MOS 6502 / MOS 6510

Deep Dive Instruction Set Part Two

CMP Compare Memory with Accumulator

A - M

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

addressing	assembler	opc	bytes	cycls
immediate	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*

* add 1 to cycles if page boundary is crossed

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

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

- Z=0, then A <> NUM and ‘BNE’
- Z=1, then A = NUM and ‘BEQ’
- C=0, then A < NUM and ‘BCC’
- C=1, then A >= NUM and ‘BCS’

CPX Compare Memory and Index X

X - M

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

addressing	assembler	opc	bytes	cycls
immediate	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	cycls
immediate	CPY #oper	C0	2	2
zeropage	CPY oper	C4	2	3
absolute	CPY oper	CC	3	4

* add 1 to cycles if page boundary is crossed

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

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

DEC Decrement Memory by One

M - 1 -> M

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

addressing	assembler	opc	bytes	cycls
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

* add 1 to cycles if page boundary is crossed

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

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

DEX Decrement Index X by One

X - 1 -> X

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

addressing	assembler	opc	bytes	cycls
implied	DEX	CA	1	2

DEY Decrement Index Y by One

Y - 1 -> Y

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

addressing	assembler	opc	bytes	cycls
implied	DEY	88	1	2

* add 1 to cycles if page boundary is crossed

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

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

EOR Exclusive-OR Memory with Accumulator

A EOR M -> A

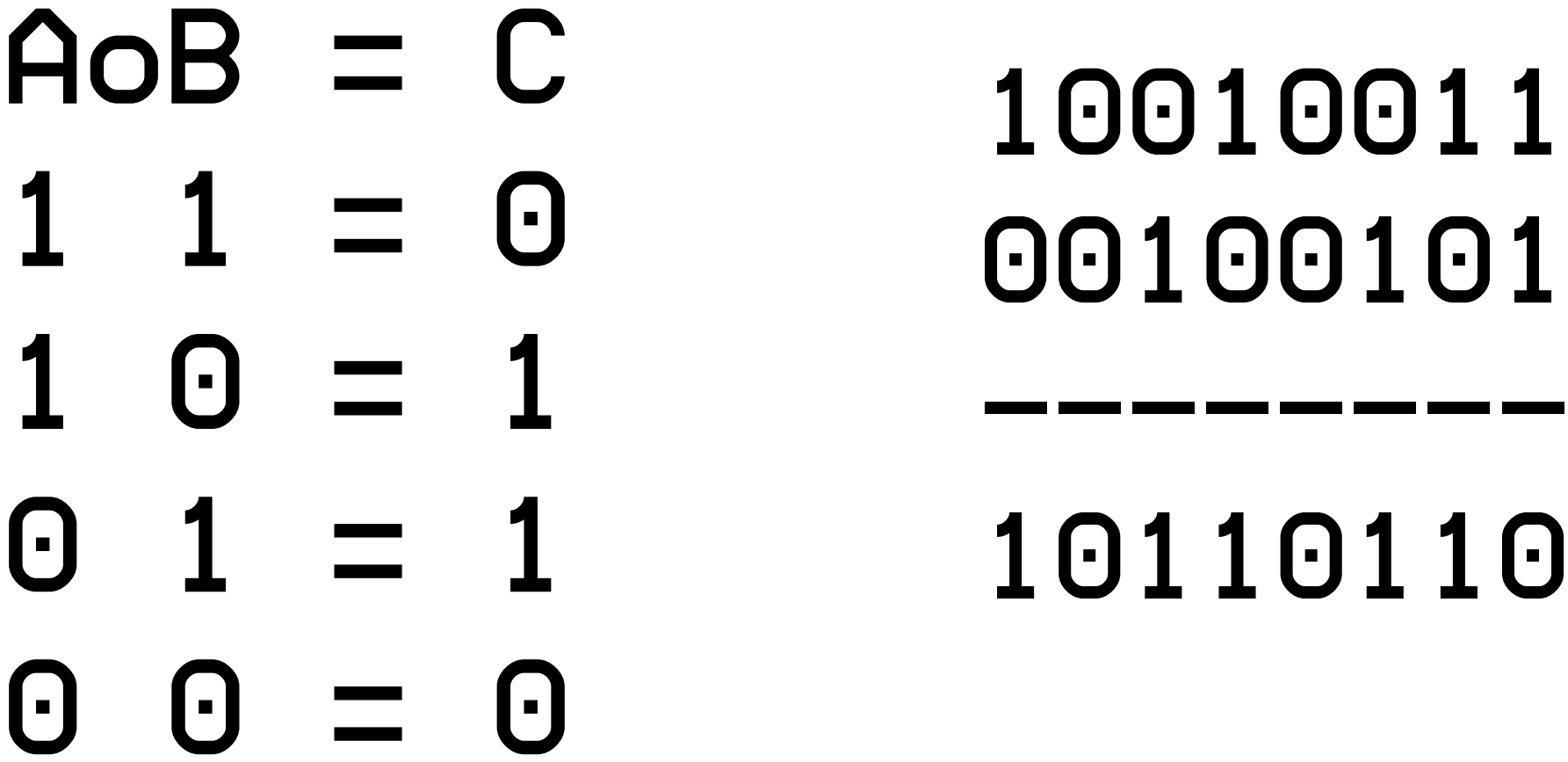
N	Z	C	I	D	V
+	+	-	-	-	-

addressing	assembler	opc	bytes	cycls
immediate	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*

* add 1 to cycles if page boundary is crossed

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

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


AoB = C
1 1 = 0
1 0 = 1
0 1 = 1
0 0 = 0
10010011
00100101
10110110

INC Increment Memory by One

M + 1 -> M

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

addressing	assembler	opc	bytes	cycls
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

* add 1 to cycles if page boundary is crossed

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

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

INX Increment Index X by One

X + 1 -> X

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

addressing	assembler	opc	bytes	cycls
implied	INX	E8	1	2

INY Increment Index Y by One

Y + 1 -> Y

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

addressing	assembler	opc	bytes	cycls
implied	INY	C8	1	2

* add 1 to cycles if page boundary is crossed

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

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

JMP Jump to New Location

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

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

addressing	assembler	opc	bytes	cycls
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	cycls
absolute	JSR oper	20	3	6

* add 1 to cycles if page boundary is crossed

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

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

LDX Load Index X with Memory

M -> X

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

addressing	assembler	opc	bytes	cycls
immediate	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*

* add 1 to cycles if page boundary is crossed

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

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

LDY Load Index Y with Memory

M -> Y

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

addressing	assembler	opc	bytes	cycls
immediate	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*

* add 1 to cycles if page boundary is crossed

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

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

LSR Shift One Bit Right (Memory or Accumulator)

0 -> ←76543210→ -> C

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

addressing	assembler	opc	bytes	cycls
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	cycls
implied	NOP	EA	1	2

* add 1 to cycles if page boundary is crossed

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

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

ORA OR Memory with Accumulator

A OR M -> A

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

addressing	assembler	opc	bytes	cycls
immediate	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*

* add 1 to cycles if page boundary is crossed

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

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

A+B	=	C
1	1	= 1
1	0	= 1
0	1	= 1
0	0	= 0

```
 10010011
 00100101
 --------
 10110111
```

PHA Push Accumulator on Stack

push A

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

addressing	assembler	opc	bytes	cycls
implied	PHA	48	1	3

PHP Push Processor Status on Stack

push SR

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

addressing	assembler	opc	bytes	cycls
implied	PHP	08	1	3

* add 1 to cycles if page boundary is crossed

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

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

PLA Pull Accumulator from Stack

pull A

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

addressing	assembler	opc	bytes	cycls
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	cycls
implied	PHP	28	1	4

* add 1 to cycles if page boundary is crossed

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

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

ROL Rotate One Bit Left (Memory or Accumulator)

C <- ←76543210← <- C

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

addressing	assembler	opc	bytes	cycls
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

* add 1 to cycles if page boundary is crossed

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

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

RTI Return from Interrupt

pull SR, pull PC

N	Z	C	I	D	V
from stack					

addressing	assembler	opc	bytes	cycls
implied	RTI	40	1	6

RTS Return from Subroutine

pull PC, PC+1 -> PC

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

addressing	assembler	opc	bytes	cycls
implied	RTS	60	1	6

* add 1 to cycles if page boundary is crossed

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

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

SBC Subtract Memory from Accumulator with Borrow

A - M - C -> A

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

addressing	assembler	opc	bytes	cycls
immediate	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*

* add 1 to cycles if page boundary is crossed

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

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

SEC Set Carry Flag

1 -> C

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

addressing	assembler	opc	bytes	cycls
implied	SEC	38	1	2

SED Set Decimal Flag

1 -> D

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

addressing	assembler	opc	bytes	cycls
implied	SED	F8	1	2

* add 1 to cycles if page boundary is crossed

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

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

SEI Set Interrupt Disable Status

1 -> I

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

addressing	assembler	opc	bytes	cycls
implied	SEI	78	1	2

STA Store Accumulator in Memory

A -> M

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

addressing	assembler	opc	bytes	cycls
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

* add 1 to cycles if page boundary is crossed

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

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

STX Store Index X in Memory

X -> M

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

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

* add 1 to cycles if page boundary is crossed

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

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

TAX Transfer Accumulator to Index X

A -> X

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

addressing	assembler	opc	bytes	cycls
implied	TAX	AA	1	2

TAY Transfer Accumulator to Index Y

A -> Y

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

addressing	assembler	opc	bytes	cycls
implied	TAY	A8	1	2

* add 1 to cycles if page boundary is crossed

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

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

TSX Transfer Stack Pointer to Index X

SP -> X

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

addressing	assembler	opc	bytes	cycls
implied	TSX	BA	1	2

TXA Transfer Index X to Accumulator

X -> A

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

addressing	assembler	opc	bytes	cycls
implied	TXA	8A	1	2

* add 1 to cycles if page boundary is crossed

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

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

TXS Transfer Index X to Stack Register

X -> SP

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

addressing	assembler	opc	bytes	cycls
implied	TXS	9A	1	2

TYA Transfer Index Y to Accumulator

Y -> A

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

addressing	assembler	opc	bytes	cycls
implied	TYA	98	1	2

* add 1 to cycles if page boundary is crossed

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

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