

ADRESOVÉ REŽIMY		délka instrukce celkem
#n	přímá hodnota (immediate)	2 B
addr8	nultá stránka 0x00xx	2 B
addr16	absolutní adresa	3 B
addr16,X	absolutní + X	3 B
addr16,Y	absolutní + Y	3 B
(addr8)	nepřímý — ukazatel v ZP	2 B
addr8,X	ZP + X (wrapuje v ZP)	2 B
(addr8),Y	ukazatel v ZP, pak + Y	2 B
Implied instrukce mají 1 B; skoky a CALL addr16 3 B; JMP/CALL (zp) 2 B.		

SYSTÉM	
NOP	nedělá nic
HLT	zastav — probudí až přerušení
OUT	výstupní registr := A
CLC	C := 0
SEC	C := 1
EI	povol přerušení
DI	zakaž přerušení
RND	A := náhodný bajt

ZÁSOBNÍK		SP je 16bitový, roste dolů
PUSH	ulož A na zásobník	—
POP	A := vrchol zásobníku	Z N
PHX / PLX	ulož / obnov X	PLX: Z N
PHY / PLY	ulož / obnov Y	PLY: Z N
PHB / PLB	ulož / obnov B (bez zničení A)	PLB: Z N
PHF	ulož bajt příznaků	—
PLF	obnov bajt příznaků	vše

16BIT UKAZATEL	
LXY #VRAM — SXY PTR	
.l: ... — ADW PTR, #40 — JMP .l	
LXY+SXY naplní ukazatel, ADW ho krokuje o řádek. Word + word stále ADD/ADC po bajtech.	

PŘESUNY DAT		M = operand dle režimu
LDA	A := M	Z N
LDX	X := M	Z N
LDY	Y := M	Z N
STA	M := A	—
STX	M := X	—
STY	M := Y	—
CPX	porovnej X s M (X – M)	C Z N
CPY	porovnej Y s M (Y – M)	C Z N

REGISTROVÉ PŘESUNY		
TAX / TXA	$A \rightarrow X / X \rightarrow A$	Z N
TAY / TYA	$A \rightarrow Y / Y \rightarrow A$	Z N
TAB / TBA	$A \rightarrow B / B \rightarrow A$	Z N
TXYS	SP := X:Y (X horní bajt)	—
TSXY	X:Y := SP	—
LXY #n16	X:Y := 16bit konstanta (X horní)	—
SXY zp	word na zp := X:Y (LE)	—
INX / DEX	$X := X \pm 1$	Z N
INY / DEY	$Y := Y \pm 1$	Z N

KTERÉ REŽIMY INSTRUKCE BEROU	
	# zp abs a,X a,Y (zp) zp,X (zp),Y
LDA	• • • • • • •
LDX	• • • • • • •
LDY	• • • • • • •
STA	• • • • • • •
STX	• • • • • • •
STY	• • • • • • •
CPX	• • • • • • •
CPY	• • • • • • •
BIT	• • • • • • •
ALU	• • • • • • •
ALU = všech osm operací ADD...CMP bere všech osm režimů.	

OTOČENÍ ZNAMÉNKA (16BIT)	
LDA #0 — SUB X_LO — STA X_LO	
LDA #0 — SBC X_HI — STA X_HI	
Dvojkový doplněk = 0 – x. Takhle míček v Arkanoidu mění směr letu.	

ALU S PAMĚTÍ		výsledek do A
ADD	A := A + M	C Z N V
ADC	A := A + M + C (16bit řetězení)	C Z N V
SUB	A := A – M (C = výpůjčka)	C Z N V
SBC	A := A – M – C	C Z N V
AND	A := A & M	Z N
OR	A := A M	Z N
XOR	A := A ^ M	Z N
CMP	porovnej A s M (A – M)	C Z N
BIT	test A & M – A přežije (jen #, zp, abs)	Z N
Pozor: každá z těchto operací si natáhne M do B — výsledek MUL/DIV v B vyzvedni (TBA) před další ALU operaci, nebo B zachraň PHB/PLB!		

AKUMULÁTOR (IMPLIED)		
INC / DEC	A := A ± 1	Z N
NOT	A := ~A (bitová negace)	Z N
SHL	posun vlevo, C := bit 7	C Z N
SHR	posun vpravo, C := bit 0	C Z N
ROL	rotace vlevo přes C	C Z N
ROR	rotace vpravo přes C	C Z N
MUL	A × B → A dolní, B horní bajt	C Z N
DIV	A ÷ B → A podíl, B zbytek	C Z N
MUL: C = horní bajt ≠ 0. DIV nulou: C = 1, A i B beze změny (bez pasti).		

PAMĚŤOVÉ RMW		zp (2 B) nebo abs (3 B)
INC / DEC m	buňka := buňka ± 1	Z N
INW / DEW m	16bit ukazatel v paměti ± 1	žádné
ADW / SBW zp,#i	16bit word ± 8bit konstanta (B := i)	žádné
SHL / SHR m	posun buňky přímo v paměti	C Z N
ROL / ROR m	rotace buňky přes C	C Z N
INW/DEW/ADW/SBW nemění příznaky — bezpečné uprostřed testů. Vícebajtové posuny řetěz přes ROL/ROR.		

SMYČKA PŘES POLE	
LDY #0	
.l: LDA (PTR),Y — INY	
CPY #N — JC .l	
CPY drží C = 1, dokud Y < N — JC je podmínka „ještě ne“.	

PODMÍNĚNÉ SKOKY		vždy addr16, 3 B
JZ / JNZ	skoč když Z = 1 / Z = 0	test Z
JC / JNC	skoč když C = 1 / C = 0	test C
JN / JP	skoč když N = 1 / N = 0	test N
JV / JNV	skoč když V = 1 / V = 0	test V
Aliasy a složené skoky po porovnání (reg – M):		
JEQ = JZ	rovná se	
JNE = JNZ	nerovná se	
JLT = JC	reg < M (výpůjčka nastala)	
JGE = JNC	reg ≥ M	
JGT / JLE	reg > M / reg ≤ M (bezznam.)	CvZ
JGES / JLTS	≥ / < znaménkově (po SUB)	NvV

ŘÍZENÍ TOKU		
JMP addr16	PC := adresa	—
JMP (zp)	skok přes ukazatel — skokové tabulky	—
CALL addr16	ulož návrat na zásobník, skoč	—
CALL (zp)	nepřímé volání přes ukazatel	—
RET	návrat z podprogramu	—
RTI	návrat z přerušení	vše
BRK	softwarové přerušení → vektor 0xFFFF	—

VEKTORY · MAPA · PASTI	
0xFFFFC	vektor RESET (start programu)
0xFFFFE	vektor IRQ (hardwarové přerušení)
0xFFFFA	vektor BRK (softwarové přerušení)
0x0000	nultá stránka (rychlé adresování, ukazatele)
0x0100	stránka zásobníku
0x0200	výchozí začátek programu (.ORG)
0x8000	video paměť (znaky · glyfy · barvy · sprity)
0xFF00	MMIO zařízení (klávesnice, zvuk, obraz, ...)
Opcody 0x00 a 0xFF jsou schválně pasti (trap); 0xFD/0xFE jsou interní pseudo-opcody.	

MUL/DIV: VYZVEDNI B HNED	
LDA Y — TAB — LDA #40 — MUL	
STA L0 — TBA — ADD #0x80 ...	
Horní bajt/zbytek čeká v B — TBA před další ALU operací, jinak o něj přijdeš.	