Advanced
Communication
Devices

Data Sheet: ACD80800

Address Resolution Logic
(8K MAC Addresses)

Rev.1.0.0.E
Last Update: September 19, 2000

Please check ACD's website for update information before starting a design
Web site: http://www.acdcorp.com/tech.html

or Contact ACD at:
Email: support@acdcorp.com
Tel: 510-354-6810
Fax: 510-354-6834

CONTENT LIST

1. SUMMARY

The ACD80800 serves as an Address Resolution Logic for ACD's switch controller chips (ACD82124, ACD82012 etc.) through a glueless ARL interface. It automatically builds up an address table and can map up to 8K MAC addresses into their associated ports.

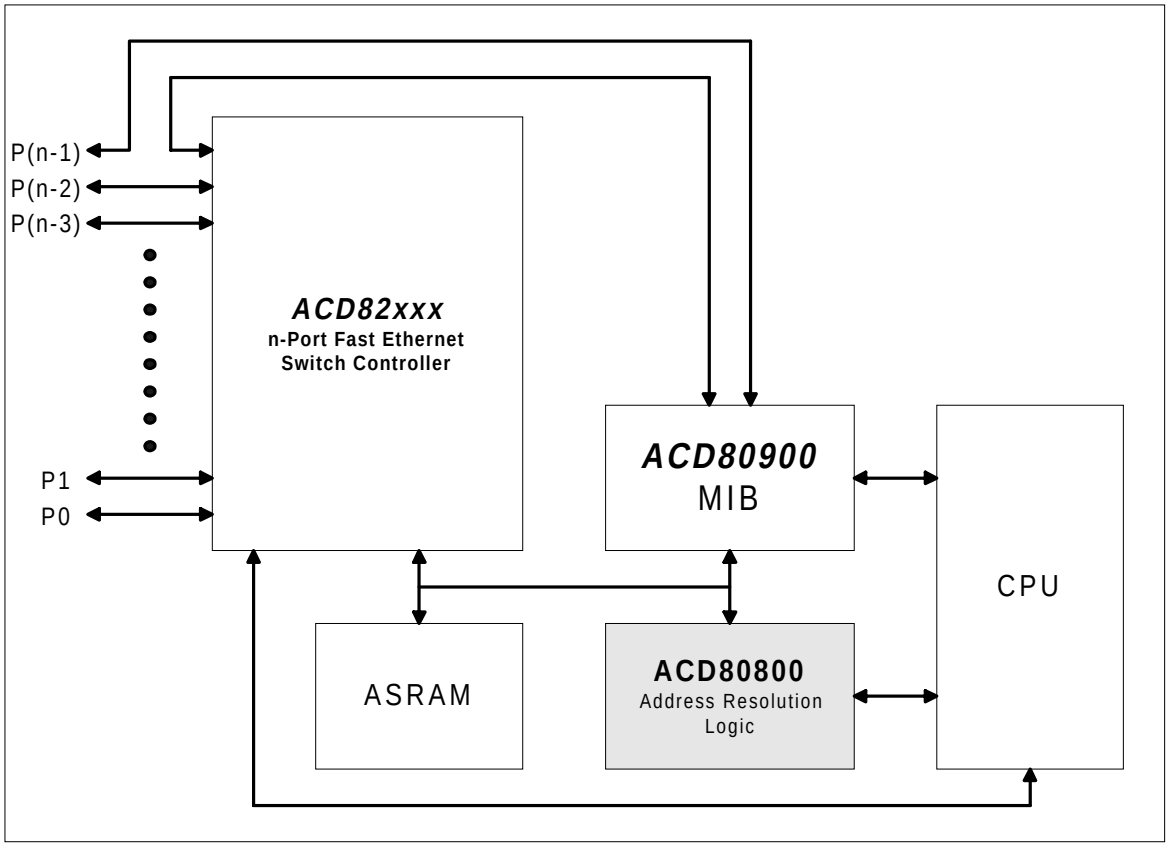
The ACD80800 can work without a CPU in a unmanaged switch system, or with a CPU and an ACD MIB (ACD80900 Management Information Base). A direct input/output interface is integrated to support a management CPU. The CPU can configure the operation mode of the ACD80800, learn all the addresses in the address table, add new addresses into the table, control security or filtering features of each address entry, etc.

The ACD80800 is designed with such a high performance that, it will never slow down the frame switching operation conducted by the ACD's switch controllers. Together with the non-blocking architecture of the ACD's switch controllers, the chip set (a ACD switch controller plus the ACD80800, plus ACD80900 in a managed switch system) can provide wire speed forwarding rate under any type of traffic load.

2. FEATURES

- Supports up to 8K MAC address lookup
- Provides Glueless ARL Interface with ACD's switch controller chip
- Provides Direct Input/Output type of interface for the management CPU
- Provides UART type of interface for the management CPU
- Wire speed address lookup time.
- Wire speed address learning time.
- Address can be automatically learned from switch without THE CPU intervention
- Address can be manually added by the CPU through the CPU interface
- Each MAC address can be secured by the CPU from being changed or aged out
- Each MAC address can be marked by the CPU from receiving any frame
- Each newly learned MAC address is notified to the CPU
- Each aged out MAC address is notified to the CPU
- Automatic address aging control, with configurable aging period
- 0.35 micron, 3.3V CMOS technology
- 128-Pin PQFP package

Figure-1: ACD80800 Used in A Managed n-Port Fast Ethernet Switch System



3. FUNCTIONAL DESCRIPTION

ACD80800 provides Address Resolution service for ACD's switch controller chip. ACD80800 provides a glueless interface with ACD's switch controller, and is used to build an address table and provide address lookup service to ACD's switch controller. *Figure 2* is a block diagram of ACD80800.

Traffic Snooping

All Ethernet frames received by ACD's switch controller have to be stored into memory buffer. As the frame data are written into memory, the status of the data shown on the data bus are displayed by ACD's switch controller through a state bus. ACD80800's Switch Controller Interface contains the signals of the data bus and the state bus. By snooping the data bus and the state bus of ACD's switch controller, ACD80800 can detect the occurrence of any destination MAC address and source MAC address embedded inside each frame.

Address Learning

Each source address caught from the data bus, together with the ID of the ingress port, is passed to the Address Learning Engine of ACD80800. The Address Learning Engine will first determine whether the frame is a valid frame. For a valid frame, it will first try to find the source address from the current address table. If that address doesn't exist, or if it does exist but the port ID associated with the MAC address is not the ingress port, the address will be learned into the address table. After an address is learned by the address learning engine, the CPU will be notified to read this newly learned address so that it can add it into the CPU's address table.

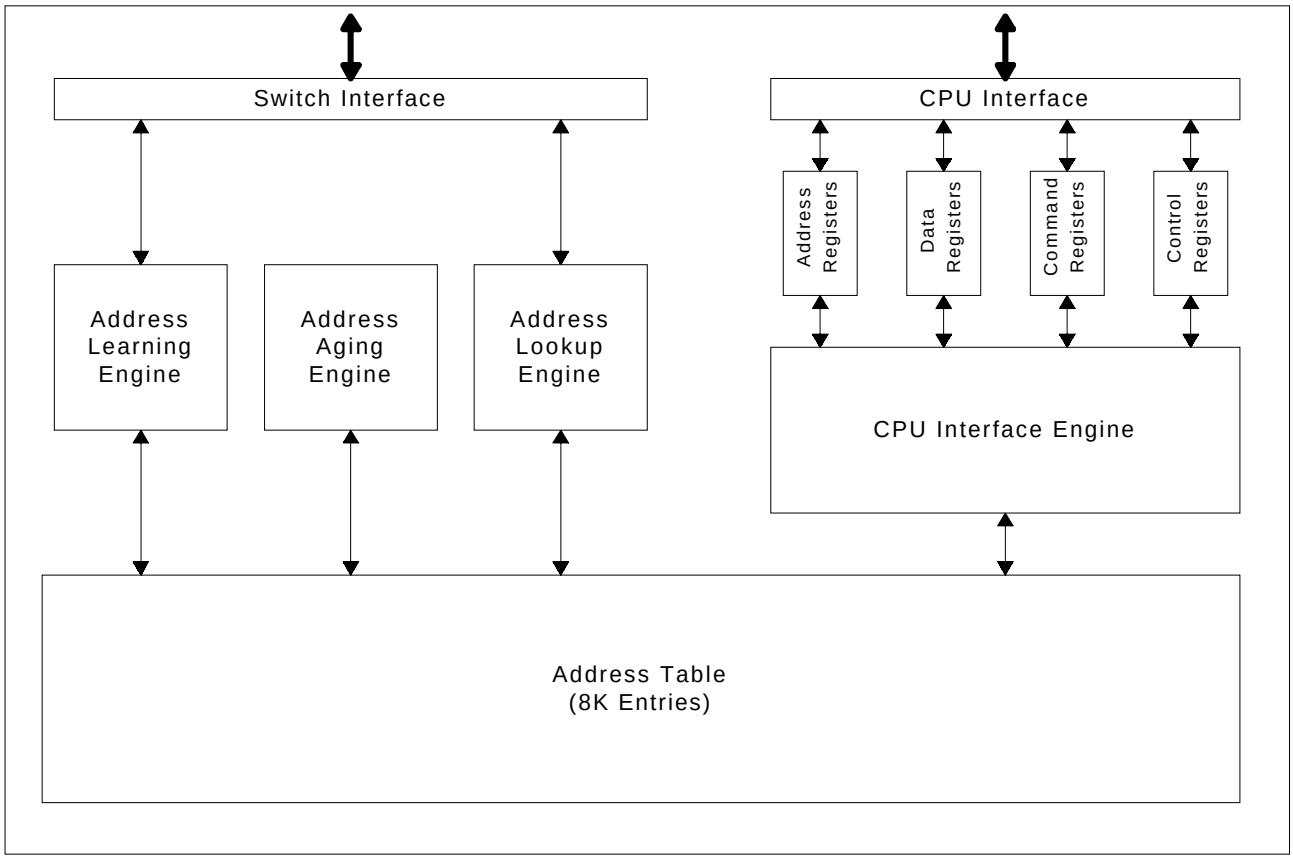
Address Aging

After each source address is learned into the address table, it has to be refreshed at least once within each address aging period. Refresh means it is caught again from the switch interface. If it has not occurred for a pre-set aging period, the address aging engine will remove the address from the address table. After an address is removed by the address aging engine, the CPU will be notified through interrupt request that it needs to read this aged out address so that it can remove this address from the CPU's address table.

Address Lookup

Each destination address is passed to the Address

Figure-2. ACD80800 Block Diagram



Lookup Engine of ACD80800. The Address Lookup Engine checks if the destination address matches with any existing address in the address table. If it does, ACD80800 returns the associated Port ID to ACD's switch controller through the output data bus. Otherwise, a no match result is passed to ACD's switch controller through the output data bus.

CPU Interface

ACD80800 provides a direct input/output type of interface for a management CPU to access various kind of registers inside ACD80800. The interface has 8-bit data bus, and 5-bit address bus. The timing of read and write operation is controlled by output enable signal and write enable signal. For details of CPU interface timing information, refer to the section of "Timing Description."

The CPU can also choose to access the registers of ACD80800 by sending commands to the UART data input line. Each command is consisted by action (read or write), register type, register index, and data. Each result of command execution is returned to the CPU through the UART data output line.

CPU Interface Registers

ACD80800 provides a bunch of registers for the control CPU. Through the registers, the CPU can read all address entries of the address table, delete particular addresses from the table, add particular addresses into the table, secure an address from being changed, set filtering on some addresses, change the hashing algorithm etc. Through a proper interrupt request signal, the CPU can be notified whenever it needs to retrieve data for a newly-learned address or an aged-out address so that the CPU can build an exact same address table learned by ACD80800.

CPU Interface Engine

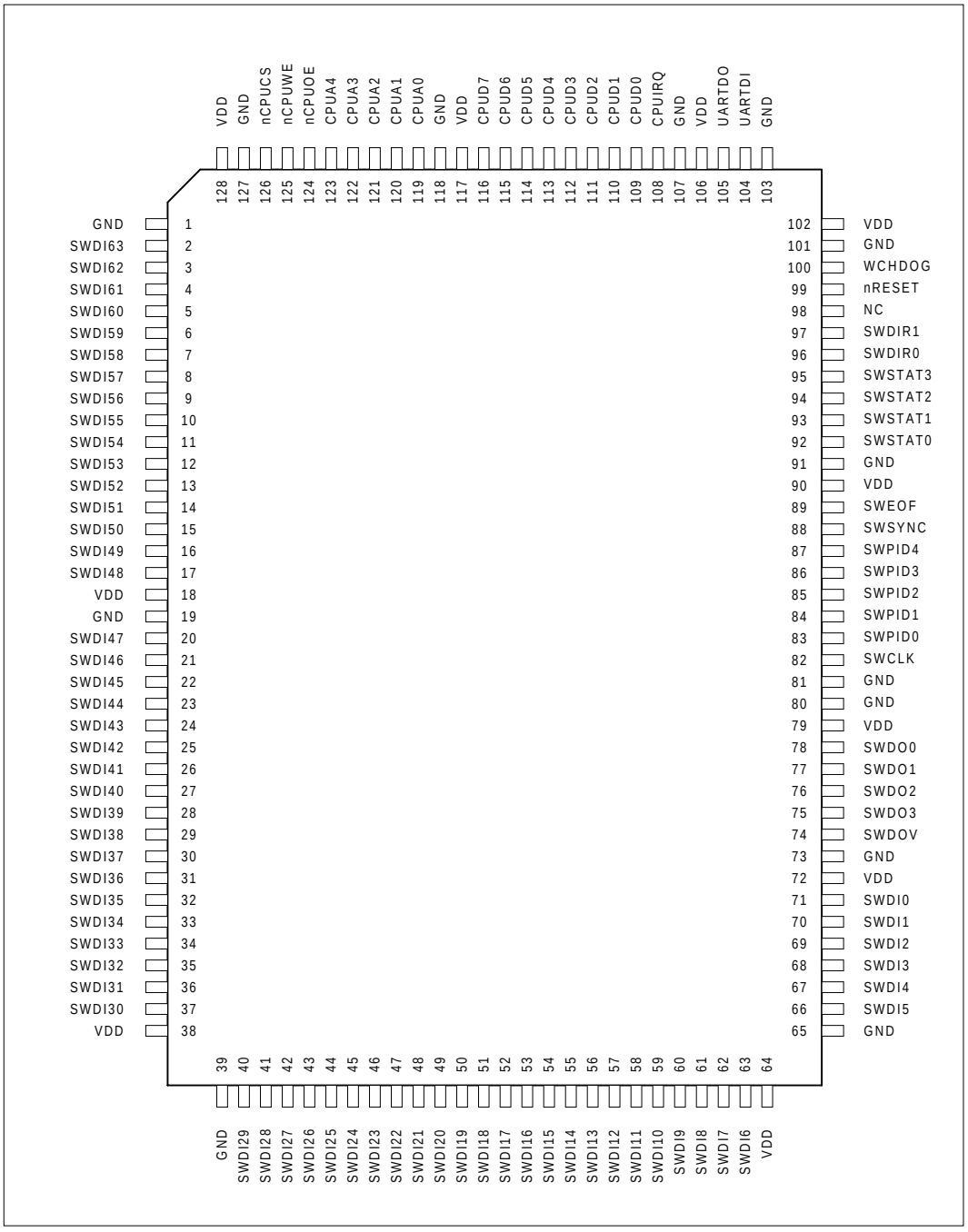
The command sent by the control CPU is executed by the CPU Interface Engine. For example, the CPU may send a command to learn the first newly-learned address. The CPU Interface Engine is responsible to find the newly-learned address from the address table, and passes it to CPU. The CPU may request to learn next newly-learned address. Then, it is again the responsibility of the CPU Interface Engine to search for next newly-learned address from the address table.

Address Table

The address table can hold up to 8K MAC addresses, together with the associated port ID, security flag, filtering flag, new flag, aging information etc. The address table resides in the embedded SRAM inside ACD80800.

4. PIN DESCRIPTIONS

Figure-3: Pin Diagram Of ACD80800 (The ARL Chip)



Pin Table

Pin	Name	Description	I/O	Pin	Name	Description	I/O
1	GND	Ground.	-	65	GND	Ground.	-
2	SWDI63	Data from switch controller chip. 3.3V	I	66	SWDI5	Data from switch controller chip. 3.3V	I
3	SWDI62	Data from switch controller chip. 3.3V	I	67	SWDI4	Data from switch controller chip. 3.3V	I
4	SWDI61	Data from switch controller chip. 3.3V	I	68	SWDI3	Data from switch controller chip. 3.3V	I
5	SWDI60	Data from switch controller chip. 3.3V	I	69	SWDI2	Data from switch controller chip. 3.3V	I
6	SWDI59	Data from switch controller chip. 3.3V	I	70	SWDI1	Data from switch controller chip. 3.3V	I
7	SWDI58	Data from switch controller chip. 3.3V	I	71	SWDI0	Data from switch controller chip. 3.3V	I
8	SWDI57	Data from switch controller chip. 3.3V	I	72	VDD	3.3V power supply.	-
9	SWDI56	Data from switch controller chip. 3.3V	I	73	GND	Ground.	-
10	SWDI55	Data from switch controller chip. 3.3V	I	74	SWDOV	Output data valid signal to switch controller chip. 3.3V	O
11	SWDI54	Data from switch controller chip. 3.3V	I	75	SWDO3	Output data to switch controller chip. 3.3V	O
12	SWDI53	Data from switch controller chip. 3.3V	I	76	SWDO2	Output data to switch controller chip. 3.3V	O
13	SWDI52	Data from switch controller chip. 3.3V	I	77	SWDO1	Output data to switch controller chip. 3.3V	O
14	SWDI51	Data from switch controller chip. 3.3V	I	78	SWDO0	Output data to switch controller chip. 3.3V	O
15	SWDI50	Data from switch controller chip. 3.3V	I	79	VDD	3.3V power supply. 3.3V	-
16	SWDI49	Data from switch controller chip. 3.3V	I	80	GND	Ground.	-
17	SWDI48	Data from switch controller chip. 3.3V	I	81	GND	Ground.	-
18	VDD	3.3V power supply.	-	82	SWCLK	50MHz reference clock signal from switch controller chip. 3.3V	I
19	GND	Ground.	-	83	SWPID0	Port ID indication signal from switch controller chip. 3.3V	I
20	SWDI47	Data from switch controller chip. 3.3V	I	84	SWPID1	Port ID indication signal from switch controller chip. 3.3V	I
21	SWDI46	Data from switch controller chip. 3.3V	I	85	SWPID2	Port ID indication signal from switch controller chip. 3.3V	I
22	SWDI45	Data from switch controller chip. 3.3V	I	86	SWPID3	Port ID indication signal from switch controller chip. 3.3V	I
23	SWDI44	Data from switch controller chip. 3.3V	I	87	SWPID4	Port ID indication signal from switch controller chip. 3.3V	I
24	SWDI43	Data from switch controller chip. 3.3V	I	88	SWSYNC	Port 0 indication signal from switch controller chip. 3.3V	I
25	SWDI42	Data from switch controller chip. 3.3V	I	89	SWEOF	End Of Frame indication signal from switch controller chip. 3.3V	I
26	SWDI41	Data from switch controller chip. 3.3V	I	90	VDD	3.3V power supply.	-
27	SWDI40	Data from switch controller chip. 3.3V	I	91	GND	Ground.	-
28	SWDI39	Data from switch controller chip. 3.3V	I	92	SWSTAT0	Data Status signal from switch controller chip. 3.3V	I
29	SWDI38	Data from switch controller chip. 3.3V	I	93	SWSTAT1	Data Status signal from switch controller chip. 3.3V	I
30	SWDI37	Data from switch controller chip. 3.3V	I	94	SWSTAT2	Data Status signal from switch controller chip. 3.3V	I
31	SWDI36	Data from switch controller chip. 3.3V	I	95	SWSTAT3	Data Status signal from switch controller chip. 3.3V	I
32	SWDI35	Data from switch controller chip. 3.3V	I	96	SWDIR0	Data direction indication signal from switch controller chip. 3.3V	I
33	SWDI34	Data from switch controller chip. 3.3V	I	97	SWDIR1	Data direction indication signal from switch controller chip. 3.3V	I
34	SWDI33	Data from switch controller chip. 3.3V	I	98	NC	Not connected.	-
35	SWDI32	Data from switch controller chip. 3.3V	I	99	nRESET	Hardware reset pin. 3.3V	I
36	SWDI31	Data from switch controller chip. 3.3V	I	100	WCHDOG	Watch dog signal. 3.3V	O
37	SWDI30	Data from switch controller chip. 3.3V	I	101	GND	Ground.	-
38	VDD	3.3V power supply.	-	102	VDD	3.3V power supply.	-
39	GND	Ground.	-	103	GND	Ground.	-
40	SWDI29	Data from switch controller chip. 3.3V	I	104	UARTDI	UART data input line. 3.3V	I
41	SWDI28	Data from switch controller chip. 3.3V	I	105	UARTDO	UART data output line. 3.3V	O
42	SWDI27	Data from switch controller chip. 3.3V	I	106	VDD	3.3V power supply.	-
43	SWDI26	Data from switch controller chip. 3.3V	I	107	GND	Ground.	-
44	SWDI25	Data from switch controller chip. 3.3V	I	108	CPUIRQ	Interrupt request. 3.3V	O
45	SWDI24	Data from switch controller chip. 3.3V	I	109	CPUD0	CPU data bus. 3.3V	I/O
46	SWDI23	Data from switch controller chip. 3.3V	I	110	CPUD1	CPU data bus. 3.3V	I/O
47	SWDI22	Data from switch controller chip. 3.3V	I	111	CPUD2	CPU data bus. 3.3V	I/O
48	SWDI21	Data from switch controller chip. 3.3V	I	112	CPUD3	CPU data bus. 3.3V	I/O
49	SWDI20	Data from switch controller chip. 3.3V	I	113	CPUD4	CPU data bus. 3.3V	I/O
50	SWDI19	Data from switch controller chip. 3.3V	I	114	CPUD5	CPU data bus. 3.3V	I/O
51	SWDI18	Data from switch controller chip. 3.3V	I	115	CPUD6	CPU data bus. 3.3V	I/O
52	SWDI17	Data from switch controller chip. 3.3V	I	116	CPUD7	CPU data bus. 3.3V	I/O
53	SWDI16	Data from switch controller chip. 3.3V	I	117	VDD	3.3V power supply.	-
54	SWDI15	Data from switch controller chip. 3.3V	I	118	GND	Ground.	-
55	SWDI14	Data from switch controller chip. 3.3V	I	119	CPUA0	CPU address bus. 3.3V	I
56	SWDI13	Data from switch controller chip. 3.3V	I	120	CPUA1	CPU address bus. 3.3V	I
57	SWDI12	Data from switch controller chip. 3.3V	I	121	CPUA2	CPU address bus. 3.3V	I
58	SWDI11	Data from switch controller chip. 3.3V	I	122	CPUA3	CPU address bus. 3.3V	I
59	SWDI10	Data from switch controller chip. 3.3V	I	123	CPUA4	CPU address bus. 3.3V	I
60	SWDI9	Data from switch controller chip. 3.3V	I	124	nCPUOE	Output Enable signal from CPU. 3.3V	I
61	SWDI8	Data from switch controller chip. 3.3V	I	125	nCPUWE	Write Enable signal from CPU. 3.3V	I
62	SWDI7	Data from switch controller chip. 3.3V	I	126	nCPUCS	Chip Select signal from CPU. 3.3V	I
63	SWDI6	Data from switch controller chip. 3.3V	I	127	GND	Ground.	-
64	VDD	3.3V power supply.	-	128	VDD	3.3V power supply.	-

5. INTERFACE DESCRIPTION

Switch Interface

Switch interface provides a communication channel between ACD's switch controller chip and ACD80800. As a frame is being received by ACD's switch controller chip , the destination address and source address of the frame are snooped from the SWDIx lines of ACD80800, with respect to the SWCLK signal. ACD80800 carries a lookup process for each destination address, and a learning process for each source address. The result of the lookup is returned to the switch controller chip through the SWDOx lines. *Table 1* shows the associated signals in the Switch Interface.

Table-1: Switch Interface

Name	Type	Description
SWDI0 ~ SWDI63	I	Input data, which can be 48-bit or 64-bit wide
SWSTAT0 ~ SWSTAT3	I	Input data state
SWEOF	I	End of frame indication signal
SWDIR0 ~ SWDIR1	I	Data direction indication signal
SWSYNC	I	Port synchronization signal
SWPID0 ~ SWPID4	I	Port-ID indication signal
SWCLK	I	Reference clock
SWDOV	O	Output data valid signal
SWDO0 ~ SWDO3	O	Output data which can be 2-bit or 4-bit wide

The SWDIx signal comes from the SRAM Data bus of ACD's switch controller chip . Since all data of the received frames have to be written into the shared memory through the Data bus, the bus can be monitored for occurrence of DA and SA values, indicated by the associated state bits. The signals in SWDIx bus can be a 48-bit or 64-bit wide data bus. For a 48-bit wide bus, the first word will be the DA and the second word will be the SA. For a 64-bit wide bus, DA is the first 48-bit of first word, SA is the last 16-bit of first word plus first 32-bit of second word.

SWDIR is a 2-bit signal to indicate the direction of the data displayed on the SWDI bus, 01 for receiving, 10 for transmitting, 00 or 11 for other states. ACD80800 only deals with the received data.

SWSTAT bus is a 4-bit signal, used to indicate the meaning (status) of the data. The 4-bit status is defined as:

- 0000 - Third to Last word
- 0001 - First word
- 0010 - Second word
- 0011 - Reserved
- 0100 - Reserved
- 0101 - Drop event
- 0110 - Jabber
- 0111 - False carrier
- 1000 - Alignment error
- 1001 - Flow control/collision*
- 1010 - Short event/excessive collision*
- 1011 - Runt/Late collision*
- 1100 - Symbol error
- 1101 - FCS error
- 1110 - Long event
- 1111 - Reserved

*Note: error type depends on SWDIR is 01 or 10.

SWSYNC is used to indicate port 0 is driving the Data bus. It is used when the bus is evenly allocated in a time division multiplexing manner, such that a monitoring device can implement a counter to indicate the ID of the port which is driving the SWDI bus, and use SWSYNC signal to reset the counter. When SWSYNC is in use, SWPID is ignored.

SWPID is used to indicate the ID of the port which is driving the Data bus. When SWPID is in use, SWSYNC is ignored.

SWEOF is used to indicate the start and end of a frame. It is always asserted when the corresponding port is idling. The start of a frame is indicated by a high-to-low transition of SWEOF signal. The end of a frame is indicated by a low to high transition of SWEOF signal.

SWCLK is used to provide timing reference of input data snooping and output data latching. The signal is also used as the system clock of the chip.

SWDOV is used to indicate the start of a lookup result package.

SWDOx is used to return the result of lookup to ACD's switch controller chip. Data is latched onto SWDOx bus with respect to the rising edge of SWCLK signal. Each result package is consisted by 5-bit source port ID, 2-bit result, and 5-bit destination port ID. The 2-bit result field is defined as 01 for match, with the port ID shown by the 5-bit destination port ID field; 10 for no match; 11 for forced disregard (filtering).

CPU Interface

The CPU interface provides a communication channel between the CPU and the ACD80800. Basically, the CPU sends command to the ACD80800 by writing into associated registers, and retrieve result from ACD80800 by reading corresponding registers. The registers are described in the section of “Register Description.” The CPU interface signals are described by *table 2*:

Table-2: CPU Interface

Name	I/O	Description
CPUA0 ~ CPUA4	I	5 address lines for register selection.
CPUD0 ~ CPUD7	I/O	8 data lines.
nCPUOE	I	Read enable signal, low active.
nCPUWE	I	Write enable signal, low active.
nCPUCS	I	Chip Select signal, low active.
CPUIRQ	O	Interrupt request signal.
UARTDI	I	UART input data line.
UARTDO	O	UART output data line.

CPUAx is the address bus used to select the registers of the ACD80800.

CPUDx is the data bus used to pass data between the CPU and the registers of the ACD80800.

nCPUOE is used to control the timing of the read operation.

nCPUWE is used to control the timing of the write operation.

nCPUCS is used to make the ACD80800 active to the nCPUOE or nCPUWE signals.

CPUIRQ is used to generate an interrupt request to the CPU. For each source of the interrupt, refer to the description of the interrupt source register.

UARTDI is used by the control CPU to send command into the ACD80800. The baud rate will be automatically detected by the ACD80800. The result will be returned through the UARTDO line with the detected baud rate. The format of the command packet is shown as follows:

Header	Address	Data	Checksum

where:

- Header is further defined as:
  - b1:b0 - read or write, 01 for read, 11 for write
  - b4:b2 - device number, 000 to 111 (0 to 7)
  - b7:b5 - device type, 010 for ARL
- Address - 8-bit value used to select the register to access
- Data - 32-bit value, only the LSB is used for write operation, all 0 for read operation
- Checksum - 8-bit value of XOR of all bytes

UARTDO is used to return the result of command execution to the CPU. The format of the result packet is shown as follows:

Header	Address	Data	Checksum

where:

- Header is further defined as:
  - b1:b0 - read or write, 01 for read, 11 for write
  - b4:b2 - device number, 000 to 111 (0 to 7)
  - b7:b5 - device type, 010 for ARL
- Address - 8-bit value for address of the selected register
- Data - 32-bit value, only the LSB is used for read operation, all 0 for write operation
- Checksum - 8-bit value of XOR of all bytes

The ACD80800 will always check the CMD header to see if both the device type and the device number matches with its setting. If not, it ignores the command and will not generate any response to this command.

Other Interface *(table 3)*

Table-3: Other Interface

Name	I/O	Description
WCHDOG	O	Alive signal from ACD80800 to indicate it is working properly.
nRESET	I	Hardware reset signal, low active.
VDD	-	3.3V power supply.
GND	-	Ground.

WCHDOG signal is used to prevent the system from hitting dead-lock by any abnormal event. Under normal condition, the output signal from the WCHDOG pin will not stay at low for longer than 10ms. If the state of

WCHDOG remains at low state, the chip is not working properly and needs to be reset.

nRESET pin is used to do a hardware reset to the ACD80800. Please note that after a hardware reset, all learned address is cleared, and the address table has to be built again.

Configuration Interface

The following table shows the Power-On-Strobed configuration setting:

Power-On-Strobed Setting

Name	Description	Shared Pin#
BIST Enable	Boot-Internal-Self-Test for optional internal RAM test	78
IC Test Enable	IC Manufacturer test use only: always pull low	77
DIO Enable	1 = Data I/O 0 = UART Mode	76
Port ID Select	1 = for 82124 or 82012 0 = reserved	75
NO CPU	11 = No CPU, 80800 will self initiate 00 = With CPU, 80800 will wait for CPU to initiate	74/111
Bus Width Selection	00 = 32 bit (reserved) 01 = 48 bit (82124 or 82012) 10 = reserved 11 = 64 bit (reserved)	110/109
UART ID	000 = ID for the only or the first 80800 on the system 001 = ID for the second 80800 on the system with two 80800s	114/113/112

Note: High=1=Enable

6. REGISTER DESCRIPTION

ACD80800 provides a bunch of registers for the CPU to access the address table inside it. Command is sent to ACD80800 by writing into the associated registers. Before the CPU can pass a command to ACD80800, it must check the result register *(register 11)* to see if the command has been done. When the Result register indicates the command has been done, the CPU may need to retrieve the result of previous command first. After that, the CPU has to write the associated parameter of the command into the Data registers. Then, the CPU can write the command type into the command register. When a new command is written into the command register, ACD80800 will change the status of the Result register to 0. The Result register will indicate the completion of the command at the end of the execution. Before the completion of the execution, any command written into the command register is ignored by ACD80800.

The registers accessible to the CPU are described by *table 4*:

Table-4: Register Description

Reg.	Name	Description
0	DataReg0	Byte 0 of data
1	DataReg1	Byte 1 of data
2	DataReg2	Byte 2 of data
3	DataReg3	Byte 3 of data
4	DataReg4	Byte 4 of data
5	DataReg5	Byte 5 of data
6	DataReg6	Byte 6 of data
7	DataReg7	Byte 7 of data
8	AddrReg0	LSB of address value
9	AddrReg1	MSB of address value
10	CmdReg	Command register
11	RsltReg	Result register
12	CfgReg	Configuration register
13	IntSrcReg	Interrupt source register
14	IntMskReg	Interrupt mask register
15	nLearnReg0	Address learning disable register for port 0 - 7
16	nLearnReg1	Address learning disable register for port 8 - 15
17	nLearnReg2	Address learning disable register for port 16 - 23
18	AgeTimeReg0	LSB of aging period register
19	AgeTimeReg 1	MSB of aging period register
20	PosCfg0	Power On Strobe configuration register 0
21	PosCfg1	Power On Strobe configuration register 1

The *DataRegX* are registers used to pass the parameter of the command to the ACD80800, and the result of the command to the CPU.

The *AddrRegX* are registers used to specify the address associated with the command.

The *CmdReg* is used to pass the type of command to the ACD80800. The command types are listed in *table 5*. The details of each command is described in the chapter of “Command Description.”

Table-5: Command List

Command	Description
0x09	Add the specified MAC address into the address table
0x0A	Set a lock for the specified MAC address
0x0B	Set a filtering flag for the specified MAC address
0x0C	Delete the specified MAC address from the address table
0x0D	Assign a port ID to the specified MAC address
0x10	Read the first entry of the address table
0x11	Read next entry of address book
0x20	Read first valid entry
0x21	Read next valid entry
0x30	Read first new page
0x31	Read next new page
0x40	Read first aged page
0x41	Read next aged page
0x50	Read first locked page
0x51	Read next locked page
0x60	Read first filtered page
0x61	Read next filtered page
0x80	Read first page with specified PID
0x81	Read next page with specified PID
0xFF	System reset

The *RstReg* is used to indicate the status of command execution. The result code is listed as follows:

- 01 - command is being executed and is not done yet
- 10 - command is done with no error
- 1x - command is done, with error indicated by x, where x is a 4-bit error code: 0001 for cannot find the entry as specified

The *CfgReg* is used to configure the way the ACD80800 works. The bit definition of CfgReg is described as:

- bit 0 - disable address aging
- bit 1 - disable address lookup
- bit 2 - disable DA cache
- bit 3 - disable SA cache
- bit 7:4 - hashing algorithm selection, default is 0000

The *IntSrcReg* is used to indicate what can cause interrupt request to CPU. The source of interrupt is listed as:

- bit 0 - aged address exists
- bit 1 - new address exists
- bit 2 - reserved
- bit 3 - reserved
- bit 4 - bucket overflowed
- bit 5 - command is done
- bit 6 - system initialization is completed
- bit 7 - self test failure

The *IntMskReg* is used to enable an interrupt source to generate an interrupt request. The bit definition is the same as IntSrcReg. A 1 in a bit enables the corresponding interrupt source to generate an interrupt request once it is set.

The *nLearnReg[2:0]* are used to disable address learning activity from a particular port. If the bit corresponding to a port is set, ACD80800 will not try to learn new addresses from that port.

The *AgeTimeReg[1:0]* are used to specify the period of address aging control. The aging period can be from 0 to 65535 units, with each unit counted as 2.684 second.

The *PosCfgReg0* is a configuration register whose default value is determined by the pull-up or pull-down status of the associated hardware pin. The bits of PosCfgReg0 is listed as follows:

- bit 0 - BISTEN, shared with ARLDO0, 0 for no self test, 1 for enable self test
- bit 1 - TESTEN, shared with ARLDO1, 0 for normal operation, 1 for production test.
- bit 2 - DIOEN, shared with ARLDO2, 0 for using UART, 1 for using DIO.
- bit 3 - SYNCEN, shared with ARLDO3, 0 for using SWPID, 1 for using SWSYNC.
- bit 4 - NOCPU*, 0 for have a control CPU, 1 for do not have a control CPU.

Note: When *NOCPU* is set as 0, ACD80800 will not start the initialization process until a System Start command is sent to the command register.

The *PosCfgReg1* is a configuration register whose default value is determined by the pull-up or pull-down status of the associated hardware pin. The bits of PosCfgReg1 is listed as follows:

- bit 1:0 - BUSMODE, shared with CPUD1:CPUD0, bus width selection, 01 for 48-bit, 10 for 64 bit.
- bit 2 - CPUGO, shared with CPUD2, only effective when NOCPU bit of PosCfgReg0 is set to 1. Setting CPUGO to 0 means

wait for the CPU to send the System Start command before the initialization process can be started.

- bit 5:3 - UARTID, shared with CPUD5:CPUD3, 3-bit ID for UART communication.

7. COMMAND DESCRIPTION

Command 09H

Description: Add the specified MAC address into the address table.

Parameter: Store the MAC address into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. Store the associated port number into DataReg6.

Result: the MAC address will be stored into the address table if there is space available. The result is indicated by the Result register.

Command 0AH

Description: Set the Lock bit for the specified MAC address.

Parameter: Store the MAC address into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB.

Result: the state machine will seek for an entry with matched MAC address, and set the Lock bit of the entry. The result is indicated by the Result register.

Command 0BH

Description: Set the Filter flag for the specified MAC address.

Parameter: Store the MAC address into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB.

Result: the state machine will seek for an entry with matched MAC address, and set the Filter bit of the entry. The result is indicated by the Result register.

Command 0CH

Description: Delete the specified MAC address from the address table.

Parameter: Store the MAC address into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB.

Result: the MAC address will be removed from the address table. The result is indicated by the Result register.

Command 0DH

Description: Assign the associated port number to the specified MAC address.

Parameter: Store the MAC address into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. Store the port number into DataReg6.

Result: the port ID field of the entry containing the specified MAC address will be changed accordingly. The result is indicated by the Result register.

Command 10H

Description: Read the first entry of the address table.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of the first entry of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag* bits are stored in DataReg7.The Read Pointer will be set to point to second entry of the address book.

Note - the Flag bits are defined as:

b7	b6	b5	b4	b3	b2	b1	b0
Rsvd	Rsvd	Filter	Lock	New	Old	Age	Valid

where:

- Filter - 1 indicates the frame heading to this address should be dropped.
- Lock - 1 indicates the entry should never be changed or aged out.
- New - 1 indicates the entry is a newly learned address.
- Old - 1 indicates the address has been aged out.
- Age - 1 indicates the address has not

been visited for current age cycle.
- Valid - 1 indicates the entry is a valid one.
- Rsvd - Reserved bits.

Command 11H

Description: Read next entry of address book.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of the address book entry pointed by Read Pointer will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer will be increased by one.

Command 20H

Description: Read first valid entry.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of first valid entry of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 21H

Description: Read next valid entry.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of next valid entry from the Read Pointer of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 30H

Description: Read first new page.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of first new entry of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 31H

Description: Read next new entry.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of next new entry from the Read Pointer of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 40H

Description: Read first aged entry.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of first aged entry of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 41H

Description: Read next aged entry.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of next aged entry from the Read Pointer of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the

MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 50H

Description: Read first locked entry.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of first locked entry of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 51H

Description: Read next locked entry.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of next locked entry from the Read Pointer of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 60H

Description: Read first filtered page.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of first filtered entry of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 61H

Description: Read next valid entry.

Parameter: None

Result: The result is indicated by the Result register. If the command is completed with no error, the content of next filtered entry from the Read Pointer of the address book will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Command 80H

Description: Read first entry with specified port number.

Parameter: Store port number into DataReg6.

Result: The result is indicated by the Result register. If the command is completed with no error, the content of first entry of the address book with the said port number will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

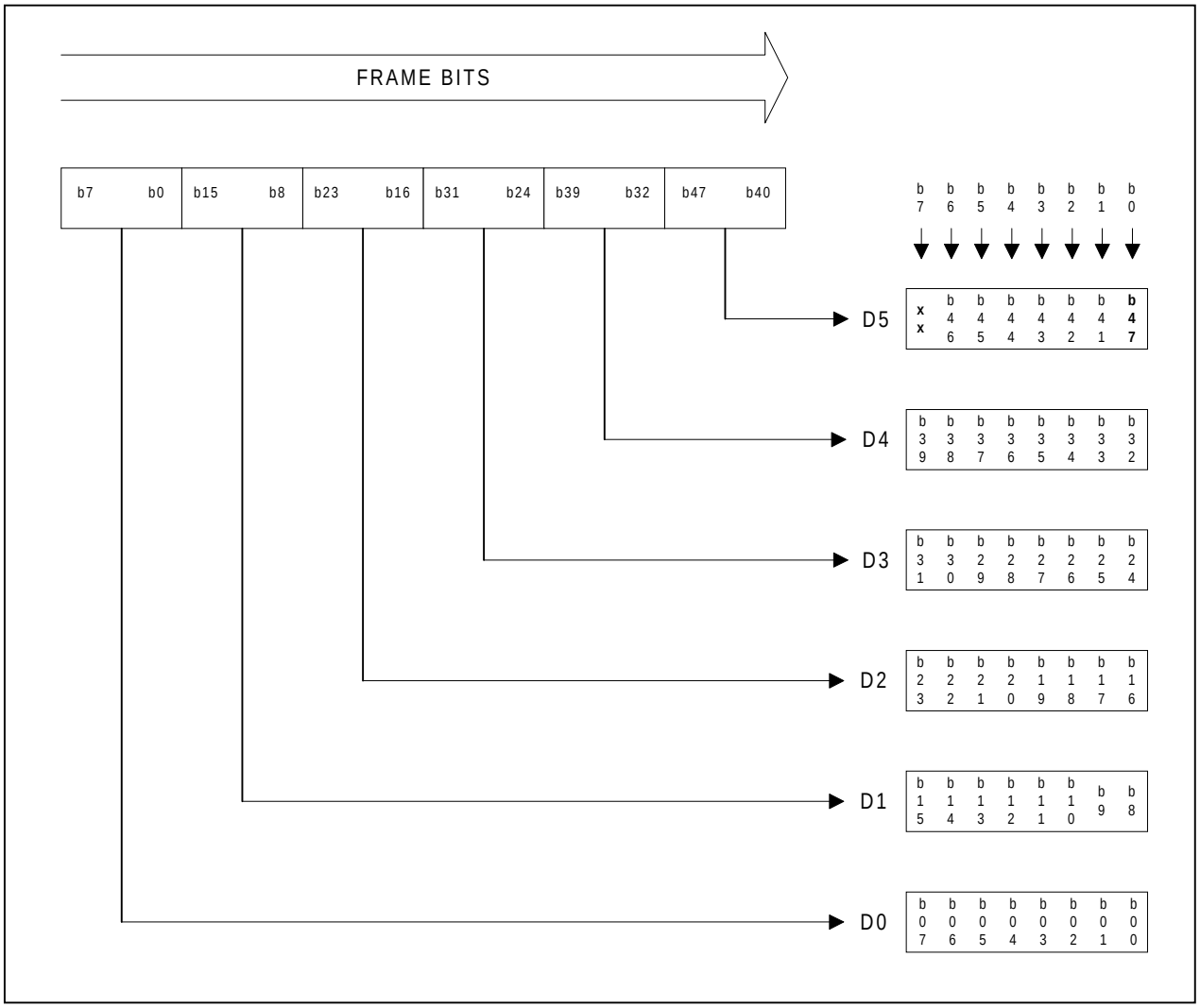
Command 81H

Description: Read next valid entry.

Parameter: Store port number into DataReg6.

Result: The result is indicated by the Result register. If the command is completed with no error, the content of next entry from the Read Pointer of the address book with the said port number will be stored into the Data registers. The MAC address will be stored into DataReg5 - DataReg0, with DataReg5 contains the MSB of the MAC address and DataReg0 contains the LSB. The port number is stored in DataReg6, and the Flag bits are stored in DataReg7. The Read Pointer is set to point to this entry.

Figure-4: Format of a 48-bit MAC Address in a Data Register



Command FFH

Description: System reset.

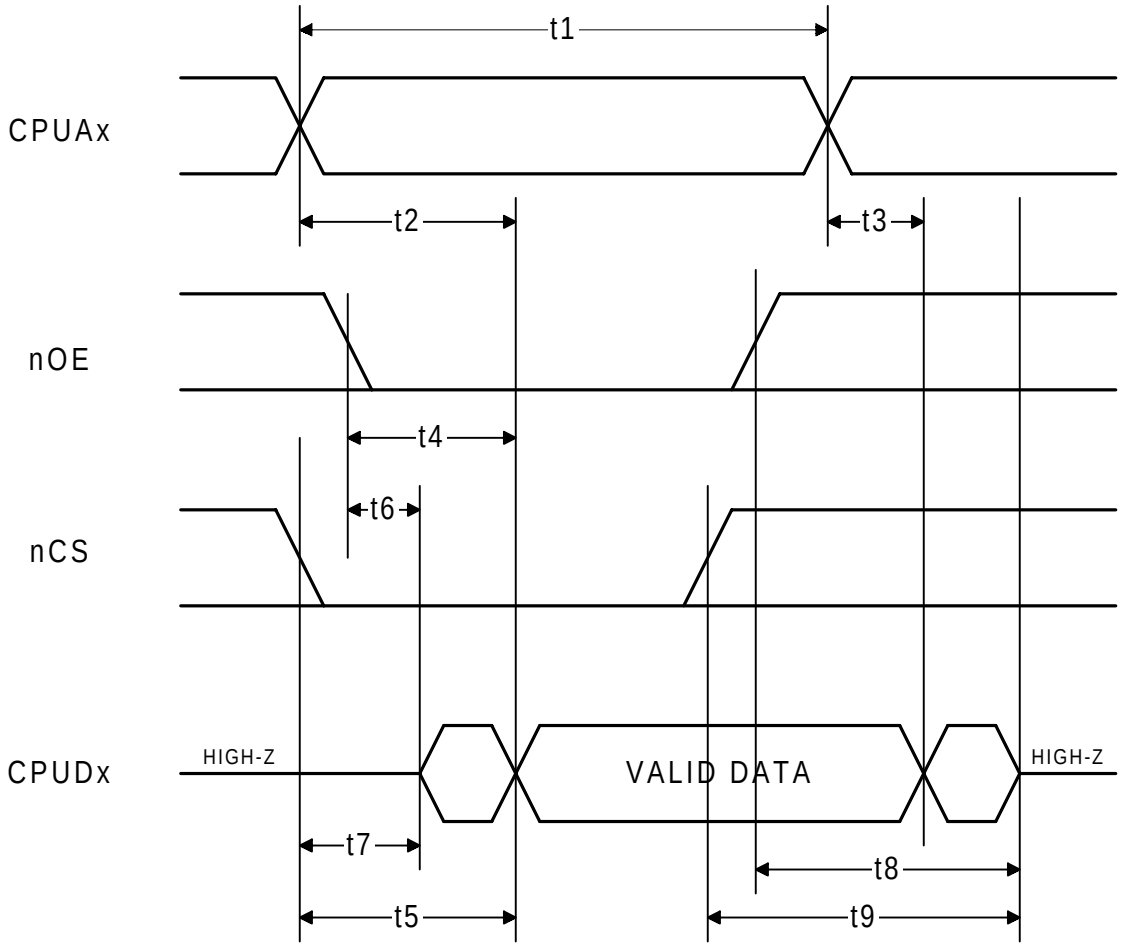
Parameter: None

Result: This command will reset the ARL system. All entries of the address book will be cleared.

***Note*: The handling of the MAC address is shown in *figure 4*. Special attention should be given to the location of *bit 47* of the MAC address, which is at *bit 0* of *D5*. The software needs to be aware of this and make the corresponding adjustment.**

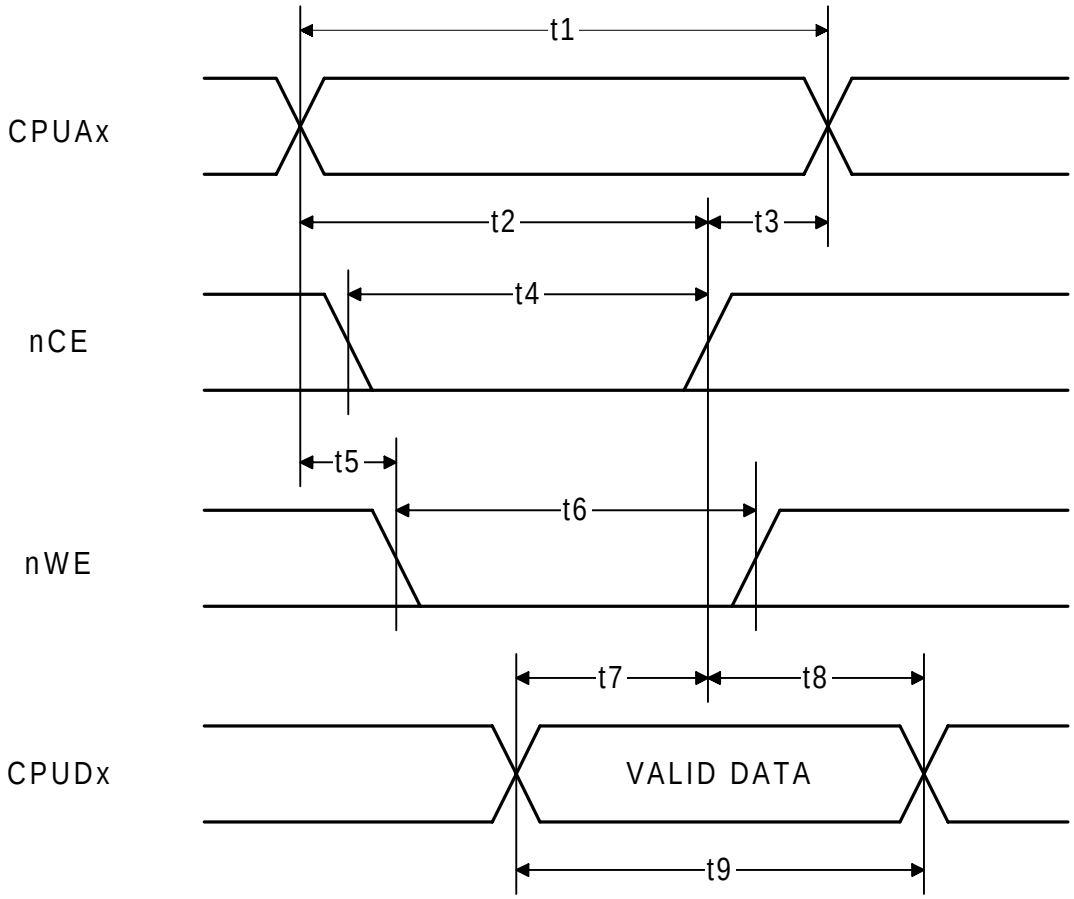
8. TIMING DESCRIPTIONS

Figure-5: Timing Of CPU Read Operation



Time	Description	Min	Typ	Max	Unit
t1	Read cycle time	50	-	-	ns
t2	Address access time	50	-	-	ns
t3	Output hold time	0	-	-	ns
t4	nOE access time	-	-	45	ns
t5	nCE access time	-	-	45	ns
t6	nOE to Low-Z output	0	-	-	ns
t7	nCE to Low-Z output	0	-	-	ns
t8	nOE to High-Z output	-	-	5	ns
t9	nCE to High-Z output	-	-	5	ns

Figure-6: Timing Of CPU Write Operation



Time	Description	Min	Typ	Max	Unit
t1	Write Cycle Time	30	-	-	ns
t2	Address Valid to Write End	30	-	-	ns
t3	Address Hold for Write End	0	-	-	ns
t4	nCE to Write End	25	-	-	ns
t5	Address Setup time	0	-	-	ns
t6	WE pulse width	25	-	-	ns
t7	Data Valid to Write End	30	-	-	ns
t8	Data Hold for Write End	0	-	-	ns

9. ELECTRICAL SPECIFICATION

Absolute Maximum Ratings

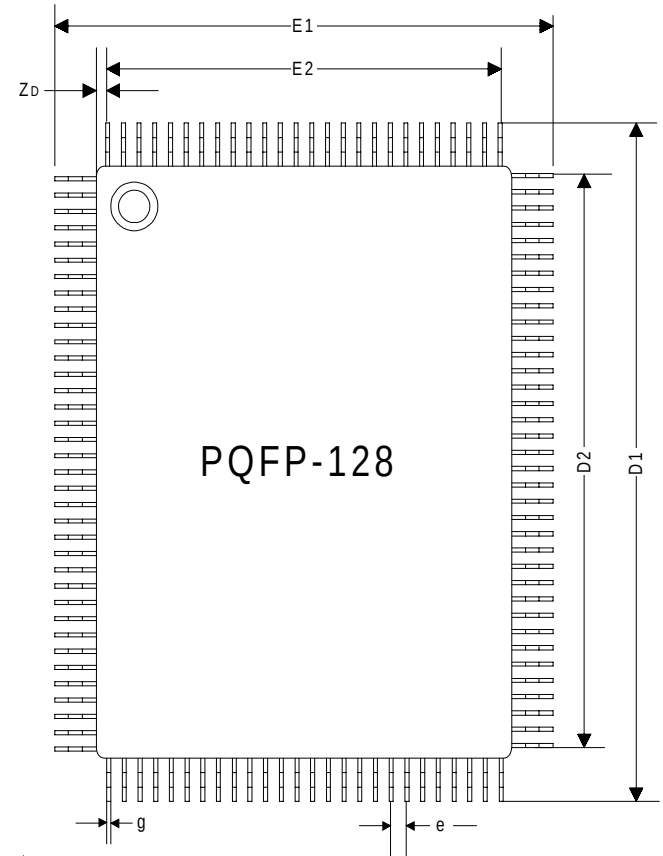
Operation at absolute maximum ratings is not implied exposure to stresses outside those listed could cause permanent damage to the device.

DC Supply voltage : VDD	-0.3V ~ +4.5V
DC input current: Iin	+/-10 mA
DC input voltage: Vin	-0.3 ~ VDD + 0.3V
DC output voltage: Vout	-0.3 ~ VDD + 0.3V

Recommended Operation Conditions

Supply voltage: VDD	3.3V+/-10%
Operating temperature: Ta	0^{o}C -70 ^{o}C
Maximum Power dissipation	900mW

10. PACKAGING



Symble	Min	Nom	Max
A1	0.25	0.33	na
A2	2.57	2.71	2.87
D1	na	23.2	na
D2	na	18.5	na
e	na	0.5	na
E1	na	17.2	na
E2	na	12.5	na
f	0.13	0.15	0.17
g	0.13	0.2	0.28
L1	0.73	0.88	1.03
L2	na	1.6	na
R1	0.13	na	na
R2	0.13	0.3	na
ZD	na	0.75	na