
MICROCOMPUTERS
MICROCOMPUTERS
MICROCOMPUTERS
MICROCOMPUTERS
SPECTRUM OF PRODUCTS

FINALLY!

a microcomputer family that delivers on all the buzzwords !!

- ✔ Lowest Cost
- ✔ Highest Performance
- ✔ Ease of Use
- ✔ Upward Expansion
- ✔ Best Addressing
- ✔ Single Supply
- ✔ Simple I/O
- ✔ Modular System
- ✔ Single Phase Clock

REV. - A

The MOS Approach

- - the third generation

The MCS6500 product line is a third generation microcomputer family combining the best features of second generation families into a product line that is both a price and performance leader.

The application of advanced but proven fabrication techniques plus superior architectural design allow for incorporation of advanced addressing features (a limitation of all second generation microprocessors such as the MC6800, 8080 and F8) in chips of smaller size allowing for expanded performance at lowest cost.

A unique feature of the MCS6500 product line is the *Microprocessor Family* which includes a range of software compatible processors with performance/cost compatibilities ranging from 8 bit microprocessors which are price competitive with the lowest cost 4-bit machines up to microprocessors which outperform the best of today's 16 bit products.

To allow for minimum I/O cost and maximum user flexibility all of the MCS6500 products are compatible with the M6800 bus structure.

COMPONENTS OF THE MICROCOMPUTER FAMILY

THE MICROPROCESSOR FAMILY – MCS6500

MCS6502 - Microprocessor Unit with On-The-Chip Clock (40-pin package)
- *External single phase input
- *RC time base input
- *Crystal time base input
- *65K Addressable bytes of memory

MCS6503 - Microprocessor Unit (28-pin package)
- *Two interrupts
- *4K Addressable bytes of memory
- *On-the-chip clock

MCS6504 - Microprocessor Unit (28-pin package)
- *One interrupt
- *8K Addressable bytes of memory
- *On-the-chip clock

MCS6405 - Micrprocessor Unit (28-pin package)
- *One interrupt
- *4K Addressable bytes of memory
- *On-the-chip clock
- *RDY Signal

MCS6506 - Microprocessor Unit (28-pin package)
- *One interrupt
- *4K Addressable bytes of memory
- *On-the-chip clock
- *Two phases off

MCS6512 - Microprocessor Unit with Two Phase Input (40-pin package)
- *Two phase clock inputs
- *65K Addressable bytes of memory

MCS6513 - Microprocessor Unit (28-pin package)
- *Two interrupts
- *4K Addressable bytes of memory
- *Two phase clock input

MCS6514 - Microprocessor Unit (28-pin package)
- *One interrupt
- *8K Addressable bytes of memory
- *Two phase clock input

MCS6515 - Microprocessor Unit (28-pin package)
- *One interrupt
- *4K Addressable bytes of memory
- *RDY Signal
- *Two phase clock input

MEMORY AND I/O SUPPORT DEVICES

MCS6530 - "Combo" Chip # 1
- *RAM - 64 x 8
- *ROM - 1024 x 8
- *I/O - 16 Bidirectional pins
- *Interval timer

MCS6532 - "Combo" Chip # 2
- *RAM - 128 x 8
- *I/O - 16 Bidirectional pins
- *Interval timer

MCS6520 - Peripheral Interface Adapter

MCS6522 - Versatile Interface Adapter
- *Features of MCS6520
- *Register for serial operations
- *Dual programmable timers
- *Latching on peripheral data ports

MCS6102 - RAM (2102 type), 1024 x 1
- *No hold time
- *Faster access time
- *Microprocessor compatible

MCS6111 - RAM (2111 type), 256 x 4
- *No hold time
- *Faster access time
- *Microprocessor compatible

MCS6540 - 2048 x 8 ROM
- * Fast access time
- *Microprocessor compatible

SOFTWARE SUPPORT

- *Cross Assembler* - Allows programming of the microprocessor at symbolic assembly level. Provides extensive error checking and cross referencing information and is available on various time-sharing services, as well as available for purchase by the user.
- *Emulator* - Utilizing an input file generated from the Cross Assembler, the emulator determines the viability of operation and calculates the timing of sections of code. "Fortran like" control facilitates rapid verification of coding in either interactive or batch mode entry as well as time-sharing.

SYSTEM DEVELOPMENT SUPPORT

- *Microcomputer Development Terminal* - **MDT**
  A high level development tool for modeling the system, the MDT allows the user to utilize a modular and flexible technique to verify the system design before committing to a finalized design. Interactive control allows the user to assemble programs, debug software through on-line editing capability and to program PROM's when the design is deemed correct. Interface to the MDT can be either through the included keyboard/display or via a provided port for use with teletype or higher speed peripherals. This unit contains a resident assembler to allow the user to engage this system as his only development tool. A standard option card allows for interactive debugging of pre-production and final production systems.
- *Teletype Input/Output Monitor* - **TIM**
  An MCS6530 programmed to allow the user to input data directly into memory from teletype, to read memory via teletype, to load the output from the cross assembler into memory and to initiate programs from any memory location. Capability includes loading and writing from a high speed port.
- *Keyboard Input/Output Monitor* - **KIM-1**
  Two MCS6530's programmed to allow the user to interface to his own applications via a keyboard, alphanumeric display, TTY and audio cassette.

DOCUMENTATION

- *Hardware Manual*
  Contains a detailed discussion of all chips in the family, how they interface, how the peripherals are controlled as well as design techniques to facilitate system operation, testing and maintenance. Specific emphasis is placed on "bringing up" a system, including testing techniques, scope synchronizing and general procedures used in trouble-shooting a system.
- *Programming Manual*
  Defines the architecture of the MCS6500 products. Defines each instruction and its effect on the internal registers with particular emphasis on the sophisticated addressing modes utilized in the MCS6500 family. Contains detailed information on programming the MCS6500 products.
- *Cross Assembler Manual*
  Discusses the concept of assembler directives and the use of the assembler in time-share and batch operations. Particular emphasis is placed on understanding and resolving error messages.
- *Emulator Manual*
  Discusses emulation techniques with emphasis on understanding errors and the process of resolving hardware and software problems using both the time-share and batch operations. This includes leading the user through the evolution of a sample program.
- *MDT Manual*
  Instructs the user in the operation of the MDT System and its application in developing a working microprocessor system.
- *TIM Manual*
  Defines how to apply the Teletype Input/Output Monitor to developing microprocessor systems.
- *KIM Manual*
  Defines how to apply the Keyboard Input/Output Monitor to developing microprocessor systems.

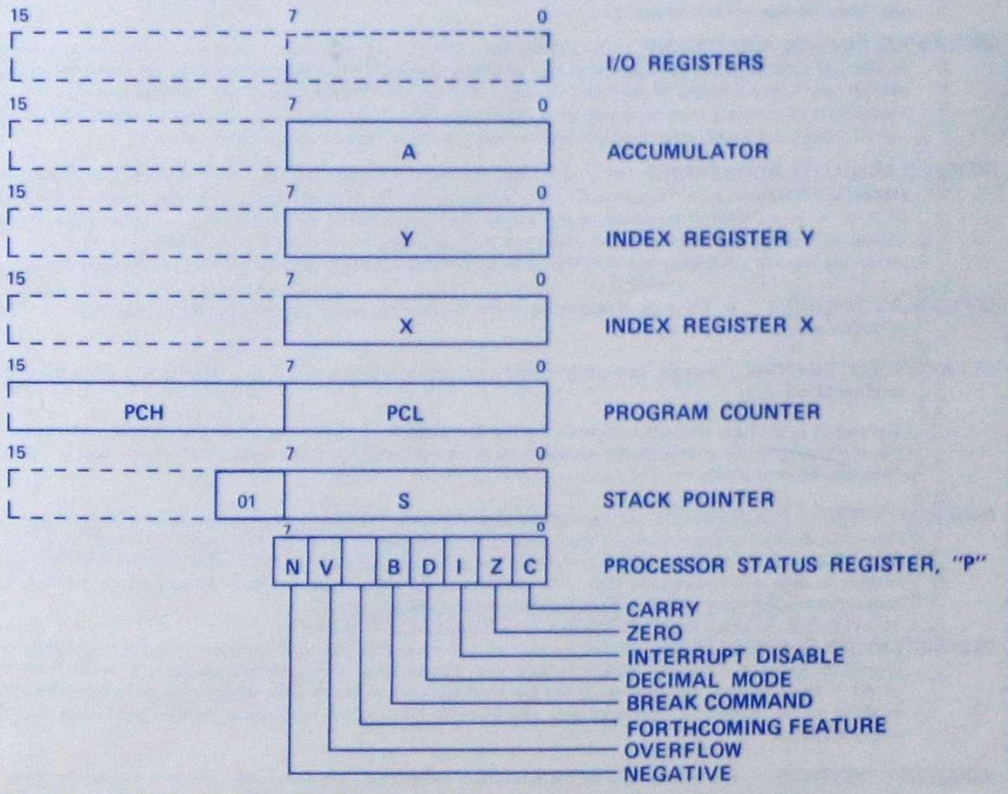
MCS6500

the first Microprocessor family

FEATURES OF THE FAMILY

- Low Cost
- 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
- On-the-Chip clock options
  - External single clock input
  - RC Time base input
  - Crystal time base input
- 40 and 28 pin package versions
- Pipeline Architecture
- Single 5 Volt Supply

PROGRAMMING MODEL MCS6500



* Solid line indicates currently available features
Dashed line indicates forthcoming members of family

POWERFUL ADDRESSING

Thirteen Addressing Modes including True Indexing

The MCS 6500 have thirteen modes of address. The first byte of each instruction is the operation code and specifies both the instruction and the addressing mode. The following table defines the addressing modes pertinent to each instruction with the corresponding execution time in clock cycles, where a 2 MHz clock frequency implies an 0.5 microsecond cycle.

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 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 ADDRESSING MODES AND RELATED EXECUTION TIMES (in clock cycles)

	Accumulator	Immediate	Zero Page	Zero Page, X	Zero Page, Y	Absolute	Absolute, X	Absolute, Y	Implied	Relative	(Indirect, X)	(Indirect), Y	Absolute Indirect
ADC	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
AND	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
ASL	2	.	5	6	.	6	7	.	.	.	.	.	.
BCC	.	.	.	.	.	.	.	.	.	2**	.	.	.
BCS	.	.	.	.	.	.	.	.	.	2**	.	.	.
BEQ	.	.	.	.	.	.	.	.	.	2**	.	.	.
BIT	.	.	3	.	.	4	.	.	.	.	.	.	.
BMI	.	.	.	.	.	.	.	.	.	2**	.	.	.
BNE	.	.	.	.	.	.	.	.	.	2**	.	.	.
BPL	.	.	.	.	.	.	.	.	.	2**	.	.	.
BRK	.	.	.	.	.	.	.	.	.	.	.	.	.
BVC	.	.	.	.	.	.	.	.	.	2**	.	.	.
BVS	.	.	.	.	.	.	.	.	.	2**	.	.	.
CLC	.	.	.	.	.	.	.	.	2	.	.	.	.
CLD	.	.	.	.	.	.	.	.	2	.	.	.	.
CLI	.	.	.	.	.	.	.	.	2	.	.	.	.
CLV	.	.	.	.	.	.	.	.	2	.	.	.	.
CMP	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
CPX	.	2	3	.	.	4	.	.	.	.	.	.	.
CPY	.	2	3	.	.	4	.	.	.	.	.	.	.
DEC	.	.	5	6	.	6	7	.	.	.	.	.	.
DEX	.	.	.	.	.	.	.	.	2	.	.	.	.
DEY	.	.	.	.	.	.	.	.	2	.	.	.	.
EØR	.	2	3	4	.	4	4*	4*	.	.	6	5	.
INC	.	.	5	6	.	6	7	.	.	.	.	.	.
INX	.	.	.	.	.	.	.	.	2	.	.	.	.
INY	.	.	.	.	.	.	.	.	2	.	.	.	.
JMP	.	.	.	.	.	3	.	.	.	.	.	.	5

	Accumulator	Immediate	Zero Page	Zero Page, X	Zero Page, Y	Absolute	Absolute, X	Absolute, Y	Implied	Relative	(Indirect, X)	(Indirect), Y	Absolute Indirect
JSR	.	.	.	.	.	6	.	.	.	.	.	.	.
LDA	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
LDX	.	2	3	.	4	4	.	4*	.	.	.	.	.
LDY	.	2	3	4	.	4	4*	.	.	.	.	.	.
LSR	2	.	5	6	.	6	7	.	.	.	.	.	.
NØP	.	.	.	.	.	.	.	.	2	.	.	.	.
ØRA	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
PHA	.	.	.	.	.	.	.	.	3	.	.	.	.
PHP	.	.	.	.	.	.	.	.	3	.	.	.	.
PLA	.	.	.	.	.	.	.	.	4	.	.	.	.
PLP	.	.	.	.	.	.	.	.	4	.	.	.	.
RØL	2	.	5	6	.	6	7	.	.	.	.	.	.
RØR	2	.	5	6	.	6	7	.	.	.	.	.	.
RTI	.	.	.	.	.	.	.	.	6	.	.	.	.
RTS	.	.	.	.	.	.	.	.	6	.	.	.	.
SBC	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
SEC	.	.	.	.	.	.	.	.	2	.	.	.	.
SED	.	.	.	.	.	.	.	.	2	.	.	.	.
SEI	.	.	.	.	.	.	.	.	2	.	.	.	.
STA	.	.	3	4	.	4	5	5	.	.	6	6	.
STX	.	.	3	.	4	4	.	.	.	.	.	.	.
STY	.	.	3	4	.	4	.	.	.	.	.	.	.
TAX	.	.	.	.	.	.	.	.	2	.	.	.	.
TAY	.	.	.	.	.	.	.	.	2	.	.	.	.
TSX	.	.	.	.	.	.	.	.	2	.	.	.	.
TXA	.	.	.	.	.	.	.	.	2	.	.	.	.
TXS	.	.	.	.	.	.	.	.	2	.	.	.	.
TYA	.	.	.	.	.	.	.	.	2	.	.	.	.

* Add one cycle if indexing across page boundary
** Add one cycle if branch is taken, Add one additional if branching operation crosses page boundary

MCS6500 MICROPROCESSOR 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
EØR	"Exclusive-or" Memory with Accumulator
INC	Increment Memory by One
INX	Increment X by One
INY	Increment 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)
NØP	No Operation
ØRA	"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
RØL	Rotate One Bit Left (Memory or Accumulator)
RØR	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 Pointer
TYA	Transfer Index Y to Accumulator

MOS TECHNOLOGY, INC.

MOS TECHNOLOGY, INC. · Otto-Hahn-Str. 40 · D-6382 Friedrichsdorf 2
Telefon (06175) 15 10 + 15 19 · Telex 04 10787 ipf