Most instructions that explicitly reference memory locations have bit patterns of the form **aaabbbcc**. The **aaa** and **cc** bits determine the opcode, and the **bbb** bits determine the addressing mode.

Instructions with **cc** = **01** are the most regular, and are therefore considered first. The **aaa** bits determine the opcode as follows:

aaa	opcode
000	ORA
001	AND
010	EOR
011	ADC
100	STA
101	LDA
110	CMP
111	SBC

And the addressing mode (**bbb**) bits:

bbb	addressing mode
000	(zero page,X)
001	zero page
010	#immediate
011	absolute
100	(zero page),Y
101	zero page,X
110	absolute,Y
111	absolute,X

Putting it all together:

	ORA	AND	EOR	ADC	STA	LDA	CMP	SBC
(zp,X)	01	21	41	61	81	A1	C1	E1
zp	05	25	45	65	85	A5	C5	E5
#	09	29	49	69		A9	C9	E9
abs	0D	2D	4D	6D	8D	AD	CD	ED
(zp),Y	11	31	51	71	91	B1	D1	F1
zp,X	15	35	55	75	95	B5	D5	F5
abs,Y	19	39	59	79	99	B9	D9	F9
abs,X	1D	3D	5D	7D	9D	BD	DD	FD

The only irregularity is the absence of the nonsensical immediate STA instruction.

Next we consider the **cc** = **10** instructions. These have a completely different set of opcodes:

aaa	opcode
000	ASL
001	ROL
010	LSR
011	ROR
100	STX
101	LDX
110	DEC
111	INC

The addressing modes are similar to the **01** case, but not quite the same:

bbb	addressing mode
000	#immediate
001	zero page
010	accumulator
011	absolute
101	zero page,X
111	absolute,X

Note that **bbb** = **100** and **110** are missing. Also, with STX and LDX, "zero page,X" addressing becomes "zero page,Y", and with LDX, "absolute,X" becomes "absolute,Y".

These fit together like this:

	ASL	ROL	LSR	ROR	STX	LDX	DEC	INC
#						A2		
zp	06	26	46	66	86	A6	C6	E6
A	0A	2A	4A	6A				
abs	0E	2E	4E	6E	8E	AE	CE	EE
zp,X/zp,Y	16	36	56	76	96	B6	D6	F6
abs,X/abs,Y	1E	3E	5E	7E		BE	DE	FE

Most of the gaps in this table are easy to understand. Immediate mode makes no sense for any instruction other than LDX, and accumulator mode for DEC and INC didn't appear until the 65C02. The slots that "STX A" and "LDX A" would occupy are taken by TXA and TAX respectively, which is exactly what one would expect. The only inexplicable gap is the absence of a "STX abs,Y" instruction.

Next, the **cc** = **00** instructions. Again, the opcodes are different:

aaa	opcode
001	BIT
010	JMP
011	JMP (abs)
100	STY
101	LDY
110	CPY
111	CPX

It's debatable whether the JMP instructions belong in this list...I've included them because they *do* seem to fit, provided one considers the indirect JMP a separate opcode rather than a different addressing mode of the absolute JMP.

The addressing modes are the same as the **10** case, except that accumulator mode is missing.

bbb	addressing mode
000	#immediate
001	zero page
011	absolute
101	zero page,X
111	absolute,X

And here's how they fit together:

	BIT	JMP	JMP()	STY	LDY	CPY	CPX
#					A0	C0	E0
zp	24			84	A4	C4	E4
abs	2C	4C	6C	8C	AC	CC	EC
zp,X				94	B4		
abs,X					BC		

Some of the gaps in this table are understandable (e.g. the lack of an immediate mode for JMP, JMP(), and STY), but others are not (e.g. the absence of "zp,X" for CPY and CPX, and the absence of "abs,X" for STY, CPY, and CPX). Note that if accumulator mode (**bbb** = **010**) were available, "LDY A" would be A8, which falls in the slot occupied by TAY, but the pattern breaks down elsewhere--TYA is 98, rather than 88, which we would expect it to be if it corresponded to the nonexistant "STY A".

No instructions have the form **aaabbb11**.

The conditional branch instructions all have the form **xxy10000**. The flag indicated by **xx** is compared with **y**, and the branch is taken if they are equal.

xx	flag
00	negative
01	overflow
10	carry
11	zero

This gives the following branches:

BPL	BMI	BVC	BVS	BCC	BCS	BNE	BEQ
10	30	50	70	90	B0	D0	F0

The remaining instructions are probably best considered simply by listing them. Here are the interrupt and subroutine instructions:

BRK	JSR abs	RTI	RTS
00	20	40	60

(JSR is the only absolute-addressing instruction that doesn't fit the **aaabbbcc** pattern.)

Other single-byte instructions:

PHP	PLP	PHA	PLA	DEY	TAY	INY	INX
08	28	48	68	88	A8	C8	E8

CLC	SEC	CLI	SEI	TYA	CLV	CLD	SED
18	38	58	78	98	B8	D8	F8

TXA	TXS	TAX	TSX	DEX	NOP
8A	9A	AA	BA	CA	EA