SGS-THOMSON MICROELECTRONICS

FUZZYSTUDIO™ W.A.R.P.-ADB 1.0

APPLICATION DEVELOPMENT BOARD

ADVANCED DATA

Key Features:

- Up to 16 input variables
- Up to 16 output variables
- Up to 256 rules (4 antecedents, 1consequent)
- W.A.R.P. programmer on board
- EPROM programmer on board
- RS232-C standard communication
- Directly connection of W.A.R.P. with control system
- Internal clock

Introduction

The Application Development Board (ADB) is a powerful tool to develop applications based on the Weight Associative Rule Processor (W.A.R.P.) and to test physical implementations of fuzzy control.

W.A.R.P.-ADB is a tool which is part of the FUZZYSTUDIO™ Basic Tool Kit.

Through the W.A.R.P.-System Development Tool (W.A.R.P.-SDT) it is then possible to load the defined rules and membership functions into the W.A.R.P. memories.

As a matter of fact, a fuzzy project can be stored on the Weight Associative Rule Processor located on the board.

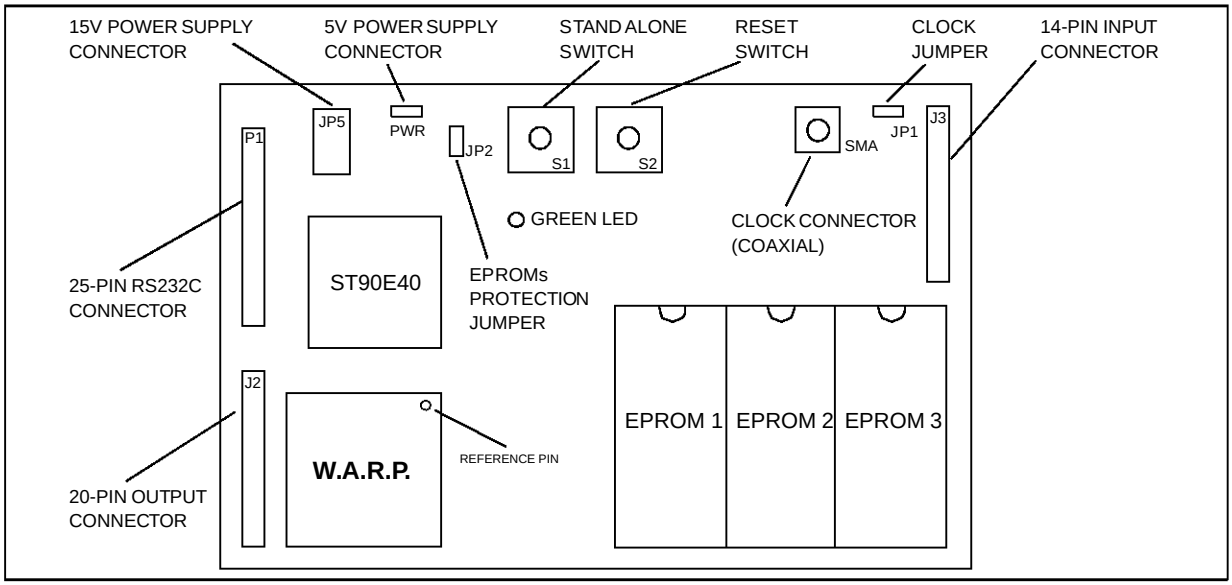
The ADB can be connected to the RS232-C port of an IBM PC 386 (or higher) and can also work stand alone.

When ADB is working connected with a PC through serial communication it is possible to control by means of the PC the following operations:

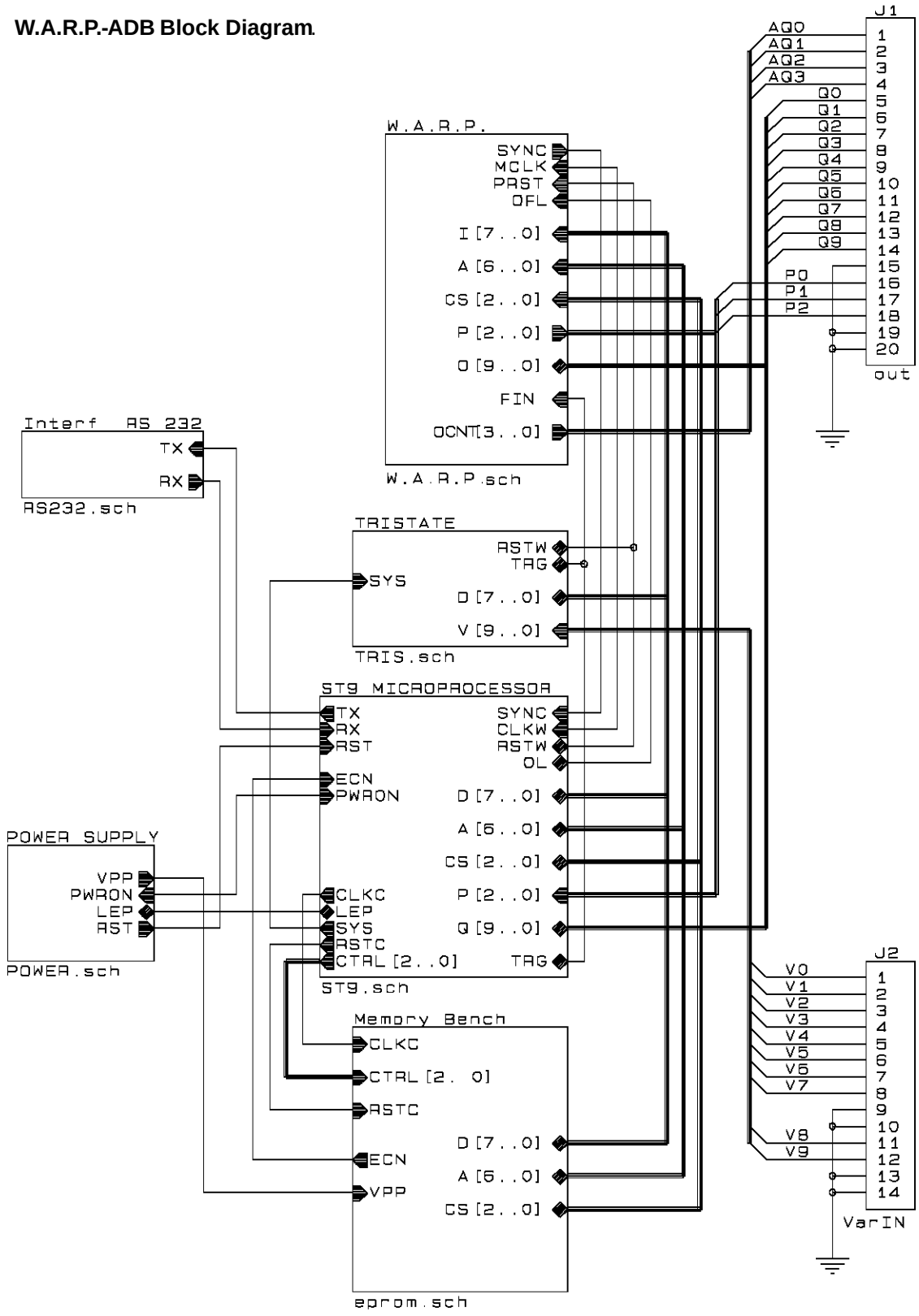
- Data Loading from PC to EPROM.
- Data Loading from PC to W.A.R.P..
- Data Loading from EPROM to W.A.R.P..
- Monitoring of Inputs and Outputs data of W.A.R.P. by PC.
- Switch to/from stand alone mode.

When ADB is working stand alone (direct control between the board and the controlled system), it is possible to utilize the W.A.R.P. signals directly from the Input/Output connectors of the board. In this case the user has to manage Inputs and Outputs because the PC is disconnected from the board.

W.A.R.P.-ADB Description.



W.A.R.P.-ADB Block Diagram.



Board Functionalities

The W.A.R.P.-ADB can provide the following functionalities:

- Direct Loading of a project generated by W.A.R.P.-SDT Compiler on W.A.R.P. memories, while remaining in W.A.R.P.-SDT environment.
- Possibility to program the EPROMs of the board from the development environment in order to insert the board in the user's control system.
- Testing of input patterns sent to W.A.R.P.. It computes the correspondent outputs while sending them back to the development environment, so allowing a real-time visualization of the outputs.
- Capability to test single patterns or pattern streams coming from files defined by the user.
- Disposability of the outputs on the external connector to test in the meantime the user's application. In this case the clock is scanned by the times of the entire project management.
- Possibility to commute the board in stand alone mode deconnecting it from the interactive functionalities with the environment, in order to allow the user a faster testing of his application. After testing his application, the user can return back to the control by the PC.
- Possibility to know the name, date and time of last updating of the project stored in the EPROMs.
- Possibility to use an external clock to syncronize the fuzzy computation to the system speed.

Board Description

The W.A.R.P.-ADB is composed by the following functional blocks.

- The fuzzy processor W.A.R.P. performing the fuzzy computation either from the inputs coming from the PC or from the user application, according to the fuzzy program loaded on W.A.R.P..
- The ST9 microcontroller managing the connection with the PC by means of RS232 interface.
- The block of EPROMs to store the database to be loaded into W.A.R.P. when the board works stand alone and the counter performs their addressment.
- The power supply block able to give 5V for the running of all the components of the board and 12.5 V for the programming of EPROMs.
- The connection block with the external application, including connectors and input three-state buffers.
- The busy led indicating when the board is working.

SGS-THOMSON MICROELECTRONICS

FUZZYSTUDIO™ W.A.R.P.-SDT 1.0

SOFTWARE DEVELOPMENT TOOL

ADVANCED DATA

W.A.R.P. Software Development Tool is a graphic environment which allows users to build projects targeted to W.A.R.P. processor, using the Fuzzy Logic terminology, in a user friendly way.

W.A.R.P.-SDT is a tool which is part of the FUZZYSTUDIO™ Basic Tool Kit.

This tool allows to program the Weight Associative Rule Processor (W.A.R.P.) fuzzy microcontroller: in accordance with the controlling specs defined by the user it generates the program to be loaded on W.A.R.P. memories.

Input and output Variables, Membership Functions and Fuzzy Rules could be graphically defined by means of linguistic terms.

The W.A.R.P. Software Development Tool guides the designer in his project definition and prevents possible mistakes generated by this new way of programming.

The diagram below shows the basic blocks of W.A.R.P.-SDT

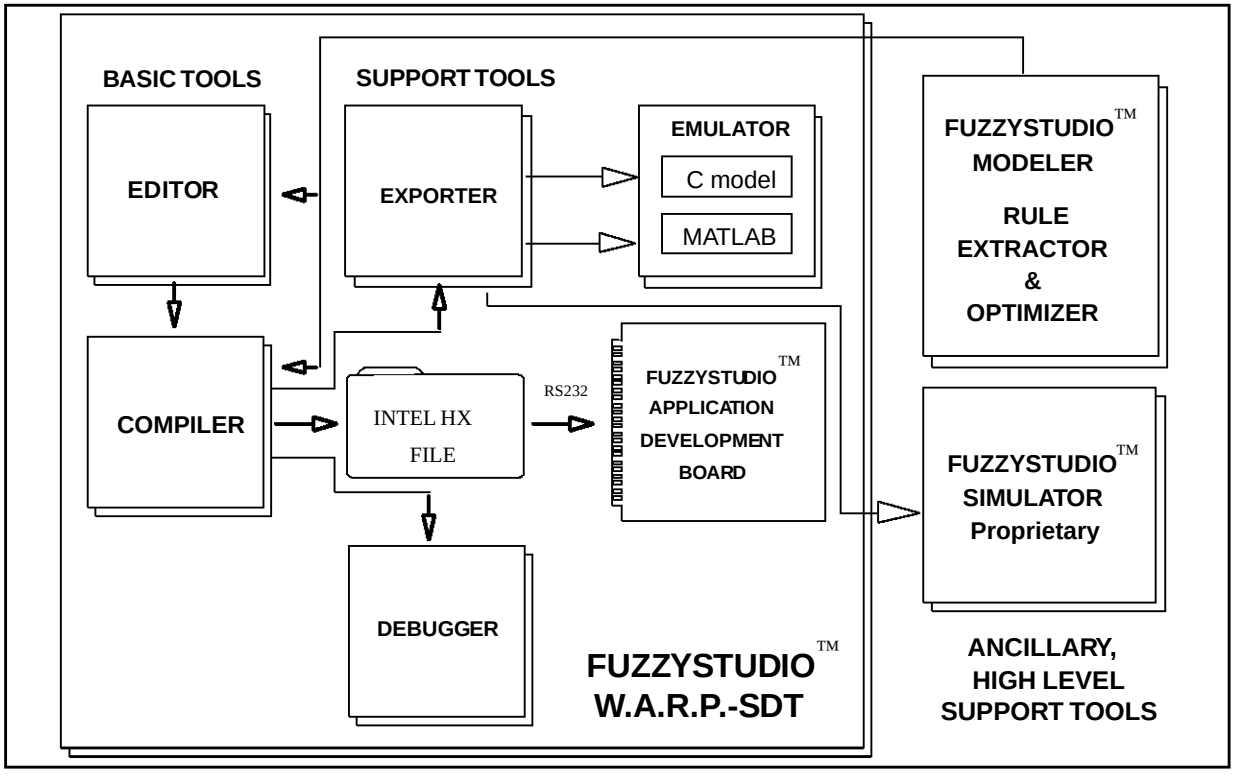
The main parts of W.A.R.P.-SDT are:

- Editor (Variable and Rule)
- Compiler
- Debugger
- Exporter
- W.A.R.P.-ADB management Tools

System Requirements

- PC IBM compatible with Intel type 80386 (or higher) processor
- 2 Mbytes RAM memory minimum (4 Mbytes suggested)
- Hard Disk with at least 2 Mbytes of free space
- VGA or higher graphics card
- Mouse
- Ms-DOS version 3.3 or higher
- Window version 3.X

Figure 1. Block Diagram



Editor

The Editor graphic environment guides the user in all the steps necessary to define the variables and the rules of the fuzzy controller.

It is composed by:

Variable Editor:

Variables are defined providing the boundaries of the Universe of Discourse, the resolution (configurable up to 7 bit for the input, fixed to 10 bit for the output) and the Membership Functions for each input and output variable.

Moreover, the Variable Editor allows a symbolic name to be specified in order to recognize a variable.

The Membership Functions associated to that variable can be defined by calling the Membership Functions Editor from the Variable Editor. It allows the user to define graphically the shapes of the Membership Functions by using the drawing tool and the mouse.

Rule Editor

The Rule Editor allows the fuzzy rules to be simply defined by using the mouse. All functionalities and keywords not allowed by the standard format of the rule are automatically disabled in order to avoid possible mistakes and guide the user in the definition of the rule.

A rule has a valid format when it has at least one antecedent and one consequent but no more than four antecedents and eight consequents .

In order to take in account rules with a greater number of antecedents and consequents the user has to split the chosen rules in many equivalent ones having standard format, meanwhile considering the -AND- and -OR- connectives.

Figure 2. VAR EDITOR window

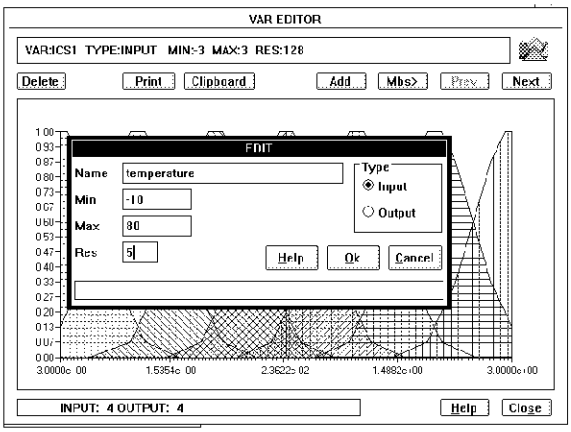


Figure 3. MEMBERSHIP EDITOR window

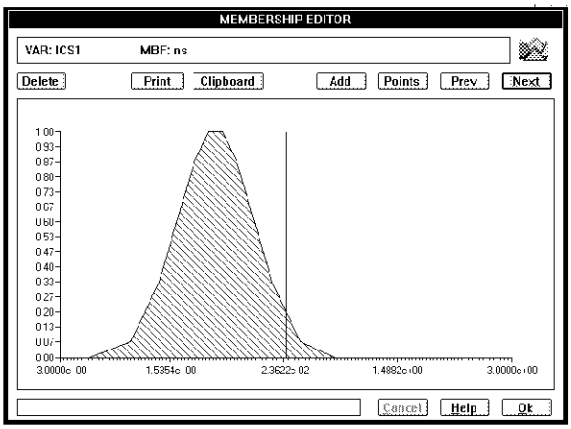
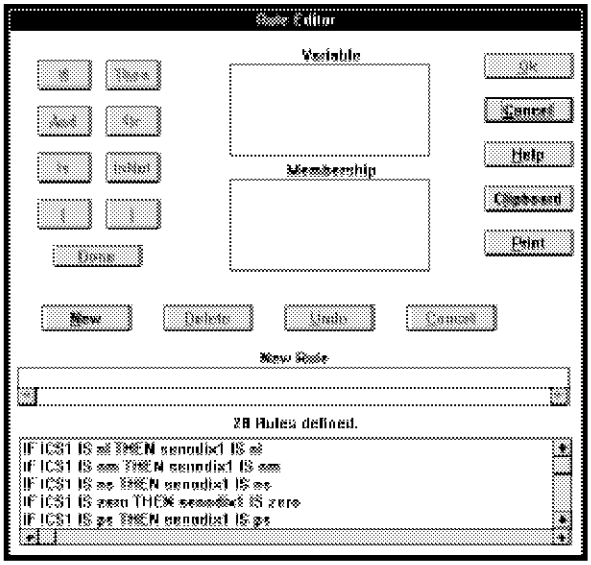


Figure 4. RULE EDITOR window



Compiler

The Compiler translates the data originated from the Editor into a data stream to be loaded in W.A.R.P. memories, so generating the machine code.

It accepts as inputs only those files containing a description, in Fuzzy Logic Language of all information regarding variables and rules.

It can also be used in DOS environment providing it an input a file containing the description of variables and rules in Fuzzy Logic Language (FLL).

The code provided by Compiler depends on W.A.R.P. memories configuration, which is determined by the type and number of input variable. W.A.R.P. memories configuration is automatically computed and optimized by the Compiler.

The compiler provides also a JEDEC format file which can be loaded into a standard EPROM programmer.

Debugger

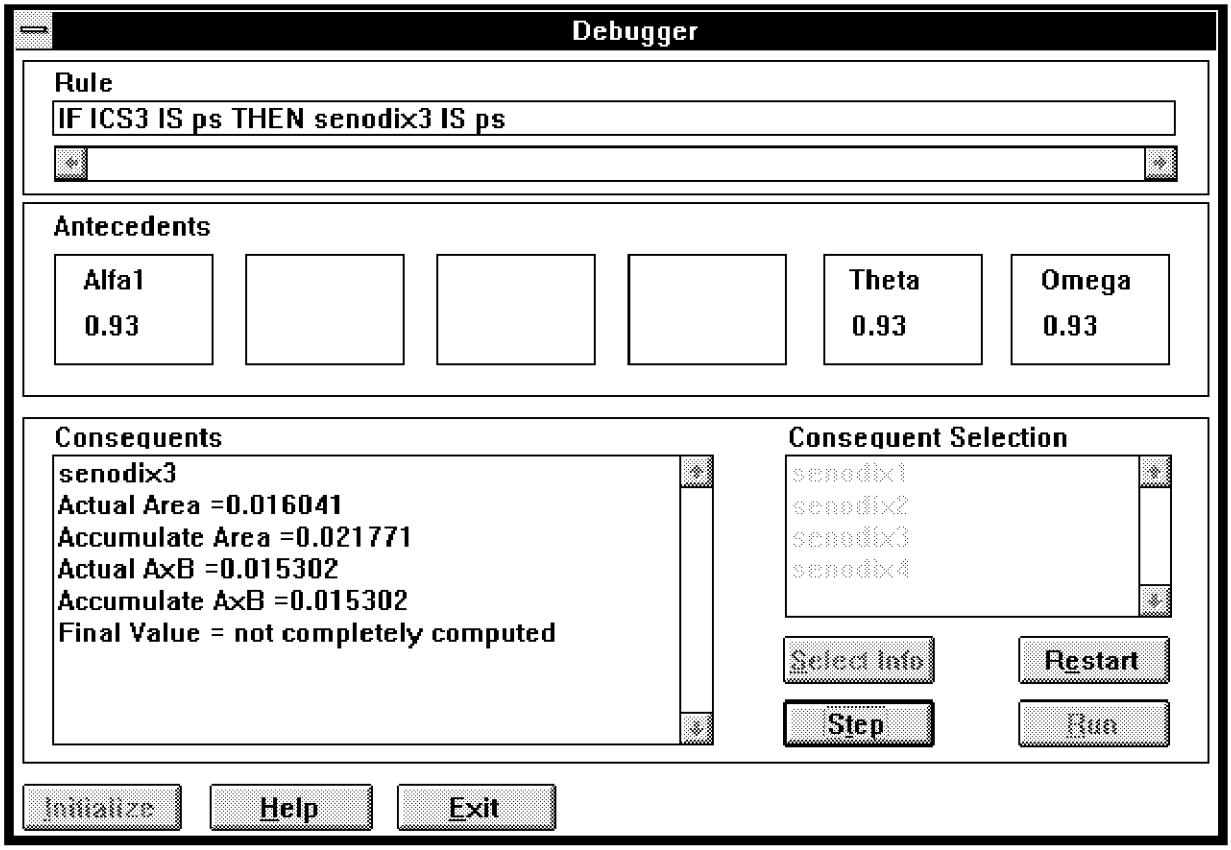
The Debugger tool provides W.A.R.P. emulation and gives the user all information he needs to perform this task.

The Debugger can either directly provide the output values or execute the computation rule by rule while monitoring the fuzzy computing values. The step-by-step analysis can be performed monitoring the only rules related to any selected outputs.

To perform an intensive testing, it is possible to execute a W.A.R.P. emulation by taking the input patterns either from a user-defined file or from any automatic tool able to generate data in a compatible format.

The output variables provided as result are visualized on the screen and stored into a file.

Figure 5. DEBUGGER window



Exporter

Exporter allows to translate a W.A.R.P.-SDT project into a format that can be imported in other simulation environments.

It can be used in C language, in the technical computing environment MATLAB, or directly treated by W.A.R.P.-SIMinternal tool, to make a closed-loop simulation and a complete control system synthesis.

The exporter provides library functions working on databases automatically generated by the compiler and used in order to initialize the structures to be used in library functions.

The above mentioned functions can be exploited inside programs developed by the user in order to perform simulation of a Weight Associative Rule Processor control system as many times as it is required. Many different databases can be used in order to simulate a control system with many fuzzy controllers.

Therefore, the user has the only task to describe the model of the controlled system, according to the fact that the controller model is automatically performed by the Exporter.

MATLAB Exporter

The MATLAB exporter allows a simulation of an entire control system in MATLAB or SIMULINK environment to be executed.

The exporter provides some library function which work on the database automatically created by W.A.R.P.-SDT Compiler. This data is used to initialize the library functions structures. These functions can be exploited inside programs developed by the user in order to perform system simulations as many times as he wants.

C Language Exporter

The usage of C MODEL EXPORTER is quite similar to the usage of ”MATLAB” exporter also with respect to library functions. C Language Exporter provides source library functions.

W.A.R.P.-SIM Exporter (Proprietary)

WARP-SIM EXPORTER produces a dabase including information to emulate W.A.R.P. in the W.A.R.P.-SIM program.

W.A.R.P.-SIM program is a tool to make a closes loop simulation and a complete control system synthesis.

W.A.R.P.-ADB management tools

The W.A.R.P.-ADB management tools allows the rules and the membership functions designed by means of W.A.R.P.-SDT to be loaded into the W.A.R.P. memories.

The W.A.R.P.-ADB can be connected to the RS232-C port of an IBM PC 386 (or higher) and can also work stand alone.

When W.A.R.P.-ADB is working connected with a PC through a serial communication it is possible to control by means of the PC the following operations:

- Data Loading from PC to W.A.R.P.
- Data Loading from PC to EPROMs.
- Data Loading from EPROMs to W.A.R.P.
- Monitoring of Inputs and Outputs data of W.A.R.P. at a reduced clock rate by PC.

Figure 6. W.A.R.P.-ADB block diagram .