VLSI TECHNOLOGY, INC.

VY86C500

12-BIT FUZZY COMPUTATIONAL ACCELERATION CORE

FEATURES

- 12-bit data resolution
- 16-bit rule base memory address bus
- 12-bit scratch memory address bus
- Complete fuzzy computation including defuzzification
- Optimized rule base format for high-speed fuzzy computation
- Flexible rule base capabilities
- Centroid, height, and alpha defuzzification
- Several premise membership function representations including piecewise linear and quadratic S, π, and Z-functions.
- Arbitrary output membership function representations
- Fuzzy operations AND, OR, and NOT
- Product-sum fuzzy inferencing
- Built-in self-test
- 182 mil x 115 mil size
- 20 MHz operating frequency
- 1-Micron CMOS technology

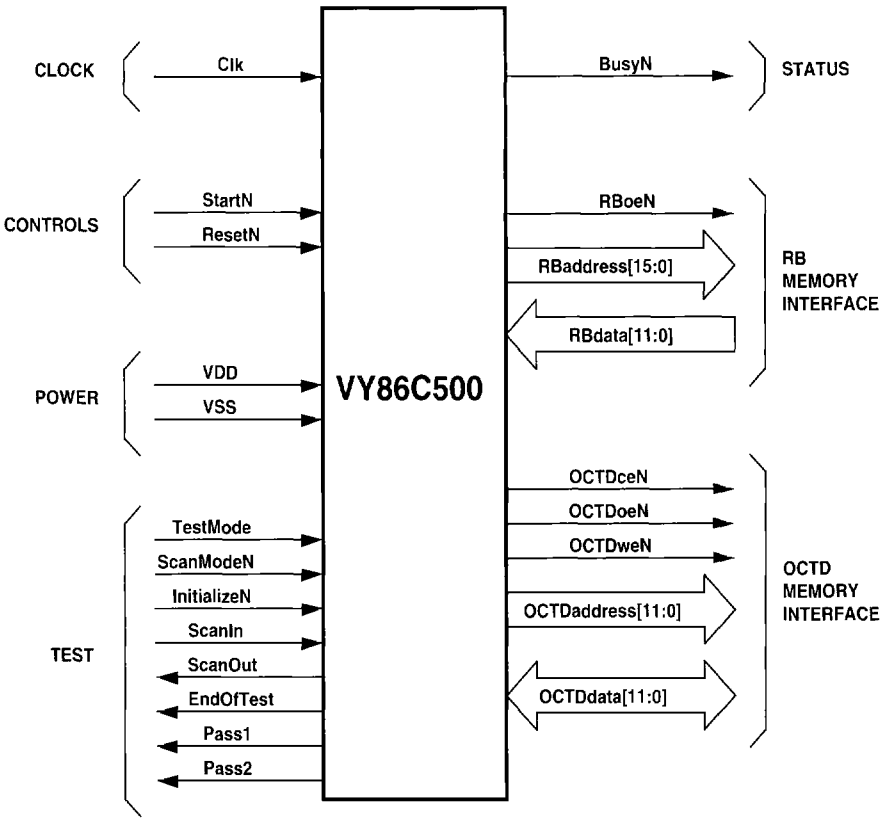
DESCRIPTION

The VY86C500 is a 12-bit general-purpose fuzzy logic FSB™ functional system block core based on Fuzzy Computational Acceleration (FCA™) technology, capable of performing true fuzzy computations in hardware. The core performs complete fuzzy computations including premise membership function evaluations, rule evaluations, conclusion membership function composition, and defuzzification.

Premise membership functions can be expressed using piecewise linear, or quadratic S-function, Z-function, or π-function representations. Membership function, overlap restrictions, and membership function "sum-to-one" restrictions do not exist, allowing for uncompromised implementations of fuzzy systems in hardware. Inferencing is performed via the product-sum method, and centroid, height, or alpha defuzzification methods may be used.

The VY86C500 offers "alpha direct-in" and "alpha direct-out" features for additional flexibility in fuzzy system design. "Alpha direct-in" allows input membership function alpha values (degrees of membership) to be calculated externally and input directly into the VY86C500.

FUNCTIONAL DIAGRAM



VLSIS005

"Alpha direct-out" allows results of rule evaluations (rule alphas) to be output directly by the VY86C500.

The VY86C500 fuzzy logic core requires a rule base memory (RB memory) and a scratch memory (OCTD memory). Theses elements can reside on-chip along with the VY86C500 core, or can exist outside of the chip containing the VY86C500 core.

The inputs and outputs of fuzzy computations are transferred between the host and the fuzzy accelerator core via a shared memory interface of the OCTD (observation, conclusion, temporary data) memory. The RB memory stores all the premise membership function, conclusion membership function, and rule information (collectively referred to as the rule base) necessary to carry out fuzzy computations. The RB memory can store multiple rule bases, space permitting.

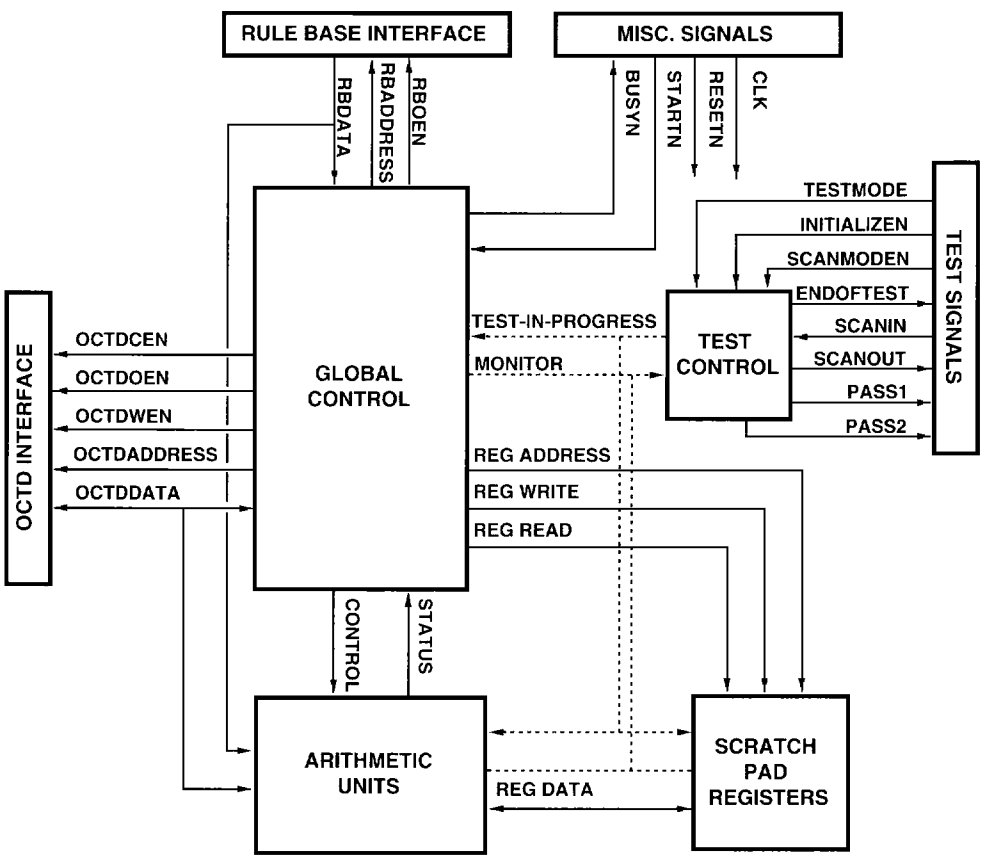
Innovative use of internal scratch pad registers and computational elements raises the fuzzy computation speed while holding the external memory size requirements to a minimum.

Fuzzy Programming Language (FPL™) files created manually or by software tools such as the TILShell can be compiled directly for the VY86C500 device using the FCA Development System (FCADS™) compiler.

All input signals are latched upon entering the VY86C500 core to minimize set-up and hold time requirements.

Built-in self-test (BIST) has been implemented with an interface similar to that of the COMPASS Design Automation BIST compilers and is compatible with the COMPASS boundary scan library.

BLOCK DIAGRAM



SIGNAL DESCRIPTIONS

Name	Type	Description
BusyN	Output	Not busy. When a rule base is being evaluated or when the FCA core is undergoing built-in self-test, this connector is asserted low. A high BusyN indicates that the VY86C500 core is idle.
Clk	Input	Clock. This connector provides the clock by which all VY86C500 core operations are timed including built-in self-test.
EndOfTest	Output	End of test (BIST). This connector indicates that the built-in self-test (BIST) sequence has completed. ResetN does not affect this signal.
InitializeN	Input	Not initialize (BIST). This connector, in conjunction with the TestMode connector, initializes the BIST logic. When this connector is asserted low for at least one clock cycle while TestMode is asserted high, the BIST circuit is reset to prepare for the execution of the BIST sequence. While this connector is held low, the BIST circuit remains in the reset state and will not begin the BIST sequence until this connector returns to its inactive high state.
OCTDaddress[11:0]	Output	OCTD address bus. This bus addresses the external OCTD memory. These connectors are driven with random patterns during self-test.
OCTDceN	Output	Not OCTD chip enable. OCTDceN should be connected to the external OCTD memory such that the OCTD memory is enabled for reading or writing when this signal is asserted low and if either OCTDoeN or OCTDweN is also asserted low. OCTDceN is driven to its inactive, high state during reset and self test.
OCTDdata[11:0]	I/O	OCTD data bus. These are bidirectional signal paths which are used for data transfers between the VY86C500 core and the external OCTD memory. These connectors are three-stated during self-test.
OCTDoeN	Output	Not OCTD output enable. OCTDoeN should be connected to the external OCTD memory, such that the OCTD memory will output data onto the OCTD data bus when both OCTDoeN and OCTDceN are asserted low. The VY86C500 accelerator will read data output by the OCTD memory at the next rising clock edge. OCTDoeN is driven to its inactive, high state during reset and self test. See Theory of Operation - OCTD Memory section (p. 14) for interface to memory without an output enable connector.
OCTDweN	Output	Not OCTD write enable. OCTDweN should be connected to the external OCTD memory such that data on the OCTD bus is written into OCTD memory while OCTDweN and OCTDceN are both asserted low or on the rising edge of OCTDweN (for clocked memory) when OCTDceN is low. The VY86C500 outputs valid address and data onto the OCTD address and OCTDdata busses, respectively, before asserting OCTDweN low. OCTDweN transitions on the falling edge of the Clk signal. OCTDaddress and OCTDdata are valid for two clock cycles from the rising edge preceding OCTDweN being asserted low. OCTDweN is driven to its inactive, high state during reset and self test.
Pass1, Pass2	Outputs	Pass 1 and Pass 2 (BIST). These connectors identify whether the fuzzy logic accelerator core has passed or failed the built-in self- test. During the initialization of the BIST circuitry, these connectors are set to the low state. At the end of the BIST sequence, these pins will go high if the test passed and will remain low if the test failed. The test is considered a pass only if both pins are high at the end of test. ResetN does not affect these signals.
RBaddress[15:0]	Output	RB address bus. This bus addresses the external RB memory. These connectors are driven with random patterns during self-test.
RBdata[11:0]	Input	RB data bus. This bus is used for data transfers from the external RB memory to the VY86C500 core.
RBoeN	Output	Not RB output enable. RBoeN should be connected to the external RB memory such that RB memory outputs data onto the RB data bus when this signal is asserted low. The VY86C500 device will read data output by the RB memory at the next rising clock edge. RBoeN is driven to its inactive, high state during reset and self test. RBoeN remains low for the duration of the fuzzy computation.

SIGNAL DESCRIPTIONS (Cont.)

Name	Type	Description
ResetN	Input	Not reset. When this connector is asserted low, the VY86C500 core is reset. Reset occurs synchronously inside the VY86C500 core following the first low to high transition of Clk after ResetN is asserted. Reset must be asserted for at least four clock cycles to ensure proper core initialization.
ScanIn	Input	Scan in (BIST). This connector is the input of the scan chain that scans through the test results registers.
ScanModeN	Input	Not scan mode (BIST). This connector selects the scan mode of the test results and enables the scanning of the test results at the end of the BIST sequence. When the end of test is reached, the shift of the test results scan chain will start when this connector is asserted low.
ScanOut	Output	Scan out (BIST). This connector is the output of the scan chain which scans through the test results registers. ResetN does not affect this signal.
StartN	Input	Not start. Asserting this connector low for one cycle starts a fuzzy computation cycle. BusyN must be high, TestMode must be low, and ResetN must be high before a fuzzy computation can be started. If these conditions are not met, the result of the fuzzy computation is not valid.
TestMode	Input	Test mode (BIST). This connector activates the built-in self-test logic. This connector must remain asserted high until the EndOfTest connector goes high and during the scanning of the test results registers. If this pin is driven low while BIST sequence is in progress, built-in self-test will be aborted and the test results will be invalid. During normal operations of the VY86C500 device, this pin must be driven low.
VDD	Input	Positive supply. All VDD pins must be connected to supply.
VSS	Input	Ground supply. All VSS pins must be connected to supply.

FUZZY LOGIC & VY86C500 TERMS AND CONCEPTS

Fuzzy logic is a branch of mathematics pioneered by Dr. Lotfi Zadeh at the University of California in the 1960's. In fuzzy logic, contrary to traditional set theory where an element either belongs or does not belong to a given set, an element can belong to a set "to a certain degree".

This allows fuzzy logic systems to be defined in simple intuitive terms and yet operate smoothly and continuously.

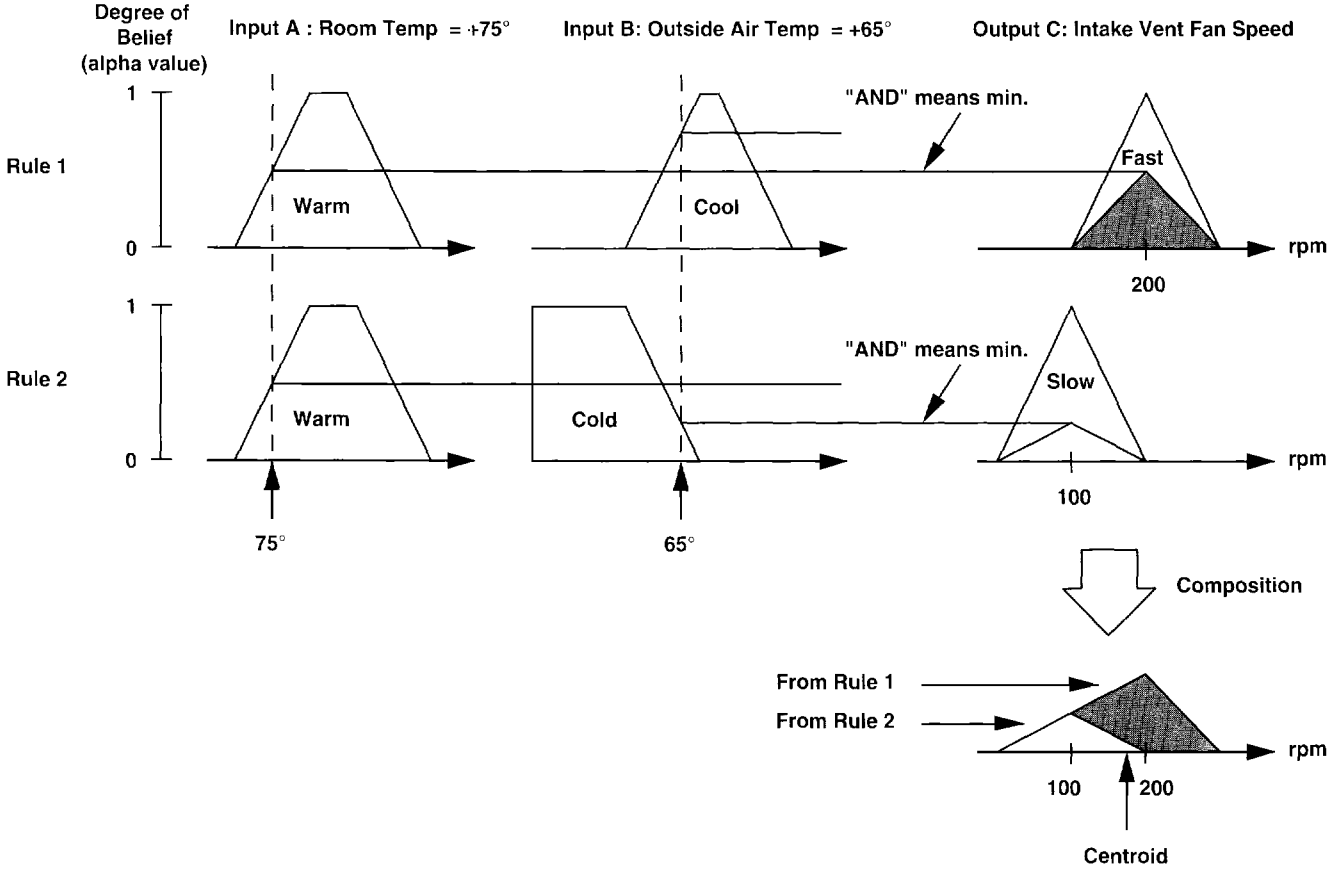
There are four main steps in performing a fuzzy computation:

1. Read in inputs and calculate the degrees of membership.
2. Evaluate rules in the system.
3. Perform conclusion membership function composition.
4. Defuzzify and output the result(s).

A diagram of the four main steps appears below. Each step is described in detail in the diagrams on the following pages.

EXAMPLE OF A FUZZY COMPUTATION

Rule 1: If Room_Temp is warm and Outside_ Temp is cool, then Intake _Fan Should_Be fast
Rule 2: If Room_Temp is warm and Outside_ Temp is cold, then Intake _Fan Should_Be slow



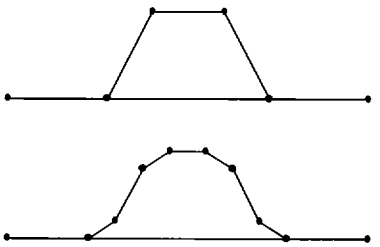
STEP 1. INPUT MEMBERSHIP FUNCTION EVALUATION

The VY86C500 device supports membership function representation in the following forms: as piecewise linear and as S-functions and its variants (Z-functions and pi-functions). To the right is a drawing showing each of the membership function representations described here and supported by the VY86C500.

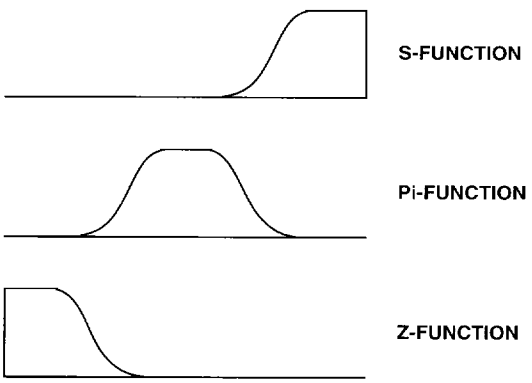
Different representations can be used with different input variables. Piecewise linear and S-functions (and its variants) can even be mixed within the same input variable at a small cost in RB memory usage. The VY86C500 fuzzy core places no restrictions on overlap of membership functions. All the membership functions may overlap if so desired.

VY86C500 PREMISE MEMBERSHIP FUNCTION REPRESENTATIVE TYPES

Examples of Piecewise Linear Representations



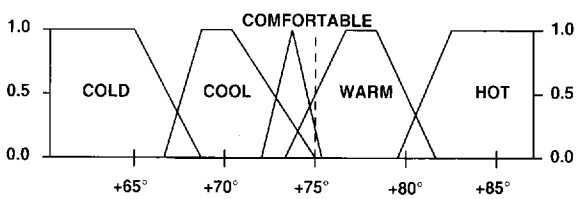
Examples of Quadratic S-Function Format Representations



A number of membership functions are defined for each input variable. The range of values for each input variable is usually plotted on the x-axis, as shown. The degrees of membership, or alpha values, are defined by membership functions and take on values between 0 and 1 and are usually plotted on the y-axis, as shown. The VY86C500 device does not require the degrees of membership to sum to 1 at any given input value.

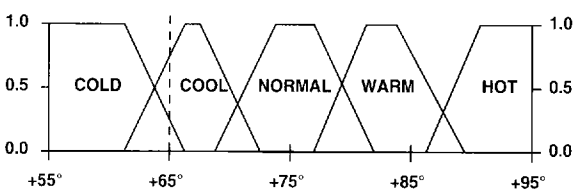
CALCULATE DEGREES OF BELIEF FOR PREMISE MEMBERSHIP FUNCTIONS

Input A: Room Temperature



In this example, since the room temperature is +75°, it will be COMFORTABLE with a degree of 0.25 and WARM with a degree of 0.5. The degrees of belief for COLD, COOL, and HOT will all be 0.

Input B: Outside Air Temperature

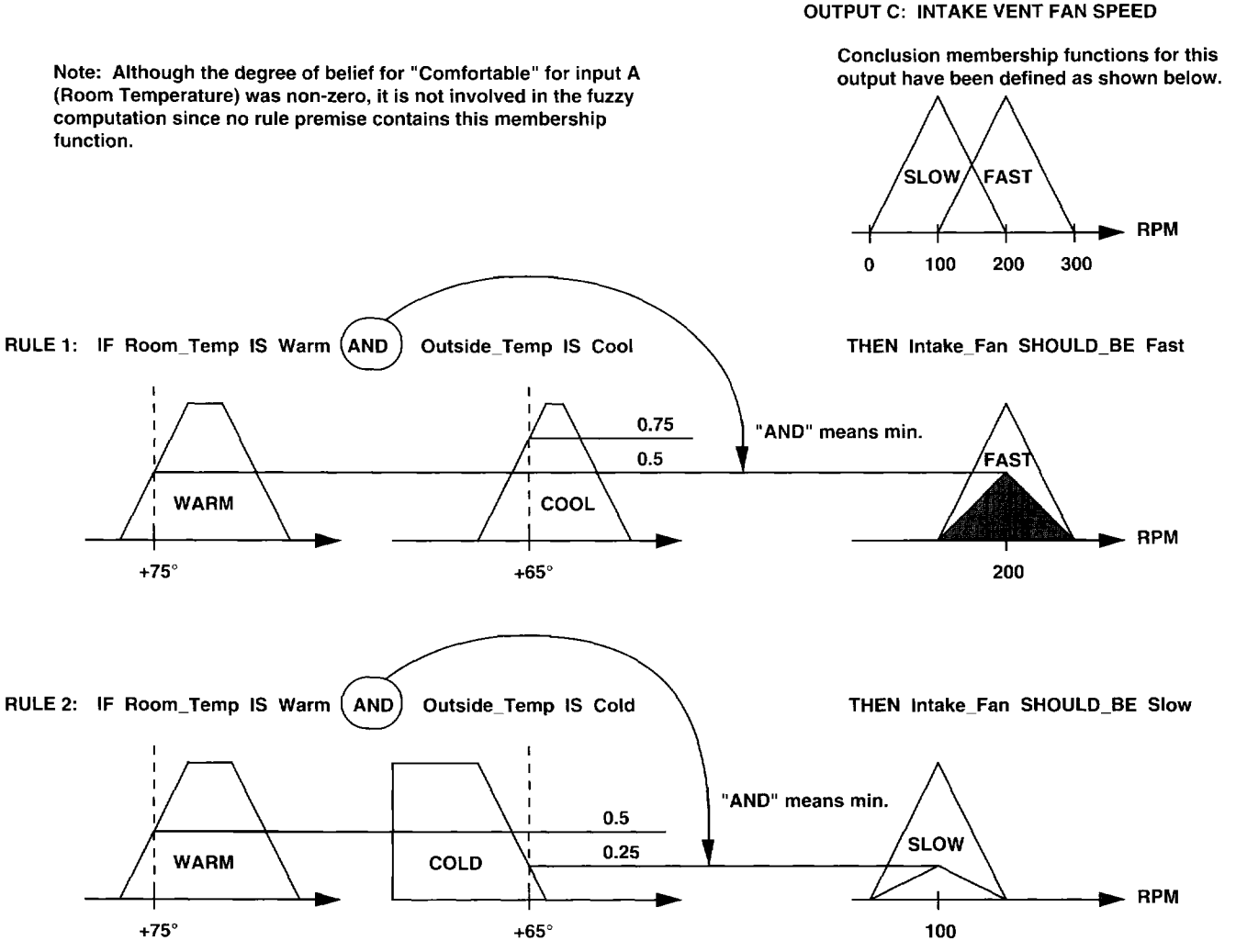


In this example, since the outside air temperature is +65°, it will be COLD with a degree of 0.25 and COOL with a degree of 0.75. The degrees of belief for NORMAL, WARM, and HOT will all be 0.

STEP 2. RULE EVALUATION

For rule statements, the VY86C500 core supports the fuzzy operators, AND, OR, and NOT. "A AND B" is equivalent to taking the minimum of the degrees of membership in A and in B. "A OR B" is equivalent to taking the maximum of the degrees of membership in A and in B. "NOT A" is equivalent to subtracting from 1 the degree of membership in A. A and B are usually associated with different input variables; however, that may not always be the case. The VY86C500 core architecture places no restrictions as to the number of membership functions allowed from the same input in a rule premise. The term, rule alpha, will refer to the value resulting from evaluating a rule.

EVALUATE RULES

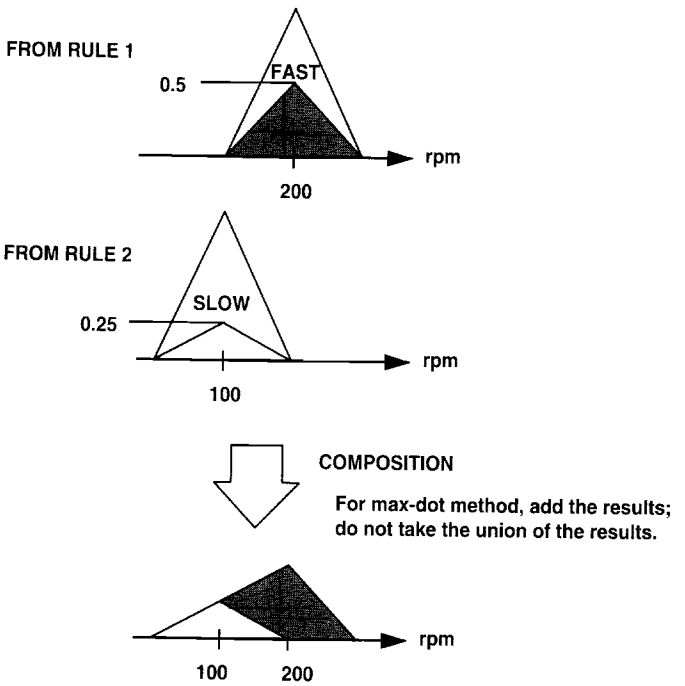


STEP 3. CONCLUSION MEMBERSHIP FUNCTION COMPOSITION

The VY86C500 core uses the product-sum method for conclusion membership function composition. The product-sum method is also known as "scaling" because the output membership function is scaled in the y-direction by a factor of the rule alpha value. This method has two desirable characteristics that for each scaled membership function, 1) the center of mass remains unchanged, and 2) the area scales by the same factor as the rule alpha value. The scaled conclusion membership functions are added to form the conclusion membership function composition. The product-sum method, along with the max-dot method is the most widely used method in the industry today.

PERFORM CONCLUSION MEMBERSHIP FUNCTION COMPOSITION

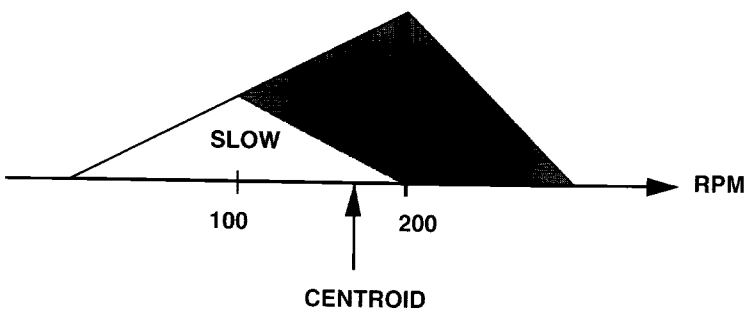
Output C: Intake Vent Fan Speed



STEP 4. DEFUZZIFICATION

Centroid, height, and alpha defuzzification methods have been implemented in the VY86C500 fuzzy logic coprocessor. As the name suggests, the centroid defuzzification method takes the scaled conclusion membership functions, adds them, and calculates the center of mass on the whole composition. The height defuzzification method is a special case of the centroid defuzzification method with each conclusion membership function having equal area. Note that height defuzzification does not refer to taking the point of maximal height of the conclusion membership function composition which is commonly referred to as the "maximal defuzzification method". The alpha defuzzification method allows the rule alphas to be output directly. In the case of the example, they will be 0.5 and 0.25. The centroid defuzzification method is the most widely used method in the industry today.

PERFORM DEFUZZIFICATION AND OUTPUT RESULT



This example fuzzy computation resulted in an output value of around 170 rpm for the intake vent fan speed.

SIGNAL DESCRIPTIONS (Cont.)

Signal Name	Pin Number	Signal Type	Signal Description
NC	60		No Connect
CLK1	62	O	1X Clock - This output is a CMOS level pulse train whose frequency is normally one-half that of the TCLK2 signal. It is used by the VL82C480, 486, and other on-board logic. Its frequency can be programmed to slower rates for Non-Turbo Mode operation.
CLKIN	63	I-C	Input Clock - CLKIN is the fundamental clock input to the VL82C480. It must be the same clock as that supplied to the 486.
M/IO#_DK0	74	IO-T	Memory Input/Output and DMA Acknowledge 0 - When HLDA is low, M/IO#_DK0 is driven by the local bus master and is decoded with D/C#_DK1 and W/R#_DK2 to indicate the type of bus cycle requested. When HLDA is high, this is an output signal which, along with D/C#_DK1 and W/R#_DK2, represents the encoded channel number being serviced at the beginning of DMA acknowledge cycles. If the VL82C480 makes ADS# active during DMA or Master Mode cycles (for local bus accesses) this signal is forced high.
D/C#_DK1	73	IO-T	Data/Control and DMA Acknowledge 1 - When HLDA is low, D/C#_DK1 is driven by the local bus master and is decoded with M/IO#_DK0 and W/R#_DK2 to indicate the type of bus cycle requested. When HLDA is high, this is an output signal which, along with M/IO#_DK0 and W/R#_DK2, represents the encoded channel number being serviced at the beginning of DMA acknowledge cycles. If the VL82C480 makes ADS# active during DMA or Master Mode cycles (for local bus accesses) this signal is forced high.
W/R#_DK2	72	IO-T	Write/Read and DMA Acknowledge 2 - When HLDA is low, W/R#_DK2 is driven by the local bus master and is decoded with D/C#_DK1 and M/IO#_DK0 to indicate the type of bus cycle requested. When HLDA is high, this is an output signal which, along with D/C#_DK1 and M/IO#_DK0, represents the encoded channel number being serviced at the beginning of DMA acknowledge cycles. If the VL82C480 makes ADS# active during DMA or Master Mode cycles (for local bus accesses) this signal indicates whether a local bus read or local bus write cycle is required.
D[31:0]	21:25, 27:31, 33:36, 38:47, 49:56	IO-T	CPU Data bus bits 31 through 0 - This is the data bus directly connected to the CPU and other external devices.
EADS#	69	IO-T	External Address [POR input] - This signal indicates a primary cache invalidation address is on the address bus. It is driven low by the VL82C480 to perform primary cache invalidations during DMA and Master Mode cycles and is also driven low for one cycle during the first T2 of a local bus write to a write-protected memory location. It is also modulated for implementation of the Non-Turbo Mode. At power-on reset (POR), this pin is used to set the polarity of the DKEN input.
FERR#	65	I-TPU	Floating Point Error - The FERR# input indicates a floating point error. When active, it generates an interrupt IRQ13 internal to the VL82C480. This input pin is active low. FERR# has an internal pull-up resistor.
HLDA	70	I-T	Hold Acknowledge - The active high HLDA signal is issued by the CPU in response to the HOLD driven by the VL82C480. It indicates that the CPU is floating its outputs to the high impedance state so that another master can take control of the bus.
HOLD	71	O	Hold Request - The active high HOLD pin is driven by the VL82C480 to the CPU. It indicates that a bus master, such as a DMA or refresh controller, is requesting control of the bus. This signal is synchronized to CLKIN.

PERFORMANCE SUMMARY

Parameter	Performance	Condition and Notes
Fuzzy Operations	Fuzzy AND, Fuzzy OR, Fuzzy NOT	
Fuzzy Inference Method	Product-sum	
Defuzzification Method	Centroid, Height, and Alpha	
Number of Inputs	1 to 2048	Note 1
Number of Outputs	1 to 4096	Note 1
Input Membership Function Shapes	Arbitrary. Expressed in piecewise linear and S-function formats	Membership function overlap is not restricted in any way. Number of points used in a piecewise linear membership function is limited by RB memory size.
Output to Membership Function Shapes	Arbitrary	
Number of Input Membership Functions	1 to 4096 per input	Note 1
Number of Output Membership Functions	1 to 4096 per output	Note 1
Number of Rules	1 to 2048 per output	Note 1
Computation Speed	72 microseconds	20 MHz, 8 inputs, 4 Outputs, 6 Points per consequents per rule, 20 rules, centroid defuzzification. Refer to information on other rule bases.
Rule Evaluation Speed	870,000 rule evaluations per second	20 MHz, 2 inputs, 1 outputs, 6 points second per membership function, 49 rules, centroid defuzzification. Refer to performance estimate table (pg. 9) for additional information on other rule bases.
Resolution	12-bit unsigned data	Input data, alpha data, and output data

Note:

1. The actual limits for these parameters are determined by OCTD memory size and RB memory size. Each of these parameters affects OCTD memory and RB memory requirements, and they must all be treated together when calculating memory requirements. The estimate equations can be used to calculate approximate memory requirements. The FCADS software outputs exact memory requirements.

SUMMARY OF TERMS

Fuzzy Logic
A generalization of the mathematical concept of set membership in which, unlike traditional Boolean logic that requires a "yes/no" type of response, an element may have only partial membership in the set. The application of fuzzy logic has spread into various areas including embedded control, aerospace, telecommunications, and pattern recognition.

Membership Function
A function describing the degree to which an element belongs to (is a member of) a set. It is customary to define membership function values between 0 and 1 with 0 being "no membership" and 1 being "full membership".

Premise Membership Function
Synonymous to input membership function.

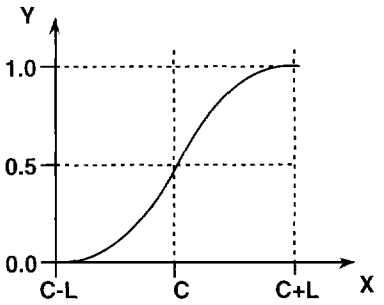
Conclusion Membership Function
Synonymous to output membership function.

Alpha Value
Synonymous to "the degree of belief/membership" or "the belief/truth value" when referring to a membership function alpha value. The rule alpha value is the result of performing fuzzy operations on several membership function alpha values.

Piecewise Linear
A method of representing membership functions with linear segments. Any arbitrary membership function shape can be approximated by a piecewise linear representation.

Quadratic S-Function
A method of representing membership functions in a quadratic form. In the case of the VY86C500 core architecture, S-function formats must begin and end at 0 or 1. The π-function and the Z-function are variants of the S-function and together will be referred to as the S-function format. The mathematical formula for an S-function is described below.

S-FUNCTION DIAGRAM



Notice that the S-function is first order continuous, but not second order continuous. The second order discontinuity of the S-function equation occurs at the point X=C.

MAIN EQUATIONS:

$$Y = \begin{cases} 0 & X \le C - L \\ \dfrac{(X - (C-L))^2}{2(L)^2} & C - L < X \le C \\ 1 - \dfrac{((C+L) - X)^2}{2(L)^2} & C < X < C + L \\ 1 & X \ge C + L \end{cases}$$

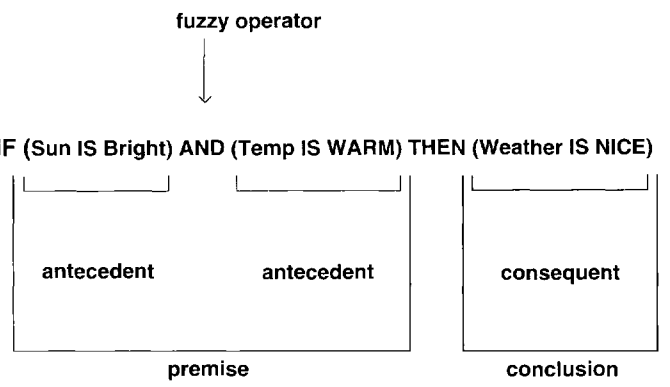
FIRST ORDER DERIVATIVES:

$$Y' = \begin{cases} 0 & X \le C - L \\ \dfrac{(X - (C-L)}{L^2} & C - L < X \le C \\ \dfrac{(C+L) - X}{L^2} & C < X < C + L \\ 0 & X \ge C + L \end{cases}$$

SECOND ORDER DERIVATIVES:

$$Y'' = \begin{cases} 0 & X \le C - L \\ \dfrac{1}{L^2} & C - L < X \le C \\ -\dfrac{1}{L^2} & C < X < C + L \\ 0 & X \ge C + L \end{cases}$$

Rule
A key element of a fuzzy rule base. There are two components to a rule: the premise (or the left-hand side or the input side) and the conclusion (or the right-hand side or the output side). The premise is comprised of antecedent(s) related by fuzzy operators, and the conclusion consists of consequent(s). An example is displayed to the right with each of the parts labeled.



Inference
The complete cycle of performing a fuzzy computation to derive the output value from the input values and the rule base. Includes input membership function evaluation, rule evaluation, conclusion membership function composition, and defuzzification.

Inference Speed
The speed at which a full inference or a full fuzzy computation is performed. Synonymous to full fuzzy computation speed. A full fuzzy computation includes input membership function evaluation, rule evaluation, conclusion membership function composition, and defuzzification.

Max-Min
A method of fuzzy inferencing. The rule alpha values are used to truncate or clip the conclusion membership functions at that value. The conclusion membership function composition is performed by union, and the maximum point is used for defuzzification.

Product-Sum
A method of fuzzy inferencing. The rule alpha values are used to scale the conclusion membership functions by that factor. The conclusion membership function composition is performed by addition, and centroid defuzzification is performed on the composition.

Defuzzification
Process in which a crisp output is derived from a fuzzy system.

Rule Base
All information regarding a fuzzy system. Includes inputs, input membership functions, outputs, output membership functions, rules.

Observation
Synonymous to "input value". Observations are the inputs to a fuzzy system.

Conclusion
i) Synonymous to "output value". Conclusions are the outputs of a fuzzy system.

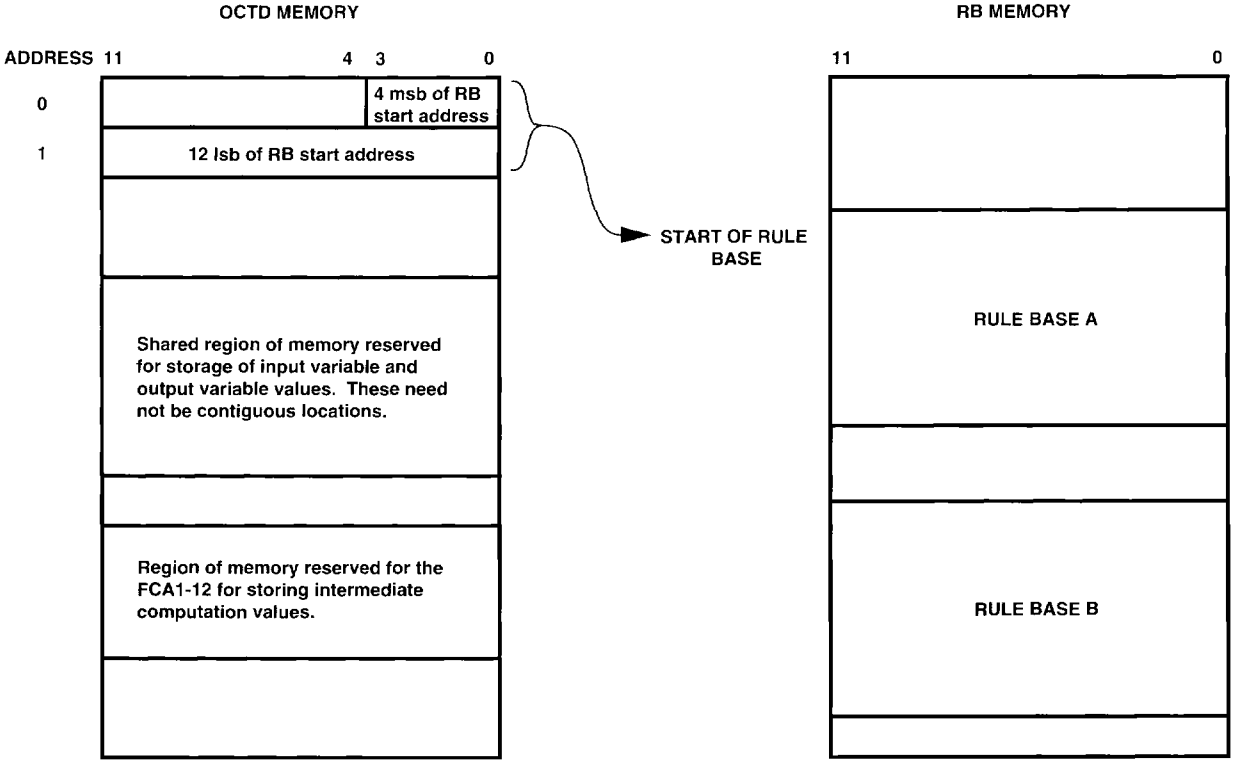
ii) The right hand portion of a rule statement following the "THEN" clause and consisting of consequent(s).

RB Memory
The external memory in which the rule base(s) are stored.

OCTD Memory
The external memory used to transfer input and output values between a host and the VY86C500 coprocessor. OCTD memory is also used to store temporary data.

OCTD MEMORY MAP
OCTD memory mapping is shown below. The only fixed locations are locations 0 and 1. The input variable and output variable locations must be defined and compiled into the rule base memory. Please see Theory of Operations section (pg. 14) for a detailed description of the OCTD memory.

OCTD MEMORY MAP DIAGRAM



THEORY OF OPERATION

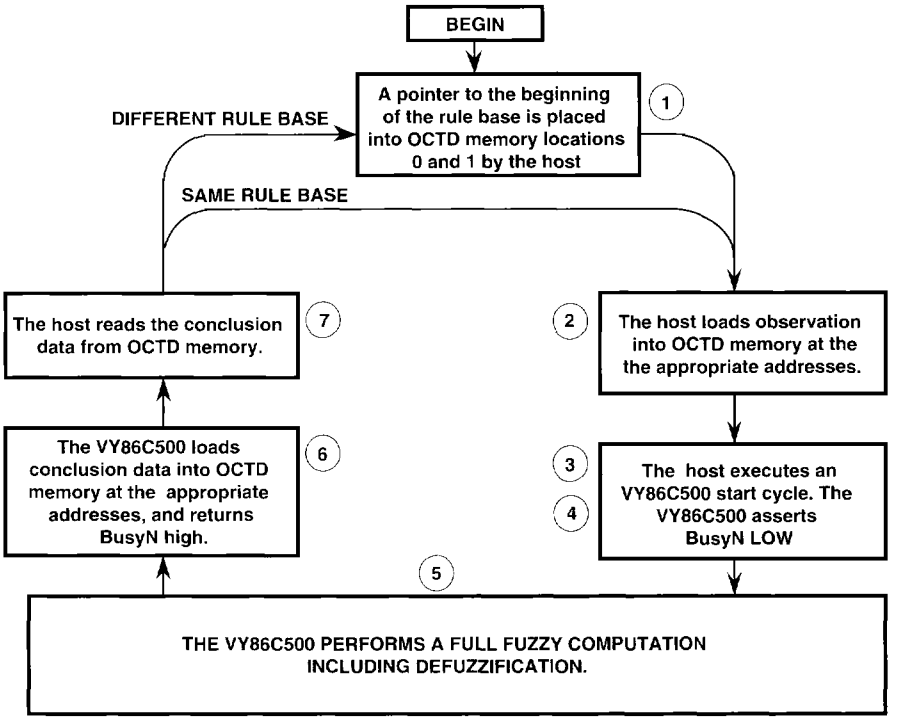
SEQUENCE OF EVENTS / HOST PROCESSOR INTERFACE

The VY86C500 device is a self-contained fuzzy computational core. Knowledge of the internal operations of the VY86C500 is not necessary to carry out a fuzzy computation as long as the rule base is properly loaded into RB memory. The host (host processor, host controller, etc.) need only issue a start signal to execute a fuzzy computation using the VY86C500 fuzzy logic accelerator. The following describes the sequence of events for fuzzy computations.

1. Prior to executing fuzzy computations, the rule base must be loaded into RB memory and a pointer to the beginning of the rule base must be placed in OCTD memory locations 0 and 1. The most significant portion is placed at location 0 and the least significant at location 1.
2. The host loads input (observation) data into OCTD memory at the appropriate addresses.
3. The host asserts the StartN connector low for one cycle to start fuzzy computation.
4. The VY86C500 device asserts the BusyN connector low.
5. The VY86C500 device performs the fuzzy computation and places the output (conclusion) data into OCTD memory at the appropriate addresses.
6. The VY86C500 device drives the BusyN connector high to signal completion of the fuzzy computation.
7. The host reads output (conclusion) data from OCTD memory at the appropriate addresses. Repeat steps 2-7 for subsequent fuzzy computations on the same rule base. Repeat steps 1-7 if evaluating a different rule base.

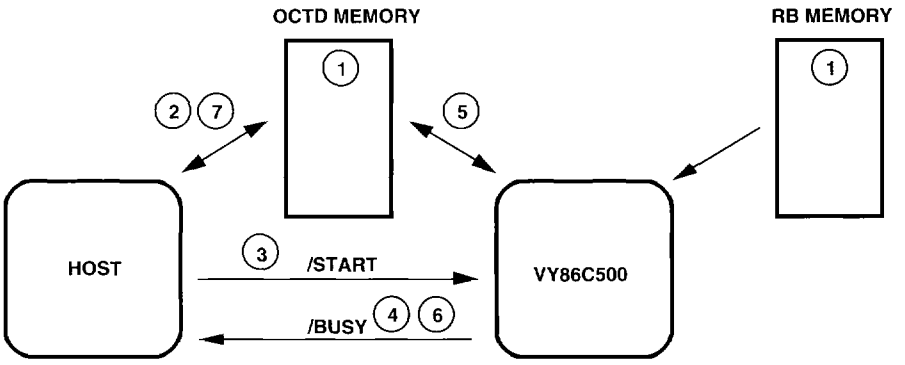
Timing diagrams for fuzzy computations can be found on pages 19-20.

SEQUENCE OF EVENTS 1



Note: The numbers inside circles correspond to steps described at left.

SEQUENCE OF EVENTS 2



Note: The numbers inside circles correspond to steps described at left

RB MEMORY/OCTD MEMORY/ PERFORMANCE SPEED

All information about a specific fuzzy system (or rule base) resides in external memory called the RB (rule base) memory. A rule base needs to be loaded into RB memory prior to the start of fuzzy computations. RB memory can be RAM or ROM. The VY86C500 device interface to the RB memory consists of one control signal (RBoeN), a 16-bit address bus (RBaddress), and a 12-bit data bus (RBdata). During execution of a fuzzy computation, the RB memory is constantly accessed by the VY86C500 device. The RB memory should not be accessed by external circuitry while BusyN is asserted low. The VY86C500 device supports up to 64K 12-bit words of RB memory.

The OCTD memory serves two purposes. One is for the transfer of input and output values between the host and the VY86C500 coprocessor. The host would load input values (observations) into predetermined locations in the OCTD memory prior to executing a fuzzy computation. During execution of a fuzzy computation, the VY86C500 accelerator would read these locations to obtain the input values. At the end of a fuzzy computation, the VY86C500 accelerator would write the output values (conclusions) into predetermined locations in the OCTD memory and drive BusyN high. The host can then access these locations to obtain the output values. The second purpose of the OCTD memory is to serve as temporary storage of intermediate computation values. This feature is useful in the rule base debug process.

The VY86C500 coprocessor interface to the OCTD memory consists of three control signals (OCTDceN, OCTDoeN, OCTDweN), a 12-bit address bus (OCTDaddress), and a 12-bit data bus (OCTDdata). For memories with only two control signals (not chip enable, and not write enable), it is suggested that 1) OCTDoeN be "and"ed with OCTDweN and be connected to the "not chip enable" of the memory, 2) OCTDoeN be inverted and be connected to the "not write enable" of the memory. The OCTD memory should not be accessed by external circuitry while BusyN is asserted low. The VY86C500 device supports up to 4K 12-bit words of OCTD memory.

Many parameters affect the fuzzy computation speed (inference speed) of the VY86C500 fuzzy logic accelerator: number of inputs, number of outputs, rules, membership function representation type, even the values of the input variables during run time. In general, the S-function format requires the least amount of RB memory and computes at the approximate speed of an eight-point piecewise linear format. (Please refer to the Estimate Table on page 9.)

The following equations can be used to estimate RB memory and OCTD memory requirements, as well as computation speed for a given rule base. The equations provide only estimates and should be treated as such. The equations do not take into account optimizations by the FCADS compiler or any special situations that might occur with a particular rule base.

These equations assume that the number of antecedents per rule equals the number of inputs and the number of consequents per rule equals the number of outputs per rule.

Definition of variables used in estimate equations:

In	=	number of inputs
Mem	=	average number of input membership functions per input
Type	=	premise membership function representation type 24 for six-point piecewise linear representations 40 for 10-point piecewise linear representations 9 for S, Z, π-functions representations
Rule	=	number of rules in the rule base
Out	=	number of outputs

Estimate equations, Out ≤ 4 :

OCTD memory size = (in 12-bit words)	((Mem + 1) * In) + Out + 2
RB memory size = (in 12-bit words)	((((Type + 2) * Mem) + 3) * In) + ((In + (2 * Out) + 1) * Rule) + Out + 8
Fuzzy Computation Speed = (in ns)	((16 * Mem + 6) * In) + (51 * Out) + 13 + (8 * Out + In + 7) * Rule) * 50 ns

Estimate equations, 4 < Out ≤ 8 :

(OCTD memory size was minimized for these estimates. Conversely, RB memory size can be reduced at the cost of OCTD memory size.)

OCTD memory size = (in 12-bit words)	((Mem + 1) * In) + Out + 2
RB memory size = (in 12-bit words)	((((Type + 2) * Mem) + 3) * In) + ((In + Out) * 2 * Rule) + Out + 10
Fuzzy Computation Speed = (in ns)	((16 * Mem + 6) * In) + (51 * Out) + 18 + (4 * Out + In + 6) * 2 * Rule) * 50 ns

The use of the FCADS software is recommended if the number of outputs exceeds eight. The FCADS provides exact size/speed values.

BIST/FAULT COVERAGE

BIST has been implemented in the VY86C500 device to eliminate the need for test pattern generation. Most aspects of the BIST for the VY86C500 device are similar to that for the COMPASS BISTRAM, including compatibility with the COMPASS boundary scan library. There are 4 inputs (TestMode, ScanModeN, InitializeN, ScanIn) and 4 outputs (EndOfTest, Pass1, Pass2, ScanOut) to the BIST circuitry. BIST is started when TestMode is asserted high and InitializeN is cycled low for one cycle. During built-in self-test the OCTDdata and RBdata busses cannot be driven externally, and StartN and ResetN should be held high. BIST automatically terminates after 1010 cycles or can be aborted at anytime by driving TestMode connector low. If BIST is aborted prematurely the test results are invalid. At the end of the BIST sequence, the EndOfTest connector asserts high. At that time, the Pass1 and Pass2 connectors will assert high if the test passed. Both Pass1 and Pass2 must be high for the BIST sequence to be considered a pass. When the EndOfTest connector is high, the test results (Pass1 and Pass2) can be scanned out. Driving ScanModeN low while TestMode is high, begins scanning of the test results registers. The ScanIn and ScanOut connectors can be connected to other test results registers to form a scan chain. When the TestMode connector is driven low, the VY86C500 device goes through a 6-cycle reset during which time a fuzzy computation or a BIST sequence can not be started. BIST for the VY86C500 device yields a fault coverage of 97% and can operate at 20 MHz. The sequence of events for a built-in self-test is outlined below.

1. The StartN and ResetN connectors must be driven high, and the RBdata and OCTDdata busses must not be driven externally.
2. The InitializeN connector is asserted low for 1 cycle while the TestMode connector is asserted high. The TestMode connector must be held high for the duration of the BIST sequence.
3. The VY86C500 device drives the BusyN and EndOfTest connectors low and self-test begins. The EndOfTest connector remains low until the BIST sequence is completed or aborted.
4. Upon completion of the BIST sequence, the VY86C500 device drives the BusyN and EndOfTest connectors high.
5. The test result appears at the Pass1 and Pass2 connectors. The BIST sequence passes only if both Pass1 and Pass2 are high. The Pass1 and Pass2 connectors hold their states until the next BIST sequence is started.
6. (Optional) The Pass1 and Pass2 test results can be scanned out at this time. Asserting the ScanModeN connector low begins scanning of the test results. The TestMode and InitializeN connectors must be held high during scanning of the test results. At the end of the scan sequence, the ScanModeN connector must be returned high.
7. The TestMode connector is driven low to return to normal mode. The VY86C500 device begins a six-cycle reset during which a fuzzy computation or a BIST sequence may not be initiated.

The BIST Sequence Timing Diagrams can be found on pages 24-25.

SOFTWARE AND TOOLS

FUZZY PROGRAMMING LANGUAGE AND FCA DEVELOPMENT SYSTEM

Fuzzy Programming Language (FPL) is a high-level language for programming fuzzy systems. The TILShell provides a graphical user interface for creating fuzzy systems and generates FPL format code. Another way of creating an FPL source is to program directly in FPL. An example of an FPL source program listing appears below.

This program estimates the function, $Z = (X^2 + Y^2) / 10$. X and Y are the inputs; Z is the output.

PROJECT ESTIMATE

```
VAR X
 TYPE signed byte¡
 MIN -20
 MAX 20

  MEMBER NB
   POINTS -20,1 -10,0
  END

  MEMBER NS
   POINTS -20,0 -10,1 0,0
  END

  MEMBER Z
   POINTS -10,0 0,1 10,0
  END

  MEMBER PS
   POINTS 0,0 10,1 20,0
  END

  MEMBER PB
   POINTS 10,0 20,1
  END
END
```

```
VAR Y
 TYPE signed byte
 MIN -20
 MAX 20

  MEMBER NB
   POINTS -20,1 -10,0
  END

  MEMBER NS
   POINTS -20,0 -10,1 0,0
  END

  MEMBER Z
   POINTS -10,0 0,1 10,0
  END

  MEMBER PS
   POINTS 0,0 10,1 20,0
  END

  MEMBER PB
   POINTS 10,0 20,1
  END
END
```

```
VAR Z
  TYPE signed byte
  MIN -10
  MAX 90

   MEMBER VVL
    POINTS -10,0 0,1 10,0
   END

   MEMBER VL
    POINTS 0,0 10,1 20,0
   END

   MEMBER LOW
    POINTS 10,0 20,1 30,0
   END
   MEMBER HI
    POINTS 30,0 40,1 50,0
   END

   MEMBER VH
    POINTS 40,0 50,1 60,0
   END

   MEMBER VVH
    POINTS 70,0 80,1 90,0
   END
 END
```

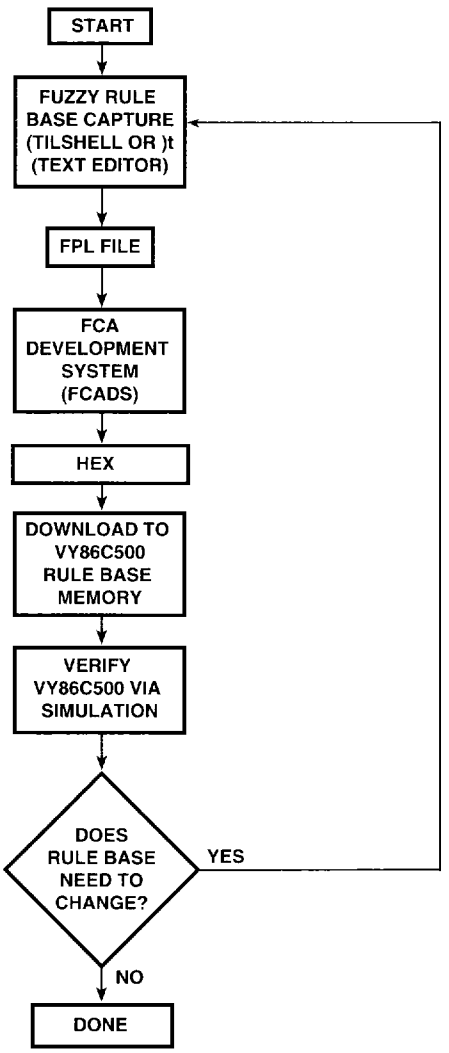
FUZZY RULES

```
RULE Rule0
 IF x IS NB AND y IS NB THEN
 z=VVH
END

RULE Rule1
 IF x IS NB AND y IS NS THEN
 z=VH
END

RULE Rule2
 IF x IS NB AND y IS Z THEN
 z=HI
END

RULE Rule3
 IF x IS NB AND y IS PS THEN
 z=VH
END

RULE Rule4
 IF x IS NB AND y IS PB THEN
 z=VVH
END

RULE Rule5
 IF x IS NS AND y IS NB THEN
 z=VH
END

RULE Rule6
 IF x IS NS AND y IS NS THEN
 z=LOW
END

RULE Rule7
 IF x IS NS AND y IS Z THEN
 z=VL
END

RULE Rule8
 IF x IS NS AND y IS PS THEN
 z=LOW
END

RULE Rule9
 IF x IS NS AND y IS PB THEN
 z=VH
END

RULE Rule10
 IF x IS Z AND y IS NB THEN
 z=HI
END

RULE Rule11
 IF x IS Z AND y IS NS THEN
 z=VL
END

RULE Rule12
 IF x IS Z AND y IS Z THEN
 z=VVL
END

RULE Rule13
 IF x IS Z AND y IS PS THEN
 z=VL
END

RULE Rule14
 IF x IS Z AND y IS PB THEN
 z=HI
END

RULE Rule15
 IF x IS PS AND y IS NB THEN
 z=VH
END

RULE Rule16
 IF x IS PS AND y IS NS THEN
 z=LOW
END

RULE Rule17
 IF x IS PS AND y IS Z THEN
 z=VL
END

RULE Rule18
 IF x IS PS AND y IS PS THEN
 z=LOW
END

RULE Rule19
 IF x IS PS AND y IS PB THEN
 z=VH
END

RULE Rule20
 IF x IS PB AND y IS NB THEN
 z=VVH
END

RULE Rule21
 IF x IS PB AND y IS NS THEN
 z=VH
END

RULE Rule22
 IF x IS PB AND y IS Z THEN
 z=HI
END

RULE Rule23
 IF x IS PB AND y IS PS THEN
 z=VH
 END

RULE Rule24
 IF x IS PB AND y IS PB THEN
 z=VVH
 END
END

CONNECT
 FROM x
 TO Rules
END

CONNECT
 FROM y
 TO Rules
END

CONNECT
 FROM Rules
 TO z
END

END
```

Once an FPL source code is created, the Fuzzy Computational Accelerator Development System (FCADS) software is used to compile FPL source code into FCA image format. The compiled output is loaded into RB memory and contains all the information necessary for the VY86C500 device to perform fuzzy computations. The VY86C500 device can begin executing fuzzy computations once the pointer to the start of the rule base is placed in locations 0 and 1 of the OCTD memory, and the input variable values are written into appropriate locations in the OCTD memory.

The diagram below illustrates the process of creating a compiled FCA image.

FCADS DIAGRAM



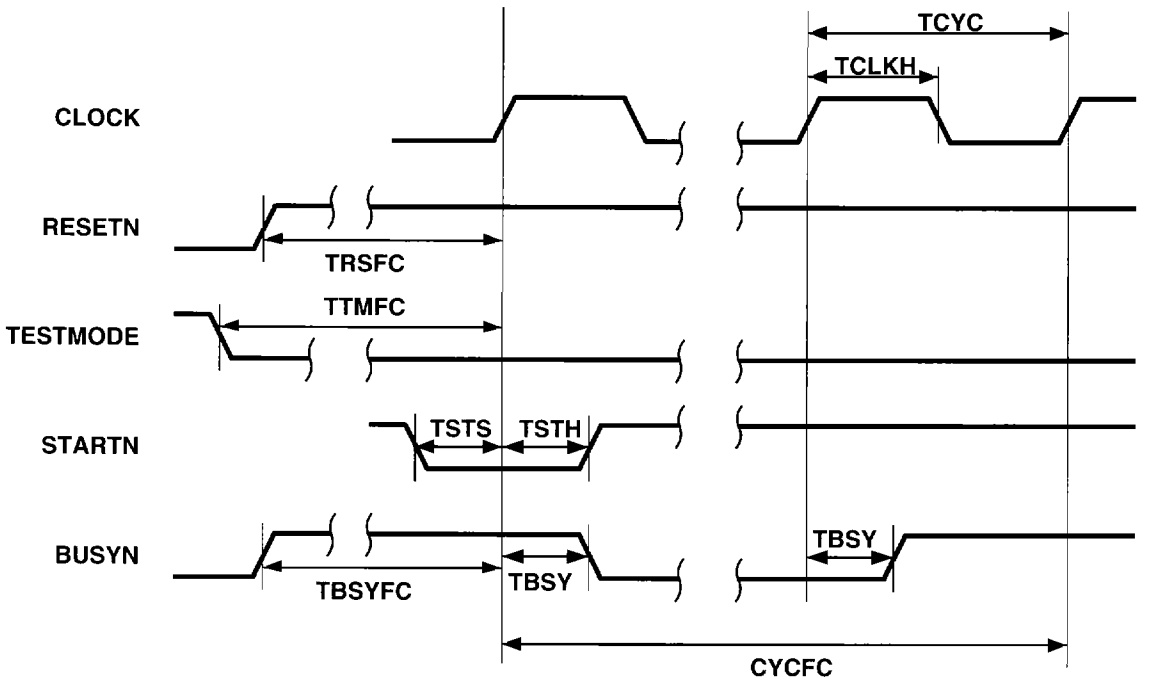
TIMING DIAGRAMS

Symbol	Parameter	Min	Max	Units	Notes
TCYC	CLK Cycle Time	50.0		ns	1
TCLKH	CLK High Time	20.0	30.0	ns	1
TRSFC	ResetN to Fuzzy Comp. Start	1 cycle + TRSS			
TTMFC	TestMode to Fuzzy Comp. Start	6 cycles + TTMS			
TSTS	StartN Setup Time	1.5		ns	
TSTH	StartN Hold Time	1.0		ns	
TBSYFC	BuzyN to Fuzzy Comp. Start	1 cycle -TBSY			
TBSY	BuzyN Delay		9.6 + 1.6 * Cld	ns	
CYCFC	Fuzzy Computation Cycles	Note 2	Note 2	cycles	2
TRSS	ReSetN Setup Time	1.5		ns	
TRSH	ReSetN Hold Time	1.0		ns	
CYCRS	Reset Cycles	4		cycles	

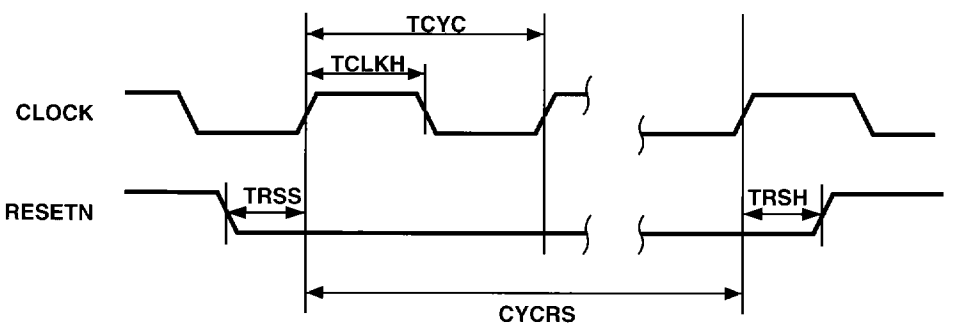
Notes:
1. CLK timings measured between clock edges at 50% of VDD.
2. Fuzzy computation cycle vary with the values of the inputs (observations) and the rule base being evaluated. Refer to section on performance for estimate numbers.

MAIN SIGNALS

FUZZY COMPUTATION



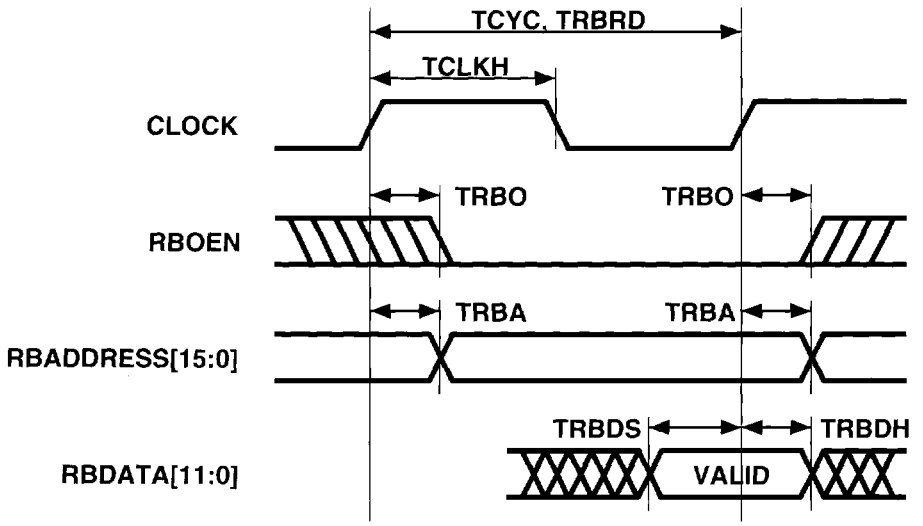
RESET



RB BUS SIGNALS

Symbol	Parameter	Min	Max	Units	Notes
TCYC	CLK Cycle Time	50.0		ns	
TCLKH	CLK High Time	20.0	30.0	ns	
TRBRD	RB Read Cycle Time	1		cycle	1
TRBO	RBoeN Delay		8.4 + 1.8 * Cld	ns	1
TRBA	RBaddress Delay		10.4 + 2.8 * Cld	ns	1
TRBDS	RBdata Setup Time	4.7		ns	1
TRBDH	RBdata Hold Time	1.4		ns	

READ CYCLE



Notes:

1. From these values, RB memory access time requirements can be calculated as follows:
   RB memory address access time = TRBRD - (TRBA + TRBDS)

OCTD BUS SIGNALS

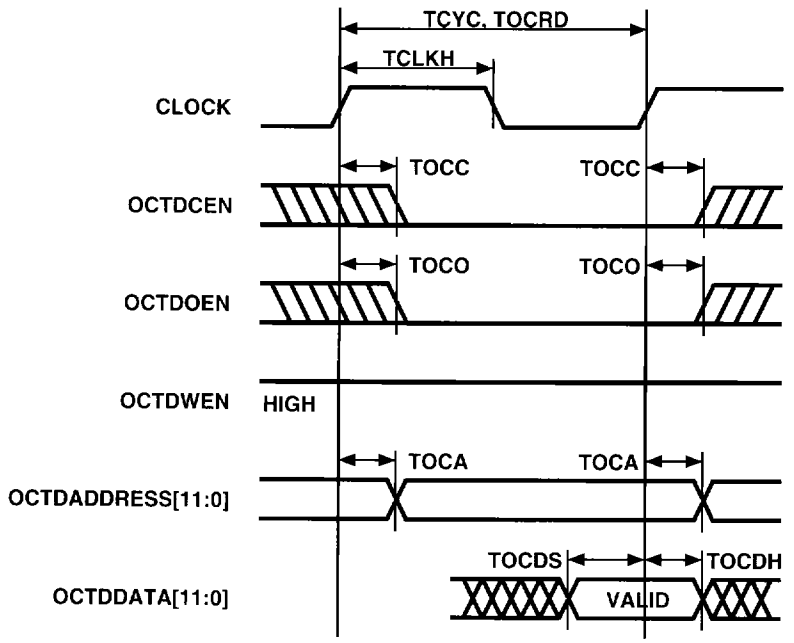
Symbol	Parameter	Min	Max	Units	Notes
TCYC	CLK Cycle Time	50.0		ns	
TCLKH	CLK High Time	20.0	30.0	ns	
TOCRD	OCTD Read Cycle Time	1		cycle	1
TOCC	OCTDceN Delay		7.3 + 2.5 * Cld	ns	1
TOCO	OCTDoeN Delay		7.6 + 2.8 * Cld	ns	1
TOCA	OCTDaddress Delay		18.1 + 1.3 * Cld	ns	1
TOCDS	OCTDdata Setup Time		4.2	ns	
TOCDH	OCTDdata Hold Time	1.4		ns	
TOCWR	OCTD Write Cycle Time	2		cycles	
TOCW	OCTDweN Delay		7.6 + 2.5 * Cld	ns	
TOCD	OCTDdata Delay		11.7 + 0.6 * Cld	ns	

Note:

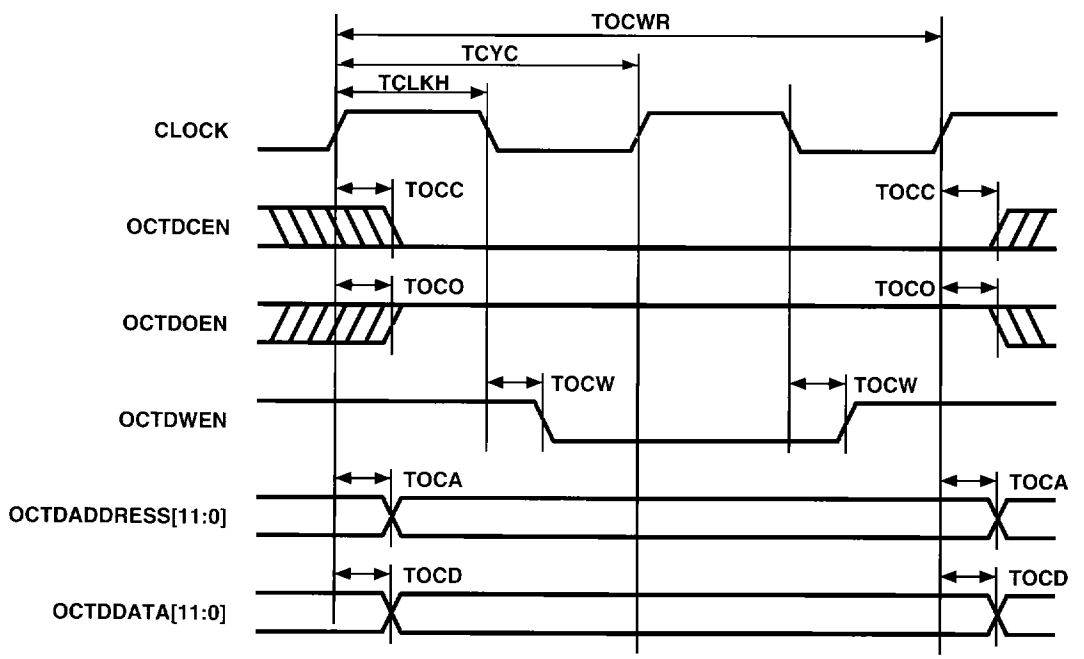
1. From these values, OCTD memory access time requirements can be calculated as follows:
   OCTD memory address access time = TOCRD - (TOCA + TOCDS)
   OCTD memory chip enable access time = TOCRD - (TOCC + TOCDS)
   OCTD memory output enable access time= TOCRD - (TOCO + TOCDS)

OCTD BUS SIGNALS (Cont.)

READ CYCLE



WRITE CYCLE

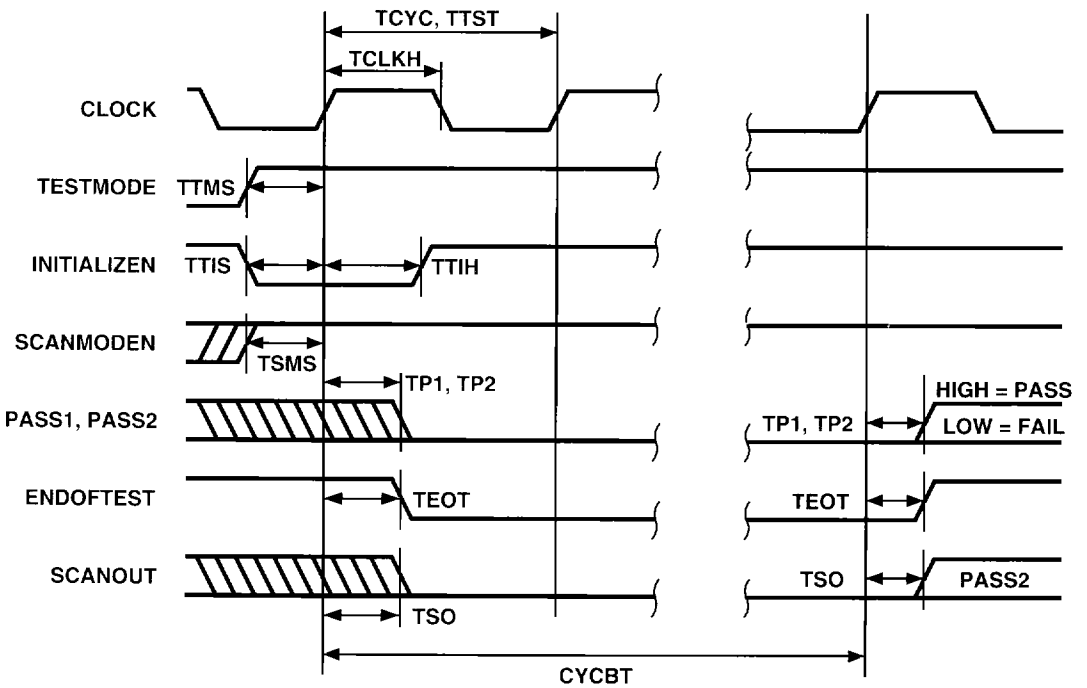


BIST SIGNALS

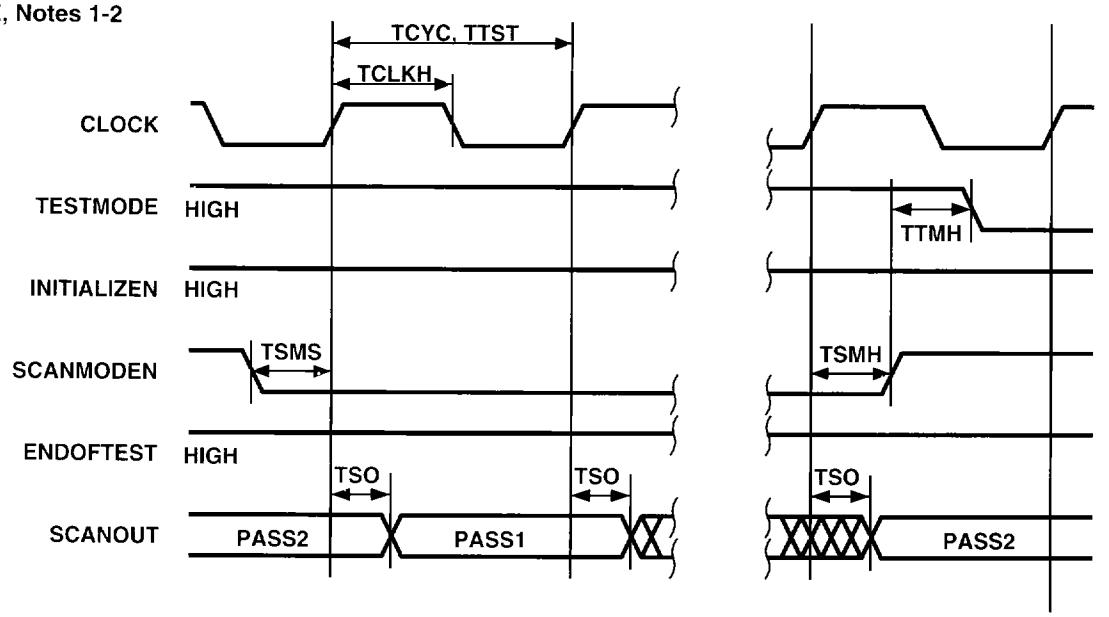
Symbol	Parameter	Min	Max	Units
TCYC	CLK Cycle Time	50.0		ns
TCLKH	CLK High Time	20.0	30.0	ns
TTST	CLK Cycle Time during Test	50.0		ns
TTMS	TestMode Setup Time	6.5		ns
TTMH	TestMode Hold Time	1.0		ns
TTIS	InitializeN Setup Time	6.0		ns
TTIH	InitializeN Hold Time	1.0		ns
TSMS	ScanModeN Setup Time	3.4		ns
TSMH	ScanModeN Hold Time	1.0		ns
TP1	Pass1 Delay		9.1 + 1.9 * Cld	ns
TP2	Pass2 Delay		9.1 + 1.9 * Cld	ns
TEOT	EndOfTest Delay		8.4 + 1.5 * Cld	ns
TSO	ScanOut Delay		3.7 + 2.9 * Cld	ns
CYCBT	BIST Cycles	997	997	cycles

BIST SIGNALS (Cont.)

SELF TEST MODE, Notes 1-2



SCAN MODE, Notes 1-2



Notes:

1. During SELF-TEST and SCAN, RESETN and STARTN must be HIGH, and OCTDADDRESS and RBADDRESS may not be driven externally.
2. SCANMODEN should not be asserted low for at least 1 cycle after ENDOFTEST returns high.

NOTES:

NOTES:

VLSI TECHNOLOGY, INC.

VY86C500

VLSI CORPORATE OFFICES

CORPORATE HEADQUARTERS • VLSI PRODUCT DIVISIONS
1109 McKay Drive • San Jose, CA 95131 • 408-434-3100

PORTABLE SYSTEMS DIVISION AND DESKTOP SYSTEMS DIVISION
8375 South River Parkway • Tempe, AZ 85284 • 602-752-8574

VLSI SUBSIDIARY

COMPASS DESIGN AUTOMATION, INC.
1865 Lundy Avenue • San Jose, CA 95131 • 408-433-4880

VLSI SALES OFFICES AND TECH CENTERS

UNITED STATES

ARIZONA
Tempe, 602-752-6450
FAX 602-752-6001

CALIFORNIA
San Jose, 408-922-5200
FAX 408-922-5252

Encino, 818-609-9981
FAX 818-609-0535

Irvine, 714-250-4900
FAX 714-250-9041

COLORADO
Fort Collins, 303-223-6944
FAX 303-223-7355

FLORIDA
Pompano Beach, 305-971-0404
FAX 305-971-2086

GEORGIA
Duluth, 404-476-8574
FAX 404-476-3790

ILLINOIS
Hoffman Estates, 708-884-0500
FAX 708-884-9394

MARYLAND
Millersville, 410-987-8777
FAX 410-987-4489

MASSACHUSETTS
Wilmington, 508-658-9501
FAX 508-657-6420

NEW JERSEY
Plainsboro, 609-799-5700
FAX 609-799-5720

NORTH CAROLINA
Durham, 919-544-1891/92
FAX 919-544-6667

TEXAS
Richardson, 214-231-6716
FAX 214-669-1413

WASHINGTON
Bellevue, 206-453-5414
FAX 206-453-5229

INTERNATIONAL

FRANCE
Palaiseau Cedex, 1-69-19-71-00
FAX 1-69-19-71-01

GERMANY
Muenchen, 89-92795-0
FAX 89-92795-145

HONG KONG
Wanchai, 852-802-7755
FAX 852-802-7959

ITALY
Argrate Brianza, 39-6056791
FAX 39-6056808

JAPAN
Tokyo, 03-3239-5211
FAX 03-3239-5215

Chuo-ku, Osaka, 06-243-6041
FAX 06-243-6960

TAIWAN
Taipei, 886-2-758-6500
FAX 886-2-757-6013

UNITED KINGDOM
Milton Keynes, 09 08/66 75 95
FAX 09 08/67 00 27

VLSI SALES OFFICES

UNITED STATES

ALABAMA
Huntsville, 205-539-5513
FAX 205-536-8622

FLORIDA
Clearwater, 813-538-0681
FAX 813-538-8379

MINNESOTA
St. Louis Park, 612-545-1490
FAX 612-545-3489

NEW YORK
E. Rochester, 716-586-0670
FAX 716-586-0672
Fishkill, 914-897-8574
FAX 914-897-2363

OHIO
Cleveland, 216-591-1160
FAX 216-292-1081

OREGON
Portland, 503-244-9882
FAX 503-245-0375

TEXAS
Austin, 512-343-8191
FAX 512-343-2759

INTERNATIONAL

JAPAN
Tokyo, 03-3262-0850
FAX 03-3262-0881

KOREA
Seoul, 82-2-563-8232
FAX 82-2-563-8233

SINGAPORE
Singapore, 65-742-2314
FAX 65-741-1979

VLSI AUTHORIZED DESIGN CENTERS

UNITED STATES

COLORADO
SIS MICROELECTRONICS, INC.
Longmont, 303-776-1667

ILLINOIS
ASIC DESIGNS
Naperville, 708-717-5841

MAINE
QUADIC SYSTEMS, INC.
South Portland, 207-871-8244

PENNSYLVANIA
INTEGRATED CIRCUIT SYSTEMS, INC.
Valley Forge, 215-666-1900

INTERNATIONAL

EIRE AND U.K.
PA TECHNOLOGY
Herts, 76-261222

SYMBIONICS
Cambridge, 223-421025

FRANCE
CETIA
Toulon Cedex, 9-42-12005

SOREP
Chateaubourg, 99-623955

INDIA
ARCUS TECHNOLOGY LTD.
Bangalore, 91-812-217307

JAPAN
ADC CORPORATION
Tokyo, 03-3492-1251

LSI SYSTEMS, INC.
Kanagawa, 0462-29-3220

NIPPON STEEL CORPORATION
Tokyo, 03-5566-2141

TOMEN ELECTRONICS
Tokyo, 03-3506-3650

PALTEK CORPORATION
Tokyo, 03-3707-5455

KOREA
ANAM VLSI DESIGN CENTER
Seoul, 82-2-553-2106

NORWAY
NORKRETS AS
Oslo, 47-2360677/8

VLSI SALES REPRESENTATIVES

UNITED STATES

ARIZONA
LUSCOMBE ENGINEERING
Scottsdale, 602-949-9333

COLORADO
LUSCOMBE ENGINEERING
Longmont, 303-772-3342

IOWA
SELTEC SALES
Cedar Rapids, 319-364-7660

KANSAS
ELECTRI-REP
Overland Park, 913-649-2168

MISSOURI
ELECTRI-REP
St. Louis, 314-993-4421

NEW YORK
ADVANCED COMPONENTS
Syracuse, 315-437-6700

OREGON
ELECTRA TECHNICAL SALES
Beaverton, 503-643-5074

UTAH
LUSCOMBE ENGINEERING
Salt Lake City, 801-565-9885

WASHINGTON
ELECTRA TECHNICAL SALES
Kirkland, 206-821-7442

INTERNATIONAL

AUSTRALIA
GEORGE BROWN GROUP
Adelaide, 61-8-352-2222
Brisbane, 61-7-252-3876
Melbourne, 61-3-878-8111
Newcastle, 61-49-69-6399
Perth, 61-9-362-1044
Sydney, 61-2-638-1888

AUSTRIA
THOMAS NEUROTH
Wien, 0043-222-82 56 45

BRAZIL
UNIAO DIGITAL COMERCIAL
Sao Paulo, 55-11 533-0967

CANADA
INTELATECH, INC.
Mississauga, 416-629-0082

HONG KONG
LESTINA INTERNATIONAL, LTD
Tsimshatsui, 852-7351736

ISRAEL
RDT ELECTRONICS ENG. LTD
Tel-Aviv, 972-3-5483737

SINGAPORE
DYNAMIC SYSTEMS PTE, LTD
Singapore, 65-742-1986

TAIWAN
PRINCETON TECH CORP.
Taipei, 886-2-917-8856

WEIKENG INDUSTRIAL CO.
Taipei, 886-2-776-3998

THAILAND
TRON ELECTRONICS CO. LTD
Bangkok, 66 2 260-3913

VLSI DISTRIBUTORS

UNITED STATES

(represented by **Arrow/Schweber** except where noted)

ALABAMA
Huntsville, 205-837-6955

ARIZONA
Phoenix, 602-431-0030

CALIFORNIA
Los Angeles, 818-880-9686
Orange County, 714-838-5422
San Diego, 619-565-4800
San Jose, 408-441-9700

COLORADO
Denver, 303-799-0258

CONNECTICUT
Wallingford, 203-265-7741

FLORIDA
Deerfield Beach, 305-429-8200
Lake Mary, 407-333-9300

GEORGIA
Atlanta, 404-497-1300

ILLINOIS
Chicago, 708-250-0500

INDIANA
Indianapolis, 317-299-2071

IOWA
Cedar Rapids, 319-395-7230

KANSAS
Kansas City, 913-541-9542

MARYLAND
Baltimore, 301-596-7800

MASSACHUSETTS
Boston, 508-658-0900

MICHIGAN
Detroit, 313-462-2290

MINNESOTA
Minnesota, 612-941-5280

MISSOURI
St. Louis, 314-567-6888

NEW JERSEY
Philadelphia, 609-596-8000
Pine Brook, 201-227-7880

NEW YORK
Happuage, 516-231-1000
Rochester, 716-427-0300

NORTH CAROLINA
Raleigh, 919-876-3132

OHIO
Dayton, 513-435-5563
Cleveland, 216-248-3990

OKLAHOMA
Tulsa, 918-252-7537

OREGON
ALMAC/ARROW
Portland, 503-629-8090

PENNSYLVANIA
Pittsburgh, 412-963-6807

TEXAS
Austin, 512-835-4180
Dallas, 214-380-6464
Houston, 713-530-4700

UTAH
Salt Lake City, 801-973-6913

WASHINGTON
ALMAC/ARROW
Seattle, 206-643-9992

WISCONSIN
Milwaukee, 414-792-0150

INTERNATIONAL

BELGIUM/LUXEMBURG
MICROTRON
Mechelen, 215-212223

CANADA
ARROW/SCHWEBER
Montreal, 514-421-7411
Ottawa, 613-226-6903
Toronto, 416-670-7769
Vancouver, 604-421-2333

SEMAD
British Columbia, 604-420-9889
Calgary, 403-252-5664
Markham, 416-475-8500
Montreal, 514-694-0860
Ottawa, 613-727-8325

DENMARK
PRESCOM A/S
Lynge, 45 42 18 98 00

ENGLAND
HAWKE COMPONENTS
Bramley, NR Basingstoke
256-880800

KUDOS-THAME LTD
Berks, 734-351010

FINLAND
COMDAX
Helsinki, 80-670277

FRANCE
ASAP s.a.
Montigny-le-Bretonneux,
1-30438233

GERMANY
DATA MODUL GmbH
Muenchen, 089-560170

ELECTRONIC 2000 AG
Muenchen, 089-420010

EAST GERMANY
ZENTRUM MIKROELEKTRONIK
Dresden, 0037-51-588464

ITALY
INTER-REP S.P.A.
Torino, 11-2165901

JAPAN
ASAHI GLASS CO. LTD
Tokyo, 03-3218-5854

TOKYO ELECTRON, LTD
Tokyo, 03-3340-8111

NETHERLANDS
TME
Aa Heeswijk-Dinther,
4139-8895

PUERTO RICO
ISLA CARIBE ELECTRO SALES
Guaynabo, 809-720-4430

SWEDEN
TRACO AB
Farsta, 8-930000

SPAIN AND PORTUGAL
SEMICONDUCTORES s.a.
Barcelona, 3-21723 40

SWITZERLAND
DECTRO SWISS ELECTRONIC DESIGN AG
Zurich, 0041-1-3868-600

1/93

130900-001 5M 1/93.