Synertek.

SY6500

8-Bit Microprocessor Family

Features

- Single 5 V ±5% Power 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
- Non-maskable Interrupt
- Use with Any Type or Speed Memory
- Bi-directional Data Bus
- Instruction Decoding and Control
- Addressable Memory Range of up to 65K Bytes
- "Ready" Input
- Direct Memory Access Capability
- Bus Compatible with MC6800
- Choice of External or On-board Clocks
- 1 MHz, 2 MHz Operation
- On-chip Clock Options
  - — External Single Clock Input
  - — Crystal Time Base Input
- 40 and 28 Pin Package Versions
- Pipeline Architecture

Description

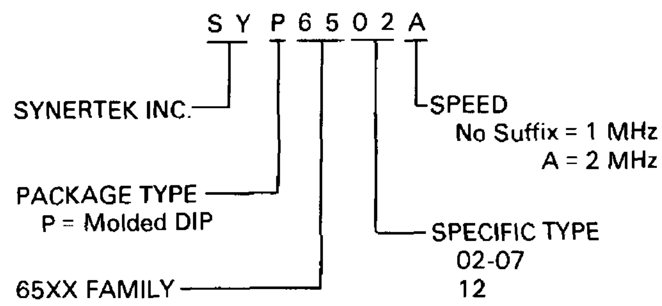
The SY6500 Series Microprocessors represent the first totally software compatible microprocessor family. This family of products includes a range of software compatible microprocessors which provide a selection of addressable memory range, interrupt input options and on-chip clock oscillators and drivers. All of the microprocessors in the SY6500 family are software compatible within the group and are bus compatible with the MC6800 product offering.

The family includes six microprocessors with on-board clock oscillators and drivers for four microprocessors driven by external clocks. The on-chip clock versions are aimed at high performance, low cost applications where single phase inputs or crystals provide the time base. The external clock versions are geared for the multi-processor system applications where maximum timing control is mandatory. All versions of the microprocessors are available in 1 MHz, 2 MHz, 3 MHz and 4 MHz maximum operating frequencies.

Members of the Family

Part Number	Clocks	Pins	IRQ	NMI	RYD	Addressing
SY6502	On-Chip	40	√	√	√	64K
SY6507	"	28			√	8K
SY6512	External	40	√	√	√	64K

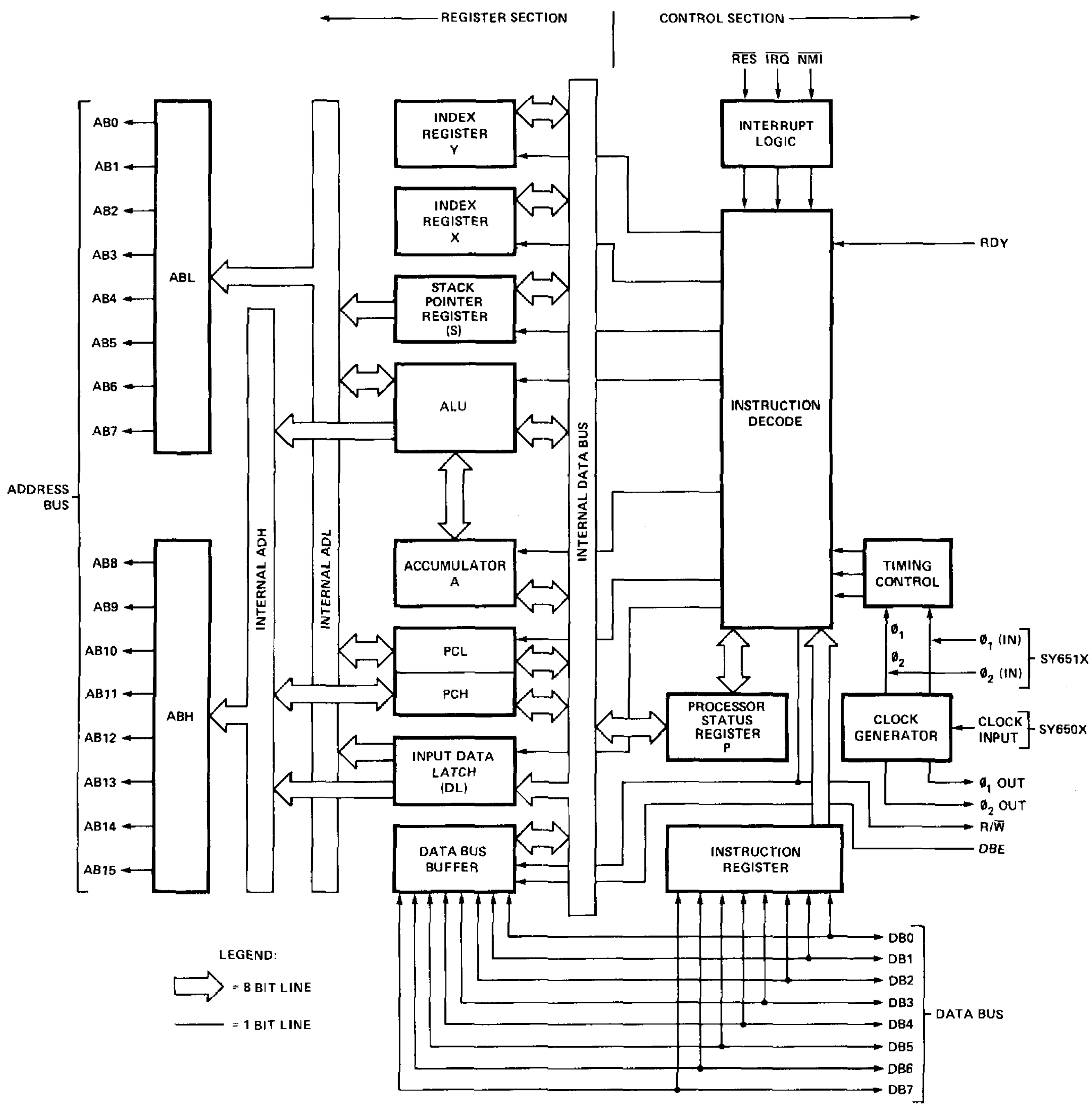
Ordering Information



Comments on the Data Sheet

The data sheet is constructed to review the basic "Common Characteristics" — those features which are common to the general family of microprocessors. Subsequent to a review of the family characteristics will be sections devoted to each member of the group with specific features of each.

SY6500 Internal Architecture



NOTE:
1. CLOCK GENERATOR IS NOT INCLUDED ON SY651X.
2. ADDRESSING CAPABILITY AND CONTROL OPTIONS VARY WITH EACH OF THE SY6500 PRODUCTS.

Absolute Maximum Ratings*

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 to +7.0	V
Input Voltage	V_{in}	-0.3 to +7.0	V
Operating Temperature	T_A	0 to +70	°C
Storage Temperature	T_{STG}	-55 to +150	°C

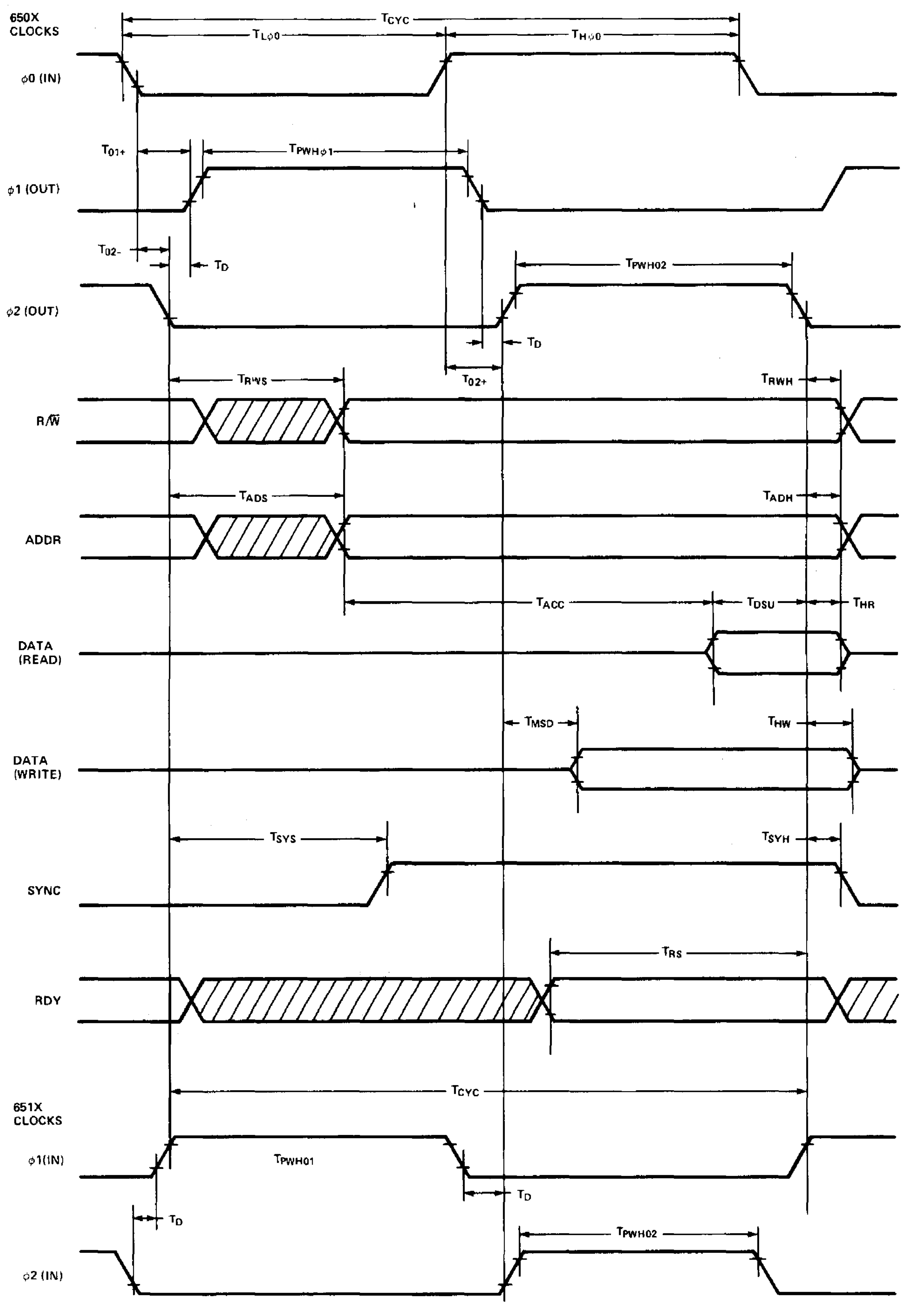
Comment*

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.

D.C. Characteristics (V_{CC} = 5.0V ±5%, T_A = 0-70°C)

($Ø_1$, $Ø_2$ applies to SY651X, $Ø_{o(in)}$ applies to SY650X)

Symbol	Characteristic	Min.	Max.	Unit
V_{IH}	Input High Voltage Logic and $Ø_o$ (in) for all 650X devices — 1,2,3 MHz Logic and $Ø_o$ (in) for all 650X devices — 4 MHz	 +2.0 +3.3	 V_{CC} V_{CC}	 V V
	$Ø_1$ and $Ø_2$ only for all 651X devices. Logic as 650X — All Speeds	V_{CC} -0.5	V_{CC} + 0.25	V
V_{IL}	Input Low Voltage Logic, $Ø_{o(in)}$ (650X) $Ø_1$, $Ø_2$ (651X)	 -0.3 -0.3	 +0.8 +0.2	 V
I_{IL}	Input Loading (V_{in} = 0 V, V_{CC} = 5.25 V) RDY, S.O.	-10	-300	μA
I_{in}	Input Leakage Current (V_{in} = 0 to 5.25 V, V_{CC} = 0) Logic (Excl. RDY, S.O.) $Ø_1$, $Ø_2$ (651X) $Ø_{o(in)}$ (650X)	 – – –	 2.5 100 10.0	 μA μA μA
I_{TSI}	Three-State (Off State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = 5.25 V) DB0-DB7	–	±10	μA
V_{OH}	Output High Voltage (I_{LOAD} = -100μAdc, V_{CC} = 4.75 V) 1, 2 MHz SYNC, DB0-DB7, A0-A15, R/$\overline{W}$	2.4	–	V
V_{OL}	Output Low Voltage (I_{LOAD} = 1.6mAdc, V_{CC} = 4.75 V) 1, 2 MHz SYNC, DB0-DB7, A0-A15, R/$\overline{W}$	–	0.4	V
P_D	Power Dissipation 1 MHz and 2 MHz (V_{CC} = 5.25V)	–	700	mW
C	Capacitance (V_{in} = 0, T_A = 25°C, f = 1 MHz)			pF
C_{in}	$\overline{RES}$, $\overline{NMI}$, RDY, $\overline{IRQ}$, S.O., DBE	–	10	
	DB0-DB7	–	15	
C_{out}	A0-A15, R/$\overline{W}$, SYNC	–	12	
$C_{Ø_{o(in)}}$	$Ø_{o(in)}$ (650X)	–	15	
$C_{Ø_1}$	$Ø_1$ (651X)	–	50	
$C_{Ø_2}$	$Ø_2$ (651X)	–	80	


650X CLOCKS
φ0 (IN)
φ1 (OUT)
φ2 (OUT)
R/W̄
ADDR
DATA (READ)
DATA (WRITE)
SYNC
RDY
651X CLOCKS
φ1(IN)
φ2 (IN)
TCYC
TLφ0
THφ0
T01+
TPWHφ1
T02-
TD
TPWH02
T02+
TRWS
TRWH
TADS
TADH
TACC
TDSU
THR
TMSD
THW
TSYS
TSYH
TRS
TPWH01

Dynamic Operating Characteristics

(V_{CC} = 5.0 ± 5%, T_A = 0° to 70°C)

Parameter	Symbol	1 MHz Min.	1 MHz Max.	2 MHz Min.	2 MHz Max.	Units
651X						
Cycle Time	T_{CYC}	1.00	40	0.50	40	μs
Ø1 Pulse Width	$T_{PWHØ_1}$	430	—	215	—	ns
Ø2 Pulse Width	$T_{PWHØ_2}$	470	—	235	—	ns
Delay Between Ø1 and Ø2	T_D	0	—	0	—	ns
Ø1 and Ø2 Rise and Fall Times[1]	T_R, T_F	0	25	0	20	ns
650X						
Cycle Time	T_{CYC}	1.00	40	0.50	40	μs
$Ø_{o(IN)}$ Low Time[2]	$T_{LØ_o}$	480	—	240	—	ns
$Ø_{o(IN)}$ High Time[2]	$T_{HØ_o}$	460	—	240	—	ns
$Ø_o$ Neg to Ø1 Pos Delay[5]	T_{01+}	10	70	10	70	ns
$Ø_o$ Neg to Ø2 Neg Delay[5]	T_{02-}	5	65	5	65	ns
$Ø_o$ Pos to Ø1 Neg Delay[5]	T_{01-}	5	65	5	65	ns
$Ø_o$ Pos to Ø2 Pos Delay[5]	T_{02+}	15	75	15	75	ns
$Ø_{o(IN)}$ Rise and Fall Time[1]	T_{RO}, T_{FO}	0	30	0	20	ns
$Ø_{1(OUT)}$ Pulse Width	$T_{PWHØ_1}$	$T_{LØ_o}$-20	$T_{LØ_o}$	$T_{LØ_o}$-20	$T_{LØ_o}$	ns
$Ø_{2(OUT)}$ Pulse Width	$T_{PWHØ_2}$	$T_{LØ_o}$-40	$T_{LØ_o}$-10	$T_{LØ_o}$-40	$T_{LØ_o}$-10	ns
Delay Between Ø1 and Ø2	T_D	5	—	5	—	ns
Ø1 and Ø2 Rise and Fall Times[1,3]	T_R, T_F	—	25	—	25	ns
650X, 651X						
R/W̄ Setup Time	T_{RWS}	—	225	—	140	ns
R/W̄ Hold Time	T_{RWH}	30	—	30	—	ns
Address Setup Time	T_{ADS}	—	225	—	140	ns
Address Hold Time	T_{ADH}	30	—	30	—	ns
Read Access Time	T_{ACC}	—	650	—	310	ns
Read Data Setup Time	T_{DSU}	100	—	50	—	ns
Read Data Hold Time	T_{HR}	10	—	10	—	ns
Write Data Setup Time	T_{MDS}	20	175	20	100	ns
Write Data Hold Time	T_{HW}	60	150	60	150	ns
Sync Setup Time	T_{SYS}	—	350	—	175	ns
Sync Hold Time	T_{SYH}	30	—	30	—	ns
RDY Setup Time[4]	T_{RS}	200	—	200	—	ns

Notes:
1. Measured between 10% and 90% points.
2. Measured at 50% points.
3. Load = 1 TTL load +30 pF.
4. RDY must never switch states within T_{RS} to end of ϕ_2.
5. Load = 100 pF.
6. The 2 MHz devices are identified by an "A" suffix.

Timing Diagram Note:
Because the clock generation for the SY650X and SY651X is different, the two clock timing sections are referenced to the main timing diagram by three reference lines marked REF 'A', REF 'B' and REF 'C'. Reference between the two sets of clock timings is without meaning. Timing parameters are referred to these lines and scale variations in the diagrams are of no consequence.

Pin Functions

Clocks (ϕ_1, ϕ_2)

The SY651X requires a two phase non-overlapping clock that runs at the V_{CC} voltage level.

The SY650X clocks are supplied with an internal clock generator. The frequency of these clocks is externally controlled. Clock generator circuits are shown elsewhere in this data sheet.

Address Bus A_0-A_{15})

(See sections on each micro for respective address lines on those devices.)

These outputs are TTL compatible, capable of driving one standard TTL load and 130 pF.

Data Bus (DB_0-DB_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 three-state buffers, capable of driving one standard TTL load and 130 pF.

Data Bus Enable (DBE)

This TTL compatible input allows external control of the three-state data output buffers and will enable the microprocessor bus driver when in the high state. In normal operation DBE would be driven by the phase two (ϕ_2) clock, thus allowing data output from microprocessor only during ϕ_2. During the read cycle, the data bus drivers are internally disabled, becoming essentially an open circuit. To disable data bus drivers externally, DBE should be held low. This signal is available on the SY6512, only.

Ready (RDY)

This input signal allows the user to halt the microprocessor on all cycles except write cycles. A negative transition to the low state during or coincident with phase one, (ϕ_1) will halt the microprocessor with the output address lines reflecting the current address being fetched. This condition will remain through a subsequent phase two (ϕ_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 opeation. Ready transitions must not be permitted during ϕ_2 time.

Interrupt Request ($\overline{IRQ}$)

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 the 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 futher 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. The RDY signal must be in the high state for any interrupt to be recognized. A 3KΩ external resistor should be used for proper wire-OR operation.

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

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

$\overline{NMI}$ is an unconditional interrupt. Following completion of the current instruction, the sequence of operations defined for $\overline{IRQ}$ will be performed, regardless of the state interrupt mask flag. 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.

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

Inputs $\overline{IRQ}$ and $\overline{NMI}$ are hardware interrupts lines that are sampled during ϕ_2 (phase 2) and will begin the appropriate interrupt routine on the ϕ_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 ϕ_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 ϕ_1 of an OP CODE fetch and stays high for the remainder of that cycle. If the RDY line is pulled low during the ϕ_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.

Reset ($\overline{RES}$)

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 V_{CC} 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/$\overline{W}$ and SYNC 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.

Read/Write (R/$\overline{W}$)

This output signal is used to control the direction of data transfers between the processor and other circuits on the data bus. A high level on R/$\overline{W}$ signifies data into the processor; a low is for the data transfer out of the processor.

Programming Characteristics

INSTRUCTION SET — ALPHABETIC SEQUENCE

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

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 specifies 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 or 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

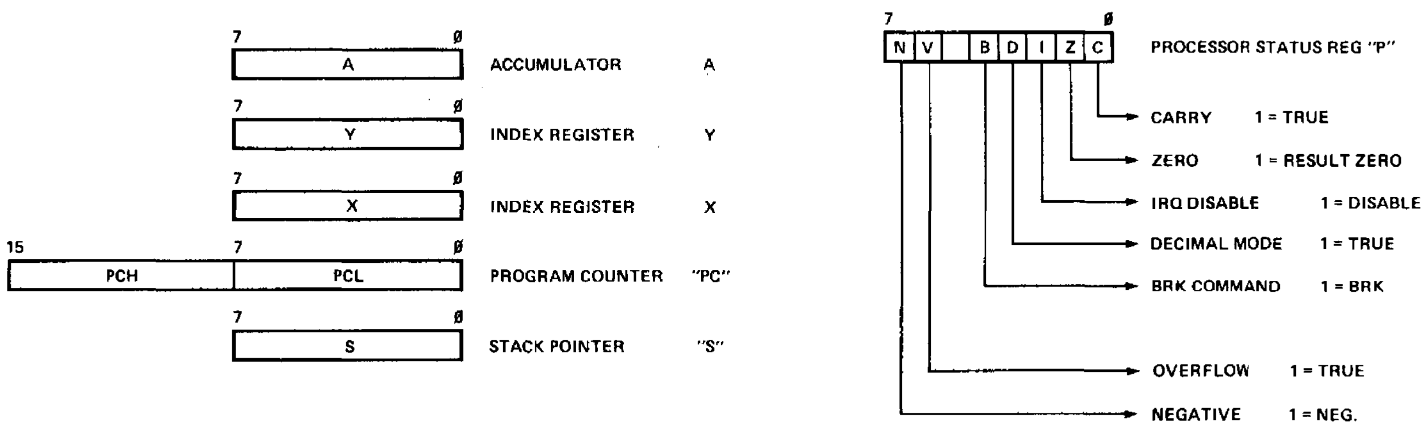
In indirect indexed addressing (referred to as [Indirect], Y), the second byte of the instruction points to a memory location in page zero. The contents of this memory location is added to the contents of the Y index register, the result being the low order eight bits of the effective address. The carry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

Absolute Indirect

The second byte of the instruction contains the lwo 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.

Programming Characteristics

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
ADC	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										✓	✓	✓	–	–	✓
AND	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										✓	✓	–	–	–	–
ASL	C ← [7 0] ← 0				0E	6	3	06	5	2	0A	2	1										16	6	2	1E	7	3													✓	✓	✓	–	–	–
BCC	BRANCH ON C=0 (2)																															90	2	2							–	–	–	–	–	–
BCS	BRANCH ON C=1 (2)																															B0	2	2							–	–	–	–	–	–
BEQ	BRANCH ON Z=1 (2)																															F0	2	2							–	–	–	–	–	–
BIT	A ∧ M				2C	4	3	24	3	2																															M_7	✓	–	–	–	M_6
BMI	BRANCH ON N=1 (2)																															30	2	2							–	–	–	–	–	–
BNE	BRANCH ON Z=0 (2)																															D0	2	2							–	–	–	–	–	–
BPL	BRANCH ON N=0 (2)																															10	2	2							–	–	–	–	–	–
BRK	(See Fig. 1)													00	7	1																									–	–	–	✓	–	–
BVC	BRANCH ON V=0 (2)																															50	2	2							–	–	–	–	–	–
BVS	BRANCH ON V=1 (2)																															70	2	2							–	–	–	–	–	–
CLC	0 → C													18	2	1																									–	–	0	–	–	–
CLD	0 → D													D8	2	1																									–	–	–	–	0	–
CLI	0 → I													58	2	1																									–	–	–	0	–	–
CLV	0 → V													B8	2	1																									–	–	–	–	–	0
CMP	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										✓	✓	✓	–	–	–
CPX	X-M	E0	2	2	EC	4	3	E4	3	2																															✓	✓	✓	–	–	–
CPY	Y-M	C0	2	2	CC	4	3	C4	3	2																															✓	✓	✓	–	–	–
DEC	M-1 → M				CE	6	3	C6	5	2													D6	6	2	DE	7	3													✓	✓	–	–	–	–
DEX	X-1 → X													CA	2	1																									✓	✓	–	–	–	–
DEY	Y-1 → Y													88	2	1																									✓	✓	–	–	–	–
EOR	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										✓	✓	–	–	–	–
INC	M + 1 → M				EE	6	3	E6	5	2													F6	6	2	FE	7	3													✓	✓	–	–	–	–
INX	X + 1 → X													E8	2	1																									✓	✓	–	–	–	–
INY	Y + 1 → Y													C8	2	1																									✓	✓	–	–	–	–
JMP	JUMP TO NEW LOC				4C	3	3																												6C	5	3				–	–	–	–	–	–
JSR	(See Fig. 2) JUMP SUB				20	6	3																																		–	–	–	–	–	–
LDA	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										✓	✓	–	–	–	–

		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
LDX	M → X (1)	A2	2	2	AE	4	3	A6	3	2																			BE	4	3							B6	4	2	✓	✓	–	–	–	–
LDY	M → Y (1)	A0	2	2	AC	4	3	A4	3	2													B4	4	2	BC	4	3													✓	✓	–	–	–	–
LSR	0 → [7 0] → C				4E	6	3	46	5	2	4A	2	1										56	6	2	5E	7	3													0	✓	✓	–	–	–
NOP	NO OPERATION													EA	2	1																									–	–	–	–	–	–
ORA	A V M → A	09	2	2	0D	4	3	05	3	2							01	6	2	11	5	2	15	4	2	1D	4	3	19	4	3										✓	✓	–	–	–	–
PHA	A → Ms S-1 → S													48	3	1																									–	–	–	–	–	–
PHP	P → Ms S-1 → S													08	3	1																									–	–	–	–	–	–
PLA	S + 1 → S Ms → A													68	4	1																									✓	✓	–	–	–	–
PLP	S + 1 → S Ms → P													28	4	1																									(RESTORED)					
ROL	← [7 0] ← [C] ←				2E	6	3	26	5	2	2A	2	1										36	6	2	3E	7	3													✓	✓	✓	–	–	–
ROR	→ [C] → [7 0] →				6E	6	3	66	5	2	6A	2	1										76	6	2	7E	7	3													✓	✓	✓	–	–	–
RTI	(See Fig. 1) RTRN INT													40	6	1																									RESTORED)					
RTS	(See Fig. 2) RTRN SUB													60	6	1																									–	–	–	–	–	–
SBC	A-M-C̄ → 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)	–	–	✓
SEC	1 → C													38	2	1																									–	–	1	–	–	–
SED	1 → D													F8	2	1																									–	–	–	–	1	–
SEI	1 → I													78	2	1																									–	–	–	1	–	–
STA	A → M				8D	4	3	85	3	2							81	6	2	91	6	2	95	4	2	9D	5	3	99	5	3										–	–	–	–	–	–
STX	X → M				8E	4	3	86	3	2																												96	4	2	–	–	–	–	–	–
STY	Y → M				8C	4	3	84	3	2													94	4	2																–	–	–	–	–	–
TAX	A → X													AA	2	1																									✓	✓	–	–	–	–
TAY	A → Y													A8	2	1																									✓	✓	–	–	–	–
TSX	S → X													BA	2	1																									✓	✓	–	–	–	–
TXA	X → A													8A	2	1																									✓	✓	–	–	–	–
TXS	X → S													9A	2	1																									–	–	–	–	–	–
TYA	Y → A													98	2	1																									✓	✓	–	–	–	–

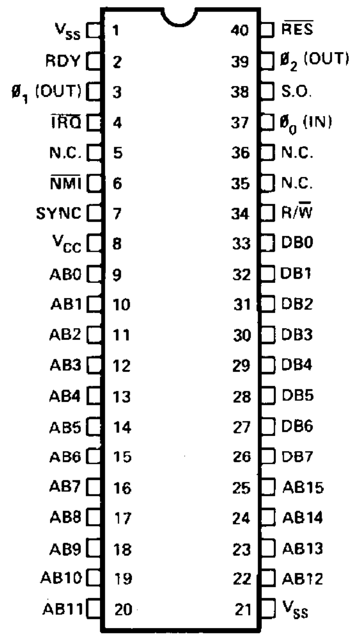
(1) ADD 1 TO "N" IF PAGE BOUNDARY IS CROSSED
(2) ADD 1 TO "N" IF BRANCH OCCURS TO SAME PAGE
ADD 2 TO "N" IF BRANCH OCCURS TO DIFFERENT PAGE
(3) CARRY NOT = BELOW
(4) IF IN DECIMAL MODE Z FLAG IS INVALID
ACCUMULATOR MUST BE CHECKED FOR ZERO RESULT

X INDEX X
Y INDEX Y
A ACCUMULATOR
M MEMORY PER EFFECTIVE ADDRESS
Ms MEMORY PER STACK POINTER

\+ ADD
– SUBTRACT
∧ AND
V OR
⊻ EXCLUSIVE OR
✓ MODIFIED

– NOT MODIFIED
M_7 MEMORY BIT 7
M_6 MEMORY BIT 6
N NO CYCLES
\# NO BYTES

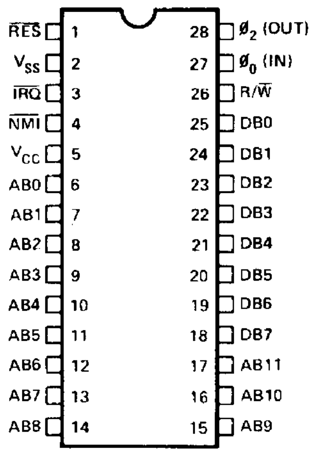
SY6502 — 40 Pin Package



Features

- 65K Addressable Bytes of Memory
- IRQ Interrupt
- NMI Interrupt
- On-the-chip Clock
  - √ TTL Level Single Phase Input
  - √ Crystal Time Base Input
- SYNC Signal
  (can be used for single instruction execution)
- RDY Signal
  (can be used for single cycle execution)
- Two Phase Output Clock for Timing of Support Chips

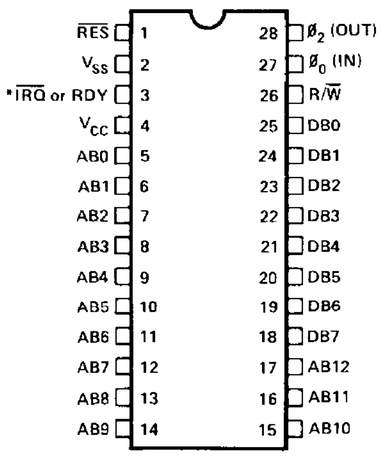
SY6503 — 28 Pin Package



Features

- 4K Addressable Bytes of Memory (AB00-AB11)
- On-the-chip Clock
- IRQ Interrupt
- NMI Interrupt
- 8 Bit Bi-Directional Data Bus

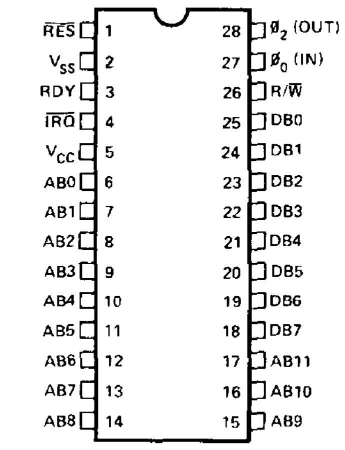
SY6504 & SY6507 — 28 Pin Package



Features

- IRQ Interrupt (6504 only)
- RDY Signal (6507 only)
- 8K Addressable Bytes of Memory (AB00-AB12)
- On-the-chip Clock
- 8 Bit Bi-Directional Data Bus

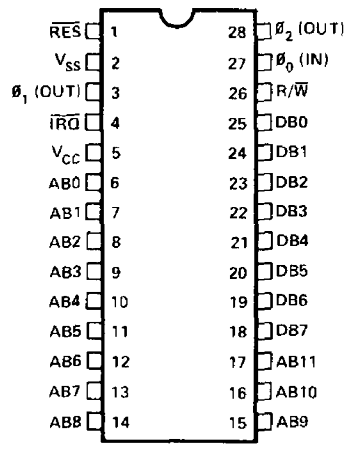
SY6505 — 28 Pin Package



Features

- 4K Addressable Bytes of Memory (AB00-AB11)
- On-the-chip Clock
- IRQ Interrupt
- RDY Signal
- 8 Bit Bi-Directional Data Bus

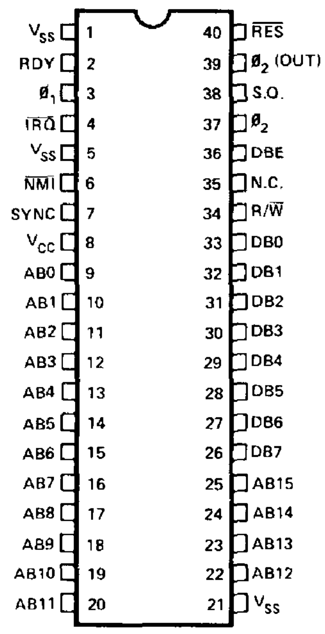
SY6506 — 28 Pin Package



Features

- 4K Addressable Bytes of Memory (AB00-AB11)
- On-the-chip Clock
- IRQ Interrupt
- Two phases off
- 8 Bit Bi-Directional Data Bus

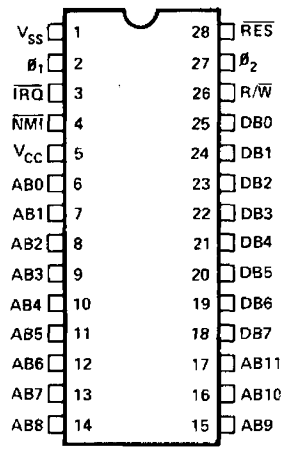
SY6512 — 40 Pin Package



Features

- 65K Addressable Bytes of Memory
- IRQ Interrupt
- NMI Interrupt
- RDY Signal
- 8 Bit Bi-Directional Data Bus
- SYNC Signal
- Two phase input
- Data Bus Enable

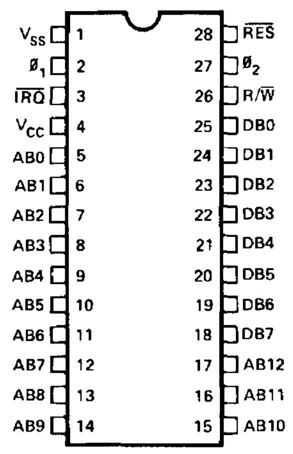
SY6513 — 28 Pin Package



Features

- 4K Addressable Bytes of Memory (AB00-AB11)
- Two phase clock input
- $\overline{IRQ}$ Interrupt
- $\overline{NMI}$ Interrupt
- 8 Bit Bi-Directional Data Bus

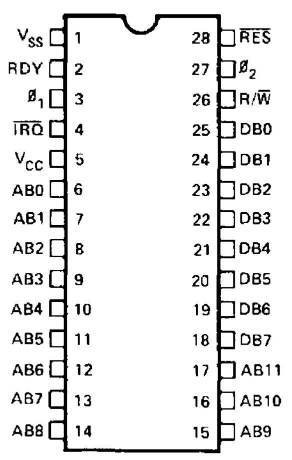
SY6514 — 28 Pin Package



Features

- 8K Addressable Bytes of Memory (AB00-AB12)
- Two phase clock input
- $\overline{IRQ}$ Interrupt
- 8 Bit Bi-Directional Data Bus

SY6515 — 28 Pin Package

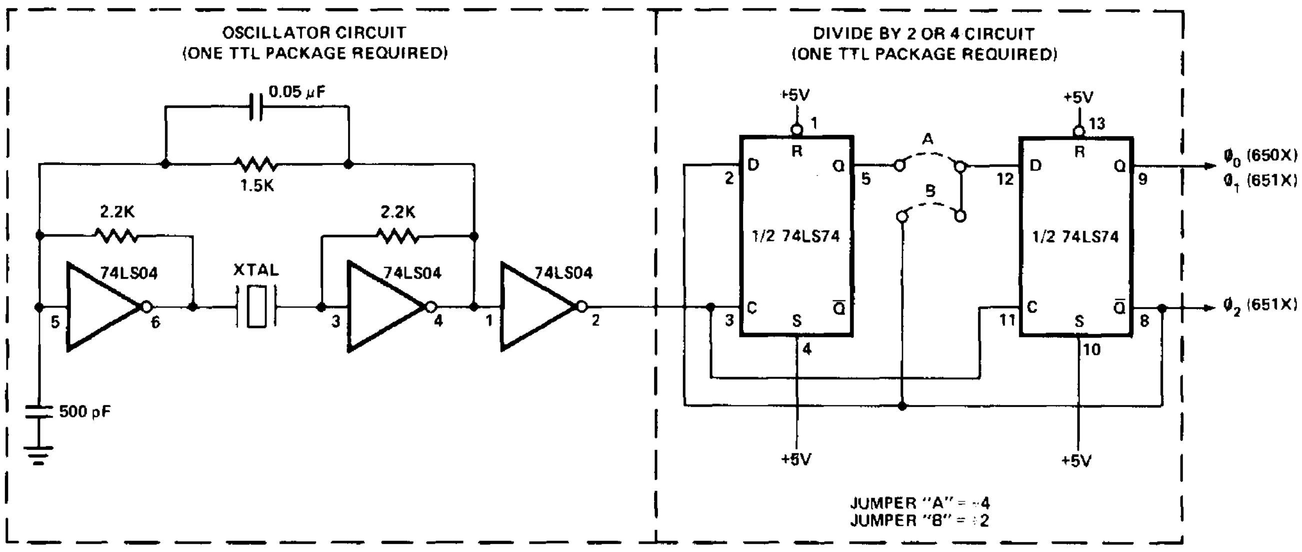


Features

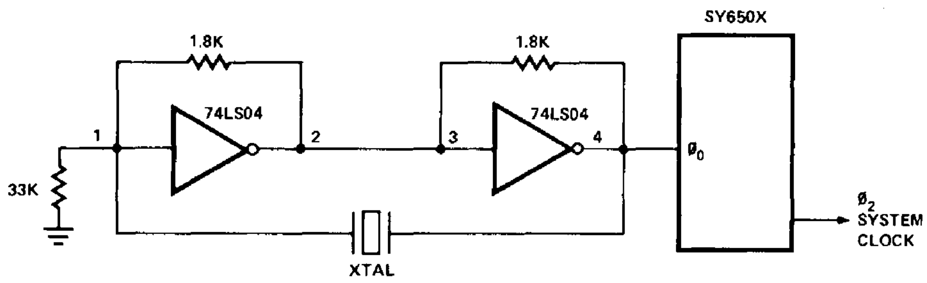
- 4K Addressable Bytes of Memory (AB00-AB11)
- Two phase clock input
- $\overline{IRQ}$ Interrupt
- 8 Bit Bi-Directional Data Bus

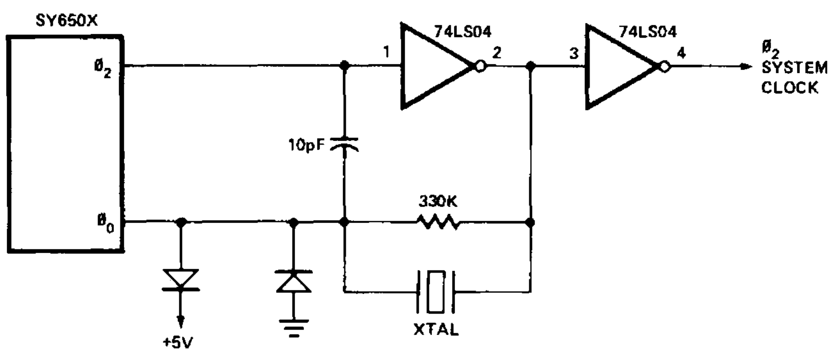
Clock Generation Circuits*

*For further details refer to Synertek SY6500 Applications Information Note AN2. Crystals used are CTS Knight MP Series or equivalents. (Series Mode)



CRYSTAL FREQUENCY	OUTPUT FREQUENCY	
	÷2	÷4
3.579545 MHz	1.7897 MHz	0.894886 MHz
4.194304 MHz	2.097152 MHz	1.048576 MHz





Synertek.

SY65C02
CMOS 8-Bit Microprocessor Family

PRELIMINARY

Features

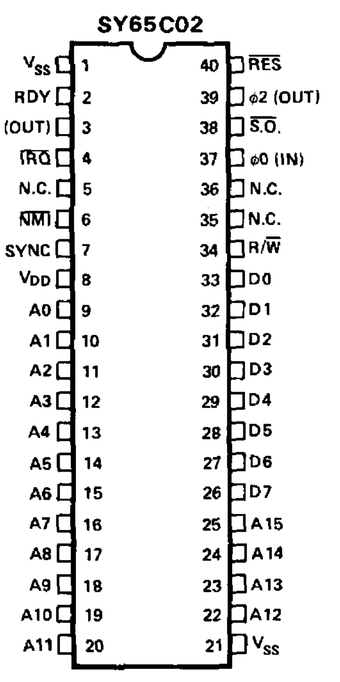
- High Performance n-Well HCMOS Family of Microprocessors
- Low Power Consumption, 4 mA at 1 MHz, 10 μA in Standby Operation Allowing Battery Operation
- Pin and Software Compatible with the NMOS 6500
- Improved Software Performance
  - — 27 New Operation Codes
  - — 15 Addressing Modes
  - — 66 Microprocessor Instructions
  - — 178 Total Operation Codes
- External or On-Board Clock Generation
  - — On-Board Clock Generator can be Driven by an External Single-Phase Clock Input, an RC Network, or a Crystal Circuit
- 1,2,3 or 4 MHz Operation
- Advanced Memory Access Timing Option
  - — Early Address Valid Allows High Speed Microprocessor Use with Slow Memories
  - — Early Write Data for Dynamic Memories
- Decimal and Binary Arithmetic
- Programmable Stack Pointer
- Variable Length Stack
- Improved Operational Capabilities

Description

The CMOS 65C02 microprocessor is compatible with the NMOS 6500 family of microprocessors. This 8-bit microprocessor unit designed in Synertek's proprietary high performance N-well silicon gate technology offers higher performance than the original NMOS 6502. The design allows for operating frequencies up to 4 MHz, and below 1 MHz further reducing its already low power consumption.

Not only is the 65C02 a low power version of the popular 6500 microprocessor, it also has these new features. Ability to tri-state the R/W line, address and data bus for DMA applications. Improved T_{ACC} specs allowing use with slower memory devices. A new optional output enhancing multiprocessing capabilityies. Two new addressing modes, an a larger instruction set providing the user with more compact programming capabilities.

Pin Configuration



Block Diagram

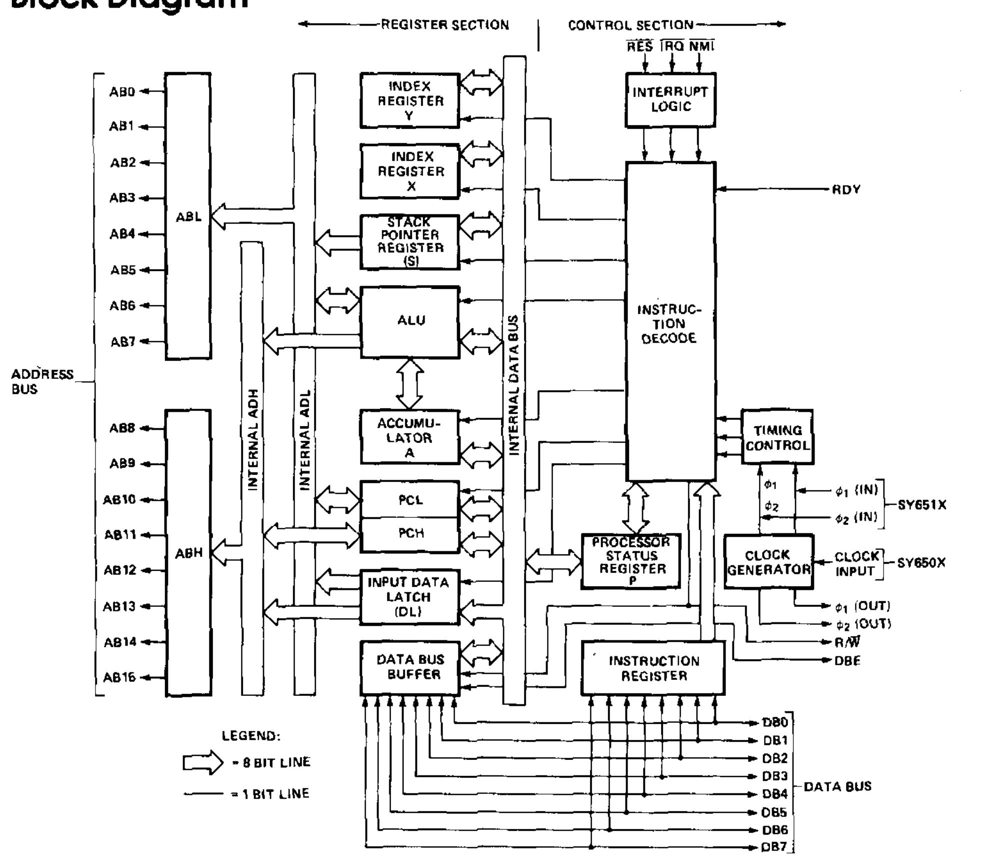


Figure 1.

Absolute Maximum Ratings

(V_{DD} = 5.0 V ± 5%, V_{SS} = 0 V, T_A = 0°C to 70°C)

Supply Voltage (V_{DD}) .... –0.3 to +7.0V
Input Voltage (V_{IN}) .... –0.3 to +7.0V
Operating Temperature (T_A) .... 0°C to +70°C
Storage Temperature (T_{STG}) .... –55°C to +150°C

Comment*

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

Pin Function

Pin	Function
A_0-A_{15}	Address Bus
D_0-D_7	Data Bus
$\overline{IRQ}$*	Interrupt Request
RDY*	Ready
$\overline{ML}$	Memory Lock
$\overline{NMI}$*	Non-Maskable Interrupt
SYNC	Synchronize
$\overline{RES}$*	Reset

Pin	Function
$\overline{SO}$*	Set Overflow
NC	No Connection
R/$\overline{W}$	Read/Write
V_{DD}	Power Supply (+5V)
V_{SS}	Internal Logic Ground
ϕ_0	Clock Input
ϕ_1, ϕ_2	Clock Output

*This pin has an optional internal pullup for a No Connect condition.

DC Characteristics

	Symbol	Min.	Typ.	Max.	Unit
Input High Voltage ϕ_0 (IN) $\overline{RES}$, $\overline{NMI}$, RDY, $\overline{IRQ}$, Data, S.O.	V_{IH}	 V_{SS} + 2.4 V_{SS} + 2.0	 — —	 V_{DD} —	 V V
Input Low Voltage ϕ_0 (IN) $\overline{RES}$, $\overline{NMI}$, RDY, $\overline{IRQ}$, Data, S.O.	V_{IL}	 V_{SS} – 0.3 —	 — —	 V_{SS} + 0.4 V_{SS} + 0.8	 V V
Input Leakage Current (V_{IN} = 0 to 5.25V, V_{DD} = 5.25V) With Pullups Without Pullups	I_{IN}	 -30 —	 — —	 +10 +1.0	 μA μA
Three State (Off State) Input Current (V_{IN} = 0.4 to 2.4V, V_{CC} = 5.25V) Data Lines	I_{TSI}	—	—	10	μA
Output High Voltage (I_{OH} = –100 μAdc, V_{DD} = 4.75V, SYNC, Data, A_0-A_{15}, R/W)	V_{OH}	V_{SS} + 2.4	—	—	V
Output Low Voltage (I_{OL} = 1.6 mAdc, V_{DD} = 4.75V, SYNC, Data, A_0-A_{15}, R/W)	V_{OL}	—	—	V_{SS} + 0.4	V
Supply Current f = 1 MHz	I_{DD}	—	—	4	mA
Supply Current f =2 MHz	I_{DD}	—	—	8	mA
Capacitance (V_{IN} = 0, T_A = 25°C, f = 1 MHz) Logic Data A_0-A_{15}, R/W, SYNC ϕ_0 (IN)	C C_{IN} C_{OUT} $C\phi_0$ (IN)	 — — — —	 — — — —	 5 10 10 10	pF

Microprocessor Operational Enhancements

Function	NMOS 6502 Microprocessor	SY65C02 Microprocessor
Indexed addressing across page boundary.	Extra read of invalid address.	Extra read of last instruction byte.
Execution of invalid op codes.	Some terminate only by reset. Results are undefined.	All are NOPs (reserved for future use). Op Code / Bytes / Cycles X2 / 2 / 2 X3, X7, XB, XF / 1 / 1 44 / 2 / 3 54, D4, F4 / 2 / 4 5C / 3 / 8 DC, FC / 3 / 4
Jump indirect, operand = XXFF.	Page address does not increment.	Page address increments and adds one additional cycle.
Read/modify/write instructions at effective address.	One read and two write cycles.	Two read and one write cycle.
Decimal flag.	Indeterminate after reset.	Initialized to binary mode (D = 0) after reset and interrupts.
Flags after decimal operation.	Invalid N, V and Z flags.	Valid flag adds one additional cycle.
Interrupt after fetch of BRK instruction.	Interrupt vector is loaded, BRK vector is ignored.	BRK is executed, then interrupt is executed.

Microprocessor Hardware Enhancements

Function	NMOS 6502	SY65C02
Assertion of Ready RDY during write operations.	Ignored.	Stops processor during ϕ_2.
Unused input-only pins ($\overline{IRQ}$, $\overline{NMI}$, RDY, $\overline{RES}$, $\overline{SO}$).	Must be connected to low impedance signal to avoid noise problems.	Connected internally by a high-resistance to V_{DD} (approximately 250K ohm).

New Instruction Mnemonics

HEX	Mnemonic	Description
80	BRA	Branch relative always [Relative]
3A	DEA	Decrement accumulator [Accum]
1A	INA	Increment accumulator [Accum]
DA	PHX	Push X on stack [Implied]
5A	PHY	Push Y on stack [Implied]
FA	PLX	Pull X from stack [Implied]
7A	PLY	Pull Y from stack [Implied]
9C	STZ	Store zero [Absolute]
9E	STZ	Store zero [ABS, X]
64	STZ	Store zero [Zero Page]
74	STZ	Store zero [ZPG, X]
1C	TRB	Test and reset memory bits with accumulator [Absolute]
14	TRB	Test and reset memory bits with accumulator [Zero page]
0C	TSB	Test and set memory bits with accumulator [Absolute]
04	TSB	Test and set memory bits with accumulator [Zero page]
89	BIT	Test immediate with accumulator [IMMEDIATE]

Additional Instruction Addressing Modes

HEX	Mnemonic	Description
72	ADC	Add memory to accumulator with carry [(ZPG)]
32	AND	"AND" memory with accumulator [(ZPG)]
3C	BIT	Test memory bits with accumulator[ABS, X]
34	BIT	Test memory bits with accumulator [ZPG, X]
D2	CMP	Compare memory and accumulator [(ZPG)]
52	EOR	"Exclusive OR" memory with accumulator [(ZPG)]
7C	JMP	Jump (New addressing mode) [ABS(IND, X)]
B2	LDA	Load accumulator with memory [(ZPG)]
12	ORA	"OR" memory with accumulator [(ZPG)]
F2	SBC	Subtract memory from accumulator with borrow [(ZPG)]
92	STA	Store accumulator in memory [(ZPG)]

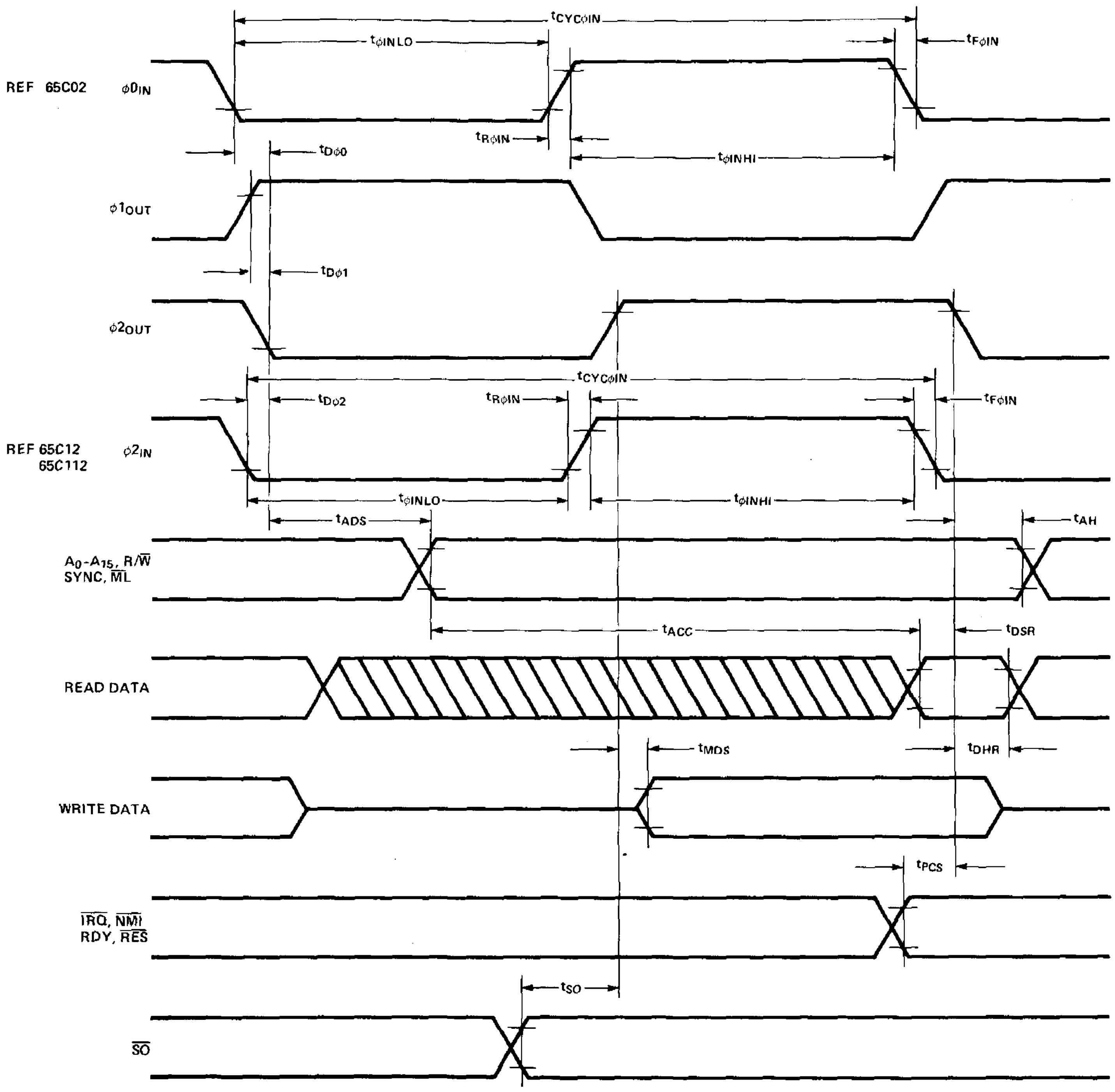


Figure 2. AC Characteristics, SY65C02

AC Characteristics, SY65C02 V_{DD} = 5.0 V ± 5%, T_A = -40°C to +85°C

Parameter	Symbol	1 MHz		2 MHz		3 MHz		4 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
Delay Time, ϕ0 (IN) to ϕ2 (OUT)	$t_{D\phi 0}$	—	100	—	100	—	100	—	100	ns
Delay Time, ϕ2 (IN) to ϕ2 (OUT)	$t_{D\phi 2}$	—	75	—	75	—	75	—	75	ns
Delay Time, ϕ1 (OUT) to ϕ2 (OUT)	$t_{D\phi 1}$	—	50	—	50	—	50	—	50	ns
Cycle Time	$t_{CYC\phi IN}$	1.0	DC	0.50	DC	0.33	DC	0.25	DC	μs
Clock Pulse Width Low	$t_{PW(\phi)INLO}$	470	—	240	—	160	—	115	—	ns
Clock Pulse Width High	$t_{PW(\phi)INHI}$	470	—	240	—	160	—	115	—	ns
Fall Time, Rise Time	$t_{F\phi IN}$, $t_{R\phi IN}$	—	25	—	25	—	15	—	15	ns
Address Hold Time	t_{AH}	30	—	30	—	15	—	10	—	ns
Address Setup Time	t_{ADS}	—	225	—	140	—	110	—	90	ns
Access Time	t_{ACC}	650	—	310	—	170	—	110	—	ns
Read Data Hold Time	t_{DHR}	10	—	10	—	10	—	10	—	ns
Read Data Setup Time	t_{DSR}	100	—	50	—	50	—	50	—	ns
Write Data Delay Time	t_{MDS}	—	175	—	100	—	75	—	70	ns
Write Data Hold Time	t_{DHW}	30	—	30	—	30	—	30	—	ns
$\overline{SO}$ Setup Time	t_{SO}	100	—	50	—	35	—	25	—	ns
Processor Control Setup Time	t_{PCS}	200	—	200	—	150	—	120	—	ns

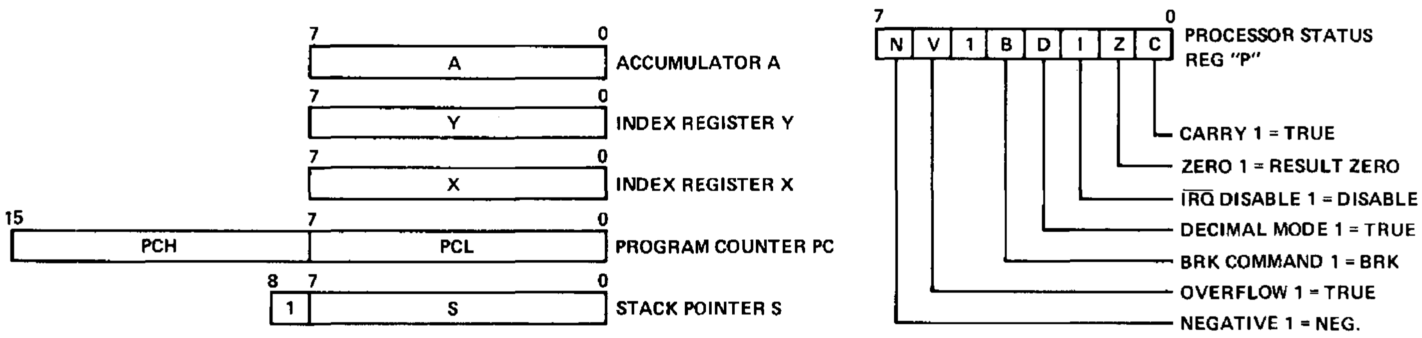


Figure 4. Microprocessor Programming Model

Functional Description

Timing Control

The timing control unit keeps track of the instruction cycle being monitored. The unit is set to zero each time an instruction fetch is executed and is advanced at the beginning of each phase one clock pulse for as many cycles as is required to complete the instruction. Each data transfer which takes place between the registers depends upon decoding the contents of both the instruction register and the timing control unit.

Program Counter

The 16-bit program counter provides the addresses which step the microprocessor through sequential instructions in a program.

Each time the microprocessor fetches an instruction from program memory, the lower byte of the program counter (PCL) is placed on the low-order bits of the address bus and the higher byte of the program counter (PCH) is placed on the high-order 8 bits. The counter is incremented each time an instruction or data is fetched from program memory.

Instruction Register and Decode

Instructions fetched from memory are gated onto the internal data bus. These instructions are latched into the instruction register, then decoded, along with timing and interrupt signals, to generate control signals for the various registers.

Arithmetic and Logic Unit (ALU)

All arithmetic and logic operations take place in the ALU including incrementing and decrementing internal registers (except the program counter). The ALU has no internal memory and is used only to perform logical and transient numerical operations.

Accumulator

The accumulator is a general purpose 8-bit register that stores the results of most arithmetic and logic operations, and in addition, the accumulator usually contains one of the two data words used in these operations.

Index Registers

There are two 8-bit index registers (X and Y), which may be used to count program steps or to provide an index value to be used in generating an effective address.

When executing an instruction which specifies indexed addressing, the CPU fetches the op code and the base address, and modifies the address by adding the index register to it prior to performing the desired operation. Pre-or post-indexing of indirect addresses is possible (see addressing modes).

Stack Pointer

The stack pointer is an 8-bit register used to control the addressing of the variable-length stack on page one. The stack pointer is automatically incremented and decremented under control of the microprocessor to perform stack manipulations under direction of either the program or interrupts ($\overline{NMI}$ and $\overline{IRQ}$). The stack allows simple implementation of nested subroutines and multiple level interrupts. The stack pointer should be initialized before any interrupts or stack operations occur.

Processor Status Register

The 8-bit processor status register contains seven status flags. Some of the flags are controlled by the program, others may be controlled both by the program and the CPU. The 6500 instruction set contains a number of conditional branch instructions which are designed to allow testing of these flags (see microprocessor programming model).

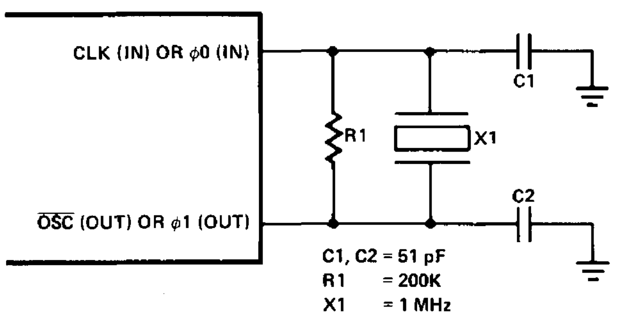


Figure 5 (a). Crystal Circuit for Internal Oscillator

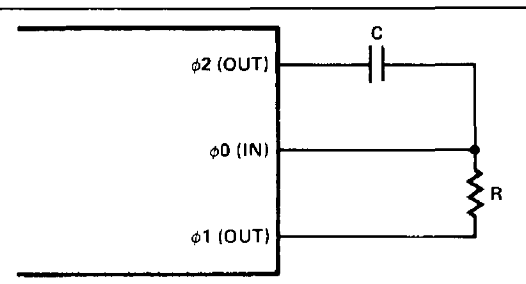


Figure 5 (b). Suggested RC Network Configuration for Internal Oscillator

Addressing Modes

Fifteen addressing modes are available to the user of the SY65C02 microprocessor. The addressing modes are described in the following paragraphs:

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

Accumulator Addressing (Accum)
This form of addressing is represented with a one byte instruction and implies an operation on the accumulator.

Immediate Addressing (Immediate)
With immediate addressing, the operand is contained in the second byte of the instruction; no further memory addressing is required.

Absolute Addressing (Absolute)
For absolute addressing, the second byte of the instruction specifies the eight low-order bits of the effective address, while the third byte specifies the eight high-order bits. Therefore, this addressing mode allows access to the total 64K bytes of addressable memory.

Zero Page Addressing (Zero Page)
Zero page addressing allows shorter code and execution times by only fetching the second byte of the instruction and assuming a zero high address byte. The careful use of zero page addressing can result in significant increase in code efficiency.

Absolute Indexed Addressing (ABS, X or ABS, Y)
Absolute indexed addressing is used in conjunction with X or Y index register and is referred to as "Absolute, X," and "Absolute, Y." The effective address is formed by adding the contents of X or 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.

Zero Page Indexed Addressing (ZPG, X or ZPG, Y)
Zero page absolute 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 eight bits of memory, and crossing of page boundaries does not occur.

Relative Addressing (Relative)
Relative addressing is used only with branch instructions; it 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.

Zero Page Indexed Indirect Addressing [(IND, X)]
With zero page indexed indirect addressing (usually referred to as indirect X) the second byte of the instruction is added to the contents of the X index register; the carry is discarded. 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.

***Absolute Indexed Indirect Addressing [ABS(IND, X)] (Jump Instruction Only)**
With absolute indexed indirect addressing the contents of the second and third instruction bytes are added to the X register. The result of this addition, points to a memory location containing the lower-order eight bits of the effective address. The next memory location contains the higher-order eight bits of the effective address.

Indirect Indexed Addressing [(IND), Y]
This form of addressing is usually 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 are 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.

***Zero Page Indirect Addressing [(ZPG)]**
In the zero page indirect addressing mode, the second byte of the instruction points to a memory location on page zero containing the low-order byte of the effective address. The next location on page zero contains the high-order byte of the effective address.

Absolute Indirect Addressing [(ABS)] (Jump Instruction Only)
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 16 bit program counter.

NOTE: * = New Address Modes

Signal Description

Address Bus (A_0-A_{15})
A_0-A_{15} forms a 16-bit address bus for memory and I/O exchanges on the data bus. The output of each address line is TTL compatible, capable of driving one standard TTL load and 130 pF.

Clocks (ϕ_0, ϕ_1, and ϕ_2)
ϕ_0 is a TTL level input that is used to generate the internal clocks in the 6502. Two full level output clocks are generated by the 6502. The ϕ_2 clock output is in phase with ϕ_0. The ϕ_1 output pin is 180° out of phase with ϕ_0. (See timing diagram.)

Data Bus (D_0-D_7)
The data lines (D_0-D_7) constitute an 8-bit bidirectional data bus used for data exchanges to and from the device and peripherals. The outputs are three-state buffers capable of driving one TTL load and 130 pF.

Interrupt Request ($\overline{IRQ}$)
This TTL compatible input requests that an interrupt sequence begin within the microprocessor. The $\overline{IRQ}$ is sampled during ϕ_2 operation; if the interrupt flag in the processor status register is zero, the current instruction is completed and the interrupt sequence begins during ϕ_1. 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 $\overline{IRQ}$s 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, transferring program control to the memory vector located at these addresses. The RDY signal must be in the high state for any interrupt to be recognized. A 3K ohm external resistor should be used for proper wire OR operation.

Memory Lock ($\overline{ML}$)
In a multiprocessor system, the $\overline{ML}$ output indicates the need to defer the rearbitration of the next bus cycle to ensure the integrity of read-modify-write instructions. $\overline{ML}$ goes low during ASL, DEC, INC, LSR, ROL, ROR, TRB, TSB memory referencing instructions. This signal is low for the modify and write cycles.

Non-Maskable Interrupt ($\overline{NMI}$)
A negative-going edge on this input requests that a non-maskable interrupt sequence be generated within the microprocessor. The $\overline{NMI}$ is sampled during ϕ_2; the current instruction is completed and the interrupt sequence begins during ϕ_1. The program counter is loaded with the interrupt vector from locations FFFA (low byte) and FFFB (high byte), thereby transferring program control to the non-maskable interrupt routine.

NOTE: Since this interrupt is non-maskable, another $\overline{NMI}$ can occur before the first is finished. Care should be taken when using $\overline{NMI}$ to avoid this.

Ready (RDY)
This input allows the user to single-cycle the microprocessor on all cycles including write cycles. A negative transition to the low state, during or coincident with phase one (ϕ_1), will halt the microprocessor with the output address lines reflecting the current address being fetched. This condition will remain through a subsequent phase two (ϕ_2) in which the ready signal is low. This feature allows microprocessor interfacing with low-speed memory as well as direct memory access (DMA).

Reset ($\overline{RES}$)
This input is used to reset the microprocessor. Reset must be held low for at least two clock cycles after V_{DD} reaches operating voltage from a power down. A positive transition on this pin will then cause an initialization sequence to begin. Likewise, after the system has been operating, a low on this line of at least two cycles will cease microprocessing activity, followed by initialization after the positive edge on $\overline{RES}$.

When a positive edge is detected, there is an initialization sequence lasting six clock cycles. Then the interrupt mask flag is set, the decimal mode is cleared, and the program counter is loaded with the restart vector from locations FFFC (low byte) and FFFD (high byte). This is the start location for program control. This input should be high in normal operation.

Read/Write (R/$\overline{W}$)
This signal is normally in the high state indicating that the microprocessor is reading data from memory or I/O bus. In the low state the data bus has valid data from the microprocessor to be stored at the addressed memory location.

Set Overflow ($\overline{SO}$)
A negative transition on this line sets the overflow bit in the status code register. The signal is sampled on the trailing edge of ϕ_1.

Synchronize (SYNC)
This output line is provided to identify those cycles during which the microprocessor is doing an OP CODE fetch. The SYNC line goes high during ϕ_1 of an OP CODE fetch and stays high for the remainder of that cycle. If the RDY line is pulled low during the ϕ_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.

Instruction Set — Alphabetical Sequence

ADC Add Memory to Accumulator with Carry
AND "AND" Memory with Accumulator
ASL Shift One Bit Left
BCC Branch on Carry Clear
BCS Branch on Carry Set
BEQ Branch on Result Zero
BIT Test Memory Bits with Accumulator
BMI Branch on Result Minus
BNE Branch on Result Not Zero
BPL Branch on Result Plus
● BRA Branch Always
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 by One
DEX Decrement Index X by One
DEY Decrement Index Y by One
EOR "Exclusive-or" Memory with Accumulator
INC Increment 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
NOP No Operation
ORA "OR" Memory with Accumulator
PHA Push Accumulator on Stack
PHP Push Processor Status on Stack
● PHX Push Index X on Stack
● PHY Push Index Y on Stack
PLA Pull Accumulator from Stack
PLP Pull Processor Status from Stack
● PLX Pull Index X from Stack
● PLY Pull Index Y from Stack
ROL Rotate One Bit Left
ROR Rotate One Bit Right
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 Bit
STA Store Accumulator in Memory
STX Store Index X in Memory
STY Store Index Y in Memory
● STZ Store Zero in Memory
TAX Transfer Accumulator to Index X
TAY Transfer Accumulator to Index Y
● TRB Test and Reset Memory Bits with Accumulator
● TSB Test and Set Memory Bits with Accumulator
TSX Transfer Stack Pointer to Index X
TXA Transfer Index X to Accumulator
TXS Transfer Index X to Stack Pointer
TYA Transfer Index Y to Accumulator

Note: ● = New Instruction

MSD \ LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRK	ORA ind, X			**TSB zpg**	ORA zpg	ASL zpg		PHP	ORA imm	ASL A		**TSB abs**	ORA abs	ASL abs		0
1	BPL rel	ORA ind, Y	**ORA ind**		**TRB zpg**	ORA zpg, X	ASL zpg, X		CLC	ORA abs, Y	**INC A**		**TRB abs**	ORA abs, X	ASL abs, X		1
2	JSR abs	AND ind, X			BIT zpg	AND zpg	ROL zpg		PLP	AND imm	ROL A		BIT abs	AND abs	ROL abs		2
3	BMI rel	AND ind, Y	**AND ind**		**BIT zpg,X**	AND zpg, X	ROL zpg, X		SEC	AND abs, Y	**DEC A**		**BIT abs,X**	AND abs, X	ROL abs, X		3
4	RTI	EOR ind, X				EOR zpg	LSR zpg		PHA	EOR imm	LSR A		JMP abs	EOR abs	LSR abs		4
5	BVC rel	EOR ind, Y	**EOR ind**			EOR zpg, X	LSR zpg, X		CLI	EOR abs, Y	**PHY**			EOR abs, X	LSR abs, X		5
6	RTS	ADC ind, X			**STZ zpg**	ADC zpg	ROR zpg		PLA	ADC imm	ROR A		JMP ind	ADC abs	ROR abs		6
7	BVS rel	ADC ind, Y	**ADC ind**		**STZ zpg,X**	ADC zpg, X	ROR zpg, X		SEI	ADC abs, Y	**PLY**		**JMP ind,X**	ADC abs, X	ROR abs, X		7
8	**BRA rel**	STA ind, X			STY zpg	STA zpg	STX zpg		DEY	**BIT imm**	TXA		STY abs	STA abs	STX abs		8
9	BCC rel	STA ind, Y	**STA ind**		STY zpg, X	STA zpg, X	STX zpg, Y		TYA	STA abs, Y	TXS		**STZ abs**	STA abs, X	**STZ abs,X**		9
A	LDY imm	LDA ind, X	LDX imm		LDY zpg	LDA zpg	LDX zpg		TAY	LDA imm	TAX		LDY abs	LDA abs	LDX abs		A
B	BCS rel	LDA ind, Y	**LDA ind**		LDY zpg, X	LDA zpg, X	LDX zpg, Y		CLV	LDA abs, Y	TSX		LDY abs, X	LDA abs, X	LDX abs, Y		B
C	CPY imm	CMP ind, X			CPY zpg	CMP zpg	DEC zpg		INY	CMP imm	DEX		CPY abs	CMP abs	DEC abs		C
D	BNE rel	CMP ind, Y	**CMP ind**			CMP zpg, X	DEC zpg, X		CLD	CMP abs, Y	**PHX**			CMP abs, X	DEC abs, X		D
E	CPX imm	SBC ind, X			CPX zpg	SBC zpg	INC zpg		INX	SBC imm	NOP		CPX abs	SBC abs	INC abs		E
F	BEQ rel	SBC ind, Y	**SBC ind**			SBC zpg, X	INC zpg, X		SED	SBC abs, Y	**PLX**			SBC abs, X	INC abs, X		F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Note: ■ = New Op Codes

Figure 6. Microprocessor Op Code Table

Operational Codes, Execution Time, and Memory Requirements

Each addressing-mode cell gives OP n #.

MNE	OPERATION		IMMEDIATE	ABSOLUTE	ZERO PAGE	ACCUM	IMPLIED	(IND, X)	(IND, Y)	ZPG, X	ZPG, Y	ABS, X	ABS, Y	RELATIVE	(ABS)	ABS (IND, X)	(ZPG)	PROCESSOR STATUS CODES 7 6 5 4 3 2 1 0 N V . B D I Z C	MNE
ADC	A + M + C → A	(1,3)	69 2 2	6D 4 3	65 3 2			61 6 2	71 5 2	75 4 2		7D 4 3	79 4 3				72 5 2	N V . . . . Z C	ADC
AND	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				32 5 2	N . . . . . Z .	AND
ASL	C ← [7 … 0] ← 0	(1)		0E 6 3	06 5 2	0A 2 1				16 6 2		1E 6 3						N . . . . . Z C	ASL
BCC	Branch if C=0	(2)												90 2 2				. . . .	BCC
BCS	Branch if C=1	(2)												B0 2 2				. . . .	BCS
BEQ	Branch if Z=1	(2)												F0 2 2				. . . .	BEQ
BIT	A ∧ M	(4)	89 2 2	2C 4 3	24 3 2					34 4 2		3C 4 3						M_7M_6 . . . . Z .	BIT
BMI	Branch if N=1	(2)												30 2 2				. . . .	BMI
BNE	Branch if Z=0	(2)												D0 2 2				. . . .	BNE
BPL	Branch if N=0	(2)												10 2 2				. . . .	BPL
BRA	Branch Always	(2)												80 2 2				. . . .	BRA
BRK	Break						00 7 1											. . . 1 . 1 . .	BRK
BVC	Branch if V=0	(2)												50 2 2				. . . .	BVC
BVS	Branch if V=1	(2)												70 2 2				 0	BVS
CLC	0 → C						18 2 1											. . . . 0 . . .	CLC
CLD	0 → D						D8 2 1											. . . . 0 . . .	CLD
CLI	0 → I						58 2 1											. . . . . 0 . .	CLI
CLV	0 → V						B8 2 1											. 0	CLV
CMP	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				D2 5 2	N . . . . . Z C	CMP
CPX	X - M		E0 2 2	EC 4 3	E4 3 2													N . . . . . Z C	CPX
CPY	Y - M		C0 2 2	CC 4 3	C4 3 2													N . . . . . Z C	CPY
DEA	A - 1 → A					3A 2 1												N . . . . . Z .	DEA
DEC	M - 1 → M	(1)		CE 6 3	C6 5 2					D6 6 2		DE 6 3						N . . . . . Z .	DEC
DEX	X - 1 → X						CA 2 1											N . . . . . Z .	DEX
DEY	Y - 1 → Y						88 2 1											N . . . . . Z .	DEY
EOR	A ⊻ M → A		49 2 2	4D 4 3	45 3 2			41 6 2	51 5 2	55 4 2		5D 4 3	59 4 3				52 5 2	N . . . . . Z .	EOR
INA	A + 1 → A					1A 2 1												N . . . . . Z .	INA
INC	M + 1 → M	(1)		EE 6 3	E6 5 2					F6 6 2		FE 6 3						N . . . . . Z .	INC
INX	X + 1 → X						E8 2 1											N . . . . . Z .	INX
INY	Y + 1 → Y						C8 2 1											N . . . . . Z .	INY
JMP	Jump to new loc			4C 3 3											6C 6 3	7C 6 3		. . . .	JMP
JSR	Jump Subroutine			20 6 3														. . . .	JSR
LDA	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				B2 5 2	N . . . . . Z .	LDA
LDX	M → X	(1)	A2 2 2	AE 4 3	A6 3 2						B6 4 2		BE 4 3					N . . . . . Z .	LDX
LDY	M → Y	(1)	A0 2 2	AC 4 3	A4 3 2					B4 4 2		BC 4 3						N . . . . . Z .	LDY
LSR	0 → [7 … 0] → C	(1)		4E 6 3	46 5 2	4A 2 1				56 6 2		5E 6 3						0 . . . . . Z C	LSR
NOP	PC + 1 → PC						EA 2 1											. . . .	NOP
ORA	A V 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				12 5 2	N . . . . . Z .	ORA
PHA	A → M_S S - 1 → S						48 3 1											. . . .	PHA
PHP	P → M_S S - 1 → S						08 3 1											. . . .	PHP
PHX	X → M_S S - 1 → S						DA 3 1											. . . .	PHX
PHY	Y → M_S S - 1 → S						5A 3 1											. . . .	PHY
PLA	S + 1 → S M_S → A						68 4 1											N . . . . . Z .	PLA
PLP	S + 1 → S M_S → P						28 4 1											N V . 1 D I Z C	PLP
PLX	S + 1 → S M_S → X						FA 4 1											. . . .	PLX
PLY	S + 1 → S M_S → Y						7A 4 1											. . . .	PLY
ROL	← [7 … 0] ← C ←	(1)		2E 6 3	26 5 2	2A 2 1				36 6 2		3E 6 3						N . . . . . Z C	ROL
ROR	→ C → [7 … 0] →	(1)		6E 6 3	66 5 2	6A 2 1				76 6 2		7E 6 3						N . . . . . Z C	ROR
RTI	Return from Inter.						40 6 1											N V . 1 D I Z C	RTI
RTS	Return from Subr.						60 6 1											. . . .	RTS
SBC	A - M - $\overline{C}$ → A	(1,3)	E9 2 2	ED 4 3	E5 3 2			E1 6 2	F1 5 2	F5 4 2		FD 4 3	F9 4 3				F2 5 2	N V . . . . Z C	SBC
SEC	1 → C						38 2 1											 1	SEC
SED	1 → D						F8 2 1											. . . . 1 . . .	SED
SEI	1 → I						78 2 1											. . . . . 1 . .	SEI
STA	A → M			8D 4 3	85 3 2			81 6 2	91 6 2	95 4 2		9D 5 3	99 5 3				92 5 2	. . . .	STA
STX	X → M			8E 4 3	86 3 2						96 4 2							. . . .	STX
STY	Y → M			8C 4 3	84 3 2					94 4 2								. . . .	STY
STZ	00 → M			9C 4 3	64 3 2					74 4 2		9E 5 3						. . . .	STZ
TAX	A → X						AA 2 1											N . . . . . Z .	TAX
TAY	A → Y						A8 2 1											N . . . . . Z .	TAY
TRB	$\overline{A}$ ∧ M → M	(4)		1C 6 3	14 5 2													M_7M_6 . . . . Z .	TRB
TSB	A V M → M	(4)		0C 6 3	04 5 2													M_7M_6 . . . . Z .	TSB
TSX	S → X						BA 2 1											N . . . . . Z .	TSX
TXA	X → A						8A 2 1											N . . . . . Z .	TXA
TXS	X → S						9A 2 1											. . . .	TXS
TYA	Y → A						98 2 1											N . . . . . Z .	TYA

Notes:

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. Add 1 to "n" if decimal mode.
4. V bit equals memory bit 6 prior to execution.
   N bit equals memory bit 7 prior to execution.

- X Index X
- Y Index Y
- A Accumulator
- M Memory per effective address
- Ms Memory per stack pointer
- + Add
- − Subtract
- ∧ And
- V Or
- ⊻ Exclusive or
- n No. Cycles
- # No. Bytes
- M_6 Memory bit 6
- M_7 Memory bit 7

Package Availability 40 Pin Molded DIP

Ordering Information

SY P 65CXX X

PREFIX

PACKAGE
P — Molded DIP

DEVICE TYPE

SPEED RANGE
No Designator — 1 MHz
A — 2 MHz
B — 3 MHz
C — 4 MHz

APPENDIX A

SUMMARY OF SINGLE CYCLE EXECUTION

This section contains an outline of the data on both the address bus and the data bus for each cycle of the various processor instructions. It tells the system designer exactly what to expect while single cycling through a program.

Note that the processor will not stop in any cycle where R/W is a 0 (write cycle). Instead, it will go right into the next read cycle and stop there. For this reason, some instructions may appear to be shorter than indicated here.

All instructions begin with T0 and the fetch of the OP CODE and continue through the required number of cycles until the next T0 and the fetch of the next OP CODE.

While the basic terminology used in this appendix is discussed in the Programming Manual, it has been defined below for ease of reference while studying Single Cycle Execution.

OP CODE--The first byte of the instruction containing the operator and mode of address.

OPERAND--The data on which the operation specified in the OP CODE is performed.

BASE ADDRESS--The address in Indexed addressing modes which specifies the location in memory to which indexing is referenced. The high order of byte of the base address (AB08 to AB15) is BAH (Base Address High) and the low order byte of the base address (AB00 to AB07) is BAL (Base Address Low).

EFFECTIVE ADDRESS--The destination in memory in which data is to be found. The effective address may be loaded directly as in the case of Page Zero and Absolute Addressing or may be calculated as in Indexing operations. The high order byte of the effective address (AB08 to AB15) is ADH and the low order byte of the effective address (AB00 to AB07) is ADL.

INDIRECT ADDRESS--The address found in the operand of instructions utilizing (Indirect),Y which contains the low order byte of the base address. IAH and IAL represent the high and low order bytes.

JUMP ADDRESS--The value to be loaded into Program Counter as a result of a Jump instruction.

A. 1. SINGLE BYTE INSTRUCTIONS

ASL	DEX	NOP	TAX	TYA
CLC	DEY	ROL	TAY	
CLD	INX	SEC	TSX	
CLI	INY	SED	TXA	
CLV	LSR	SEI	TXS	

These single byte instructions require two cycles to execute. During the second cycle the address of the next instruction in program sequence will be placed on the address bus. However, the OP CODE which appears on the data bus during the second cycle will be ignored. This same instruction will be fetched on the following cycle at which time it will be decoded and executed. The ASL, ROL and LSR instructions apply to the accumulator mode of address.

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	OP CODE (Discarded)	1	
T0	PC + 1	OP CODE	1	Next Instruction

A. 2. INTERNAL EXECUTION ON MEMORY DATA

ADC	CMP	EOR	LDY
AND	CPX	LDA	ORA
BIT	CPY	LDX	SBC

The instructions listed above will execute by performing operations inside the microprocessor using data fetched from the effective address. This total operation requires three steps. The first step (one cycle) is the OP CODE fetch. The second (zero to four cycles) is the calculation of an effective address. The final step is the fetching of the data from the effective address. Execution of the instruction takes place during the fetching and decoding of the next instruction.

A. 2.1. Immediate Addressing (2 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Data	1	Fetch Data
T0	PC + 2	OP CODE	1	Next Instruction

A. 2.2. Zero Page Addressing (3 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Effective Address
T2	00, ADL	Data	1	Fetch Data
T0	PC + 2	OP CODE	1	Next Instruction

A. 2.3. Absolute Addressing (4 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order Effective Address byte
T2	PC + 2	ADH	1	Fetch high order Effective Address byte
T3	ADH, ADL	Data	1	Fetch Data
T0	PC + 3	OP CODE	1	Next Instruction

A. 2.4. Indirect, X Addressing (6 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	00, BAL + X	ADL	1	Fetch low order byte of Effective Address
T4	00, BAL + X + 1	ADH	1	Fetch high order byte of Effective Address
T5	ADH, ADL	Data	1	Fetch Data
T0	PC + 2	OP CODE	1	Next Instruction

A. 2.5. Absolute, X or Absolute, Y Addressing (4 or 5 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch low order byte of Base Address
T2	PC + 2	BAH	1	Fetch high order byte of Base Address
T3	ADL: BAL + index register	Data*	1	Fetch data (no page crossing)
	ADH: BAH + C			Carry is Ø or 1 as required from previous add operation
T4*	ADL: BAL + index register	Data	1	Fetch data from next page
	ADH: BAH + 1			
T0	PC + 3	OP CODE	1	Next Instruction

*If the page boundary is crossed in the indexing operation, the data fetched in T3 is ignored. If page boundary is not crossed, the T4 cycle is bypassed.

A. 2.6. Zero Page, X or Zero Page, Y Addressing Modes (4 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	00, BAL + index register	Data	1	Fetch Data (no page crossing)
T0	PC + 2	OP CODE	1	Next Instruction

A. 2.7. Indirect, Y Addressing Mode (5 or 6 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	IAL	1	Fetch Page Zero Indirect Address
T2	00, IAL	BAL	1	Fetch low order byte of Base Address
T3	00, IAL + 1	BAH	1	Fetch high order byte of Base Address
T4	ADL: BAL + Y	Data*	1	Fetch Data from same page
	ADH: BAH + C			Carry is 0 or 1 as required from previous add operation
T5*	ADL: BAL + Y	Data	1	Fetch Data from next page
	ADH: BAH + 1			
T0	PC + 2	OP CODE	1	Next Instruction

*If page boundary is crossed in indexing operation, the data fetch in T4 is ignored. If page boundary is not crossed, the T5 cycle is bypassed.

A. 3. STORE OPERATIONS

STA
STX
STY

The specific steps taken in the Store Operations are very similar to those taken in the previous group (Internal execution on memory data). However, in the Store Operation, the fetch of data is replaced by a WRITE (R/W = 0) cycle. No overlapping occurs and no shortening of the instruction time occurs on indexing operations.

A. 3.1. Zero Page Addressing (3 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Zero Page Effective Address
T2	00, ADL	Data	0	Write internal register to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 3.2. Absolute Addressing (4 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order byte of Effective Address
T2	PC + 2	ADH	1	Fetch high order byte of Effective Address
T3	ADH, ADL	Data	0	Write internal register to memory
T0	PC + 3	OP CODE	1	Next Instruction

A. 3.3. Indirect, X Addressing (6 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	00, BAL + X	ADL	1	Fetch low order byte of Effective Address
T4	00, BAL + X + 1	ADH	1	Fetch high order byte of Effective Address
T5	ADH, ADL	Data	0	Write internal register to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 3.4. Absolute, X or Absolute, Y Addressing (5 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch low order byte of Base Address
T2	PC + 2	BAH	1	Fetch high order byte of Base Address
T3	ADL: BAL + index register ADH: BAH + C	Data (Discarded)	1	
T4	ADH, ADL	Data	0	Write internal register to memory
T0	PC + 3	OP CODE	1	Next Instruction

A. 3.5. Zero Page, X or Zero Page, Y Addressing Modes (4 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	ADL: BAL + index register	Data	0	Write internal register to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 3.6. Indirect, Y Addressing Mode (6 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	IAL	1	Fetch Page Zero Indirect Address
T2	00, IAL	BAL	1	Fetch low order byte of Base Address
T3	00, IAL + 1	BAH	1	Fetch high order byte of Base Address
T4	ADL: BAL + Y	Data (Discarded)	1	
	ADH: BAH			
T5	ADH, ADL	Data	0	Write Internal Register to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 4. READ--MODIFY--WRITE OPERATIONS

ASL LSR
DEC ROL
INC ROR

The Read--Modify--Write operations involve the loading of operands from the operand address, modification of the operand and the resulting modified data being stored in the original location.

Note: The ROR instruction will be available on MCS650X microprocessors after June, 1976.

A. 4.1. Zero Page Addressing (5 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Page Zero Effective Address
T2	00, ADL	Data	1	Fetch Data
T3	00, ADL	Data	0	
T4	00, ADL	Modified Data	0	Write modified Data back to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 4.2. Absolute Addressing (6 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order byte of Effective Address
T2	PC + 2	ADH	1	Fetch high order byte of Effective Address
T3	ADH, ADL	Data	1	
T4	ADH, ADL	Data	0	
T5	ADH, ADL	Modified Data	0	Write modified Data back into memory
TØ	PC + 3	OP CODE	1	Next Instruction

A. 4.3. Zero Page, X Addressing (6 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded	1	
T3	ADL: BAL + X (without carry)	Data	1	Fetch Data
T4	ADL: BAL + X (without carry)	Data	0	
T5	ADL: BAL + X (without carry)	Modified Data	0	Write modified Data back into memory
TØ	PC + 2	OP CODE	1	Next Instruction

A. 4.4. Absolute, X Addressing (7 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch low order byte of Base Address
T2	PC + 2	BAH	1	Fetch high order byte of Base Address
T3	ADL: BAL + X	Data (Discarded)	1	
	ADH: BAH + C			
T4	ADL: BAL + X	Data	1	Fetch Data
	ADH: BAH + C			
T5	ADH, ADL	Data	0	
T6	ADH, ADL	Modified Data	0	Write modified Data back into memory
T0	PC + 3	OP CODE	1	New Instruction

A. 5. MISCELLANEOUS OPERATIONS

BCC	BRK	PHP
BCS	BVC	PLA
BEQ	BVS	PLP
BMI	JMP	RTI
BNE	JSR	RTS
BPL	PHA	

A. 5.1. Push Operation--PHP, PHA (3 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	OP CODE (Discarded)	1	
T2	Stack Pointer*	Data	0	Write Internal Register into Stack
T0	PC + 1	OP CODE	1	Next Instruction

*Subsequently referred to as "Stack Ptr."

A. 5.2. Pull Operations--PLP, PLA (4 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	OP CODE (Discarded)	1	
T2	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr. + 1	Data	1	Fetch Data from Stack
T0	PC + 1	OP CODE	1	Next Instruction

A. 5.3. Jump to Subroutine--JSR (6 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order byte of Subroutine Address
T2	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr.	PCH	0	Push high order byte of program counter to Stack
T4	Stack Ptr. - 1	PCL	0	Push low order byte of program counter to Stack
T5	PC + 2	ADH	1	Fetch high order byte of Subroutine Address
T0	Subroutine Address (ADH, ADL)	OP CODE	1	Next Instruction

A. 5.4. Break Operation--(Hardware Interrupt)-BRK (7 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch BRK OP CODE (or force BRK)
T1	PC + 1 (PC on hardware interrupt)	Data (Discarded)	1	
T2	Stack Ptr.	PCH	0	Push high order byte of program counter to Stack
T3	Stack Ptr. - 1	PCL	0	Push low order byte of program counter to Stack
T4	Stack Ptr. - 2	P	0	Push Status Register to Stack
T5	FFFE (NMI-FFFA) (RES-FFFC)	ADL	1	Fetch low order byte of interrupt vector
T6	FFFF (NMI-FFFB) (RES-FFFD)	ADH	1	Fetch high order byte of interrupt vector
T0	Interrupt Vector (ADH, ADL)	OP CODE	1	Next Instruction

A. 5.5. Return from Interrupt-RTI (6 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Data (Discarded)	1	
T2	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr. + 1	Data	1	Pull P from Stack
T4	Stack Ptr. + 2	Data	1	Pull PCL from Stack
T5	Stack Ptr. + 3	Data	1	Pull PCH from Stack
T0	PCH, PCL	OP CODE	1	Next Instruction

A. 5.6. Jump Operation--JMP

A.5.6.1. Absolute Addressing Mode (3 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order byte of Jump Address
T2	PC + 2	ADH	1	Fetch high order byte of Jump Address
T0	ADH, ADL	OP CODE	1	Next Instruction

A.5.6.2. Indirect Addressing Mode (5 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	IAL	1	Fetch low order byte of Indirect Address
T2	PC + 2	IAH	1	Fetch high order byte of Indirect Address
T3	IAH, IAL	ADL	1	Fetch low order byte of Jump Address
T4	IAH, IAL + 1	ADH	1	Fetch high order byte of Jump Address
T0	ADH, ADL	OP CODE	1	Next Instruction

A. 5.7. Return from Subroutine--RTS (6 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Data (Discarded)	1	
T2	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr. + 1	PCL	1	Pull PCL from Stack
T4	Stack Ptr. + 2	PCH	1	Pull PCH from Stack
T5	PCH, PCL (from Stack)	Data (Discarded)	1	
T0	PCH, PCL + 1	OP CODE	1	Next Instruction

A. 5.8. Branch Operation--BCC, BCS, BEQ, BMI, BNE, BPL, BVC, BVS (2, 3, or 4 cycles)

Tn	Address Bus	Data Bus	R/W	Comments
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Offset	1	Fetch Branch Offset
T2*	PC + 2 + offset (w/o carry)	OP CODE	1	Offset Added to Program Counter
T3**	PC + 2 + offset (with carry)	OP CODE	1	Carry Added

*Skip if branch not taken

**Skip if branch not taken; skip if branch operation doesn't cross page boundary.