



DISTINCTIVE CHARACTERISTICS

- Fixed point 16 and 32 bit operations
- Floating point 32 bit operations
- Binary data formats
- Add, Subtract, Multiply and Divide
- Trigonometric and inverse trigonometric functions
- Square roots, logarithms, exponentiation
- Float to fixed and fixed to float conversions
- Stack-oriented operand storage
- DMA or programmed I/O data transfers
- End signal simplifies concurrent processing
- General purpose 8-bit data bus interface
- Standard 24 pin package
- +12 volt and +5 volt power supplies
- Advanced N-channel silicon gate MOS technology
- 100% MIL-STD-883 reliability assurance testing

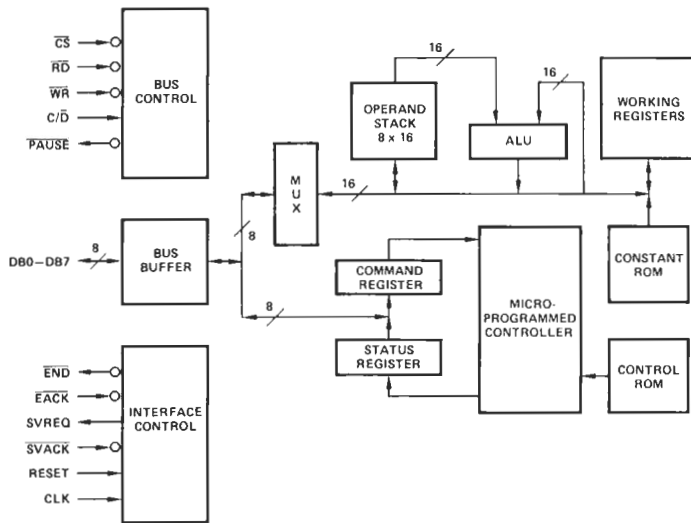
GENERAL DESCRIPTION

The Am9511 Arithmetic Processing Unit (APU) is a monolithic MOS/LSI device that provides high performance fixed and floating point arithmetic and a variety of floating point trigonometric and mathematical operations. It may be used to enhance the computational capability of a wide variety of processor-oriented systems.

All transfers, including operand, result, status and command information, take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack and a command is issued to perform operations on the data in the stack. Results are then available to be retrieved from the stack, or additional commands may be entered.

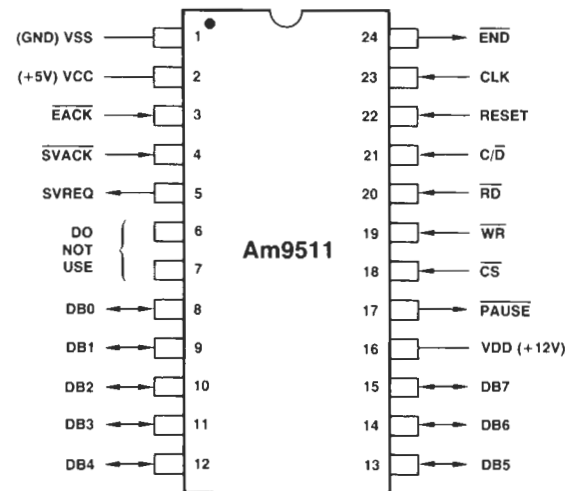
Transfers to and from the APU may be handled by the associated processor using conventional programmed I/O, or may be handled by a direct memory access controller for improved performance. Upon completion of each command, the APU issues an end of execution signal that may be used as an interrupt by the CPU to help coordinate program execution.

BLOCK DIAGRAM



MOS-046

CONNECTION DIAGRAM Top View



Pin 1 is marked for orientation.

MOS-047

ORDERING INFORMATION

Package Type	Ambient Temperature	Clock Frequency		
		2MHz	3MHz	4MHz
Hermetic DIP	$0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$	Am9511DC	Am9511-1DC	Am9511-4DC
	$-55^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$	Am9511DM	Am9511-1DM	

INTERFACE SIGNAL DESCRIPTION

VCC: +5 Volt power supply

VDD: +12 Volt power supply

VSS: Ground

CLK (Clock, Input)

An external timing source should be applied to the CLK pin. The Clock input may be asynchronous to the Read and Write control signals.

RESET (Reset, Input)

The active high Reset signal provides initialization for the chip. Reset terminates any operation in progress, clears the status register and places the Am9511 into the idle state. Stack contents are not affected by Reset. The Reset should be active for at least 5 clock periods following stable supply voltages and stable clock input. There is no internal power-on reset.

\overline{CS} (Chip Select, Input)

\overline{CS} is an active low input signal which conditions the read and write signals and thus enables communication with the data bus.

C/\overline{D} (Command/Data, Input)

In conjunction with the \overline{RD} and \overline{WR} signals, the C/\overline{D} control line establishes the type of transfers that are to be performed on the data bus.

C/\overline{D}	\overline{RD}	\overline{WR}	Function
0	1	0	Enter data byte into stack
0	0	1	Read data byte from stack
1	1	0	Enter command
1	0	1	Read status

\overline{RD} (Read, Input)

The active low Read signal is conditioned by \overline{CS} and indicates that information is to be transferred from internal locations to the data bus. \overline{RD} and \overline{WR} are mutually exclusive.

\overline{WR} (Write, Input)

The active low Write signal is conditioned by \overline{CS} and indicates that information is to be transferred from the data bus into internal locations. \overline{RD} and \overline{WR} are mutually exclusive.

\overline{EACK} (End Acknowledge, Input)

This active low input clears the end of execution output signal (\overline{END}). If \overline{EACK} is tied low, the \overline{END} output will be a pulse that is less than one clock period wide.

\overline{SVACK} (Service Acknowledge, Input)

This active low input clears the service request output (\overline{SVREQ}).

\overline{END} (End Execution, Output)

This active low, open-drain output indicates that execution of the previously entered command is complete. It can be used as an interrupt request and is cleared by \overline{EACK} , RESET or any read or write access to the Am9511.

\overline{SVREQ} (Service Request, Output)

This active high output signal indicates that command execution is complete and that post execution service was requested in the previous command byte. It is cleared by \overline{SVACK} , by RESET, or by the end of a subsequent command that does not request service.

\overline{PAUSE} (Pause, Output)

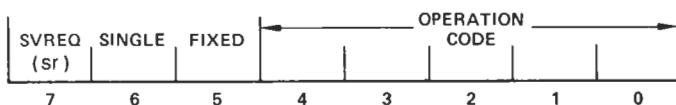
This active low output indicates that the Am9511 is unable to complete communication with the data bus. When an attempt is made to read data, write data or to enter a new command while the Am9511 is executing a command, \overline{PAUSE} goes low until execution of the current command is complete. It also goes low during valid read operations to compensate for the data stack access time. (See "Pause Operation" section.)

DB0-DB7 (Bidirectional Data Bus, I/O)

These eight bidirectional lines provide for transfer of commands, status and data between the Am9511 and the CPU. The Am9511 will drive the data bus only when \overline{CS} and \overline{RD} are low.

COMMAND STRUCTURE

Each command entered into the Am9511 consists of a single 8-bit byte having the format illustrated below:



Bits 0-4 select the operation to be performed as shown in the table. Bits 5-6 select the data format for the operation. If bit 5 is a 1, a fixed point data format is specified. If bit 5 is a 0, floating point format is specified. Bit 6 selects the precision of

the data to be operated on by fixed point commands (if bit 5 = 0, bit 6 must be 0). If bit 6 is a 1, single-precision (16-bit) operands are indicated; if bit 6 is a 0, double-precision (32-bit) operands are indicated. Results are undefined for all illegal combinations of bits in the command byte. Bit 7 indicates whether a service request is to be issued after the command is executed. If bit 7 is a 1, the service request output (\overline{SVREQ}) will go high at the conclusion of the command and will remain high until reset by a low level on the service acknowledge pin (\overline{SVACK}) or until completion of execution of a succeeding command where bit 7 is 0. Each command issued to the Am9511 requests post execution service based upon the state of bit 7 in the command byte. When bit 7 is a 0, \overline{SVREQ} remains low.

COMMAND SUMMARY

Command Code								Command Mnemonic	Command Description
7	6	5	4	3	2	1	0		
FIXED POINT 16 BIT									
sr	1	1	0	1	1	0	0	SADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	0	1	SSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	1	0	SMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	1	1	1	0	1	1	0	SMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	1	1	0	1	1	1	1	SDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FIXED POINT 32 BIT									
sr	0	1	0	1	1	0	0	DADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	0	1	DSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	1	0	DMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	0	1	1	0	1	1	0	DMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	0	1	0	1	1	1	1	DDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FLOATING POINT 32 BIT									
sr	0	0	1	0	0	0	0	FADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	0	1	FSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	0	FMUL	Multiply NOS by TOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	1	FDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
DERIVED FLOATING POINT FUNCTIONS									
sr	0	0	0	0	0	0	1	SQRT	Square Root of TOS. Result in TOS.
sr	0	0	0	0	0	1	0	SIN	Sine of TOS. Result in TOS.
sr	0	0	0	0	0	1	1	COS	Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	0	0	TAN	Tangent of TOS. Result in TOS.
sr	0	0	0	0	1	0	1	ASIN	Inverse Sine of TOS. Result in TOS.
sr	0	0	0	0	1	1	0	ACOS	Inverse Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	1	1	ATAN	Inverse Tangent of TOS. Result in TOS.
sr	0	0	0	1	0	0	0	LOG	Common Logarithm (base 10) of TOS. Result in TOS.
sr	0	0	0	1	0	0	1	LN	Natural Logarithm (base e) of TOS. Result in TOS.
sr	0	0	0	1	0	1	0	EXP	Exponential (e^x) of TOS. Result in TOS.
sr	0	0	0	1	0	1	1	PWR	NOS raised to the power in TOS. Result in NOS. Pop Stack.
DATA MANIPULATION COMMANDS									
sr	0	0	0	0	0	0	0	NOP	No Operation
sr	0	0	1	1	1	1	1	FIXS	Convert TOS from floating point to 16-bit fixed point format.
sr	0	0	1	1	1	1	0	FIXD	Convert TOS from floating point to 32-bit fixed point format.
sr	0	0	1	1	1	0	1	FLTS	Convert TOS from 16-bit fixed point to floating point format.
sr	0	0	1	1	1	0	0	FLTD	Convert TOS from 32-bit fixed point to floating point format.
sr	1	1	1	0	1	0	0	CHSS	Change sign of 16-bit fixed point operand on TOS.
sr	0	1	1	0	1	0	0	CHSD	Change sign of 32-bit fixed point operand on TOS.
sr	0	0	1	0	1	0	1	CHSF	Change sign of floating point operand on TOS.
sr	1	1	1	0	1	1	1	PTOS	Push 16-bit fixed point operand on TOS to NOS (Copy)
sr	0	1	1	0	1	1	1	PTOD	Push 32-bit fixed point operand on TOS to NOS. (Copy)
sr	0	0	1	0	1	1	1	PTOF	Push floating point operand on TOS to NOS. (Copy)
sr	1	1	1	1	0	0	0	POPS	Pop 16-bit fixed point operand from TOS. NOS becomes TOS.
sr	0	1	1	1	0	0	0	POPD	Pop 32-bit fixed point operand from TOS. NOS becomes TOS.
sr	0	0	1	1	0	0	0	POPF	Pop floating point operand from TOS. NOS becomes TOS.
sr	1	1	1	1	0	0	1	XCHS	Exchange 16-bit fixed point operands TOS and NOS.
sr	0	1	1	1	0	0	1	XCHD	Exchange 32-bit fixed point operands TOS and NOS.
sr	0	0	1	1	0	0	1	XCHF	Exchange floating point operands TOS and NOS.
sr	0	0	1	1	0	1	0	PUPI	Push floating point constant " π " onto TOS. Previous TOS becomes NOS.

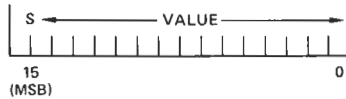
NOTES:

1. TOS means Top of Stack. NOS means Next on Stack.
2. AMD Application Brief "Algorithm Details for the Am9511 APU" provides detailed descriptions of each command function, including data ranges, accuracies, stack configurations, etc.
3. Many commands destroy one stack location (bottom of stack) during development of the result. The derived functions may destroy several stack locations. See Application Brief for details.
4. The trigonometric functions handle angles in radians, not degrees.
5. No remainder is available for the fixed-point divide functions.
6. Results will be undefined for any combination of command coding bits not specified in this table.

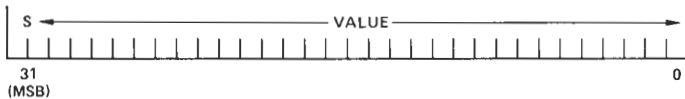
DATA FORMATS

The Am9511 Arithmetic Processing Unit handles operands in both fixed point and floating point formats. Fixed point operands may be represented in either single (16-bit operands) or double precision (32-bit operands), and are always represented as binary, two's complement values.

16-BIT FIXED POINT FORMAT



32-BIT FIXED POINT FORMAT



The sign (positive or negative) of the operand is located in the most significant bit (MSB). Positive values are represented by a sign bit of zero ($S = 0$). Negative values are represented by the two's complement of the corresponding positive value with a sign bit equal to 1 ($S = 1$). The range of values that may be accommodated by each of these formats is $-32,768$ to $+32,767$ for single precision and $-2,147,483,648$ to $+2,147,483,647$ for double precision.

Floating point binary values are represented in a format that permits arithmetic to be performed in a fashion analogous to operations with decimal values expressed in scientific notation.

$$(5.83 \times 10^2)(8.16 \times 10^1) = (4.75728 \times 10^4)$$

In the decimal system, data may be expressed as values between 0 and 10 times 10 raised to a power that effectively shifts the implied decimal point right or left the number of places necessary to express the result in conventional form (e.g., 47,572.8). The value-portion of the data is called the mantissa. The exponent may be either negative or positive.

The concept of floating point notation has both a gain and a loss associated with it. The gain is the ability to represent the significant digits of data with values spanning a large dynamic range limited only by the capacity of the exponent field. For

example, in decimal notation if the exponent field is two digits wide, and the mantissa is five digits, a range of values (positive or negative) from 1.0000×10^{-99} to $9.9999 \times 10^{+99}$ can be accommodated. The loss is that only the significant digits of the value can be represented. Thus there is no distinction in this representation between the values 123451 and 123452, for example, since each would be expressed as: 1.2345×10^5 . The sixth digit has been discarded. In most applications where the dynamic range of values to be represented is large, the loss of significance, and hence accuracy of results, is a minor consideration. For greater precision a fixed point format could be chosen, although with a loss of potential dynamic range.

The Am9511 is a binary arithmetic processor and requires that floating point data be represented by a fractional mantissa value between .5 and 1 multiplied by 2 raised to an appropriate power. This is expressed as follows:

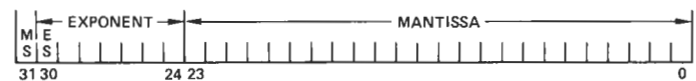
$$\text{value} = \text{mantissa} \times 2^{\text{exponent}}$$

For example, the value 100.5 expressed in this form is 0.11001001×2^7 . The decimal equivalent of this value may be computed by summing the components (powers of two) of the mantissa and then multiplying by the exponent as shown below:

$$\begin{aligned} \text{value} &= (2^{-1} + 2^{-2} + 2^{-5} + 2^{-8}) \times 2^7 \\ &= (0.5 + 0.25 + 0.03125 + 0.00290625) \times 128 \\ &= 0.78515625 \times 128 \\ &= 100.5 \end{aligned}$$

FLOATING POINT FORMAT

The format for floating point values in the Am9511 is given below. The mantissa is expressed as a 24-bit (fractional) value; the exponent is expressed as an unbiased two's complement 7-bit value having a range of -64 to $+63$. The most significant bit is the sign of the mantissa (0 = positive, 1 = negative), for a total of 32 bits. The binary point is assumed to be to the left of the most significant mantissa bit (bit 23). All floating point data values must be normalized. Bit 23 must be equal to 1, except for the value zero, which is represented by all zeros.

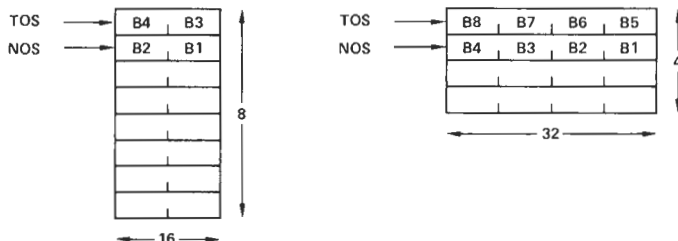


The range of values that can be represented in this format is $\pm(2.7 \times 10^{-20}$ to $9.2 \times 10^{18})$ and zero.

FUNCTIONAL DESCRIPTION

Stack Control

The user interface to the Am9511 includes access to an 8 level 16-bit wide data stack. Since single precision fixed point operands are 16 bits in length, eight such values may be maintained in the stack. When using double precision fixed point or floating point formats four values may be stored. The stack in these two configurations can be visualized as shown below:



Data are written onto the stack, eight bits at a time, in the order shown (B1, B2, B3, ...). Data are removed from the stack in reverse byte order (B8, B7, B6, ...). Data should be transferred into or out of the stack in multiples of the number of bytes appropriate to the chosen data format.

Data Entry

Data entry is accomplished by bringing the chip select (\overline{CS}), the command/data line (C/\overline{D}), and \overline{WR} low, as shown in the timing diagram. The entry of each new data word "pushes down" the previously entered data and places the new byte on the top of stack (TOS). Data on the bottom of the stack prior to a stack entry are lost.

Data Removal

Data are removed from the stack in the Am9511 by bringing chip select (\overline{CS}), command/data (C/\overline{D}), and \overline{RD} low as shown in the timing diagram. The removal of each data word redefines TOS so that the next successive byte to be removed becomes TOS. Data removed from the stack rotates to the bottom of the stack.

Command Entry

After the appropriate number of bytes of data have been entered onto the stack, a command may be issued to perform an operation on that data. Commands which require two operands for execution (e.g., add) operate on the TOS and NOS values. Single operand commands operate only on the TOS.

Commands are issued to the Am9511 by bringing the chip select (\overline{CS}) line low, command/data (C/\overline{D}) line high, and \overline{WR} line low as indicated by the timing diagram. After a command is issued, the CPU can continue execution of its program concurrently with the Am9511 command execution.

Command Completion

The Am9511 signals the completion of each command execution by lowering the End Execution line (\overline{END}). Simultaneously, the busy bit in the status register is cleared and the Service Request bit of the command register is checked. If it is a "1" the service request output level (\overline{SVREQ}) is raised. \overline{END} is cleared on receipt of an active low End Acknowledge (\overline{EACK}) pulse. Similarly, the service request line is cleared by recognition of an active low Service Acknowledge (\overline{SVACK}) pulse.

Pause Operation

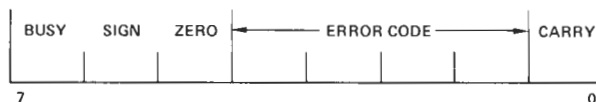
An active low Pause (\overline{PAUSE}) is provided. This line is high in its quiescent state and is pulled low by the Am9511 under the following conditions:

1. A previously initiated operation is in progress (device busy) and Command Entry has been attempted. In this case, the \overline{PAUSE} line will be pulled low and remain low until completion of the current command execution. It will then go high, permitting entry of the new command.
2. A previously initiated operation is in progress and stack access has been attempted. In this case, the \overline{PAUSE} line will be pulled low, will remain in that state until execution is complete, and will then be raised to permit completion of the stack access.
3. The Am9511 is not busy, and data removal has been requested. \overline{PAUSE} will be pulled low for the length of time necessary to transfer the byte from the top of stack to the interface latch, and will then go high, indicating availability of the data.
4. The Am9511 is not busy, and a data entry has been requested. \overline{PAUSE} will be pulled low for the length of time required to ascertain if the preceding data byte, if any has been written to the stack. If so \overline{PAUSE} will immediately go high. If not, \overline{PAUSE} will remain low until the interface latch is free and will then go high.
5. When a status read has been requested, \overline{PAUSE} will be pulled low for the length of time necessary to transfer the status to the interface latch, and will then be raised to permit completion of the status read. Status may be read whether or not the Am9511 is busy.

When \overline{PAUSE} goes low, the APU expects the bus and bus control signals present at the time to remain stable until \overline{PAUSE} goes high.

Device Status

Device status is provided by means of an internal status register whose format is shown below:



- BUSY:** Indicates that Am9511 is currently executing a command (1 = Busy).
- SIGN:** Indicates that the value on the top of stack is negative (1 = Negative).
- ZERO:** Indicates that the value on the top of stack is zero (1 = Value is zero).
- ERROR CODE:** This field contains an indication of the validity of the result of the last operation. The error codes are:
- 0000 – No error
 - 1000 – Divide by zero
 - 0100 – Square root or log of negative number
 - 1100 – Argument of inverse sine, cosine, or e^x too large
 - XX10 – Underflow
 - XX01 – Overflow
- CARRY:** Previous operation resulted in carry or borrow from most significant bit. (1 = Carry/Borrow, 0 = No Carry/No Borrow)

If the BUSY bit in the status register is a one, the other status bits are not defined; if zero, indicating not busy, the operation is complete and the other status bits are defined as given above.

Read Status

The Am9511 status register can be read by the CPU at any time (whether an operation is in progress or not) by bringing the chip select (\overline{CS}) low, the command-data line (C/\overline{D}) high, and lowering \overline{RD} . The status register is then gated onto the data bus and may be input by the CPU.

EXECUTION TIMES

Timing for execution of the Am9511 command set is shown in the table below. Speeds are given in terms of clock cycles and should be multiplied by the clock period being used to arrive at time values. Where substantial variation of execution times is possible, the minimum and maximum values are shown; otherwise, typical values are given. Variations are data dependent. Some boundary conditions that will cause shorter execution times are not taken into account. The listing is in alphabetical order by mnemonic.

Total execution times may require allowances for operand transfer into the APU, command execution, and result retrieval from the APU. Except for command execution, these times will be heavily influenced by the nature of the data, the control interface used, the speed of memory, the CPU used, the priority allotted to DMA and Interrupt operations, the size and number of operands to be transferred, and the use of chained calculations, etc.

COMMAND EXECUTION TIMES

Command Mnemonic	Clock Cycles	Command Mnemonic	Clock Cycles
ACOS	6304-8284	LOG	4474-7132
ASIN	6230-7938	LN	4298-6956
ATAN	4992-6536	NOP	4
CHSD	26-28	POPD	12
CHSF	16-20	POPF	12
CHSS	22-24	POPS	10
COS	3840-4878	PTOD	20
DADD	20-22	PTOF	20
DDIV	196-210	PTOS	16
DMUL	194-210	PUPI	16
DMUU	182-218	PWR	8290-12032
DSUB	38-40	SADD	16-18
EXP	3794-4878	SDIV	84-94
FADD	54-368	SIN	3796-4808
FDIV	154-184	SMUL	84-94
FIXD	90-336	SMUU	80-98
FIXS	90-214	SQRT	782-870
FLTD	56-342	SSUB	30-32
FLTS	62-156	TAN	4894-5886
FMUL	146-168	XCHD	26
FSUB	70-370	XCHF	26
		XCHS	18

As mentioned, the above clock cycle execution times can be converted to μsec by multiplying by the clock period used. Several examples (minimums) are shown below:

Command	Am9511	Am9511-1	Am9511-4
ARC COSINE (ACOS)	6304 μsec	3152μsec	1576 μsec
COSINE (COS)	3840 μsec	1920μsec	960 μsec
EXPONENTIATION (EXP)	3794 μsec	1897μsec	949 μsec
SINGLE MULTIPLY LOWER (SMUL)	84 μsec	42μsec	21 μsec
SINGLE ADD (SADD)	16 μsec	8μsec	4 μsec

MAXIMUM RATINGS beyond which useful life may be impaired

Storage Temperature	-65°C to +150°C
Ambient Temperature Under Bias	-55°C to +125°C
VDD with Respect to VSS	-0.5V to +15.0V
VCC with Respect to VSS	-0.5V to +7.0V
All Signal Voltages with Respect to VSS	-0.5V to +7.0V
Power Dissipation (Package Limitation)	2.0W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	Ambient Temperature	VSS	VCC	VDD
Am9511DC	0°C ≤ T _A ≤ +70°C	0V	+5.0V ±5%	+12V ±5%
Am9511DM	-55°C ≤ T _A ≤ +125°C	0V	+5.0V ±10%	+12V ±10%

ELECTRICAL CHARACTERISTICS Over Operating Range (Note 1)

Parameters	Description	Test Conditions	Min.	Typ.	Max.	Units
VOH	Output HIGH Voltage	I _{OH} = -200μA	3.7			Volts
VOL	Output LOW Voltage	I _{OL} = 3.2mA			0.4	Volts
VIH	Input HIGH Voltage		2.0		VCC	Volts
VIL	Input LOW Voltage		-0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ V _I ≤ VCC			±10	μA
IOZ	Data Bus Leakage	VO = 0.4V			10	μA
		VO = VCC			10	
ICC	VCC Supply Current	T _A = +25°C		50	80	mA
		T _A = 0°C			85	
		T _A = -55°C			90	
IDD	VDD Supply Current	T _A = +25°C		50	80	mA
		T _A = 0°C			85	
		T _A = -55°C			90	
CO	Output Capacitance	f _c = 1.0MHz, Inputs = 0V		8	10	pF
CI	Input Capacitance			5	8	pF
CIO	I/O Capacitance			10	12	pF

SWITCHING CHARACTERISTICS over operating range (Notes 2, 3)

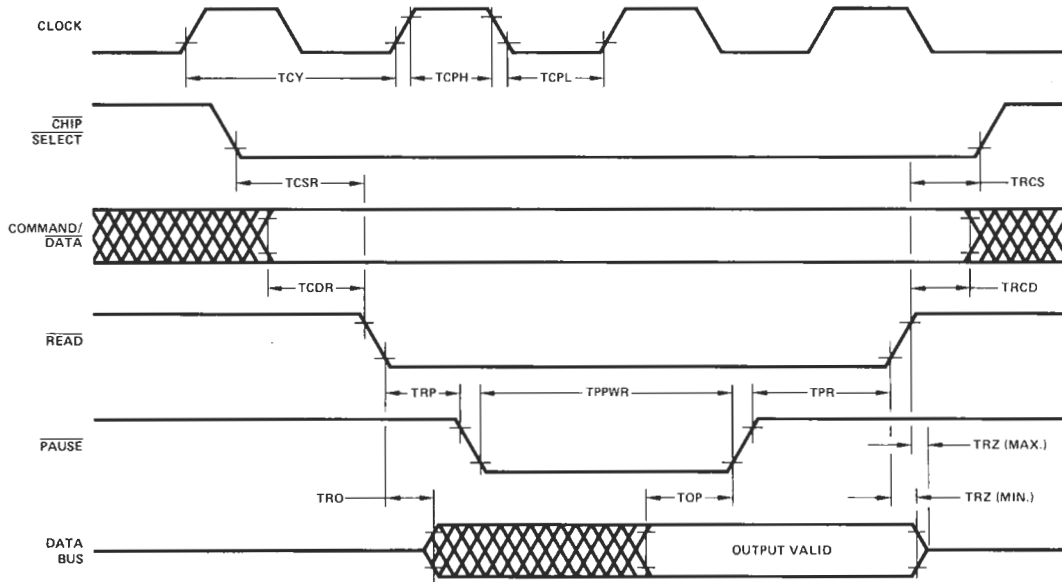
Parameters	Description	Am9511		Am9511-1		(Preliminary) Am9511-4		Units	
		Min.	Max.	Min.	Max.	Min.	Max.		
TAPW	$\overline{\text{EACK}}$ LOW Pulse Width	100		75		50		ns	
TCDR	$\text{C}/\overline{\text{D}}$ to $\overline{\text{RD}}$ LOW Set up Time	0		0		0		ns	
TCDW	$\text{C}/\overline{\text{D}}$ to $\overline{\text{WR}}$ LOW Set up Time	0		0		0		ns	
TCPH	Clock Pulse HIGH Width	200		140		100		ns	
TCPL	Clock Pulse LOW Width	240		160		120		ns	
TCSR	$\overline{\text{CS}}$ LOW to $\overline{\text{RD}}$ LOW Set up Time	50		25		25		ns	
TCSW	$\overline{\text{CS}}$ LOW to $\overline{\text{WR}}$ LOW Set up Time	50		25		25		ns	
TCY	Clock Period	480	5000	330	3300	250	2500	ns	
TDW	Data Bus Stable to $\overline{\text{WR}}$ HIGH Set up Time		150		100		100	ns	
TEAE	$\overline{\text{EACK}}$ LOW to $\overline{\text{END}}$ HIGH Delay		200		175		150	ns	
TEPW	$\overline{\text{END}}$ LOW Pulse Width (Note 4)	400		300		200		ns	
TOP	Data Bus Output Valid to $\overline{\text{PAUSE}}$ HIGH Delay	0		0		0		ns	
TPPWR	$\overline{\text{PAUSE}}$ LOW Pulse Width Read (Note 5)	Data	$3.5\text{TCY}+50$	$5.5\text{TCY}+200$	$3.5\text{TCY}+50$	$5.5\text{TCY}+200$	$3.5\text{TCY}+50$	$5.5\text{TCY}+200$	ns
		Status	$1.5\text{TCY}+50$	$3.5\text{TCY}+200$	$1.5\text{TCY}+50$	$3.5\text{TCY}+200$	$1.5\text{TCY}+50$	$3.5\text{TCY}+200$	
TPPWW	$\overline{\text{PAUSE}}$ LOW Pulse Width Write (Note 8)		50		50		50	ns	
TPR	$\overline{\text{PAUSE}}$ HIGH to $\overline{\text{RD}}$ HIGH Hold Time	0		0		0		ns	
TPW	$\overline{\text{PAUSE}}$ HIGH to $\overline{\text{WR}}$ HIGH Hold Time	0		0		0		ns	
TRCD	$\overline{\text{RD}}$ HIGH to $\text{C}/\overline{\text{D}}$ Hold Time	0		0		0		ns	
TRCS	$\overline{\text{RD}}$ HIGH to $\overline{\text{CS}}$ HIGH Hold Time	0		0		0		ns	
TRO	$\overline{\text{RD}}$ LOW to Data Bus ON Delay	50		50		25		ns	
TRP	$\overline{\text{RD}}$ LOW to $\overline{\text{PAUSE}}$ LOW Delay (Note 6)		150		100		100	ns	
TRZ	$\overline{\text{RD}}$ HIGH to Data Bus OFF Delay	50	200	50	150	25	100	ns	
TSAPW	$\overline{\text{SVACK}}$ LOW Pulse Width	100		75		50		ns	
TSAR	$\overline{\text{SVACK}}$ LOW to $\overline{\text{SVREQ}}$ LOW Delay		300		200		150	ns	
TWCD	$\overline{\text{WR}}$ HIGH to $\text{C}/\overline{\text{D}}$ Hold Time	60		30		30		ns	
TWCS	$\overline{\text{WR}}$ HIGH to $\overline{\text{CS}}$ HIGH Hold Time	60		30		30		ns	
TWD	$\overline{\text{WR}}$ HIGH to Data Bus Hold Time	20		20		20		ns	
TWI	Write Inactive Time (Note 8)	Command	3TCY		3TCY		3TCY	ns	
		Data	4TCY		4TCY		4TCY		
TWP	$\overline{\text{WR}}$ LOW to $\overline{\text{PAUSE}}$ LOW Delay (Note 6)		150		100		100	ns	

NOTES:

- Typical values are for $T_A = 25^\circ\text{C}$, nominal supply voltages and nominal processing parameters.
- Switching parameters are listed in alphabetical order.
- Test conditions assume transition times of 20ns or less, output loading of one TTL gate plus 100pF and timing reference levels of 0.8V and 2.0V.
- $\overline{\text{END}}$ low pulse width is specified for $\overline{\text{EACK}}$ tied to VSS. Otherwise TEAE applies.
- Minimum values shown assume no previously entered command is being executed for the data access. Status may be read at any time.
- $\overline{\text{PAUSE}}$ is pulled low for both command and data operations.
- TEX is the execution time of the current command (see the Command Execution Times table).
- $\overline{\text{PAUSE}}$ low pulse width is less than 50ns when writing into the data port or the control port as long as the duty cycle requirement (TWI) is observed and no previous command is being executed. TWI may be safely violated as long as the extended TPPWW that results is observed.

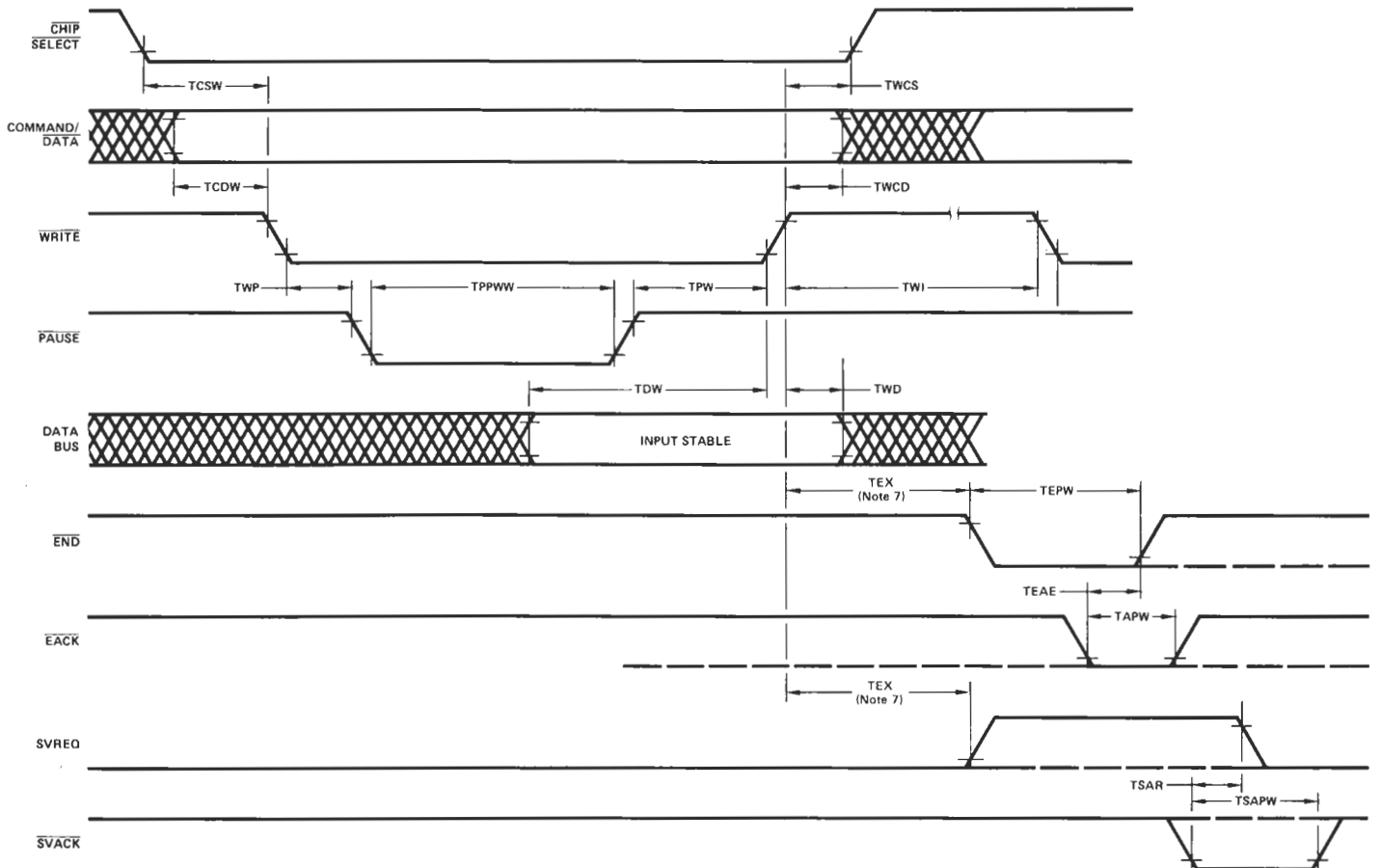
SWITCHING WAVEFORMS

READ OPERATIONS



MOS-048

WRITE OPERATIONS



MOS-049

APPLICATION INFORMATION

The diagram in Figure 2 shows the interface connections for the Am9511 APU with operand transfers handled by an Am9517 DMA controller, and CPU coordination handled by an Am9519 Interrupt Controller. The APU interrupts the CPU to indicate that a command has been completed. When the performance enhancements provided by the DMA and Interrupt

operations are not required, the APU interface can be simplified as shown in Figure 1. The Am9511 APU is designed with a general purpose 8-bit data bus and interface control so that it can be conveniently used with any general 8-bit processor.

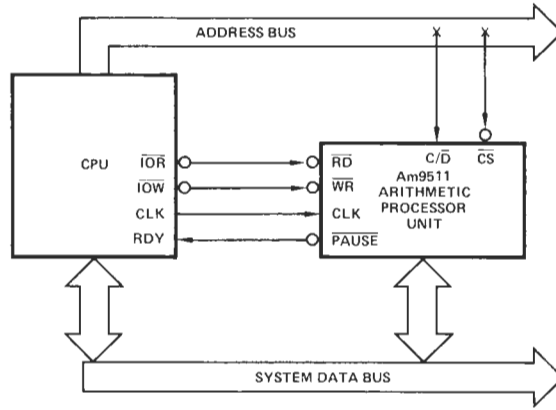


Figure 1. Am9511 Minimum Configuration Example.

MOS-050

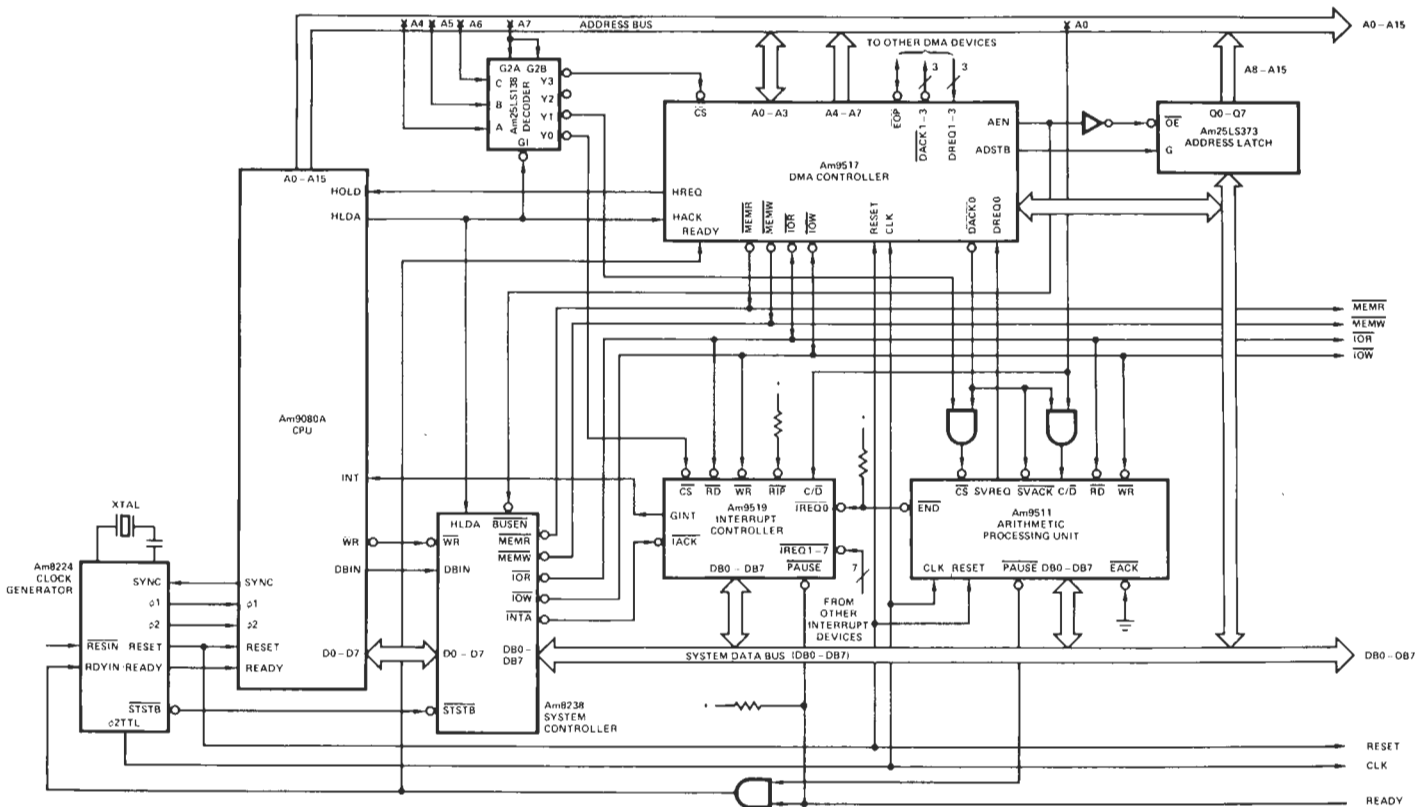
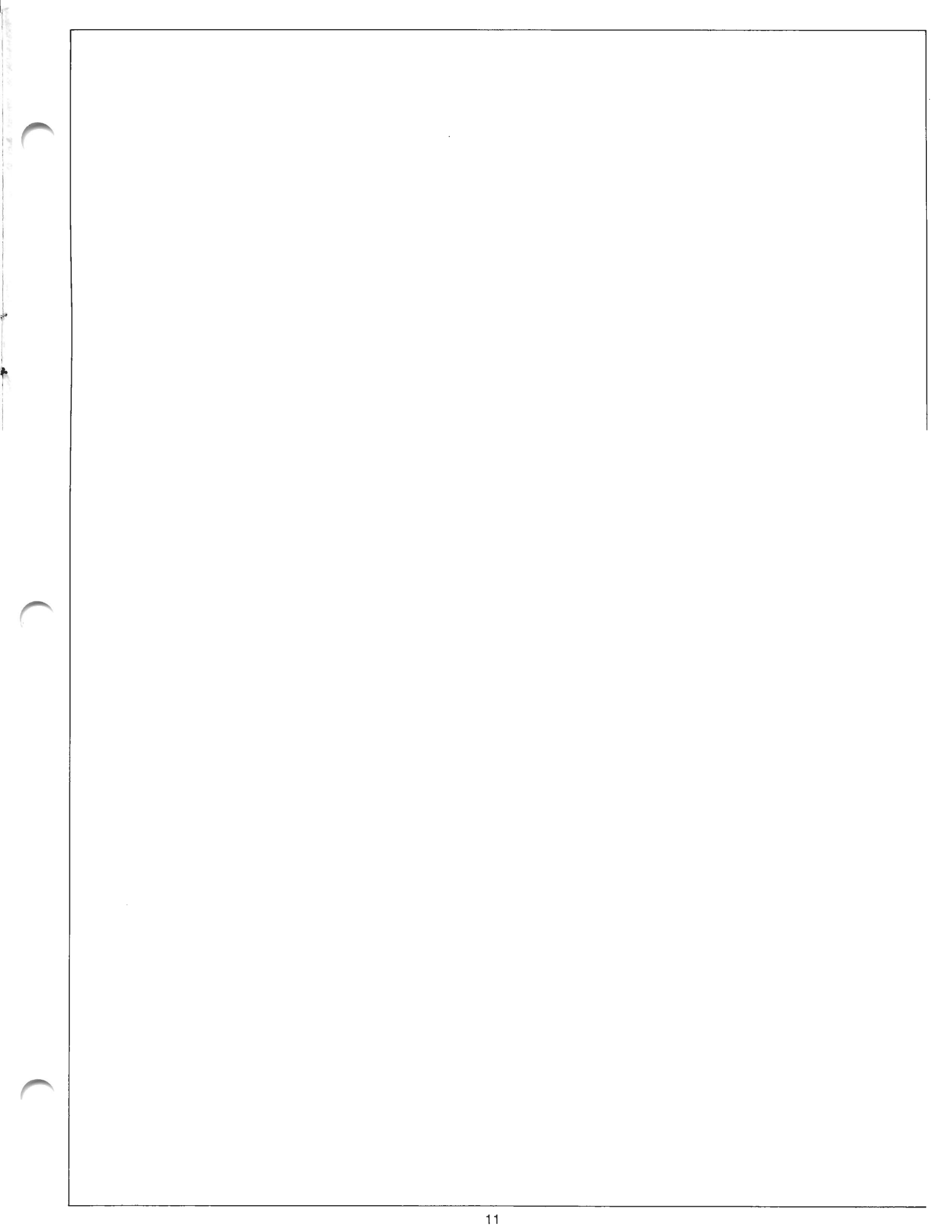


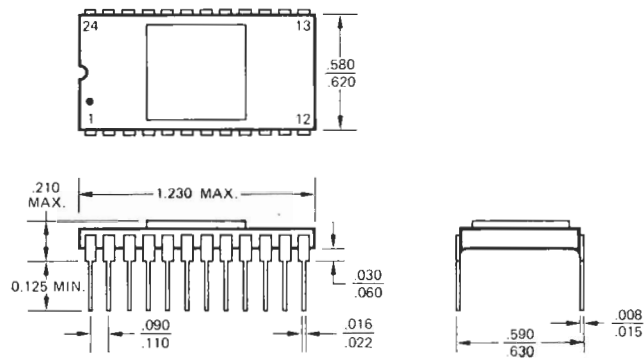
Figure 2. Am9511 High Performance Configuration Example.

MOS-051



PHYSICAL DIMENSIONS
Dual-In-Line

24-Pin Side-Brazed



**ADVANCED
MICRO
DEVICES, INC.**
901 Thompson Place
Sunnyvale
California 94086
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306
TOLL FREE
(800) 538-8450