C: Array and String

ор 0х01	name BRK	cycles 1	adrmoo imp	eflags NVBDIZC -1-1-			
functio (S)-=	inter	PC:=(\$F	FFE)	$ \begin{array}{l} {\rm logic} \ PC = PC + 1 \\ $			
Examp	Example BRK						

$\mathtt{B}\mathrm{R}\mathrm{K}$

ор 0х00	name BRK	cycles 1	adr imp	flags NVBDIZC -1-1-			
descrip functio (S)-=	inter	rrupt PC:=(\$F	FFE)	$\begin{array}{l} \operatorname{logic} & \operatorname{PC} = \operatorname{PC} + 1 \\ & \operatorname{bPoke}(\operatorname{SP},\operatorname{PC}.\operatorname{h}) \\ & \operatorname{SP} = \operatorname{SP} - 1 \\ & \operatorname{bPoke}(\operatorname{SP},\operatorname{PC}.1) \\ & \operatorname{SP} = \operatorname{SP} - 1 \\ & \operatorname{bPoke}(\operatorname{SP},\ (\operatorname{P} \backslash \$10)\) \\ & \operatorname{SP} = \operatorname{SP} - 1 \\ & \operatorname{l} = \operatorname{bPeek}(\backslash \$\operatorname{FFFE}) \\ & \operatorname{h} = \operatorname{bPeek}(\backslash \$\operatorname{FFFF}) << 8 \\ & \operatorname{PC} = \operatorname{h} \operatorname{l} \end{array}$			
exampl	example BRK						

ор 0х01	name ORA	adr izx	bytes 2	cycles 2	function A:=A or adr
flags		DIZC *-		P	$A = A \mid M$ $C \cdot N = A \cdot 7$ $C \cdot Z = (A = = 0)$? 1:0
descrip or v	tion vith ac	cumul	ator		
example ORA (\$A5,X)				c-fct	cpu_6502_ora_izy

$\begin{array}{c} \text{example} \\ \text{BRK} \end{array}$		ORA (\$A5,X)	ORA \$AB
ic	PC = PC + 1 $bPoke (SP, PC. h)$ $SP = SP - 1$ $bPoke (SP, PC. 1)$ $SP = SP - 1$ $bPoke (SP, (P $10))$ $SP = SP - 1$ $bPoke (SP (P $10))$ $SP = SP - 1$ $1 = bPeek ($FFFE)$ $h = bPeek ($FFFF) < < 8$ $PC = h 1$	$A = A \mid M$ P.N = A.7 P.Z = (A==0) ? 1:0	$A = A \mid M$ P.N = A.7 P.Z = (A==0) ? 1:0
$\begin{array}{ll} {\rm flags} & {\rm logic} \\ {\rm NVBDIZC} \end{array}$	-1-1-	NVBDIZC **-	NVBDIZC **-
function interrupt	(S)-=:PC,P PC:=(\$FFFE)	or with accumulator A:=A or adr	or with accumulator $A:=A$ or adr
cycles adr"-mode 0 imp		izx	dz
$ \begin{array}{c} \text{cycles} \\ 0 \end{array} $		9	က
$\underset{1}{\text{bytes}}$		62	7
$\begin{array}{ccc} \text{opcode} & \text{mnmonic} & \text{bytes} \\ \text{0x00} & \text{BRK} & 1 \end{array}$		ORA	ORA
$\begin{array}{c} \rm opcode \\ 0x00 \end{array}$		$ig egin{array}{c} \mathbf{Text1} \ 0 \mathrm{x} 01 \end{array}$	0x05

Table 1: opcode table

ор 0х00	name BRK	adr imp	bytes 1	cycles 0	function (S)-=:PC,P PC:=(\$FFFE)
flags descrip	NVBD 1- tion errupt			b S b S b S 1	CC = PC + 1 PPoke(SP,PC.h) P = SP - 1 PPoke(SP,PC.l) P = SP - 1 PPoke(SP, (P \$10)) P = SP - 1 = bPeek(\$FFFE) = bPeek(\$FFFF)<<8 CC = h 1
exampl		RK		c-fct	<pre>void cpu_6502_BRK_imp();</pre>

```
name
                  \displaystyle \mathop{\mathrm{imp}}^{\mathrm{adr}}
                                              \frac{\rm logic}{\rm PC}\,=\,{\rm PC}\,+\,1
                           NVBDIZC
0x00
         BRK
                            -1-1--
                                                     bPoke(SP,PC.h)
                                                     SP = \hat{SP} - 1
                            ^{\rm bytes}_{\rm 1b}
                                     cycles
example
                                                     bPoke(SP,PC.1)
          BRK
                                       1c
                                                    SP = SP - 1
bPoke(SP, (P|\$10))
                                                     SP = SP - 1
        interrupt
                                                     l = bPeek($FFFE)
                                                     h = bPeek($FFFF)<<8
\stackrel{\mathrm{description}}{\text{(S)-=:PC,P PC:=(\$FFFE)}}
                                                    PC = h \mid l
void cpu_6502_BRK_imp();
```

```
name
               \overset{\mathrm{adr}}{\mathbf{i}}\mathrm{mp}
                                       logic PC = PC + 1
                         NVBDIZC
0x00
       BRK
                         --1-1--
                                            bPoke(SP,PC.h)
                                            SP = SP - 1
                       bytes 1
example
                               cycles
                                            bPoke(SP,PC.1)
       BRK
                                            SP = SP - 1
bPoke(SP, (P|$10))
description
                                            SP = SP - 1
             interrupt
                                            1 = bPeek($FFFE)
                                            h = bPeek($FFFF) << 8
                                            PC = h | 1
  (S)-=:PC,P PC:=($FFFE)
                                       comment
\stackrel{c-fct}{\text{void cpu\_6502\_BRK\_imp();}}
```

```
name
                                           logic A = A | M
                 \operatorname{adr}
                            NVBDIZC
0x01
        0RA
                  izx
                            *---*-
                                                 P.N = A.7
                                                  P.Z = (A==0) ? 1:0
{\stackrel{\rm example}{{\rm ORA}}} \ (\${\rm A5,X})
                          \quad \text{bytes} \quad
                                   cycles
                                     6
                                           comment
description
      or with accumulator
formal
            A := A \text{ or adr}
 void cpu_6502_ORA_izx();
```

 $_{
m bho1@bfh.ch}$

ор 0х05	name ORA	adr z p		DIZC *-	logic A = A M P.N = A.7
	$ \begin{array}{ccc} \text{example} & \text{bytes} & \text{cycles} \\ \text{ORA $AB} & 2 & 3 \end{array} $			P.Z = (A==0) ? 1:0 comment	
descrip		accur	nulato	r	
formal A:=A or adr					
c-fct void cpu_6502_ORA_zp();				p();	

ор 0х06	name ASL	adr z p		DIZC **	logic P.C = B.7 B = (B << 1) & \$FE
exampl A	\widehat{A} SL $\$AB$ 2 5			P.N = B.7 P.Z = (B==0) ? 1:0	
descrip		netic sh	ift left	;	
formal	ad	r := adr	*2		
c-fct voi	c-fct void cpu_6502_ASL_zp();				

ор 0х08	name PHP	adr imp	NVBDIZC		logic bPoke(SP,P) SP = SP - 1
exampl	$\begin{array}{ccc} \text{example} & \text{bytes} & \text{cycles} \\ \text{PHP} & 1 & 3 \end{array}$				comment
descrip pus	tion h proce	essor s	tatus (SR)	
formal	((S)-:=I)		
c-fct void cpu_6502_PHP_imp();					

op 0x09	orane oran	adr imm		DIZC *-	logic A = A M P.N = A.7
exampl OF	$ \begin{array}{c cccc} \text{example} & \text{bytes} & \text{cycles} \\ \text{ORA} & \#\$\text{AB} & 2 & 2 \\ \end{array} $			P.Z = (A==0) ? 1:0 comment	
	description or with accumulator				
formal	formal A:=A or adr				
<pre>c-fct void cpu_6502_ORA_imm();</pre>				m();	

op 0x0A	name ASL	adr imp		DIZC **	logic P.C = B.7 B = (B << 1) & \$FE
exampl	e ASL		bytes 1	cycles 2	P.N = B.7 P.Z = (B==0) ? 1:0
descrip		etic sh	ift left		
formal	ad	r:=adr	·*2		
c-fct void	cpu_	6502_ <i>I</i>	ASL_im	p();	

op 0x0D	name ORA	adr abs	NVBI	DIZC *-	logic A = A M P.N = A.7
exampl OR	e A \$AB	BCD	bytes cycles 3 4		P.Z = (A==0) ? 1:0 comment
	description or with accumulator				
formal	formal A:=A or adr				
c-fct void cpu_6502_ORA_abs();				s();	

op 0x0E	name ASL	adr abs		DIZC **	logic P.C = B.7 B = (B << 1) & \$FE
exampl AS	e L \$AB	CD	bytes cycles 6		P.N = B.7 P.Z = (B==0) ? 1:0
descrip		etic sh	ift left		
formal	ad	r:=adr	*2		
c-fct void	cpu_	6502_ <i>I</i>	ASL_ab	s();	

ор 0х10	name BPL	adr rel	NVBDIZC		logic	(P.	N ==	0)	GOTO	(PC+M)
exampl B	e PL \$A	В	bytes cycles 2 3							
descrip branc	tion h on p	lus (ne	gative							
formal	formal branch on N=0									
c-fct void cpu_6502_BPL_rel();										

ор 0х11	name ORA	adr izy		DIZC *-	logic A = A M P.N = A.7
exampl OR	TO A (CAE) SZ I ČA I ČE F			P.Z = (A==0) ? 1:0 comment	
	description or with accumulator			r	
formal	A:=A or adr				
c-fct void	<pre>c-fct void cpu_6502_ORA_izy();</pre>				

ор 0х15	name ORA	adr z px		DIZC *-	logic A = A M P.N = A.7
exampl OR	e A \$AI	3,X	bytes cycles 2		P.Z = (A==0) ? 1:0 comment
	description or with accumulator				
formal	formal A:=A or adr				
c-fct void	c-fct void cpu_6502_ORA_zpx();				

ор 0х16	name ASL	adr z px		DIZC **	logic P.C = B.7 B = (B << 1) & \$FE
exampl AS	EL \$AE	3,X	bytes cycles 2 6		P.N = B.7 P.Z = (B==0) ? 1:0
	description arithmetic shift left				
formal	ad	r:=adr	*2		
c-fct void	cpu_	6502_ <i>I</i>	SL_zp	x();	

ор 0х18	name CLC	adr imp	NVBDIZC 0		logic P.C = 0
exampl	$\begin{array}{c c} \text{example} & \text{bytes} \\ \text{CLC} & 1 & 2 \end{array}$				
descrip	description clear carry				
formal		C:=0			
c-fct void	c-fct void cpu_6502_CLC_imp();				

 $_{
m bho1@bfh.ch}$

```
logic A = A | M
      name
              adr
aby
^{\mathrm{op}}
                      NVBDIZC
0x19 | 0RA
                                        P.N = A.7
                                        P.Z = (A==0) ? 1:0
example
ORA $ABCD,Y
                     bytes
                            cycles
                             4
                                   comment
description
     or with accumulator
formal
         A := A \text{ or adr}
c-fct
void cpu_6502_ORA_aby();
```

op 0x1D	name ORA	adr a bx	NVBI	DIZC *-	logic A = A M P.N = A.7
exampl ORA	e \$AB(CD,X	bytes cycles 4		P.Z = (A==0) ? 1:0 comment
	description or with accumulator				
formal	A:=A or adr				
c-fct void	c-fct void cpu_6502_ORA_abx();				

op 0x1E	name ASL	adr abx		DIZC **	logic P.C = B.7 B = (B << 1) & \$FE
exampl ASL	e \$ABC	D,X	bytes 3	cycles 7	P.N = B.7 P.Z = (B==0) ? 1:0
	description arithmetic shift left				Comment
formal	ad	r:=adr	*2		
c-fct void	c-fct void cpu_6502_ASL_abx();				

```
logic t = PC - 1
        name
                \operatorname{adr}
                          NVBDIZC
0x20
        JSR
                 \mathtt{abs}
                                               bPoke(SP,t.h)
                                               SP = SP - 1
                                 cycles 6
_{\rm JSR~\$ABCD}^{\rm example}
                        \overset{\mathrm{bytes}}{3}
                                               bPoke(SP,t.1)
                                               SP = SP - 1
PC = $A5B6
description jump subroutine
                                         comment
formal
       (S)-:=PC PC:=adr
 void cpu_6502_JSR_abs();
```

7 bho1@bfh.ch

ор 0х24	name BIT	adr z p		DIZC	logic t = A & M P.N = t.7 P.V = t.6
exampl B	e IT \$A	·			P.Z = (t==0) ? 1:0
descrip		bit test	t		
formal N:	=b7 V:	=b6 Z	:=A&a	adr	
c-fct voi	c-fct void cpu_6502_BIT_zp();				

ор 0х25	name AND	adr z p		DIZC *-	logic A = A & M P.N = A.7
exampl A		man la la F			P.Z = (A==0) ? 1:0 comment
	description and (with accumulator)			or)	
formal	A	=A&a	dr		
c-fct voi	d cpu_	6502_	AND_zp	o();	

ор 0x26	name ROL	adr z p		DIZC **	logic t = B.7 B = (B << 1) & \$FE
exampl R	$ \begin{array}{ccc} \text{xample} & \text{bytes} & \text{cycles} \\ \text{ROL \$AB} & 2 & 5 \\ \end{array} $			B = B P.C P.C = t P.Z = (B==0) ? 1:0	
descrip	description rotate left				P.N = B.7
formal	adr:	=adr*:	2+C		
c-fct voi	c-fct void cpu_6502_ROL_zp();				

ор 0х28	name PLP	adr imp	NVBI	DIZC ****	logic SP = SP + 1 P = bPeek(SP)
exampl	$\begin{array}{c c} \mathrm{example} & \mathrm{bytes} & \mathrm{cycles} \\ \mathrm{PLP} & 1 & 4 \end{array}$				comment
	description pull processor status (SR)			SR)	
formal	F	P:=+(S	5)		
c-fct void	c-fct void cpu_6502_PLP_imp();				

ор 0х29	name AND	adr imm	NVBDIZC **-		logic A = A & M P.N = A.7
	VD #\$.	AB	bytes cycles 2		P.Z = (A==0) ? 1:0
descrip	description and (with accumulator)			or)	
formal	formal A:=A&adr				
c-fct void	c-fct void cpu_6502_AND_imm();				

op 0x2A	name ROL	adr imp		DIZC **	logic t = B.7 B = (B << 1) & \$FE
exampl	ROL		bytes 1	cycles 2	B = B P.C P.C = t P.Z = (B==0) ? 1:0 P.N = B.7
descrip		otate le	eft		comment
formal	formal adr:=adr*2+C				
c-fct void	cpu_	6502_F	ROL_im	p();	

op 0x2C	name BIT	adr abs		DIZC *-	logic t = A & M P.N = t.7
exampl BI	е Г \$АВ	CD	bytes 3	cycles 4	P.V = t.6 P.Z = (t==0) ? 1:0
descrip	description bit test				comment
formal N:	formal N:=b7 V:=b6 Z:=A&adr				
c-fct void	c-fct void cpu_6502_BIT_abs();				

op 0x2E	name ROL	adr abs	*	DIZC **	logic t = B.7 B = (B << 1) & \$FE B = B P.C
exampl RO descrip	L \$AB	CD	bytes 3	cycles 6	P.C = t P.Z = (B==0) ? 1:0 P.N = B.7
formal		otate le	eft		comment
	adr:=adr*2+C				
<pre>c-fct void cpu_6502_ROL_abs();</pre>				s();	

ор 0х30	name BMI	adr rel	NVBI	NVBDIZC		if ent	(P.N	[==	1)	GOTO	(PC+M)
exampl B	e MI \$A	В	bytes 2								
descrip branc	tion h on m	ninus (1	negativ	re set)							
formal	formal branch on N=1										
c-fct void	c-fct void cpu_6502_BMI_rel();										

ор 0х31	name AND	adr izy		DIZC *-	logic A = A & M P.N = A.7
exampl AN	ė (\$A5	5),X	bytes cycles 2 5		P.Z = (A==0) ? 1:0 comment
descrip	tion id (wit	h accu	cumulator)		
formal	formal A:=A&adr				
c-fct void	c-fct void cpu_6502_AND_izy();				

```
logic A = A & M
        name
                \mathop{\mathtt{zpx}}^{\mathrm{adr}}
^{\mathrm{op}}
                           NVBDIZC
0x35 AND
                                                P.N = A.7
                                                P.Z = (A==0) ? 1:0
example
AND $AB,X
                                 cycles
                                   4
                                          comment
description and (with accumulator)
formal
            A := A \& adr
\overline{\mathrm{c-fct}}
 void cpu_6502_AND_zpx();
```

```
logic t = B.7
             adr
zpx
op
      name
                      NVBDIZC
0x36 ROL
                                       B = (B << 1) & FE
                                       B = B | P.C
example
ROL $AB,X
                    \overset{\text{bytes}}{2}
                           cycles
                                       P.C = t
                             6
                                       P.Z = (B==0) ? 1:0
                                       P.N = B.7
description
           rotate left
                                  comment
formal
        adr := adr *2 + C
c-fct
void cpu_6502_ROL_zpx();
```

ор 0х38	name SEC	adr imp		DIZC 1	logic P.C = 1
exampl	e SEC		bytes 1	cycles 2	
descrip	description set carry				
formal		C:=1			
c-fct void	c-fct void cpu_6502_SEC_imp();				

ор 0х39	name AND	adr aby		DIZC *-	logic A = A & M P.N = A.7
exampl AND	°\$ABC	CD,Y	bytes 3	cycles 4	P.Z = (A==0) ? 1:0
descrip		h accu	mulato	or)	
formal	formal A:=A&adr				
c-fct void	c-fct void cpu_6502_AND_aby();				

```
logic t = B.7
op
       name
                 \operatorname{adr}
                           NVBDIZC
0x3E ROL
                 \mathbf{a} \mathbf{b} \mathbf{x}
                                                 B = (B << 1) & FE
                                                 B = B \mid P.C
                                  cycles 7
                         \overset{\mathrm{bytes}}{3}
_{\rm ROL~\$ABCD,X}^{\rm example}
                                                 P.C = t
                                                 P.Z = (B==0) ? 1:0
                                                P.N = B.7
description
             rotate left
                                           comment
formal
          adr := adr *2 + C
c-fct
void cpu_6502_ROL_abx();
```

ор 0х40	name RTI	adr imp		DIZC ****	logic SP = SP - 1 P = bPeek(SP) SP = SP - 1
exampl	e RTI		bytes 1	cycles 6	1 = bPeek(SP) SP = SP - 1
	lescription return from interrupt			t	h = bPeek(SP) << 8 PC = h 1 comment
formal	P,PC:=+(S)				
c-fct void	cpu_	6502_F	RTI_im	p();	

ор 0х41	name EOR	adr izx		DIZC *-	logic A = A ^ M P.N = A.7
exampl EO	e R (\$A5	(\$A5,X) bytes cycles 6			P.Z = (A==0) ? 1:0 comment
descrip	escription clusive or (with accumulator)			ılator)	
formal	formal A:=A exor adr				
c-fct void	c-fct void cpu_6502_EOR_izx();				

ор 0х45	EOR	adr z p		DIZC *-	logic A = A ^ M P.N = A.7
examp E	OR \$A	В	bytes cycles 2 3		P.Z = (A==0) ? 1:0
descri exclu	description exclusive or (with accumulator)		ılator)		
forma	formal A:=A exor adr				
c-fct voi	c-fct void cpu_6502_EOR_zp();			o();	

ор 0х46	name LSR	adr z p		DIZC **	logic P.N = 0 P.C = B.0
exampl L	e SR \$A	В	bytes cycles 2 5		B = (B >> 1) & \$7F P.Z = (B==0) ? 1:0
descrip	tion logical shift right			comment	
adr:=adr/2					
c-fct void cpu_6502_LSR_zp();				o();	

ор 0х48	name PHA	adr imp	NVBDIZC		logic bPoke(SP,A) SP = SP - 1
exampl	e PHA		bytes 1	cycles 3	comment
descrip	description push accumulator				
formal	((S)-:= A	A		
c-fct void	cpu_	6502_F	PHA_im	p();	

ор 0х49	name EOR	adr imm		DIZC *-	logic A = A ~ M P.N = A.7
exampl EC	$\stackrel{\text{example}}{\text{EOR}} \#\$ AB \qquad \stackrel{\text{bytes}}{2} \stackrel{\text{cycles}}{2}$			P.Z = (A==0) ? 1:0 comment	
descrip exclus	description exclusive or (with accumulator)			ılator)	
formal	formal A:=A exor adr				
c-fct void	c-fct void cpu_6502_EOR_imm();			m();	

op 0x4A	name LSR	adr imp		DIZC **	logic P.N = 0 P.C = B.O
exampl	e LSR		bytes cycles 1 2		B = (B >> 1) & \$7F P.Z = (B==0) ? 1:0
descrip	description logical shift right				
formal	ad	r:=adr	-/2		
c-fct void	cpu_	6502_I	SR_im	p();	

op 0x4C	name JMP	adr abs	NVBDIZC		logic PC = M
exampl JM	e P \$AB	CD	bytes cycles 3		
descrip	tion	jump			
formal	F	C:=ac	lr		
c-fct void cpu_6502_JMP_abs();					

op 0x4D	name EOR	adr abs		DIZC *-	logic A = A ~ M P.N = A.7
exampl EO	$\begin{array}{c c} \text{ample} \\ \text{EOR $ABCD} & \begin{array}{c} \text{bytes} \\ 3 \end{array} & \begin{array}{c} \text{cycles} \\ 4 \end{array} \end{array}$			P.Z = (A==0) ? 1:0	
descrip exclus	description exclusive or (with accumulator)			ılator)	
formal	A:=	A exor	adr		
c-fct void	cpu_	6502_E	EOR_ab	s();	

op 0x4E	name LSR	adr abs		DIZC **	logic P.N = 0 P.C = B.O
exampl LSI	e R \$AB	CD	bytes cycles 6		B = (B >> 1) & \$7F P.Z = (B==0) ? 1:0
descrip	description logical shift right				
formal	formal adr:=adr/2				
c-fct void	cpu_	6502_I	SR_ab	s();	

ор 0x50	name BVC	^{adr} rel	NVBDIZC		logic	if	(P.V	==	0)	GOTO	(PC+M)
					comn	nent					
exampl B	e VC \$A	.В	bytes cycles 2 3								
descrip br		n overi	low cle								
branch on V=0											
c-fct void cpu_6502_BVC_rel();											

ор 0х51	name EOR	adr izy		DIZC *-	logic A = A ~ M P.N = A.7
exampl EO	e (\$A5	5),X	bytes cycles 5		P.Z = (A==0) ? 1:0
	description exclusive or (with accumulator)				
formal	A :=	A exor	adr		
c-fct void	<pre>c-fct void cpu_6502_EOR_izy();</pre>				

ор 0х55	name EOR	adr z px		DIZC *-	logic A = A ^ M P.N = A.7
exampl EC	e R \$AI	3,X	bytes cycles 2 4		P.Z = (A==0) ? 1:0 comment
	description exclusive or (with accumulator)			ılator)	
formal	A:=	A exor	adr		
c-fct void	l cpu_	6502_E	EOR_zp	x();	

ор 0х56	name LSR	adr z px		DIZC **	logic P.N = 0 P.C = B.O
exampl LS	e R \$AE	3,X	bytes cycles 2 6		B = (B >> 1) & \$7F P.Z = (B==0) ? 1:0
descrip	description logical shift right				
formal	formal adr:=adr/2				
c-fct void	cpu_	6502_I	SR_zp	x();	

ор 0х58	name CLI	adr imp	NVBDIZC		logic P.I = 0
exampl	e CLI		bytes cycles 1 2		comment
	description clear interrupt disable			le	
formal I:=0					
<pre>c-fct void cpu_6502_CLI_imp();</pre>					

ор 0х59	name EOR	adr a by		DIZC *-	logic A = A ~ M P.N = A.7
exampl EOR	$ \overset{\text{nple}}{\text{DR \$ABCD,Y}} \overset{\text{bytes}}{3} \overset{\text{cycles}}{4} $			P.Z = (A==0) ? 1:0 comment	
	description exclusive or (with accumulator)				
formal	A:=	A exor	adr		
c-fct void	cpu_	6502_E	EOR_ab	y();	

^{ор} 0х5D	name EOR	adr a bx		DIZC *-	logic A = A ~ M P.N = A.7
exampl EOR	s \$ABC	CD,X	bytes cycles 3		P.Z = (A==0) ? 1:0 comment
	description exclusive or (with accumulator)			ılator)	
formal	A:=	A exor	adr		
c-fct void cpu_6502_EOR_abx();					

op 0x5E	name LSR	adr abx		DIZC **	logic P.N = 0 P.C = B.O
exampl LSR	e\$ABC	D,X	bytes 3	cycles 7	B = (B >> 1) & \$7F P.Z = (B==0) ? 1:0
descrip	description logical shift right				
formal	$^{ m formal}$ ${ m adr}:={ m adr}/2$				
c-fct void	cpu_	6502_I	.SR_ab	x();	

ор 0х60	name RTS	adr imp	NVBI	DIZC	logic SP = SP + 1 1 = bPeek(SP)
exampl	eRTS		bytes 1	cycles 6	SP = SP + 1 h = bPeek(SP)<<8 PC = (h 1) +1
	description return from subroutine			ne	comment
formal	PC:=+(S)				
c-fct void cpu_6502_RTS_imp();					

```
logic t = A + M + P.C
      name
op
             \operatorname{adr}
                     NVBDIZC
0x61 ADC
             izx
                                      P.V = (A.7!=t.7) ? 1:0
                                      P.N = A.7
^{\rm example}_{\rm ADC}~(\$A5,\!X)
                           cycles
                    bytes
                                      P.Z = (t==0) ? 1:0
                                      IF (P.D)
                                        t = bcd(A) + bcd(M) + P.C
description
                                        P.C = (t>99) ? 1:0
        add with carry
                                        P.C = (t>255) ? 1:0
formal
                                      A = t \& 0xFF
          A := A + adr
c-fct
void cpu_6502_ADC_izx();
```

```
\frac{1}{\text{logic}} \text{ t = A + M + P.C}
^{\mathrm{op}}
       name
               adr
                         NVBDIZC
0x65 ADC
                                            P.V = (A.7!=t.7) ? 1:0
                                            P.N = A.7
                               cycles
\mathop{\mathrm{ADC}}^{\mathrm{example}} \mathrm{AB}
                       bytes
                                            P.Z = (t==0) ? 1:0
                                            IF (P.D)
                                              t = bcd(A) + bcd(M) + P.C
\operatorname{description}
                                               P.C = (t>99) ? 1:0
         add with carry
                                            ELSE
                                              P.C = (t>255) ? 1:0
formal
                                            A = t & 0xFF
           A := A + adr
                                       comment
c-fct
 void cpu_6502_ADC_zp();
```

```
\frac{1}{\text{logic}} t = B.0
       name
               \operatorname{adr}
                        NVBDIZC
0x66 ROR
                                            B = (B >> 1) & $7F
                                            B = B \mid ((P.C) ? \$80:\$00)
                               cycles
                       \overset{\text{bytes}}{2}
example
                                            P.C = t
    ROR $AB
                                5
                                            P.Z = (B==0) ? 1:0
                                            P.N = B.7
\operatorname{description}
          rotate right
                                      comment
formal
      adr:=adr/2+C*128
 void cpu_6502_ROR_zp();
```

```
name
               \displaystyle \mathop{\mathrm{imp}}^{\mathrm{adr}}
                                       logic SP = SP + 1
^{\mathrm{op}}
                         NVBDIZC
0x68 PLA
                                             A = bPeek(SP)
                                             P.N = A.7
example
                       bytes
                               cycles
                                             P.Z = (A==0) ? 1:0
       PLA
                         1
                                       comment
description
       pull accumulator
formal
             A := +(S)
c-fct
 void cpu_6502_PLA_imp();
```

```
\frac{\text{logic}}{\text{t}} = A + M + P.C
op
       name
               adr
                         NVBDIZC
0x69 ADC imm
                                             P.V = (A.7!=t.7) ? 1:0
                                             P.N = A.7
                               cycles 2
\overset{\mathrm{example}}{\mathrm{ADC}} \ \#\$\mathrm{AB}
                       \overset{\text{bytes}}{2}
                                             P.Z = (t==0) ? 1:0
                                             IF (P.D)
                                               t = bcd(A) + bcd(M) + P.C
description
                                               P.C = (t>99) ? 1:0
         add with carry
                                             ELSE
                                               P.C = (t>255) ? 1:0
formal
                                             A = t & 0xFF
            A{:=}A{+}adr
                                       \operatorname{comment}
c-fct
void cpu_6502_ADC_imm();
```

op 0x6A	name ROR	adr imp		DIZC **	logic t = B.0 B = (B >> 1) & \$7F B = B ((P.C) ? \$80:\$00)
exampl	ROR		bytes 1	cycles 2	P.C = t P.Z = (B==0) ? 1:0 P.N = B.7
descrip		tate rig	ght		comment
formal	adr:=a	adr/2+	C*128	1	
c-fct void cpu_6502_ROR_imp();					

op 0x6C	name JMP	adr ind	NVBDIZC		logic PC = M
	$\begin{array}{c c} \text{example} & \text{bytes} \\ \text{JMP \$ABCD} & 3 & 5 \end{array}$				
descrip	lescription jump				
formal	F	C:=ac	lr		
c-fct void	cpu_	6502_J	JMP_in	d();	

```
name adr
                              logic t = A + M + P.C
op
                   NVBDIZC
0x6D ADC abs
                                  P.V = (A.7!=t.7) ? 1:0
                                  P.N = A.7
                        cycles
                                  P.Z = (t==0) ? 1:0
 ADC $ABCD
                                   IF (P.D)
                                    t = bcd(A) + bcd(M) + P.C
description
                                    P.C = (t>99) ? 1:0
       add with carry
                                    P.C = (t>255) ? 1:0
formal
                                   A = t & 0xFF
         A := A + adr
                              comment
_{\mathrm{c-fct}}
void cpu_6502_ADC_abs();
```

```
op
       name
              adr
                                    logic t = B.0
                       NVBDIZC
0x6E ROR
              \mathtt{abs}
                                         B = (B >> 1) & $7F
                                         B = B \mid ((P.C) ? \$80:\$00)
                            cycles
\mathop{\rm ROR}\limits_{\rm ROR} \$ {\rm ABCD}
                     bytes
                                         P.C = t
                              6
                                         P.Z = (B==0) ? 1:0
                                         P.N = B.7
description
         rotate right
                                    comment
formal
      adr = adr/2 + C*128
void cpu_6502_ROR_abs();
```

ор 0х70	name BVS	adr rel	NVBDIZC		logic		(P.	٧	==	1)	GOTO	(PC+M)
exampl B	e VS \$A	В	bytes cycles 2									
descrip b		on over	rflow se	flow set								
formal	branch on V=1											
c-fct void cpu_6502_BVS_rel();												

```
_{\mathrm{name}}
              \operatorname{adr}
                                    logic t = A + M + P.C
                       NVBDIZC
0x71 ADC
                                         P.V = (A.7!=t.7) ? 1:0
                                         P.N = A.7
                     \overset{\text{bytes}}{2}
                             cycles
_{\rm ADC~(\$A5),X}^{\rm example}
                                         P.Z = (t==0) ? 1:0
                                         IF (P.D)
                                          t = bcd(A) + bcd(M) + P.C
description
                                           P.C = (t>99) ? 1:0
        add with carry
                                         ELSE
                                           P.C = (t>255) ? 1:0
formal
                                         A = t & 0xFF
           A := A + adr
                                    comment
void cpu_6502_ADC_izy();
```

 $19 \\ \hspace{3cm} {\tt bho1@bfh.ch}$

```
\mathop{\mathtt{zpx}}^{\mathrm{adr}}
                                    \frac{1}{\log ic} t = A + M + P.C
op
      name
                       NVBDIZC
0x75 \mid ADC
                                         P.V = (A.7!=t.7) ? 1:0
                                         P.N = A.7
^{\rm example}_{\rm ADC~\$AB,X}
                             cycles
                                         P.Z = (t==0) ? 1:0
                                         IF (P.D)
                                           t = bcd(A) + bcd(M) + P.C
description
                                           P.C = (t>99) ? 1:0
        add with carry
                                           P.C = (t>255) ? 1:0
formal
                                          A = t & 0xFF
           A := A + adr
                                    comment
c-fct
void cpu_6502_ADC_zpx();
```

```
name
^{\mathrm{op}}
                \operatorname{adr}
                                        logic t = B.0
                          NVBDIZC
                \mathbf{z} \mathbf{p} \mathbf{x}
0x76 \mid ROR
                                              B = (B >> 1) & $7F
                                              B = B \mid ((P.C) ? \$80:\$00)
                                cycles
_{\rm ROR~\$AB,X}^{\rm example}
                        bytes
                                              P.C = t
                                              P.Z = (B==0) ? 1:0
                                              P.N = B.7
description
           rotate right
                                        comment
formal
      adr = adr/2 + C*128
 void cpu_6502_ROR_zpx();
```

ор 0х78	name SEI	adr imp	NVBDIZC		logic P.I = 1
exampl	e SEI		bytes cycles 1 2		
descrip	description set interrupt disable)	
formal		I:=1			
c-fct void	cpu_	6502_S	SEI_im	p();	

```
_{\mathrm{name}}
             adr
                                  logic t = A + M + P.C
                      NVBDIZC
0x79 ADC
              aby
                                       P.V = (A.7!=t.7) ? 1:0
                                       P.N = A.7
                           cycles
_{\rm ADC~\$ABCD,Y}^{\rm example}
                    \overset{\text{bytes}}{3}
                                       P.Z = (t==0) ? 1:0
                                       IF (P.D)
                                        t = bcd(A) + bcd(M) + P.C
description
                                         P.C = (t>99) ? 1:0
        add with carry
                                       ELSE
                                         P.C = (t>255) ? 1:0
formal
                                       A = t & 0xFF
          A := A + adr
                                  comment
void cpu_6502_ADC_aby();
```

 ${\tt 20} \\$

```
name adr
                                          \frac{\text{logic}}{\text{t}} = A + M + P.C
^{\mathrm{op}}
                           NVBDIZC
0x7D \mid ADC \mid abx
                                                 P.V = (A.7!=t.7) ? 1:0
                                                 P.N = A.7

P.Z = (t==0) ? 1:0
^{\rm example}_{\rm ADC}\, {\rm \$ABCD, X}
                                  cycles
                                                 IF (P.D)
                                                   t = bcd(A) + bcd(M) + P.C
P.C = (t>99) ? 1:0
{\it description}
         add with carry
                                                  P.C = (t>255) ? 1:0
formal
                                                 A = t & 0xFF
            A := A + adr
                                           \operatorname{comment}
_{\mathrm{c-fct}}
void cpu_6502_ADC_abx();
```

```
logic t = B.0
op
     name
             \operatorname{adr}
                    NVBDIZC
0x7E ROR
             abx
                                     B = (B >> 1) & $7F
                                     B = B \mid ((P.C) ? \$80:\$00)
                         cycles 7
                   bytes 3
ROR $ABCD,X
                                     P.C = t
                                     P.Z = (B==0) ? 1:0
                                     P.N = B.7
description
        rotate right
                                comment
formal
     adr = adr/2 + C*128
c-fct
void cpu_6502_ROR_abx();
```

ор 0x81	name STA	adr izx	NVBDIZC		logic M = A
exampl ST	e (\$A5	5,X)	bytes 2	cycles 6	
descrip	description store accumulator				
formal	i	adr:=A	1		
c-fct void cpu_6502_STA_izx();					

ор 0x84	name STY	adr z p	NVBDIZC		logic M = Y
	$ \begin{array}{c c} \text{example} & \text{bytes} \\ \text{STY $AB} & 2 & 3 \end{array} $				
descrip	description store Y				
formal	$\operatorname{adr} := Y$				
c-fct voi	d cpu_	6502_	STY_zp	o();	

ор 0x85	name STA	adr z p	NVBDIZC		logic M = A
					comment
exampl S'	$ \begin{array}{c c} \text{example} & \text{bytes} & \text{cycles} \\ \text{STA $AB} & 2 & 3 \\ \end{array} $				
descrip	description store accumulator				
formal	formal adr:=A				
c-fct void cpu_6502_STA_zp();					

ор 0x86	name STX	adr z p	NVBDIZC		logic M = X
exampl S'	$ \begin{array}{c c} \text{example} & \text{bytes} \\ \text{STX \$AB} & 2 & 3 \end{array} $				
descrip	description store X				
formal	á	adr:=X	[
c-fct voi	d cpu_	6502_	STX_zŗ	p();	

88	name DEY	adr imp	*	NVBDIZC **-	
exampl	DEY		bytes 1	cycles 2	
descrip		remen	t Y		
formal	Y:=Y-1				
c-fct void	c-fct void cpu_6502_DEY_imp();				

op 0x8A	name TXA	adr imp	NVBI	DIZC *-	logic A = X P.N = A.7
exampl	e TXA		bytes 1	cycles 2	P.Z = (A==0) ? 1:0 comment
	description transfer X to accumulator			ator	
formal	formal A:=X				
c-fct void	cpu_	6502_1	TXA_im	p();	

 ${\small 22} \\$ bho1@bfh.ch

ор 0x8С	name STY	adr abs	NVBI	DIZC	logic M = Y
					comment
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$, ·			
descrip	description store Y				
formal	i	adr:=Y	7		
c-fct void	<pre>c-fct void cpu_6502_STY_abs();</pre>				

op 0x8D	name STA	adr abs	NVBI	DIZC	logic M = A
exampl ST	$\begin{array}{c c} \text{example} & \text{bytes} \\ \text{STA \$ABCD} & 3 & 4 \end{array}$		l * 4		
descrip	description store accumulator				
formal	;	adr:=A	L		
<pre>c-fct void cpu_6502_STA_abs();</pre>				s();	

op 0x8E	name STX	adr abs	NVBI	DIZC	logic M = X
	$ \begin{array}{ccc} \text{xample} & \text{bytes} \\ \text{STX $ABCD} & 3 & 4 \end{array} $				
descrip		store X	X.		
$\operatorname*{adr}:=X$					
c-fct void cpu_6502_STX_abs();					

ор 0х90	name BCC	adr rel	NVBDIZC		logic		(P.	C ==	0)	GOTO	(PC+M)
exampl B	e CC \$A	В	bytes cycles 2 3								
descrip		on car	ry clea	ry clear							
formal	branch on C=0										
c-fct void	c-fct void cpu_6502_BCC_rel();										

 $23 \\ \hspace{3.5cm} \texttt{bho1@bfh.ch}$

ор 0х91	name STA	adr izy	NVBDIZC		logic M = A
					comment
example STA (\$A5),X bytes cycles 6					
descrip	description store accumulator				
formal	ŧ	adr:=A	L		
c-fct void	<pre>c-fct void cpu_6502_STA_izy();</pre>				

ор 0х94	name STY	adr z px	NVBDIZC		$\frac{\text{logic}}{M} = Y$
exampl ST	$ \begin{array}{c c} {\rm example} \\ {\rm STY~\$AB,X} \end{array} \begin{array}{c c} {\rm bytes} \\ 2 \end{array} \begin{array}{c} {\rm cycles} \\ 4 \end{array} $				
descrip	description store Y				
formal	;	adr:=Y	7		
c-fct void	<pre>c-fct void cpu_6502_STY_zpx();</pre>				

ор 0х95	name STA	adr z px	NVBDIZC		logic M = A
exampl ST	$\begin{array}{c c} \text{example} & \text{bytes} & \text{cycles} \\ \text{STA $AB,X} & 2 & 4 \end{array}$				
descrip	description store accumulator				
formal	ŧ	adr:=A	1		
c-fct void	c-fct void cpu_6502_STA_zpx();				

ор 0х96	name STX	adr z py	NVBDIZC		logic M = X comment
exampl ST	$ \begin{array}{c cccc} \text{example} & \text{bytes} & \text{cycles} \\ \text{STX $AB,X} & 2 & 4 \\ \end{array} $				
descrip	description store X				
formal	ŧ	adr:=X	[
c-fct void	<pre>c-fct void cpu_6502_STX_zpy();</pre>				

 $24 \\ \hspace{3.5cm} \texttt{bho1@bfh.ch}$

ор 0х98	name TYA	adr imp	NVBDIZC **-		hogiment
$ \begin{array}{c c} \text{example} & \text{bytes} & \text{cycles} \\ \text{TYA} & 1 & 2 \end{array} $					
	description transfer Y to accumulator			itor	
formal	formal A:=Y				
c-fct void cpu_6502_TYA_imp();					

ор 0х99	name STA	adr aby	NVBDIZC		logic M = A
exampl STA	example STA \$ABCD,Y 3 cycles 5				
descrip	description store accumulator				
formal	ŧ	adr:=A	L		
c-fct void cpu_6502_STA_aby();				y();	

ор 0х9А	name TXS	adr imp	NVBI	DIZC	logic SP = X
exampl	$ \begin{array}{c c} \text{example} & \text{bytes} & \text{cycles} \\ \text{TXS} & 1 & 2 \end{array} $				
descrip tran	description transfer X to stack pointer			nter	
formal		S:=X			
c-fct void	<pre>c-fct void cpu_6502_TXS_imp();</pre>				

op 0x9D	name STA	adr a bx	NVBDIZC		logic M = A
exampl STA	example STA \$ABCD,X bytes 3 5				
descrip	description store accumulator				
formal	ŧ	adr:=A	L		
c-fct void	<pre>c-fct void cpu_6502_STA_abx();</pre>				

 ${}^{\rm bho1@bfh.ch}$

ор 0 хА0	name LDY	adr imm	NVBDIZC **-		logic Y = M P.N = Y.7
exampl LI	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			P.Z = (Y==0) ? 1:0	
descrip	description load Y				
formal	formal Y:=adr				
c-fct void	c-fct void cpu_6502_LDY_imm();				

op 0xA1	name LDA	adr izx		DIZC *-	logic A = M P.N = A.7 P.Z = (A==0) ? 1:0
exampl LD.	$\stackrel{\text{ample}}{\text{LDA}}$ (\$A5,X) $\begin{vmatrix} \text{bytes} \\ 2 \end{vmatrix} \stackrel{\text{cycles}}{6}$			comment	
descrip	description load accumulator				
formal	formal A:=adr				
c-fct void	c-fct void cpu_6502_LDA_izx();				

op 0xA2	name LDX	^{adr} imm	NVBDIZC **-		logic X = M P.N = X.7 P.Z = (X==0) ? 1:0
exampl LI				comment (X==0) ! 1:0	
descrip	description				
formal					
c-fct void	c-fct void cpu_6502_LDX_imm();				

op 0xA4	name LDY	adr z p		DIZC *-	$ \begin{array}{rcl} \operatorname{logic} & Y &= & M \\ P & N &= & Y &. 7 \end{array} $
	$ \begin{array}{c cccc} \text{example} & \text{bytes} & \text{cycles} \\ \text{LDY $AB} & 2 & 3 \\ \end{array} $			P.Z = (Y==0) ? 1:0	
descrip	description load Y				
formal	-	Y:=ad:	r		
c-fct voi	c-fct void cpu_6502_LDY_zp();				

 26

ор 0хА5	name LDA	adr z p		DIZC *-	logic A = M P.N = A.7
	$ \begin{array}{c cccc} \text{example} & \text{bytes} & \text{cycles} \\ \text{LDA $AB} & 2 & 3 \\ \end{array} $			P.Z = (A==0) ? 1:0 comment	
descrip	description load accumulator				
formal A:=adr					
<pre>c-fct void cpu_6502_LDA_zp();</pre>					

op 0xA6	name LDX	adr z p	NVBDIZC **-		$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
exampl L	e DX \$A	В	bytes cycles 2 3		P.Z = (X==0) ? 1:0
descrip	description				
formal	-	X:=adı	r		
<pre>c-fct void cpu_6502_LDX_zp();</pre>				o();	

op 0xA8	name TAY	adr imp		DIZC *-	$ \begin{array}{rcl} \log & Y &= & A \\ P & N &= & Y &. 7 \end{array} $
exampl	$\begin{array}{c c} \text{example} & \text{bytes} & \text{cycles} \\ \text{TAY} & 1 & 2 \end{array}$			P.Z = (Y==0) ? 1:0	
	description transfer accumulator to Y			o Y	
formal		Y:=A			
c-fct void	cpu_	6502_1	TAY_im	p();	

op 0xA9	name LDA	^{adr} imm	NVBDIZC **-		logic A = M P.N = A.7
exampl LL	$\stackrel{\text{example}}{\text{LDA}} \#\$ \text{AB} \qquad \stackrel{\text{bytes}}{2} \stackrel{\text{cycles}}{2}$			P.Z = (A==0) ? 1:0	
descrip	description load accumulator				
formal	formal A:=adr				
c-fct void	c-fct void cpu_6502_LDA_imm();				

 ${\tt 27} \\$ {\tt bho1@bfh.ch}

op 0xAA	name TAX	adr imp		DIZC *-	$ \begin{array}{cccc} \log ic & X &= A \\ P \cdot N &= X \cdot 7 \end{array} $
exampl	TEAST L'A L'A F			P.Z = (X==0) ? 1:0	
	$ \begin{array}{c} {\rm description} \\ {\rm transfer~accumulator~to~X} \end{array} $			o X	
formal		X:=A			
c-fct void cpu_6502_TAX_imp();					

op 0xAC	name LDY	adr abs	NVBDIZC **-		logic Y = M P.N = Y.7
exampl LD				P.Z = (Y==0) ? 1:0 comment	
descrip	description load Y				
formal					
c-fct void	c-fct void cpu_6502_LDY_abs();				

op 0xAD	name LDA	adr abs		DIZC *-	logic A = M P.N = A.7
exampl LD.				P.Z = (A==0) ? 1:0	
descrip		accum	ılator		
formal	_	A:=ad:	r		
c-fct void cpu_6502_LDA_abs();				s();	

op 0xAE	LDX	adr abs		DIZC *-	$\begin{array}{c} \text{logic} X = M \\ P.N = X.7 \end{array}$
	$ \begin{array}{c cccc} \text{example} & \text{bytes} & \text{cycles} \\ \text{LDX \$ABCD} & 3 & 4 \\ \end{array} $			P.Z = (X==0) ? 1:0	
descrip	description				
formal					
c-fct void	<pre>c-fct void cpu_6502_LDX_abs();</pre>				

 28

op 0xB0	name BCS	adr rel	NVBDIZC		logic i:	(P.	C =	= :	1)	GOTO	(PC+M)
exampl B	e CS \$A	В	bytes cycles 2		Comme						
descrip	description branch on carry set										
formal branch on C=1											
<pre>c-fct void cpu_6502_BCS_rel();</pre>											

^{ор} 0хВ1	name LDA	adr izy		DIZC *-	logic A = M P.N = A.7
exampl LD	$ \begin{array}{c cccc} xample & bytes & cycles \\ LDA (\$A5), X & 2 & 5 \\ \end{array} $			P.Z = (A==0) ? 1:0 comment	
descrip	description load accumulator				
formal	formal A:=adr				
c-fct void	c-fct void cpu_6502_LDA_izy();				

^{ор} 0хВ4	name LDY	adr z px		DIZC	logic Y = M P.N = Y.7 P.Z = (Y==0) ? 1:0
exampl LD	Y \$AE	3,X	$\begin{bmatrix} \text{bytes} & \text{cycles} \\ 2 & 4 \end{bmatrix}$		comment
description load Y					
formal Y:=adr					
c-fct void cpu_6502_LDY_zpx();				x();	

op 0xB5	name LDA	adr z px	NVBDIZC **-		logic A = M P.N = A.7
exampl LD	e A \$AE	\$AB,X bytes cycles 4			P.Z = (A==0) ? 1:0
descrip	description load accumulator				
formal	_	A:=adi	r		
c-fct void	cpu_	6502_I	.DA_zp	x();	

 $_{\rm bho1@bfh.ch}$

op 0xB6	name LDX	adr z py		DIZC *-	$\begin{array}{ccc} \log ic & X &= & M \\ & P \cdot N &= & X \cdot 7 \end{array}$
exampl LD	$\stackrel{ ext{table}}{\text{DX $AB,X}} \stackrel{ ext{bytes}}{2} \stackrel{ ext{cycles}}{4}$		· .	P.Z = (X==0) ? 1:0 comment	
descrip	description				
formal					
c-fct void	<pre>c-fct void cpu_6502_LDX_zpy();</pre>				

ор 0хВ8	name CLV	adr imp	NVBI	DIZC	$\begin{array}{ll} \text{logic} P \cdot V = 0 \\ \\ \text{comment} \end{array}$
exampl	e CLV		bytes 1	cycles 2	
descrip	description clear overflow				
formal		V:=0			
c-fct void	c-fct void cpu_6502_CLV_imp();				

^{ор} 0хВ9	name LDA	adr aby		DIZC	logic A = M P.N = A.7 P.Z = (A==0) ? 1:0
exampl LDA	*\$ABC	CD,Y	bytes 3	cycles 4	comment
descrip	description load accumulator				
formal	_	A := adi	r		
c-fct void cpu_6502_LDA_aby();					

op 0xBA	name TSX	adr imp		DIZC *-	$\begin{array}{ccc} \log ic & X & = & SP \\ & P \cdot N & = & X \cdot 7 \end{array}$
exampl	eTSX	X bytes cycles 1 2			P.Z = (X==0) ? 1:0 comment
	description transfer stack pointer to X			o X	
formal		X:=S			
c-fct void	cpu_0	6502_T	SX_im	p();	

op name LDY	aui		DIZC *-	logic Y = M P.N = Y.7
example LDY \$AE	CD,X bytes cycles 4			P.Z = (Y==0) ? 1:0
description	load Y	7		
formal	Y:=ad	r		
<pre>c-fct void cpu_6502_LDY_abx();</pre>				

op 0xBD	name LDA	adr a bx	NVBI		logic A = M P.N = A.7
exampl LDA	e \$ABC	CD,X	bytes cycles 3 4		P.Z = (A==0) ? 1:0 comment
descrip	description load accumulator				
formal	formal A:=adr				
c-fct void	c-fct void cpu_6502_LDA_abx();				

op 0xBE	name LDX	adr aby		DIZC *-	logic X = M P.N = X.7
exampl LDX	$\begin{array}{c cccc} xample & bytes & cycles \\ LDX \$ABCD, Y & 3 & 4 & \end{array}$			P.Z = (X==0) ? 1:0	
descrip	description				
formal					
c-fct void	c-fct void cpu_6502_LDX_aby();				

op 0xC0	name CPY	^{adr} imm		DIZC **	logic t = Y - M P.N = t.7
exampl CF	e Y #\$.	$\frac{\text{bytes}}{2}$ $\frac{\text{cycles}}{2}$			P.C = (Y>=M) ? 1:0 P.Z = (t==0) ? 1:0
description compare with Y			ith Y		
formal	formal Y-adr				
c-fct void	c-fct void cpu_6502_CPY_imm();				

```
logic t = A - M
       name
op
                adr
                          NVBDIZC
0xC1 CMP izx
                                              P.N = t.7

P.C = (A>=M) ? 1:0

P.Z = (t==0) ? 1:0
example CMP ($A5,X)
                        bytes 2
                                cycles
                                 6
                                        comment
description compare (with accumulator)
formal
               A-adr
_{
m c-fct}^{
m c-fct} void cpu_6502_CMP_izx();
```

op 0xC4	name CPY	adr z p		DIZC **	logic t = Y - M P.N = t.7
exampl C	e PY \$A	В.	bytes 2	cycles 3	P.C = (Y>=M) ? 1:0 P.Z = (t==0) ? 1:0
descrip	description compare with Y				comment
formal		Y-adr			
c-fct voi	c-fct void cpu_6502_CPY_zp();				

op 0xC5	name CMP	adr z p		DIZC **	logic t = A - M P.N = t.7
exampl C	e MP \$A	ъВ	bytes 2	cycles 3	P.C = (A>=M) ? 1:0 P.Z = (t==0) ? 1:0
	description compare (with accumulator)			ator)	Comment
formal	formal A-adr				
c-fct voi	c-fct void cpu_6502_CMP_zp();				

op 0xC6	name DEC	adr z p		DIZC *-	logic M = (M - 1) & \$FF P.N = M.7
exampl D	e EC \$A	.B	bytes cycles 2 5		P.Z = (M==0) ? 1:0
descrip	decrement				
formal	formal adr:=adr-1				
c-fct voi	c-fct void cpu_6502_DEC_zp();				

op 0xC8	name INY	adr imp		DIZC	logic Y = Y + 1 P.Z = (Y==0) ? 1:0
exampl	e INY		bytes 1	cycles 2	P.N = Y.7 comment
descrip	description increment Y				
formal Y:=Y+1					
<pre>c-fct void cpu_6502_INY_imp();</pre>					

op 0xC9	name CMP	^{adr} imm		DIZC **	logic t = A - M P.N = t.7
exampl CN	$\begin{array}{c} \text{example} \\ \text{CMP } \#\$\text{AB} \\ \end{array} \begin{array}{c} \text{bytes} \\ 2 \\ \end{array}$		cycles 2	P.C = $(A >= M)$? 1:0 P.Z = $(t == 0)$? 1:0	
	description compare (with accumulator)			ator)	comment
formal	formal A-adr				
c-fct void cpu_6502_CMP_imm();				m();	

op 0xCA	name DEX	adr imp		DIZC *-	logic X = X - 1 P.Z = (X==0) ? 1:0
exampl	DEX bytes cycles 2			P.N = X.7	
descrip	description decrement X				
formal	2	X:=X-	1		
c-fct void	cpu_	6502_I	EX_im	p();	

op 0xCC	name CPY	adr a bs		DIZC **	logic t = Y - M P.N = t.7
example CP	e Y \$AB	CD	bytes 3	cycles 4	P.C = (Y>=M) ? 1:0 P.Z = (t==0) ? 1:0
descript	description compare with Y				
formal	formal Y-adr				
c-fct void	c-fct void cpu_6502_CPY_abs();				

op 0xCD	name CMP	adr abs		DIZC **	logic t = A - M P.N = t.7
	example CMP \$ABCD bytes 3		cycles 4	P.C = (A >= M) ? 1:0 P.Z = (t==0) ? 1:0	
					comment
	description compare (with accumulator)		ator)		
formal	formal A-adr				
c-fct void	<pre>c-fct void cpu_6502_CMP_abs();</pre>				

op 0xCE	name DEC	adr abs		DIZC *-	logic M = (M - 1) & \$FF P.N = M.7
exampl DE	e C \$AB	CD	bytes 3	cycles 6	P.Z = (M==0) ? 1:0
descrip	description decrement				
formal	formal adr:=adr-1				
c-fct void	<pre>c-fct void cpu_6502_DEC_abs();</pre>				

op 0xD0	name BNE	adr rel	NVBDIZC		logic		(P.	Z =	= 0)	GOTO	(PC+M)	
exampl B	$\begin{array}{c cccc} \text{Ample} & \text{bytes} & \text{cycles} \\ \text{BNE $AB} & 2 & 3 \end{array}$											
descrip branch		ot equa	l (zero	l (zero clear)								
formal	formal branch on Z=0											
c-fct void cpu_6502_BNE_rel();												

op 0xD1	name CMP	adr izy		DIZC **	logic t = A - M P.N = t.7
exampl CM	e P (\$A	5),X	bytes 2	cycles 5	P.C = (A>=M) ? 1:0 P.Z = (t==0) ? 1:0 comment
descrip COM]	description compare (with accumulator)			ator)	comment
formal	formal A-adr				
c-fct void cpu_6502_CMP_izy();				y();	

ор 0хD6	name DEC	adr z px		DIZC *-	logic M = (M - 1) & \$FF P.N = M.7
exampl DE	e C \$AE	3,X	bytes 2	cycles 6	P.Z = (M==0) ? 1:0 comment
descrip	description decrement				
formal	formal adr:=adr-1				
<pre>c-fct void cpu_6502_DEC_zpx();</pre>				x();	

op 0xD8	name CLD	adr imp		DIZC D	logic P.D = 0
exampl	e CLD		bytes 1	cycles 2	
descrip		ar deci	mal		
formal		D:=0			
c-fct void	cpu_	6502_0	CLD_im	p();	

op 0xD9	name CMP	adr aby		DIZC **	logic t = A - M P.N = t.7
exampl CMF	$\begin{array}{c c} \text{example} \\ \text{CMP $\$ABCD,Y} & 3 & 4 \\ \end{array}$				P.C = (A>=M) ? 1:0 P.Z = (t==0) ? 1:0
descrip COM]	tion pare (v	vith ac	cumula	ator)	
formal		A-adr			
c-fct void	cpu_	6502_0	MP_ab	y();	

 $_{\rm bho1@bfh.ch}$

```
name
^{\mathrm{op}}
               adr
                                       _{
m logic} t = A - M
                         NVBDIZC
0xDD CMP
                \mathtt{a}\mathrm{b}\mathrm{x}
                          *---**
                                             P.N = t.7
                                             P.C = (A>=M) ? 1:0
\stackrel{\rm example}{\rm CMP~\$ABCD,X}
                        bytes
                               cycles
                                             P.Z = (t==0) ? 1:0
                                        comment
description
 compare (with accumulator)
formal
               A-adr
c-fct
 void cpu_6502_CMP_abx();
```

```
logic M = (M - 1) & $FF
op
        name
                 adr
                           NVBDIZC
0xDE DEC
                 abx
                                                P.N = M.7
                                                P.Z = (M==0) ? 1:0
                                  cycles 7
_{\mathrm{DEC}}^{\mathrm{example}}\$\mathrm{ABCD,X}
                         \overset{\text{bytes}}{3}
                                           comment
description
             decrement
formal
             adr:=adr-1
\mathrm{c\text{--}fct}
 void cpu_6502_DEC_abx();
```

op 0xE0	name CPX	adr imm		DIZC **	logic t = X - M P.N = t.7
exampl CF	$\begin{array}{c c} \text{example} & \text{bytes} \\ \text{CPX } \#\$\text{AB} & 2 & 2 \\ \end{array}$				P.C = (X>=M) ? 1:0 P.Z = (t==0) ? 1:0
descrip		oare wi	th X		
formal		X-adr			
c-fct void	cpu_	6502_0	PX_im	m();	

```
logic IF (P.D)
op
       _{\mathrm{name}}
               \operatorname{adr}
                        NVBDIZC
0xE1 SBC
                izx
                                             t = bcd(A) - bcd(M) - !P.C
                                              P.V = (t>99 \ OR \ t<0) ? 1:0
                               cycles
\overset{\mathrm{example}}{\mathrm{SBC}}\ (\$\mathrm{A5,X})
                       \overset{\text{bytes}}{2}
                                            ELSE
                                 6
                                               t = A - M - !P.C
                                              P.V = (t>127 OR t<-128) ? 1:0
description subtract with carry
                                            P.C = (t>=0) ? 1:0
                                            P.N = t.7
                                            P.Z = (t==0) ? 1:0
formal
                                            A = t & 0xFF
            A := A - adr
                                      comment
 void cpu_6502_SBC_izx();
```

```
logic t = X - M
       name
               adr
zp
^{\mathrm{op}}
                         NVBDIZC
0xE4 CPX
                                            P.N = t.7
                                            P.C = (X>=M) ? 1:0

P.Z = (t==0) ? 1:0
example
CPX $AB
                               cycles
                                       \operatorname{comment}
description
        compare with X
formal
               X-adr
c-fct
 void cpu_6502_CPX_zp();
```

```
logic IF (P.D)
op
       name
               adr
                        NVBDIZC
0xE5 SBC
                zp
                                             t = bcd(A) - bcd(M) - !P.C
P.V = (t>99 OR t<0) ? 1:0
example
SBC $AB
                              cycles 3
                      \overset{\text{bytes}}{2}
                                           ELSE
                                             t = A - M - !P.C
                                             P.V = (t>127 OR t<-128) ? 1:0
description
                                           P.C = (t>=0) ? 1:0
     subtract with carry
                                           P.N = t.7
                                           P.Z = (t==0) ? 1:0
formal
                                           A = t \& 0xFF
            A := A - adr
                                      \operatorname{comment}
\mathrm{c\text{--}fct}
 void cpu_6502_SBC_zp();
```

op 0xE6	name INC	adr z p		DIZC *-	logic M = (M + 1) & \$FF P.N = M.7
exampl II	e NC \$A	В	bytes 2	cycles 5	P.Z = (M==0) ? 1:0
descrip		creme	nt		
formal	ad	r:=adr	+1		
c-fct voi	d cpu_	6502_	INC_zp	o();	

op 0xE8	name INX	adr imp	NVBI	DIZC *-	logic X = X + 1 P.Z = (X==0) ? 1:0
exampl	eINX		bytes 1	cycles 2	P.N = X.7 comment
descrip		remen	t X		
formal	У	X:=X+	1		
c-fct void	cpu_	6502_1	NX_im	p();	

```
^{\mathrm{op}}
      name
              \operatorname{adr}
                                    ^{
m logic} IF (P.D)
                       NVBDIZC
0xE9 SBC imm
                                           t = bcd(A) - bcd(M) - !P.C
                                            P.V = (t>99 OR t<0) ? 1:0
\overset{\text{example}}{\text{SBC}} \#\$ AB
                             cycles
                                          ELSE
                                           t = A - M - !P.C
                                           P.V = (t>127 OR t<-128) ? 1:0
description
                                          P.C = (t>=0) ? 1:0
     subtract with carry
                                          P.N = t.7
                                          P.Z = (t==0) ? 1:0
formal
                                          A = t & 0xFF
           A := A - adr
                                    \operatorname{comment}
c-fct
void cpu_6502_SBC_imm();
```

op 0xEA	name NOP	adr imp	NVBDIZC		logic ~none~
exampl	e NOP		bytes 1	cycles 2	
descrip		operat	ion	1	
formal					
c-fct void	cpu_	6502_N	IOP_im	p();	

op 0xEC	name CPX	adr abs		DIZC **	logic t = X - M P.N = t.7
exampl CP:	e X \$AB	BCD	bytes 3	cycles 4	P.C = (X>=M) ? 1:0 P.Z = (t==0) ? 1:0
descrip		pare wi	ith X		
formal		X-adr			
c-fct void	cpu_	6502_0	CPX_ab	s();	

```
logic IF (P.D)
        name
                adr
                          NVBDIZC
0xED SBC
                 abs
                                               t = bcd(A) - bcd(M) - !P.C
P.V = (t>99 OR t<0) ? 1:0
                        \overset{\text{bytes}}{3}
                                cycles
{\stackrel{\rm example}{\rm SBC}~\$ABCD}
                                              ELSE
                                                t = A - M - !P.C
P.V = (t>127 OR t<-128) ? 1:0
description subtract with carry
                                              P.C = (t>=0) ? 1:0
                                              P.N = t.7
                                              P.Z = (t==0) ? 1:0
formal
                                              A = t & 0xFF
             A := A - adr
                                        comment
 void cpu_6502_SBC_abs();
```

op 0xEE	name INC	adr abs		DIZC	logic M = (M + 1) & \$FF P.N = M.7
exampl IN0	e \$AB	CD	bytes 3	cycles 6	P.Z = (M==0) ? 1:0
descrip		creme	nt		
formal	ad	r:=adr	+1		
c-fct void	cpu_	6502_1	INC_ab	s();	

op 0xF0	name BEQ	adr rel	NVBDIZC		logic		(P.2	Z ==	1)	GOTO	(PC+M)
exampl B	$\begin{array}{c cccc} \text{ample} & \text{bytes} & \text{cycles} \\ \text{BEQ \$AB} & 2 & 2 \\ \end{array}$										
descrip bra	tion nch on										
formal	bran										
c-fct void	cpu_										

```
logic IF (P.D)
             adr
izy
^{\mathrm{op}}
      name
                     NVBDIZC
0xF1 SBC
                                        t = bcd(A) - bcd(M) - !P.C
                                         P.V = (t>99 \ OR \ t<0) ? 1:0
SBC ($A5),X
                    \overset{\mathrm{bytes}}{2}
                           cycles
                                       ELSE
                                        t = A - M - !P.C
                                        P.V = (t>127 OR t<-128) ? 1:0
description subtract with carry
                                       P.C = (t>=0) ? 1:0
                                       P.N = t.7
                                      P.Z = (t==0) ? 1:0
formal
                                       A = t & 0xFF
           A := A - adr
                                  comment
_{\rm c-fct}
void cpu_6502_SBC_izy();
```

```
adr
zpx
                                       logic IF (P.D)
^{\mathrm{op}}
        _{\mathrm{name}}
                         NVBDIZC
0xF5 SBC
                                              t = bcd(A) - bcd(M) - !P.C
                                               P.V = (t>99 \ OR \ t<0) ? 1:0
                               cycles
                       \overset{\mathrm{bytes}}{2}
_{\mathrm{SBC}~\$\mathrm{AB,X}}^{\mathrm{example}}
                                             ELSE
                                               t = A - M - !P.C
                                               P.V = (t>127 OR t<-128) ? 1:0
description subtract with carry
                                             P.C = (t>=0) ? 1:0
                                             P.N = t.7

P.Z = (t==0) ? 1:0
formal
                                             A = t & 0xFF
             A := A - adr
                                       comment
c-fct
 void cpu_6502_SBC_zpx();
```

```
\mathop{\mathtt{zpx}}^{\mathrm{adr}}
                                       logic M = (M + 1) & $FF
op
       name
                         NVBDIZC
0xF6 INC
                                            P.N = M.7
                                            P.Z = (M==0) ? 1:0
_{\rm INC~\$AB,X}^{\rm example}
                       bytes
                               cycles
                                 6
                                       comment
description
            increment
formal
           adr = adr + 1
c-fct
 void cpu_6502_INC_zpx();
```

op 0xF8	name SED	adr imp	NVBDIZC		logic P.D = 1
exampl	e SED		bytes 1	cycles 2	
descrip		t decin	nal		
formal		D:=1			
c-fct void	cpu_	6502_S	SED_im	p();	

```
^{\mathrm{op}}
      name
              adr
                                   ^{
m logic} IF (P.D)
                      NVBDIZC
0xF9 SBC
              aby
                                         t = bcd(A) - bcd(M) - !P.C
                                          P.V = (t>99 \ OR \ t<0) ? 1:0
                            cycles
_{\mathrm{SBC}~\$\mathrm{ABCD,Y}}^{\mathrm{example}}
                     bytes
                                        ELSE
                                          t = A - M - !P.C
                                          P.V = (t>127 OR t<-128) ? 1:0
description
                                        P.C = (t>=0) ? 1:0
     subtract with carry
                                        P.N = t.7
                                        P.Z = (t==0) ? 1:0
formal
                                        A = t & 0xFF
           A := A - adr
                                   comment
c-fct
void cpu_6502_SBC_aby();
```

```
logic IF (P.D)
^{\mathrm{op}}
      name
             adr
                     NVBDIZC
0xFD SBC
             abx
                                        t = bcd(A) - bcd(M) - !P.C
                                        P.V = (t>99 \ OR \ t<0) ? 1:0
                    bytes 3
                          cycles
_{\rm SBC~\$ABCD,X}^{\rm example}
                                      ELSE
                                        t = A - M - !P.C
                                        P.V = (t>127 OR t<-128) ? 1:0
description subtract with carry
                                      P.C = (t>=0) ? 1:0
                                      P.N = t.7
                                      P.Z = (t==0) ? 1:0
formal
                                      A = t & 0xFF
          A := A - adr
                                 comment
 void cpu_6502_SBC_abx();
```

op 0xFE	name INC	adr abx		DIZC *-	logic M = (M + 1) & \$FF P.N = M.7
exampl INC	e \$ABC	D,X	bytes 3	cycles 7	P.Z = (M==0) ? 1:0
descrip		creme	nt		
formal	ad	r:=adr	+1		
c-fct void	cpu_	6502_1	NC_ab	x();	

	example	BRK										ORA (\$A5,X)				ORA \$AB		
	logic		bPoke (SP, PC.h)	SP = SP - 1	bPoke(SP, PC.1)	SP = SP - 1	bPoke(SP, (P \\$10))	SP = SP - 1	1 = bPeek(\$FFFE)	h = bPeek(\$FFFF) < < 8	PC = h l		P.N = A.7	P.Z = (A==0) ? 1:0			P.N = A.7	P.Z = (A==0) ? 1:0
		NVBDIZC										NVBDIZC	* * *			NVBDIZC **-		
Text1	function	interrupt	PC:=(\$FFFE)									or with accumu-	$\begin{array}{c} \text{lator} \\ \text{A:=A or adr} \end{array}$		•	or with accumu-	A:=A or adr	
Text1 Text1	adr"-mode											izx				$^{ m ZD}$		
Text1	cycles	0										9			,	n		
Text3	bytes	1										2			,	21		
Text2	mnmonic	BRK										ORA				ORA		
Text1	opcode	0x00										0x01			1	0x05		

Table 2: Caption