saving Return Address

push (PC+2),
(PC+1) -> PCL
(PC+2) -> PCH

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
absolute	JSR oper	20	3	6

LDA Load Accumulator with Memory

M -> A

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
immidiate	LDA #oper	A9	2	2
zeropage	LDA oper	A5	2	3
zeropage,X	LDA oper,X	B5	2	4
absolute	LDA oper	AD	3	4
absolute,X	LDA oper,X	BD	3	4*
absolute,Y	LDA oper,Y	B9	3	4*
(indirect,X)	LDA (oper,X)	A1	2	6
(indirect),Y	LDA (oper),Y	B1	2	5*

LDX Load Index X with Memory

M -> X

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
immidiate	LDX #oper	A2	2	2
zeropage	LDX oper	A6	2	3
zeropage,Y	LDX oper,Y	B6	2	4
absolute	LDX oper	AE	3	4
absolute,Y	LDX oper,Y	BE	3	4*

LDY Load Index Y with Memory

M -> Y

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
immidiate	LDY #oper	A0	2	2
zeropage	LDY oper	A4	2	3
zeropage,X	LDY oper,X	B4	2	4
absolute	LDY oper	AC	3	4
absolute,X	LDY oper,X	BC	3	4*

LSR Shift One Bit Right (Memory or Accumulator)

0 -> [76543210] -> C

N	Z	C	I	D	V
-	+	+	-	-	-

addressing	assembler	opc	bytes	cyles
accumulator	LSR A	4A	1	2
zeropage	LSR oper	46	2	5
zeropage,X	LSR oper,X	56	2	6
absolute	LSR oper	4E	3	6
absolute,X	LSR oper,X	5E	3	7

NOP No Operation

```
     ---                              N Z C I D V
                                      - - - - - -

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     implied       NOP           EA    1     2
```

ORA OR Memory with Accumulator

```
     A OR M -> A                      N Z C I D V
                                      + + - - - -

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     immidiate     ORA #oper     09    2     2
     zeropage      ORA oper      05    2     3
     zeropage,X    ORA oper,X    15    2     4
     absolute      ORA oper      0D    3     4
     absolute,X    ORA oper,X    1D    3     4*
     absolute,Y    ORA oper,Y    19    3     4*
     (indirect,X)  ORA (oper,X)  01    2     6
     (indirect),Y  ORA (oper),Y  11    2     5*
```

PHA Push Accumulator on Stack

```
     push A                           N Z C I D V
                                      - - - - - -

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     implied       PHA           48    1     3
```

PHP Push Processor Status on Stack

```
     push SR                          N Z C I D V
                                      - - - - - -

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     implied       PHP           08    1     3
```

PLA Pull Accumulator from Stack

```
     pull A                           N Z C I D V
                                      + + - - - -

     addressing    assembler    opc  bytes  cyles
     --------------------------------------------
     implied       PLA           68    1     4
```

PLP Pull Processor Status from Stack

pull SR N Z C I D V
         from stack

addressing	assembler	opc	bytes	cyles
implied	PHP	28	1	4

ROL Rotate One Bit Left (Memory or Accumulator)

C <- [76543210] <- C N Z C I D V
                      + + + - - -

addressing	assembler	opc	bytes	cyles
accumulator	ROL A	2A	1	2
zeropage	ROL oper	26	2	5
zeropage,X	ROL oper,X	36	2	6
absolute	ROL oper	2E	3	6
absolute,X	ROL oper,X	3E	3	7

ROR Rotate One Bit Right (Memory or Accumulator)

C -> [76543210] -> C N Z C I D V
                      + + + - - -

addressing	assembler	opc	bytes	cyles
accumulator	ROR A	6A	1	2
zeropage	ROR oper	66	2	5
zeropage,X	ROR oper,X	76	2	6
absolute	ROR oper	6E	3	6
absolute,X	ROR oper,X	7E	3	7

RTI Return from Interrupt

pull SR, pull PC N Z C I D V
                  from stack

addressing	assembler	opc	bytes	cyles
implied	RTI	40	1	6

RTS Return from Subroutine

pull PC, PC+1 -> PC N Z C I D V
                     - - - - - -

addressing	assembler	opc	bytes	cyles
implied	RTS	60	1	6

SBC Subtract Memory from Accumulator with Borrow

A - M - C -> A

N	Z	C	I	D	V
+	+	+	-	-	+

addressing	assembler	opc	bytes	cyles
immidiate	SBC #oper	E9	2	2
zeropage	SBC oper	E5	2	3
zeropage,X	SBC oper,X	F5	2	4
absolute	SBC oper	ED	3	4
absolute,X	SBC oper,X	FD	3	4*
absolute,Y	SBC oper,Y	F9	3	4*
(indirect,X)	SBC (oper,X)	E1	2	6
(indirect),Y	SBC (oper),Y	F1	2	5*

SEC Set Carry Flag

1 -> C

N	Z	C	I	D	V
-	-	1	-	-	-

addressing	assembler	opc	bytes	cyles
implied	SEC	38	1	2

SED Set Decimal Flag

1 -> D

N	Z	C	I	D	V
-	-	-	-	1	-

addressing	assembler	opc	bytes	cyles
implied	SED	F8	1	2

SEI Set Interrupt Disable Status

1 -> I

N	Z	C	I	D	V
-	-	-	1	-	-

addressing	assembler	opc	bytes	cyles
implied	SEI	78	1	2

STA Store Accumulator in Memory

A -> M

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
zeropage	STA oper	85	2	3
zeropage,X	STA oper,X	95	2	4
absolute	STA oper	8D	3	4
absolute,X	STA oper,X	9D	3	5
absolute,Y	STA oper,Y	99	3	5
(indirect,X)	STA (oper,X)	81	2	6
(indirect),Y	STA (oper),Y	91	2	6

STX Store Index X in Memory

X -> M

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
zeropage	STX oper	86	2	3
zeropage,Y	STX oper,Y	96	2	4
absolute	STX oper	8E	3	4

STY Sore Index Y in Memory

Y -> M

N	Z	C	I	D	V
-	-	-	-	-	-

addressing	assembler	opc	bytes	cyles
zeropage	STY oper	84	2	3
zeropage,X	STY oper,X	94	2	4
absolute	STY oper	8C	3	4

TAX Transfer Accumulator to Index X

A -> X

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	TAX	AA	1	2

TAY Transfer Accumulator to Index Y

A -> Y

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	TAY	A8	1	2

TSX Transfer Stack Pointer to Index X

SP -> X

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	TSX	BA	1	2

TXA Transfer Index X to Accumulator

X -> A

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	TXA	8A	1	2

TXS Transfer Index X to Stack Register

X -> SP

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	TXS	9A	1	2

TYA Transfer Index Y to Accumulator

Y -> A

N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cyles
implied	TYA	98	1	2

* add 1 to cycles if page boundery is crossed

** add 1 to cycles if branch occurs on same page
add 2 to cycles if branch occurs to diff page

Legend to Flags:
+ .... modified
- .... not modified
1 .... set
0 .... cleared
M6 .... memory bit 6
M7 .... memory bit 7