COMMODORE SEMICONDUCTOR GROUP
a division of Commodore Business Machines, Inc.
950 Rittenhouse Road, Norristown, PA 19403 • 215/666-7950 • TWX 510-660-4168

NMOS

6509 MICROPROCESSOR WITH MEMORY MANAGEMENT

DESCRIPTION

The 6509 is a low-cost microprocessor capable of solving a broad range of small-systems and memory management problems at minimum cost to the user.

A memory management system allows for up to One Mega-Byte of memory for ease in down loading languages, operating systems or other data.

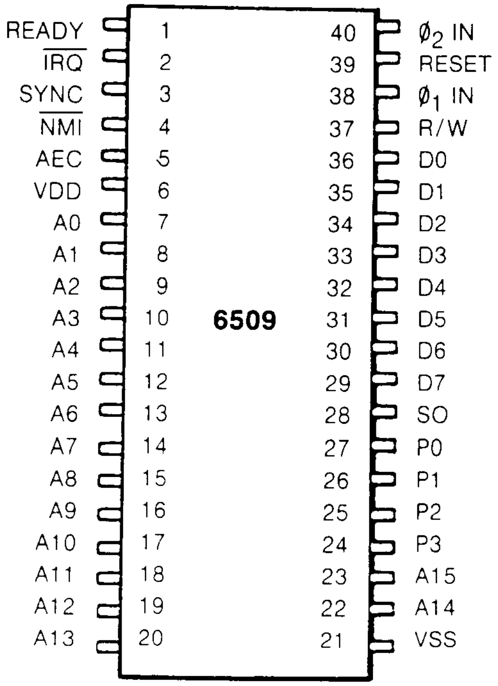
The Three-State sixteen-bit Address Bus allows Direct Memory Accessing (DMA) and multi-processor systems sharing a common memory while the Four-bit Extended Address Register allows for up to one Mega-byte of data storage.

The internal processor architecture is identical to the Commodore Semiconductor Group 6502 to provide software compatibility.

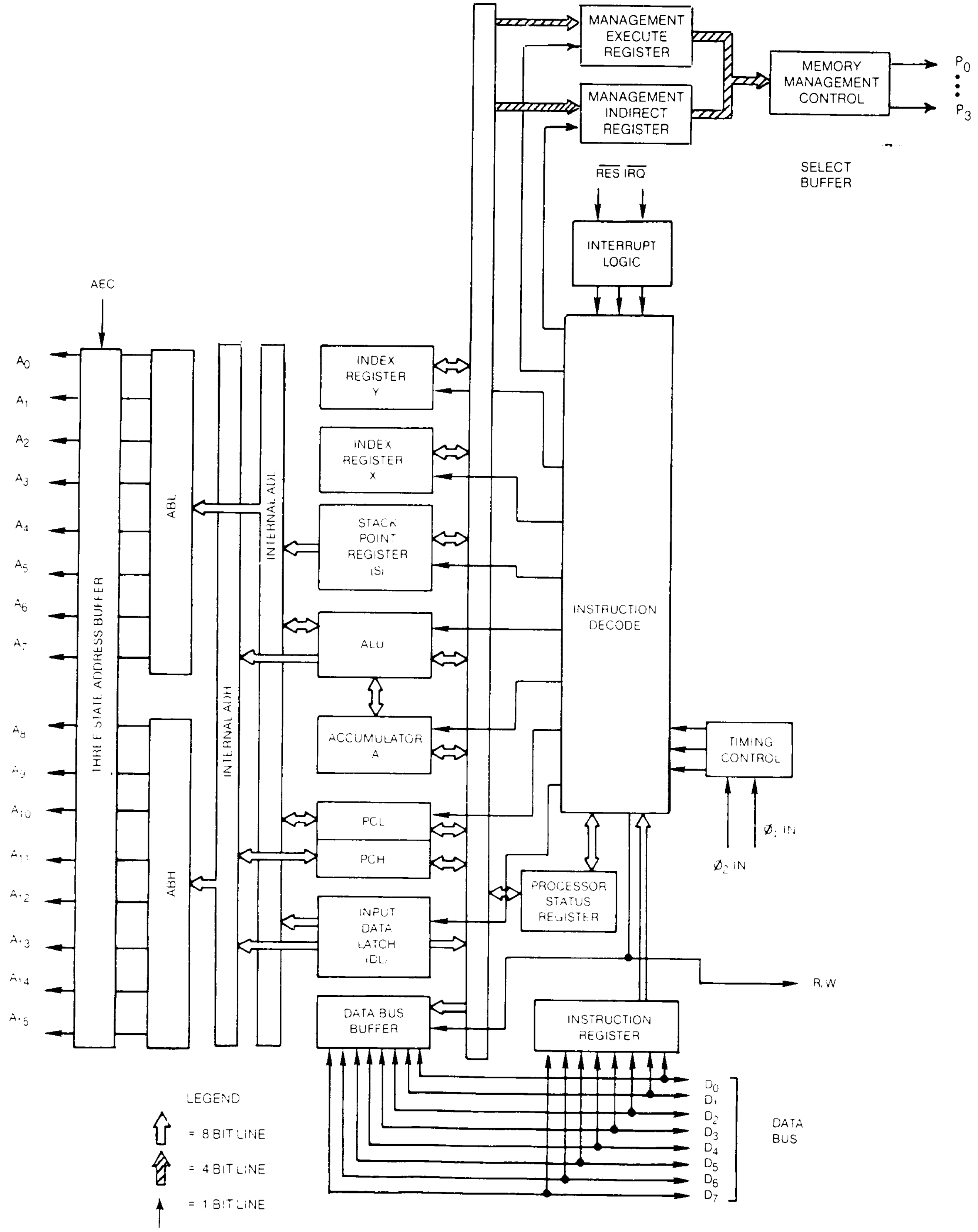
FEATURES OF THE 6509

- Memory management
- On board clock logic
- Addressable memory range of up to 1 M bytes
- Single +5 volt supply
- N channel, silicon gate, depletion load technology
- Eight bit parallel processing
- 56 Instructions
- Decimal and binary arithmetic
- Thirteen addressing modes
- True indexing capability
- Programmable stack pointer
- Variable length stack
- Interrupt capability
- 8 Bit Bi-Directional Data Bus
- Program Addressable memory range of up to 65K bytes
- Direct memory access capability
- Bus compatible with M6800
- Pipeline architecture
- 1 MHz, 2 MHz, and 3 MHz operation
- Use with any type or speed memory

PIN CONFIGURATION



10/86



6509 BLOCK DIAGRAM

6509 CHARACTERISTICS

MAXIMUM RATINGS

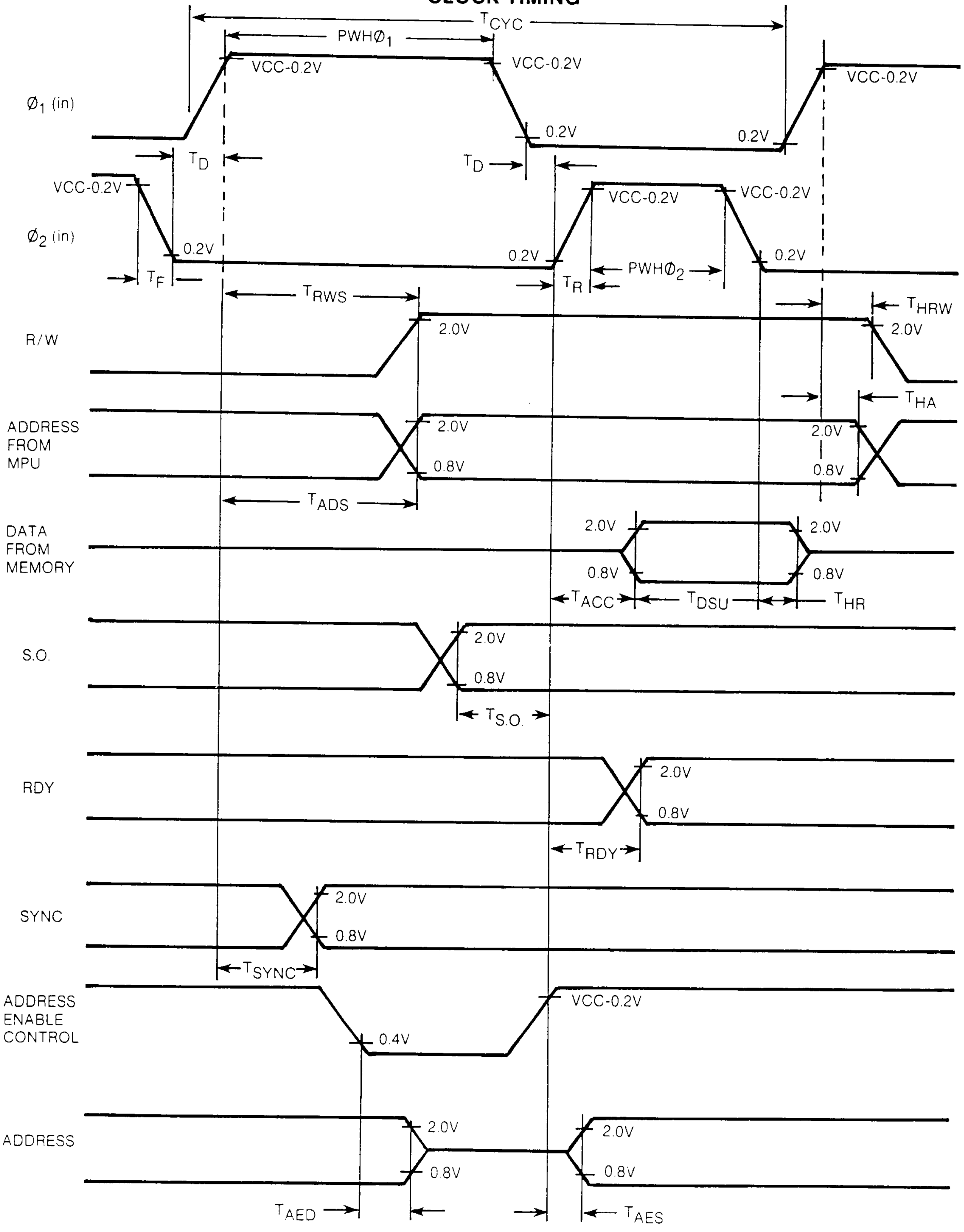
RATING	SYMBOL	VALUE	UNIT
SUPPLY VOLTAGE	V_{CC}	−0.3 to + 7.0	Vdc
INPUT VOLTAGE	Vin	−0.3 to + 7.0	Vdc
OPERATING TEMPERATURE	T_A	0 to + 70	C
STORAGE TEMPERATURE	T_{STG}	−55 to + 150	C

This device contains input protection against damage due to high static voltages or electric fields; however, precautions should be taken to avoid application of voltages higher than the maximum rating.

ELECTRICAL CHARACTERISTICS (Vcc = 5.0V ± 5%, Vss = 0, T_A = 0 to + 70 C)

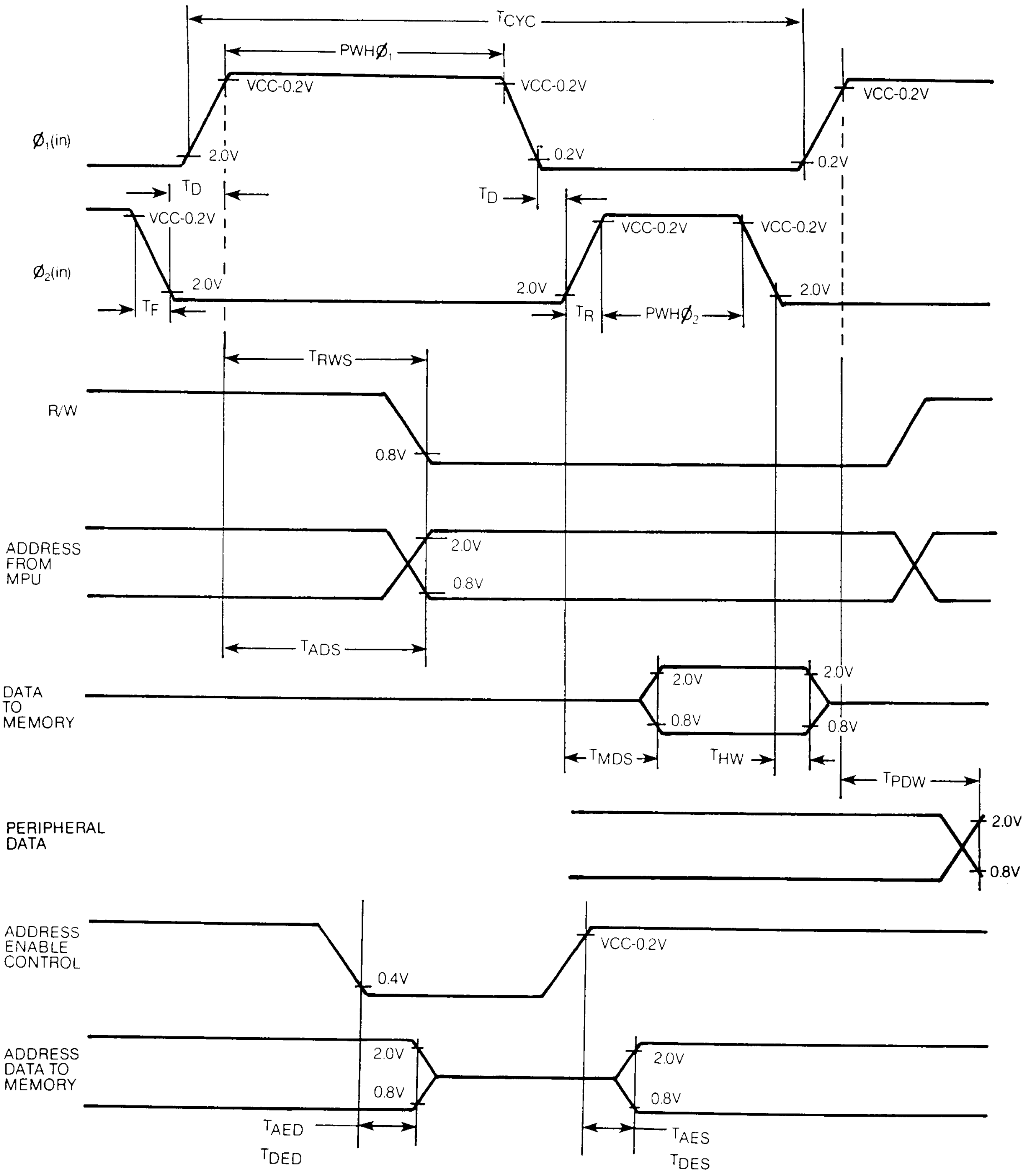
CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT
Input High Voltage Ø$_1$, Ø$_2$(in)	VIH	Vcc - 0.2	—	Vcc + 1.0V	Vdc
Input High Voltage $\overline{RES}$, P_0-P_7, $\overline{IRQ}$, Data		2.0	—	—	Vdc
Input Low Voltage Ø$_1$, Ø$_2$(in)	VIL	—	—	0.2	Vdc
$\overline{RES}$, P_0-P_7, $\overline{IRQ}$, Data		—	—	0.8	Vdc
Input Leakage Current (V_{in} = 0 to 5.25V, Vcc = 5.25V) Logic	Iin	—	—	2.5	µA
Ø$_1$, Ø$_2$(in)		—	—	100	µA
Three State (Off State) Input Current (V_{in} = 0.4 to 2.4V, Vcc = 5.25V) Data Lines	ITSI	—	—	10	µA
Output High Voltage (I_{OH} = −100µAdc, Vcc = 4.75V) Data, A0-A15, R/W, P_0-P_7	VOH	2.4	—	—	Vdc
Out Low Voltage (I_{OL} = 1.6mAdc, Vcc = 4.75V) Data, A0-A15, R/W, P_0-P_7	VOL	—	—	+0.60	Vdc
Power Supply Current	ICC	—	125		mA
Capacitance V_{in} = 0, T_A = 25 C, f = 1 MHz)	C				pF
Logic, P_0-P_7	C_{in}	—	—	10	
Data		—	—	15	
A0-A15, R/W	C_{out}	—	—	12	
Ø$_1$	CØ$_1$	—	30	50	
Ø$_2$	CØ$_2$	—	50	80	

CLOCK TIMING



TIMING FOR READING DATA FROM MEMORY OR PERIPHERALS

CLOCK TIMING



TIMING FOR WRITING DATA TO MEMORY OR PERIPHERALS

AC CHARACTERISTICS

ELECTRICAL CHARACTERISTICS (VCC = 5V + 5%, VSS = OV, T_A = 0° -70°C)

Clock Timing

		1 MHz TIMING		2 MHz TIMING		3 MHz TIMING		
CHARACTERISTIC	SYMBOL	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	UNITS
Cycle Time	T_{CYC}	1000	—	500	—	333	—	ns
Clock Pulse Width $\phi 1$ (Measured at VCC - 0.2V) $\phi 2$	$PWH\phi_1$ $PWH\phi_2$	430 470	— —	215 235	— —	145 165	— —	ns ns
Fall Time, Rise Time (Measured from 0.2V to VCC - 0.2V)	T_F, T_R	—	25	—	15	—	15	ns
Delay Time between Clocks (Measured at 0.2V)	T_D	0	—	0	—	0	—	ns

Read/Write Timing (Load = 1 TTL)

		1 MHz TIMING		2 MHz TIMING		3 MHz TIMING		
CHARACTERISTIC	SYMBOL	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	UNITS
Read, Write Setup Time from 6509	T_{RWS}	—	300	—	150	—	125	ns
Address Setup Time from 6509	T_{ADS}	—	300	—	150	—	125	ns
Memory Read Access Time	T_{ACC}	—	575	—	300	—	250	ns
Data Stability Time Period	T_{DSU}	100	—	50	—	50	—	ns
Data Hold Time-Read	T_{HR}	10	—	10	—	10	—	ns
Data Hold Time-Write	T_{HW}	10	—	10	—	10	—	ns
Data Setup Time from 6509	T_{MDS}	—	200	—	100	—	100	ns
Address Hold Time	T_{HA}	10	—	10	—	10	—	ns
R/W Hold Time	T_{HRW}	10	—	10	—	10	—	ns
Port Output Data Valid (Memory Management)	T_{PDW}	—	300	—	150	—	125	μs
RDY Setup Time	T_{RDY}	100	—	50	—	15	—	ns
S.O. Setup Time	$T_{S.O.}$	100	—	50	—	50	—	ns
SYNC Setup Time from 6509	T_{SYNC}	—	350	—	175	—	120	ns
Address Enable Setup Time	T_{AES}	—	75	—	75	—	75	ns
Data Enable Setup Time	T_{DES}	—	120	—	120	—	120	ns
Address Disable *See Note 1	T_{AED}	—	120	—	120	—	120	ns
Data Disable *See Note 1	T_{DED}	—	130	—	130	—	130	ns

*Note 1 — 1TTL Load
CL = 30pf

SIGNAL DESCRIPTION

Clocks (ϕ_1, ϕ_2)

The 6509 requires a two phase non-overlapping clock that runs at the Vcc voltage level.

Address Bus (A_0-A_{15})

The tri-state outputs are TTL compatible, capable of driving one standard TTL load and 130 pf.

Data Bus (D_0-D_7)

Eight pins are used for the data bus. This is a Bi-Directional bus, transferring data to and from the device and peripherals. The outputs are tri-state buffers capable of driving one standard TTL load and 130 pf.

Reset

This input is used to reset or start the microprocessor from a power down condition. During the time that this line is held low, writing to or from the microprocessor is inhibited. When a positive edge is detected on the input, the microprocessor will immediately begin the reset sequence.

After a system initialization time of six clock cycles, the mask interrupt flag will be set and the microprocessor will load the program counter from the memory vector locations FFFC and FFFD. This is the start location for program control.

After Vcc reaches 4.75 volts in a power up routine, reset must be held low for at least two clock cycles. At this time the R/W signal will become valid.

When the reset signal goes high following these two clock cycles, the microprocessor will proceed with the normal reset procedure detailed above.

Interrupt Request ($\overline{\text{IRQ}}$)

This TTL level input requests that an interrupt sequence begin within the microprocessor. The microprocessor will complete the current instruction being executed before recognizing the request. At that time, the interrupt mask bit in the Status Code Register will be examined. If the interrupt mask flag is not set, the microprocessor will begin an interrupt sequence. The Program Counter and Processor Status Register are stored in the stack. The microprocessor will then set the interrupt mask flag high so that no further interrupts may occur. At the end of this cycle, the program counter low will be loaded from address FFFE, and program counter high from location FFFF, therefore transferring program control to the memory vector located at these addresses.

Address Enable Control (AEC)

The Address Bus, Data Bus, and R/W signal are valid only when the Address Enable Control line is high. When low, the Address Bus is in a high-impedance state. This feature allows easy DMA and multiprocessor systems.

Read/Write (R/W)

This signal is generated by the microprocessor to control the direction of data transfers on the Data Bus. This line is high except when the microprocessor is writing to memory or a peripheral device.

Ready (RDY)

This input signal allows the user to single cycle the microprocessor on all cycles except write cycles. A negative transition to the low state during or coincident with phase one ($\emptyset_1$) and up to 100ns after phase two ($\emptyset_2$) will halt the microprocessor with the output address lines reflecting the current address being fetched. This condition will remain through a subsequent phase two ($\emptyset_2$) in which the Ready signal is low. This feature allows microprocessor interfacing with low speed PROMS as well as fast (max. 2 cycle) Direct Memory Access (DMA). If Ready is low during a write cycle, it is ignored until the following read operation.

Non-Maskable Interrupt ($\overline{\text{NMI}}$)

A negative going edge on this input requests that a non-maskable interrupt sequence be generated within the microprocessor.

$\overline{\text{NMI}}$ is an unconditional interrupt. Following completion of the current instruction, the sequence of operations defined for $\overline{\text{IRQ}}$ will be performed, regardless of the interrupt mask flag status. The vector address loaded into the program counter, low and high, are locations FFFA and FFFB respectively, thereby transferring program control to the memory vector located at these addresses. The instructions loaded at these locations cause the microprocessor to branch to a non-maskable interrupt routine in memory.

$\overline{\text{NMI}}$ also requires an external 3K resistor to V_{CC} for proper wire-OR operations.

Inputs $\overline{\text{IRQ}}$ and $\overline{\text{NMI}}$ are hardware interrupt lines that are sampled during $\emptyset_2$ (phase 2) and will begin the appropriate interrupt routine on the $\emptyset_1$ (phase 1) following the completion of the current instruction.

Set Overflow Flag (.S.O.)

A NEGATIVE going edge on this input sets the overflow bit in the Status Code Register. This signal is sampled on the trailing edge of $\emptyset_1$.

SYNC

This output line is provided to identify those cycles in which the microprocessor is doing an OP CODE fetch. The SYNC line goes high during $\emptyset_1$ of an OP CODE fetch and stays high for the remainder of that cycle. If the RDY line is pulled low during the $\emptyset_1$ clock pulse in which SYNC went high, the processor will stop in its current state and will remain in the state until the RDY line goes high. In this manner, the SYNC signal can be used to control RDY to cause single instruction execution.

Memory Management Control (P_0-P_3)

The four extended address pins, P_0-P_3, enable the processor to align to one of sixteen banks of 64K memory space. The use of the instructions; Load A indirect Y ($B1_{16}$) and Store A indirect Y (91_{16}), transfers the processor from the normal execute mode to the indirect mode. In the execute mode, the processor is aligned to a particular memory bank. The indirect mode aligns the processor to a predetermined memory block. The contents of the extended address registers is controlled by software with the execute register at location 0000 and indirect register at 0001.

During fetch and execution of the indirect mode instructions, the processor remains in execute mode until data transfer is to occur. At this time the processor switches to the indirect mode aligning itself to the new memory block. After one cycle the processor returns to the execute mode.

The upper four bits of the data bus are logic "0"s during a read/write to the extended address registers. Also, these registers are set to logic "1"s during reset.

The extended address pins are not under control of AEC and are not tri-statable.

ADDRESSING MODES

Accumulator Addressing

This form of addressing is represented with a one byte instruction, implying an operation on the accumulator.

Immediate Addressing

In immediate addressing, the operand is contained in the second byte of the instruction, with no further memory addressing required.

Absolute Addressing

In absolute addressing, the second byte of the instruction specified the eight low order bits of the effective address while the third byte specifies the eight high order bits. Thus, the absolute addressing mode allows access to the entire 65K bytes of addressable memory.

Zero Page Addressing

The zero page instructions allow for shorter code and execution times by only fetching the second byte of the instruction and assuming a zero high address byte. Careful use of the zero page can result in significant increase in code efficiency.

Indexed Zero Page Addressing

(X, Y indexing). This form of addressing is used in conjunction with the index register and is referred to as "Zero Page, X" or "Zero Page, Y." The effective address is calculated by adding the second byte to the contents of the index register. Since this is a form of "Zero Page" addressing, the content of the second byte references a location in page zero. Additionally, due to the "Zero Page" addressing nature of this mode, no carry is added to the high order 8 bits of memory and crossing of page boundaries does not occur.

Indexed Absolute Addressing

(X, Y indexing). This form of addressing is used in conjunction with X and Y index register and is referred to as "Absolute, X," and "Absolute, Y." The effective address is formed by adding the contents of X and Y to the address contained in the second and third bytes of the instruction. This mode allows the index register to contain the index or count value and the instruction to contain the base address. This type of indexing allows any location referencing and the index to modify multiple fields resulting in reduced coding and execution time.

Implied Addressing

In the implied addressing mode, the address containing the operand is implicitly stated in the operation code of the instruction.

Relative Addressing

Relative addressing is used only with branch instructions and establishes a destination for the conditional branch.

The second byte of the instruction becomes the operand which is an "Offset" added to the contents of the lower eight bits of the program counter when the counter is set at the next instruction. The range of the offset is -128 to +127 bytes from the next instruction.

Indexed Indirect Addressing

In indexed indirect addressing (referred to as [Indirect, X]), the second byte of the instruction is added to the contents of the X index register, discarding the carry. The result of this addition points to a memory location on page zero whose contents is the low order eight bits of the effective address. The next memory location in page zero contains the high order eight bits of the effective address. Both memory locations specifying the high and low order bytes of the effective address must be in page zero.

Indirect Indexed Addressing

In indirect indexed addressing (referred to as [Indirect, Y]), the second byte of the instruction points to a memory location in page zero. The contents of this memory location is added to the contents of the Y index register, the result being the low order eight bits of the effective address. The carry from this addition is added to the contents of the next page zero memory location, the result being the high order eight bits of the effective address.

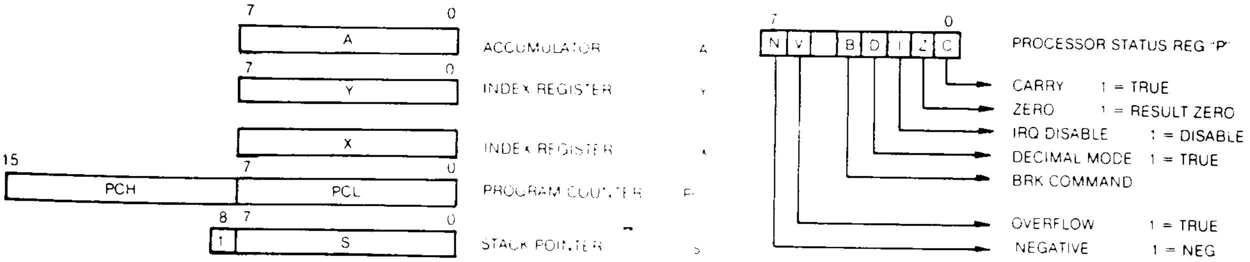
Absolute Indirect

The second byte of the instruction contains the low order eight bits of a memory location. The high order eight bits of that memory location is contained in the third byte of the instruction. The contents of the fully specified memory location is the low order byte of the effective address. The next memory location contains the high order byte of the effective address which is loaded into the sixteen bits of the program counter.

INSTRUCTION SET—ALPHABETIC SEQUENCE

ADC	Add Memory to Accumulator with Carry
AND	"AND" Memory with Accumulator
ASL	Shift Left One Bit (Memory or Accumulator)
BCC	Branch on Carry Clear
BCS	Branch on Carry Set
BEQ	Branch on Result Zero
BIT	Test Bits in Memory with Accumulator
BMI	Branch on Result Minus
BNE	Branch on Result not Zero
BPL	Branch on Result Plus
BRK	Force Break
BVC	Branch on Overflow Clear
BVS	Branch on Overflow Set
CLC	Clear Carry Flag
CLD	Clear Decimal Mode
CLI	Clear Interrupt Disable Bit
CLV	Clear Overflow Flag
CMP	Compare Memory and Accumulator
CPX	Compare Memory and Index X
CPY	Compare Memory and Index Y
DEC	Decrement Memory by One
DEX	Decrement Index X by One
DEY	Decrement Index Y by One
EOR	"Exclusive-or" Memory with Accumulator
INC	Increment Memory by One
INX	Increment Index X by One
INY	Increment Index Y by One
JMP	Jump to New Location
JSR	Jump to New Location Saving Return Address
LDA	Load Accumulator with Memory
LDX	Load Index X with Memory
LDY	Load Index Y with Memory
LSR	Shift One Bit Right (Memory or Accumulator)
NOP	No Operation
ORA	"OR" Memory with Accumulator
PHA	Push Accumulator on Stack
PHP	Push Processor Status on Stack
PLA	Pull Accumulator from Stack
PLP	Pull Processor Status from Stack
ROL	Rotate One Bit Left (Memory or Accumulator)
ROR	Rotate One Bit Right (Memory or Accumulator)
RTI	Return from Interrupt
RTS	Return from Subroutine
SBC	Subtract Memory from Accumulator with Borrow
SEC	Set Carry Flag
SED	Set Decimal Mode
SEI	Set Interrupt Disable Status
STA	Store Accumulator in Memory
STX	Store Index X in Memory
STY	Store Index Y in Memory
TAX	Transfer Accumulator to Index X
TAY	Transfer Accumulator to Index Y
TSX	Transfer Stack Pointer to Index X
TXA	Transfer Index X to Accumulator
TXS	Transfer Index X to Stack Register
TYA	Transfer Index Y to Accumulator

PROGRAMMING MODEL



INSTRUCTION SET — OP CODES, Execution Time, Memory Requirements

INSTRUCTIONS		IMMEDIATE			ABSOLUTE			ZERO PAGE			ACCUM.			IMPLIED			(IND. X)			(IND). Y			Z. PAGE, X			ABS. X			ABS. Y			RELATIVE			INDIRECT			Z. PAGE, Y			CONDITION CODES					
MNEMONIC	OPERATION	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	N	Z	C	I	D	V
A D C	A + M + C → A (4)(1)	69	2	2	6D	4	3	65	3	2							61	6	2	71	5	2	75	4	2	7D	4	3	79	4	3										✓	✓	✓	-	-	✓
A N D	A ∧ M → A (1)	29	2	2	2D	4	3	25	3	2							21	6	2	31	5	2	35	4	2	3D	4	3	39	4	3										✓	✓	-	-	-	-
A S L	C ← [7 0] ← 0				0E	6	3	06	5	2	0A	2	1										16	6	2	1E	7	3													✓	✓	✓	-	-	-
B C C	BRANCH ON C = 0 (2)																															90	2	2							-	-	-	-	-	-
B C S	BRANCH ON C = 1 (2)																															B0	2	2							-	-	-	-	-	-
B E Q	BRANCH ON Z = 1 (2)																															F0	2	2							-	-	-	-	-	-
B I T	A ∧ M				2C	4	3	24	3	2																															M_7	✓	-	-	-	M_6
B M I	BRANCH ON N = 1 (2)																															30	2	2							-	-	-	-	-	-
B N E	BRANCH ON Z = 0 (2)																															D0	2	2							-	-	-	-	-	-
B P L	BRANCH ON N = 0 (2)																															10	2	2							-	-	-	-	-	-
B R K														00	7	1																									-	-	-	1	-	-
B V C	BRANCH ON V = 0 (2)																															50	2	2							-	-	-	-	-	-
B V S	BRANCH ON V = 1 (2)																															70	2	2							-	-	-	-	-	-
C L C	0 → C													18	2	1																									-	-	0	-	-	-
C L D	0 → D													D8	2	1																									-	-	-	-	0	-
C L I	0 → I													58	2	1																									-	-	-	0	-	-
C L V	0 → V													B8	2	1																									-	-	-	-	-	0
C M P	A − M (1)	C9	2	2	CD	4	3	C5	3	2							C1	6	2	D1	5	2	D5	4	2	DD	4	3	D9	4	3										✓	✓	✓	-	-	-
C P X	X − M	E0	2	2	EC	4	3	E4	3	2																															✓	✓	✓	-	-	-
C P Y	Y − M	C0	2	2	CC	4	3	C4	3	2																															✓	✓	✓	-	-	-
D E C	M − 1 → M				CE	6	3	C6	5	2													D6	6	2	DE	7	3													✓	✓	-	-	-	-
D E X	X − 1 → X													CA	2	1																									✓	✓	-	-	-	-
D E Y	Y − 1 → Y													88	2	1																									✓	✓	-	-	-	-
E O R	A ⊻ M → A (1)	49	2	2	4D	4	3	45	3	2							41	6	2	51	5	2	55	4	2	5D	4	3	59	4	3										✓	✓	-	-	-	-
I N C	M + 1 → M				EE	6	3	E6	5	2													F6	6	2	FE	7	3													✓	✓	-	-	-	-
I N X	X + 1 → X													E8	2	1																									✓	✓	-	-	-	-
I N Y	Y + 1 → Y													C8	2	1																									✓	✓	-	-	-	-
J M P	JUMP TO NEW LOC				4C	3	3																												6C	5	3				-	-	-	-	-	-
J S R	JUMP SUB				20	6	3																																		-	-	-	-	-	-
L D A	M → A (1)	A9	2	2	AD	4	3	A5	3	2							A1	6	2	B1	5	2	B5	4	2	BD	4	3	B9	4	3										✓	✓	-	-	-	-
L D X	M → X (1)	A2	2	2	AE	4	3	A6	3	2																			BE	4	3							B6	4	2	✓	✓	-	-	-	-
L D Y	M → Y (1)	A0	2	2	AC	4	3	A4	3	2													B4	4	2	BC	4	3													✓	✓	-	-	-	-
L S R	0 → [7 0] → C				4E	6	3	46	5	2	4A	2	1										56	6	2	5E	7	3													0	✓	✓	-	-	-
N O P	NO OPERATION													EA	2	1																									-	-	-	-	-	-
O R A	A ∨ M → A (1)	09	2	2	0D	4	3	05	3	2							01	6	2	11	5	2	15	4	2	1D	4	3	19	4	3										✓	✓	-	-	-	-
P H A	A → M_S S − 1 → S													48	3	1																									-	-	-	-	-	-
P H P	P → M_S S − 1 → S													08	3	1																									-	-	-	-	-	-
P L A	S + 1 → S M_S → A													68	4	1																									✓	✓	-	-	-	-
P L P	S + 1 → S M_S → P													28	4	1																									(RESTORED)					
R O L	← [7 0] ← [C] ←				2E	6	3	26	5	2	2A	2	1										36	6	2	3E	7	3													✓	✓	✓	-	-	-
R O R	→ [C] → [7 0] →				6E	6	3	66	5	2	6A	2	1										76	6	2	7E	7	3													✓	✓	✓	-	-	-
R T I	RTRN INT													40	6	1																									(RESTORED)					
R T S	RTRN SUB													60	6	1																									-	-	-	-	-	-
S B C	A − M − C → A (1)	E9	2	2	ED	4	3	E5	3	2							E1	6	2	F1	5	2	F5	4	2	FD	4	3	F9	4	3										✓	✓	(3)	-	-	✓
S E C	1 → C													38	2	1																									-	-	1	-	-	-
S E D	1 → D													F8	2	1																									-	-	-	-	1	-
S E I	1 → I													78	2	1																									-	-	-	1	-	-
S T A	A → M				8D	4	3	85	3	2							81	6	2	91	6	2	95	4	2	9D	5	3	99	5	3										-	-	-	-	-	-
S T X	X → M				8E	4	3	86	3	2																												96	4	2	-	-	-	-	-	-
S T Y	Y → M				8C	4	3	84	3	2													94	4	2																-	-	-	-	-	-
T A X	A → X													AA	2	1																									✓	✓	-	-	-	-
T A Y	A → Y													A8	2	1																									✓	✓	-	-	-	-
T S X	S → X													BA	2	1																									✓	✓	-	-	-	-
T X A	X → A													8A	2	1																									✓	✓	-	-	-	-
T X S	X → S													9A	2	1																									-	-	-	-	-	-
T Y A	Y → A													98	2	1																									✓	✓	-	-	-	-

(1) ADD 1 TO "N" IF PAGE BOUNDARY IS CROSSED

(2) ADD 1 TO "N" IF BRANCH OCCURS TO SAME PAGE
ADD 2 TO "N" IF BRANCH OCCURS TO DIFFERENT PAGE

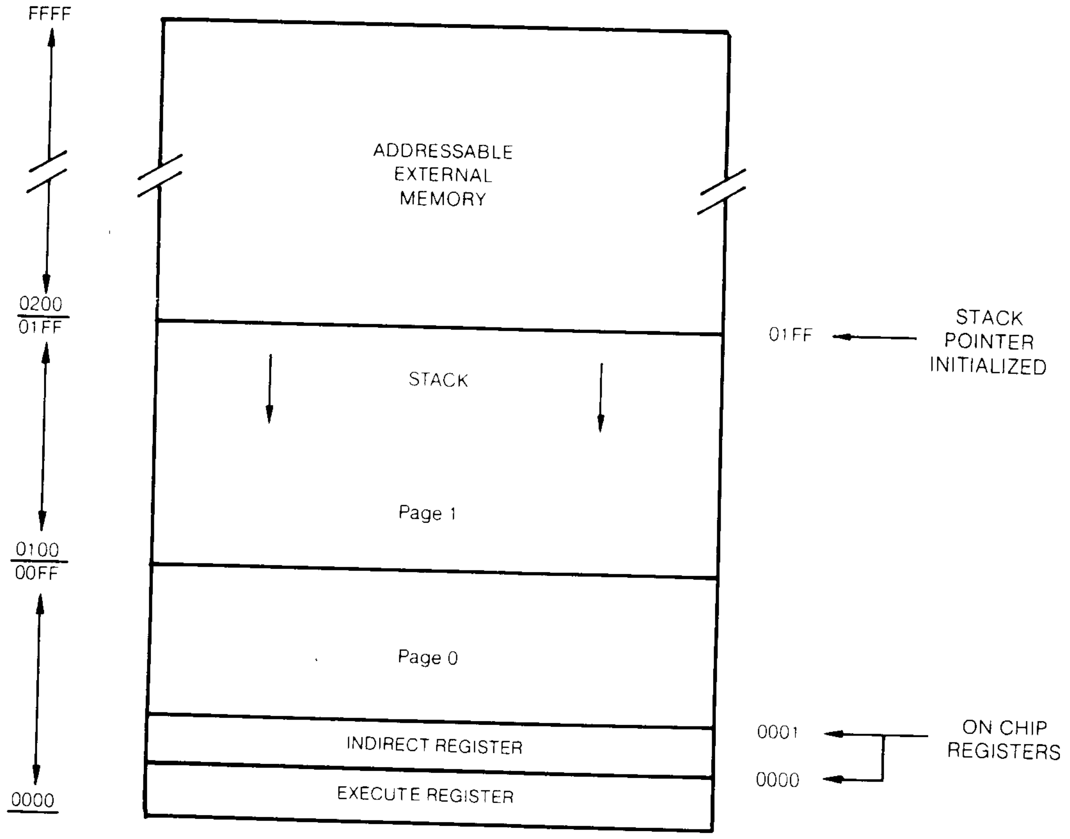
(3) CARRY NOT = BORROW

(4) IF IN DECIMAL MODE Z FLAG IS INVALID
ACCUMULATOR MUST BE CHECKED FOR ZERO RESULT

X	INDEX X	+	ADD	⊻	EXCLUSIVE OR	N	NO. CYCLES
Y	INDEX Y	−	SUBTRACT	✓	MODIFIED	#	NO. BYTES
A	ACCUMULATOR	∧	AND	-	NOT MODIFIED		
M	MEMORY PER EFFECTIVE ADDRESS	∨	OR	M_7	MEMORY BIT 7		
M_S	MEMORY PER STACK POINTER			M_6	MEMORY BIT 6		

Note: Commodore Semiconductor Group cannot assume liability for the use of undefined OP Codes

6509 MEMORY MAP



COMMODORE SEMICONDUCTOR GROUP reserves the right to make changes to any products herein to improve reliability, function or design. COMMODORE SEMICONDUCTOR GROUP does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.