The MCS650X Microprocessor Family Concept

The MCS6501 - MCS6505 represent the first five members of the MCS650X 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. The family includes the 40 pin MCS6501 for clock compatibility with the MC6800 microprocessor, the 40 pin MCS6502 with the same features as the MCS6501 but including an on-chip clock, and the 28 pin MCS6503, 4 and 5 providing in addition to the on-chip clock a set of options allowing the user to tailor his microprocessor to suit the particular need. All of the microprocessors in the MCS6501 - MCS6505 group are software compatible within the group and are bus compatible with the M6800 product offering.

Features of the MCS6501 - MCS6505

- Single five volt supply
- N channel, silicon gate, depletion load technology
- Eight bit parallel processing
- 55 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

Members of the Family

**MCS6501 - 40 pin package**
- Compatible with MC6800
- 65K addressable bytes of memory

**MCS6502 - 40 pin package**
- 65K addressable bytes of memory
- On-the-chip clock
  - External single phase input
  - RC time base input
  - Crystal time base input

**MCS6503 - 28 pin package**
- On-the-chip clock
- 4K addressable bytes
- Two interrupts

**MCS6504 - 28 pin package**
- On-the-chip clock
- 8K addressable bytes
- One interrupt

**MCS6505 - 28 pin package**
- On-the-chip clock
- 4K addressable bytes
- One interrupt, RDY signal
Comments on the Data Sheet

This data sheet describes the first five members of the MCS650X microprocessor family. The data sheet is constructed to review first the basic "Common Characteristics" - those features which, unless specifically stated otherwise, are common to all of the MCS6501 - MCS6505 microprocessors. Subsequent to a review of the family characteristics will be sections devoted to each member of the group with specific features of each.

COMMON CHARACTERISTICS

MCS6501 - MCS6505 Internal Architecture
**COMMON CHARACTERISTICS**

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

**ELECTRICAL CHARACTERISTICS** (Vcc = 5.0V ± 5%, Vss = 0, TA = 25°C)

<table>
<thead>
<tr>
<th>CHARACTERISTIC</th>
<th>SYMBOL</th>
<th>MIN.</th>
<th>TYP.</th>
<th>MAX.</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input High Voltage</td>
<td>VIH</td>
<td>2.4</td>
<td>Vcc</td>
<td>-</td>
<td>Vdc</td>
</tr>
<tr>
<td>Input Low Voltage</td>
<td>VIL</td>
<td>0.3</td>
<td>Vss</td>
<td>-</td>
<td>Vdc</td>
</tr>
<tr>
<td>Input High Threshold Voltage</td>
<td>VIH</td>
<td>2.0</td>
<td>-</td>
<td>-</td>
<td>Vdc</td>
</tr>
<tr>
<td>Input Low Threshold Voltage</td>
<td>VIL</td>
<td>-</td>
<td>Vss</td>
<td>0.8</td>
<td>Vdc</td>
</tr>
<tr>
<td>Input Leakage Current</td>
<td>IL</td>
<td>-</td>
<td>100</td>
<td>0</td>
<td>uA</td>
</tr>
<tr>
<td>Three-State (Off State) Input Current</td>
<td>TSII</td>
<td>-</td>
<td>10</td>
<td>-</td>
<td>uA</td>
</tr>
<tr>
<td>Output High Voltage</td>
<td>VOH</td>
<td>Vcc</td>
<td>-</td>
<td>-</td>
<td>Vdc</td>
</tr>
<tr>
<td>Output Low Voltage</td>
<td>VOL</td>
<td>Vss</td>
<td>-</td>
<td>0.4</td>
<td>Vdc</td>
</tr>
<tr>
<td>Power Dissipation</td>
<td>PD</td>
<td>.25</td>
<td>.70</td>
<td>-</td>
<td>W</td>
</tr>
<tr>
<td>Capacitance</td>
<td>C</td>
<td>10</td>
<td>15</td>
<td>50</td>
<td>pF</td>
</tr>
<tr>
<td>Cycle Time</td>
<td>TCYC</td>
<td>1.0</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Clock Pulse Width</td>
<td>PWH</td>
<td>430</td>
<td>470</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Fall Time</td>
<td>TF</td>
<td>-</td>
<td>25</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Delay Time between Clocks</td>
<td>TD</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Read/Write Setup Time from MCS650X</td>
<td>TWRS</td>
<td>100</td>
<td>300</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Address Setup Time from MCS650X</td>
<td>TADS</td>
<td>200</td>
<td>300</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Memory Read Access Time</td>
<td>TACC</td>
<td>575</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Data Stability Time Period</td>
<td>TDUS</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Data Hold Time</td>
<td>TH</td>
<td>30</td>
<td>30</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Enable High Time for DBE Input</td>
<td>TER</td>
<td>470</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Data Setup Time from MCS650X</td>
<td>TMDS</td>
<td>150</td>
<td>200</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>RDY Setup Time</td>
<td>TRDY</td>
<td>470</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Bus Available Setup Time from MCS650X</td>
<td>TBA</td>
<td>350</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>SYNC Setup Time from MC1630X</td>
<td>TSYNC</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
</tbody>
</table>
COMMON CHARACTERISTICS

Two Phase Clock Timing - MCS6501

Timing for Reading Data from Memory or Peripherals

Timing for Writing Data to Memory or Peripherals
The MCS6501 requires a two phase non-overlapping clock that runs at the Vcc voltage level.
The MCS6502, 3, 4 and 5 clocks are supplied with an internal clock generator. The frequency of these clocks is externally controlled. Details of this feature are discussed in the MCS6502 portion of this data sheet.

Address Bus (A1-A15) (See sections on MCS6503, 4 and 5 for respective address lines on those devices.)
These outputs are TTL compatible, capable of driving one standard TTL load and 130pf.

Data Bus (D0-D7)
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 130pf.

Data Bus Enable (DBE) (MCS6501 only)
This TTL compatible input allows external control of the tri-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 (Q2) clock, thus allowing data output from microprocessor only during Q2. 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.

Ready (RDY) (MCS6501, MCS6502, MCS6503 only)
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 (Q1) will halt the microprocessor with the output address lines reflecting the current address being fetched. This condition will remain through a subsequent phase two (Q2) in which the Ready signal is high. 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.

Bus Available (BA) (MCS6501 only)
During normal operation the Bus Available signal will be in the low state, when in the high state it indicates that the microprocessor has stopped and that all buses are available. This situation will occur if the RDY signal is low and the microprocessor is not in a Write state.

Interrupt Request (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 FF9F, and program counter high from location FF7F, 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 (NMI) (MCS6501, MCS6502, MCS6503 only)
A negative going edge on this input requests that a non-maskable interrupt sequence be generated within the microprocessor.

NMI is an unconditional interrupt. Following completion of the current instruction, the sequence of operations defined for IRQ will be performed, regardless of the state interrupt mask flag. The vector address loaded into the program counter, low and high, are locations FF8F and FF9F respectively. The instructions loaded at these locations cause the microprocessor to branch to a non-maskable interrupt routine in memory. NMI also requires an external 3kΩ register to Vcc for proper wire-OR operations.

Inputs IRQ and NMI are hardware interrupts lines that are sampled during Q2 (phase 2) and will begin the appropriate interrupt routine on the Q2 (phase 1) following the completion of the current instruction.

Set Overflow Flag (S.O.) (MCS6502 only)
This TTL level input signal allows external control of the overflow bit in the Status Code Register.

SYNC (MCS6502 only)
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 Q2 of an OP CODE fetch and stays high for the remainder of that cycle. If the RDY line is pulled low during the Q2 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
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 FF8F and FF9F. 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 and (BA or 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.

MCS6501 – MCS6505 Signal Description
**INSTRUCTION SET – ALPHABETIC SEQUENCE**

<table>
<thead>
<tr>
<th>ADDRESSES</th>
<th>DEC</th>
<th>DEX</th>
<th>DMY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC Add Memory to Accumulator with Carry</td>
<td>Decrement Memory by One</td>
<td>Decrement Index X by One</td>
<td>Decrement Index Y by One</td>
</tr>
<tr>
<td>ADR “AND” Memory with Accumulator</td>
<td>DEX Decrement Index X by One</td>
<td>DMY Decrement Index Y by One</td>
<td></td>
</tr>
<tr>
<td>ASL Shift left One Bit (Memory or Accumulator)</td>
<td>EOR “Exclusive-or” Memory with Accumulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCC Branch on Carry Clear</td>
<td>INC Increment Memory by One</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCS Branch on Carry Set</td>
<td>INX Increment Index X by One</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSR Branch on Result Zero</td>
<td>INY Increment Index Y by One</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT Test Bits in Memory with Accumulator</td>
<td>JMP Jump to New Location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMI Branch on Result Negative</td>
<td>JSR Jump to New Location Saving Return Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPL Branch on Result Plus</td>
<td>LDA Load Accumulator with Memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BVC Branch on Overflow Clear</td>
<td>LDX Load Index X with Memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BVS Branch on Overflow Set</td>
<td>LDY Load Index Y with Memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLC Clear Carry Flag</td>
<td>LSR Shift One Bit Right (Memory or Accumulator)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLD Clear Decimal Mode</td>
<td>NOP No Operation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLI Clear Interrupt Disable Bit</td>
<td>ORA “Or” Memory with Accumulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLV Clear Overflow Flag</td>
<td>PLA Pull Accumulator from Stack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMP Compare Memory and Accumulator</td>
<td>PLP Pull Processor Status from Stack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPX Compare Memory and Index X</td>
<td>PUS Push Accumulator on Stack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPY Compare Memory and Index Y</td>
<td>PHP Push Processor Status on Stack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLY Pull Accumulator from Stack</td>
<td>PLA Pull Accumulator from Stack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTS Return from Subroutine</td>
<td>ROL Rotate One Bit Left (Memory or Accumulator)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI Return from Interrupt</td>
<td>ROR Rotate One Bit Right (Memory or Accumulator)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEC Set Carry Flag</td>
<td>RTS Return from Subroutine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEP Set Decimal Mode</td>
<td>SBC Subtract Memory from Accumulator with Borrow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIH Set Interrupt Enable Status</td>
<td>SEI Set Interrupt Disable Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STA Store Accumulator in Memory</td>
<td>SRET Return from Subroutine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STX Store Index X in Memory</td>
<td>STA Store Accumulator in Memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STY Store Index Y in Memory</td>
<td>TAX Transfer Accumulator to Index X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAX Transfer Accumulator to Index X</td>
<td>TAY Transfer Accumulator to Index Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSX Transfer Stack Pointer to Index X</td>
<td>TSX Transfer Index X to Stack Pointer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSB Transfer Index X to Accumulator</td>
<td>TXA Transfer Index X to Accu 1ator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TXS Transfer Index X to Stack Pointer</td>
<td>TXA Transfer Index X to Accumulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSX Transfer Index X to Stack Pointer</td>
<td>TXA Transfer Index X to Accumulator</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COMMON CHARACTERISTICS**

- **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 address of the second byte references an index at zero. Additionally, due to the "Zero Page" addressing nature of this mode, any carry 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 value or count value and the instruction to contain the base address. This type of addressing allows any location referencing the index to modify multiple fields resulting in reduced code 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 ADDRESSING**
  - In indirect address 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 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.
The MCS6501 is a bus compatible replacement for the MC6800 microprocessor. As such, this product uses a two phase high level (5 volt) clock input consistent with the MC6800 device requirements. The MCS6501 while pinout compatible with the M6800 address bus does not have the three-state buffers on the address pins. As such the MCS6501 always has valid addresses on the address bus, with this feature allowing the single cycle mode of operation.

Features of MCS6501

* 65K Addressable Bytes of Memory
* IRQ Interrupt
* NMI Interrupt
* RDY Signal (can be used for single cycling)

MCS6501 Pinout Designations

* VMA is connected internally to VCC. The VMA signal is not required on the MCS6501 as on the MC6800, since the MCS6501 always puts out known addresses on the address bus.
The MCS6502 combines the MC6800 bus compatibility features of the MCS6501 with an on-the-chip clock oscillator and driver which eliminates the requirement for a two phase 5 volt clock input. This feature allows the chip to be driven from a single TTL level input clock, or an RC time base or Crystal time base. Additionally, the MCS6502 has a SYNC line output which signals each time an OP CODE fetch is being performed, thus allowing single instruction execution.

**MCS6502 Pin Designations**

* 65K Addressable Bytes of Memory
* **IRQ** Interrupt
* **NMI** Interrupt
* On-the-chip Clock
  √ TTL Level Single Phase Input
  √ RC Time Base Input
  √ Crystal Time Base Input

**Features of MCS6502**

* **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
TIME BASE GENERATION OF INPUT CLOCK

CRYSTAL
(Suggested ranges for $R_F, C_F$: $0 < R_F < 500\, \text{k} \Omega$, $2\, \text{pf} < C_F < 12\, \text{pf}$.)

Parallel Mode Crystal Controlled Oscillator

Serial Mode Crystal Controlled Oscillator

RC

RC Network Time Base Generation
### Features of MCS6503

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

### Features of MCS6504

- *8K Addressable Bytes of Memory (AB00-AB12)*
- *On-the-chip Clock*
- *IRQ Interrupt*
- *8 Bit Bi-Directional Data Bus*

### Features of MCS6505

- *4K Addressable Bytes of Memory (AB00-AB11)*
- *On-the-chip Clock*
- *IRQ Interrupt*
- *RDY Signal*
- *8 Bit Bi-Directional Data Bus*
TIME BASE GENERATION OF INPUT CLOCK

CRYSTAL (Suggested ranges for $R_f$, $C_f$: $0 < R_f < 500 \Omega$, $2 \text{pF} < C_f < 12 \text{pF}$.)

Parallel Mode Crystal Controlled Oscillator

Serial Mode Crystal Controlled Oscillator

RC

RC Network Time Base Generation