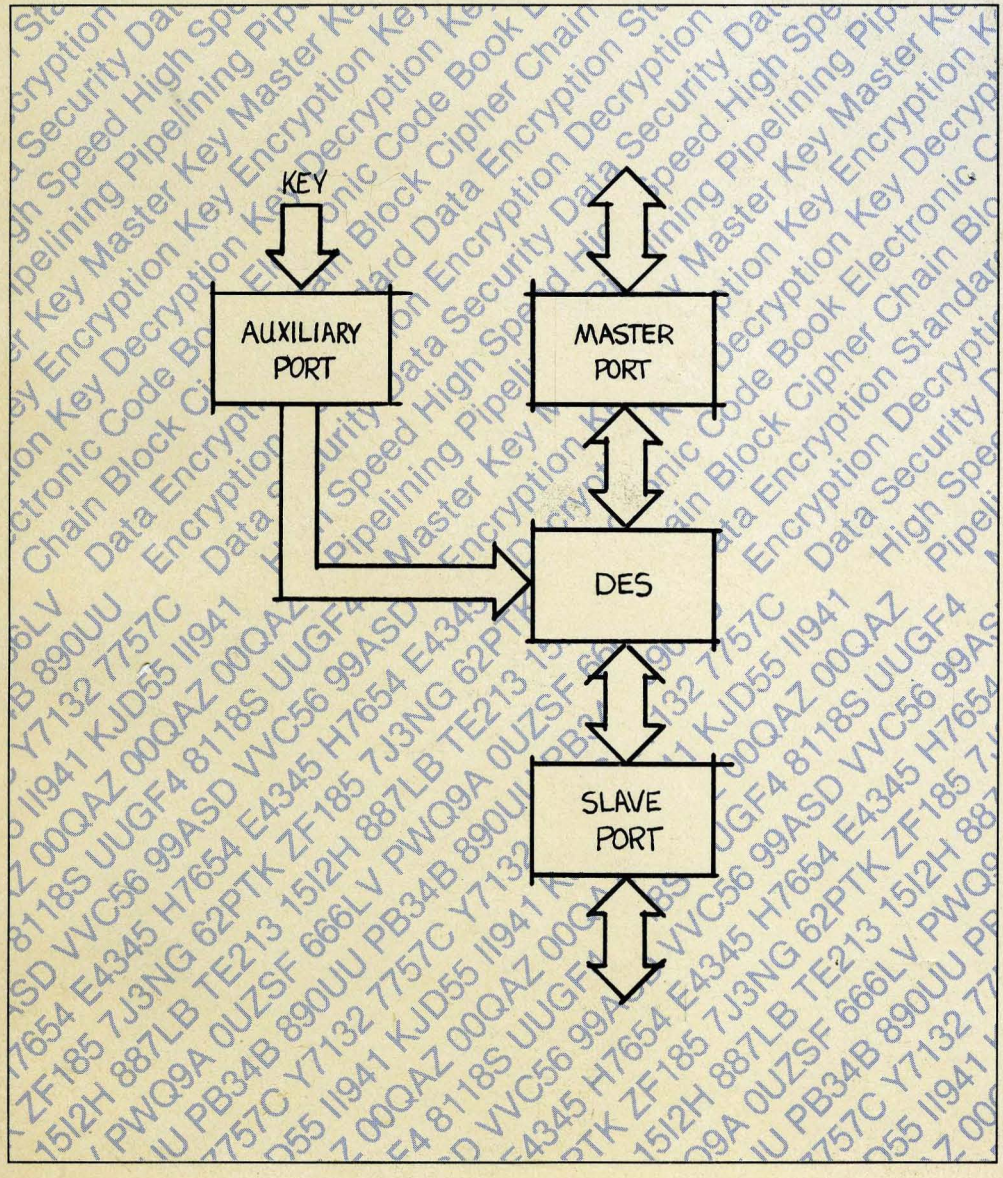
Data Ciphering Processors
Am9518, Am9568, AmZ8068

Technical Manual



ADVANCED MICRO DEVICES

Advanced Micro Devices

Am9518/AmZ8068/Am9568 Data Ciphering Processors

Technical Manual

901 Thompson Place, P.O. Box 3453, Sunnyvale, California 94088
(408) 732-2400 TWX: 910-339-9280 TELEX: 34-6306

Printed in U.S.A.

ACKNOWLEDGEMENTS:

This technical manual was written by Juergen Stelbrink, Headquarters Applications Engineer.

Chapter 4.12 HIGH SPEED SERIAL DATA CIPHERING IN NETWORK SYSTEMS was contributed by Al Sussman, Field Applications Engineer in Burlington, Massachusetts.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

Cryptography is almost as old as civilization. The human desire for privacy when communicating leads inevitably to cryptography. Webster's Dictionary describes cryptography as: "the art or practice of preparing messages in a form intended to prevent their being read by those not privy to secrets of the form; also: the science of devising methods and means for this". The word cryptography combines the Greek "kryptos" (secret) and "graphos" (writing).

The Spartans established one of the first military cryptographic systems in the fifth century B.C. They developed a simple tool consisting of a strip of parchment wrapped around a staff of wood. The original message was written on the parchment down the length of the staff. Once unwrapped, the message becomes unreadable and can be transferred by messenger to the receiver, who decrypts the message by rewrapping it around a staff of the same thickness. The Spartans used it to transfer secret information during the Persian Wars.

There are two basic kinds of encrypting or ciphering methods: transposition and substitution. Data ciphering by transposition takes the characters of the original message (the plain text) and scrambles them to form the encrypted message (the cipher text). The scrambling changes the position of characters in the text only and not the characters themselves. "CIPHER" written as "HCERPI" is an example of transposition ciphering.

The substitution method replaces each character of the original text by another character, number or special symbol. Julius Caesar designed a cryptographic algorithm where the characters were shifted a fixed number of positions; for a shift of three positions,an "a" becomes a "d" and a "b" becomes an "e". His name is substituted as "Mxolxv Fdhvdu". He employed this algorithm to protect an exchange of letters with Cicero during the Gallic Wars.

The fundamental weakness of Caesar's algorithm is that it always encrypts the same letter in the same manner. Codebreaking techniques introduced in the second half of the nineteenth century take advantage of the fact that each language has its own character frequency spectrum. The most common letter in the English language is the "e"; the most frequently recurring double letters are "th". Spectrum analysis can easily break Caesar's code.

More sophisticated algorithms developed in the Renaissance eliminated the weakness of Caesar's code. The encrypted character becomes a function of the original character and its position in the text. The same character in two different text locations is replaced by different encrypted characters.

Chapter 1

German intelligence in World War I employed a code where a list of words organized in a dictionary were linked to a set of numbers. The linkage was not organized in numerical or alphabetical order; it was a giant substitution. In January 1917, the German Foreign Minister, A. Zimmermann, sent a top-secret encrypted telegram to his ambassador in Washington. The British Post Office intercepted this wireless telegram and sent it to the codebreaking branch of British Naval Intelligence. The decoding of the "Zimmermann telegram" was probably the most important single codebreaking task in intelligence history. It caused the United States to join the war.

Until the early Sixties, most cryptographic equipment was based on complicated machines consisting of many mechanical disks and gears. Today, the use of electronic devices increased the capabilities of cryptography. The algorithms are now more sophisticated; but, on the other hand, cryptoanalysts are also able to break more sophisticated codes using computers.

The extensive use of data communication over radio or telephone lines makes it easy for someone to listen to masses of sensitive information without being detected. Great quantities of confidential data, stored on disks or transmitted over various communication links, need protection from unauthorized access. Using any home computer with a modem, an outsider can dial many phone numbers automatically to find a connection where a computer system answers. By trying random passwords he might then gain access to the system, but this access would be worthless if the sensitive data were stored in encrypted form.

A U.S. government department, the National Bureau of Standards, developed an algorithm designed to protect sensitive computer data. Advanced Micro Devices implemented this algorithm into silicon. The result, the Data Ciphering Processor (DCP), is a one-chip 40-pin LSI device, best suited for use in high-speed electronic data ciphering systems and certified by the National Bureau of Standards. The two major application areas of this device are:

- to protect mass data storage (files on tape or disk),

- to protect data communication links to keep the transferred information private (voice encryption, home banking, bank tellers, satellite communication).

This handbook is organized into three parts. First, it gives the reader an overview about data ciphering in general and the DES algorithm supported by the DCP in particular. Differences between two cryptography systems, the public and the private key system, are discussed. Further, it outlines the differences between the three members of the AMD DCP family.

Chapter 3 provides a detailed description of all features and functions of the DCP. It introduces the reader to the internal structure of the DCP and explains the data ciphering instruction set. Timing information can be found in Appendix E. Detailed program flowcharts show the operation of the DCP in the different modes.

Chapter 4 addresses the system designer, providing hints and ideas for designing the DCP into a specific system environment. It shows interfaces to most 8-bit and 16-bit microprocessors. Chapter 4.11 shows what is probably the simplest data ciphering system. It consists of a microcomputer and a DCP built in a "black box". This box provides data ciphering inserted in a serial communication line, for example between a terminal and a modem. Chapter 4.12 shows an application of the DCP in high-speed, serial data-communication environments such as Ethernet.

CHAPTER 2. DATA CIPHERING

The data ciphering algorithm supported by the DCP was tested and accepted by the US government. The technique works by passing original data through a circuit whose output is a complex, non-linear function of the data and a user-supplied, 56-bit key, involving XORing, substitution, block swapping, and key subset selection. The resultant encoded data is called "cipher text".

It is virtually impossible to regenerate the original data without knowing the key. The DES specifies that the algorithm be implemented in hardware rather than software for maximum security. The DCP can execute both encryption and decryption. The device can hold three different keys: one for encryption, one to decrypt a received encoded message and a third one called Master Key to generate session keys or to transfer keys over the line. Refer to Chapter 3.2 (Master Key Register) for more information about the usage of the Master Key. Each key is entered into the DCP as a series of eight bytes, each byte consisting of seven key bits and one parity bit. The chip checks the parity on each byte of the key as entered. To enhance system security, the keys cannot be read back.

The DCP supports three data encryption modes to satisfy the requirements of most applications. Electronic Code Book (ECB) is best suited for high-speed disk applications. Chain Block Cipher (CBC) provides an extra degree of data security over ECB in that it detects any insertion or deletion in the cipher text. It also implements one of the basic cryptography rules: Never encode the same message the same way twice. Data ciphering in disk applications cannot follow this rule because it requires that records be decrypted randomly. The third data ciphering mode is Cipher Feedback (CFB). It is designed for medium-speed, character-based applications. Data is handled on a byte-by-byte basis without waiting to form 64-bit blocks, as in the other two methods.

2.1 DATA ENCRYPTION STANDARD

In January 1977, the National Bureau of Standards published a Data Encryption Standard (DES) in the Federal Information Processing Standards Publication (FIPS PUB 46). The DES specifies an algorithm to be implemented in electronic hardware devices to protect computer data cryptographically. That publication provides a complete description of the mathematical background of the DES algorithm.

Although the DES encryption/decryption algorithm is public information, the individual privacy is insured with a private key. The user can chose any 56-bit key; thus, he can select one of 7.2×10^{16} possible keys. The same key is used for encryption and decryption. The DES is a private key system.

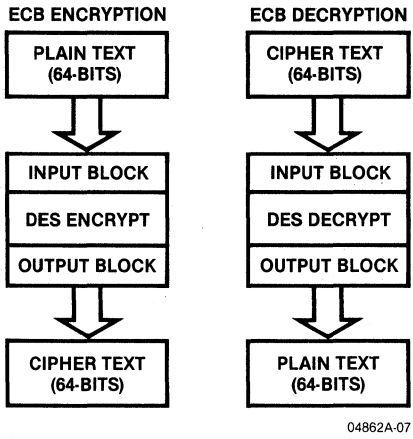


Figure 2.1. Electronic Codebook (ECB) Mode

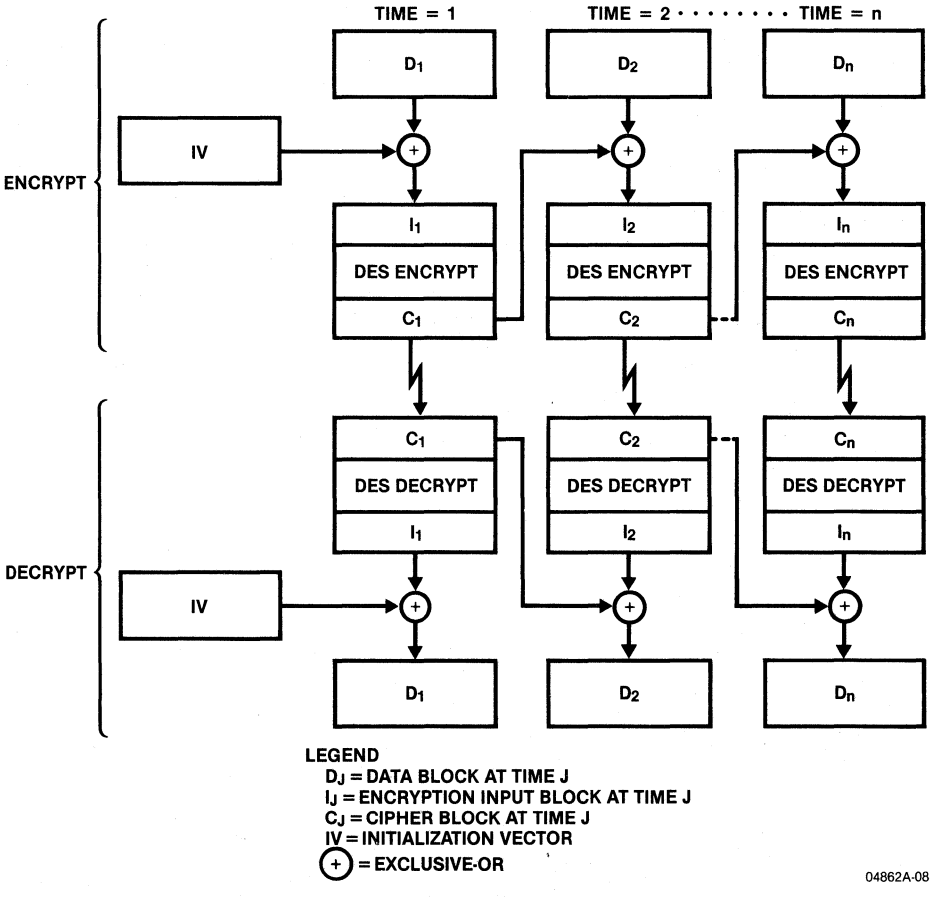


Figure 2.2. Cipher Block Chaining (CBC) Mode

The DES algorithm takes a data block through 18 data-manipulation stages. Sixteen of these stages are identical. They execute complex series of bit manipulations depending on the key.

The first and the last stage do only simple bit transpositions. This overview of the internal operation makes it obvious that this algorithm is well-suited for implementation in electronic hardware.

The DES algorithm translates a 64-bit binary block into a unique 64-bit output block. It is important for some applications that this ciphering algorithm does not add information. Input and output blocks have the same length. Each bit of the result is a function of each and any bit of the input data as well as the key. In other words, a change of any single input bit has approximately equal probability of changing any output bit.

The National Bureau of Standards has defined four implementations of the DES algorithm to be used in a wide variety of applications. These implementations are called Modes of Operation.

Advanced Micro Devices' Data Ciphering Processor was certified by the National Bureau of Standards in January 1981 (see Appendix D). The DCP has passed the DES test and 4 million iterations of the Monte Carlo test. (Since the DES is a complex nonlinear algorithm, it cannot be fully tested with a limited set of test vectors. To verify the correct hardware implementation, the National Bureau of Standards has created a statistical procedure -- the Monte Carlo routine.)

Modes of Operation

The National Bureau of Standards has defined four implementations of the DES algorithm. Each of them is designed for specific applications.

ECB The Electronic Code Book (Figure 2.1) is a direct implementation of the DES algorithm. The analogy to a code book arises because the same plain text always generates the same ciphered text for a given cryptographic key. The DCP determines the codebook entries each time. A single bit error or change, in either the input text block or the key, causes an average bit error rate of 50% for its output block. However, an error in one text block will not affect any other block. In other words, there is no error extention between ECB blocks.

The input and output block size is 64 bits. Since data blocks are independently ciphered, this mode is qualified for disk applications.

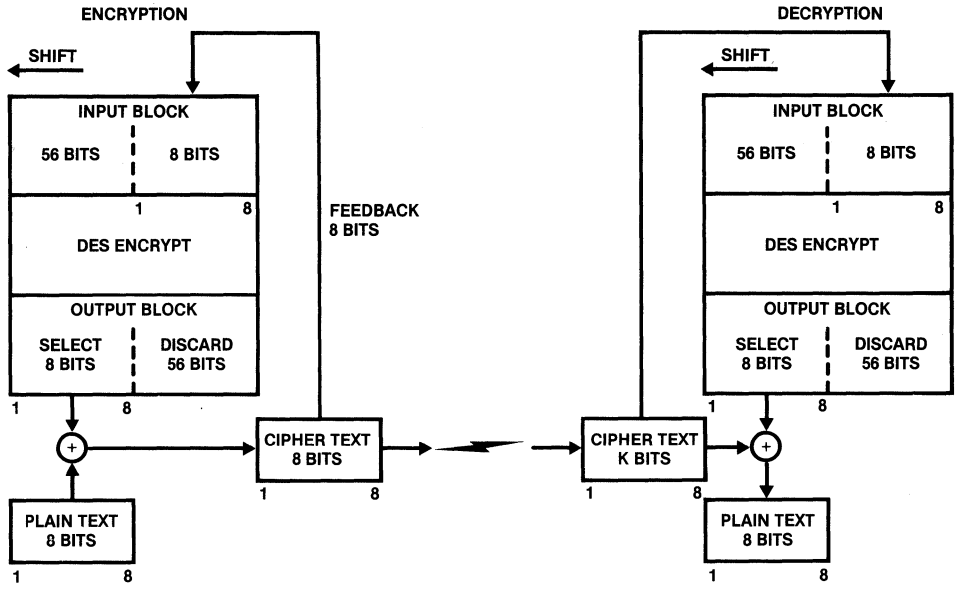


Figure 2.3. 8-Bit Cipher Feedback (CFB) Mode

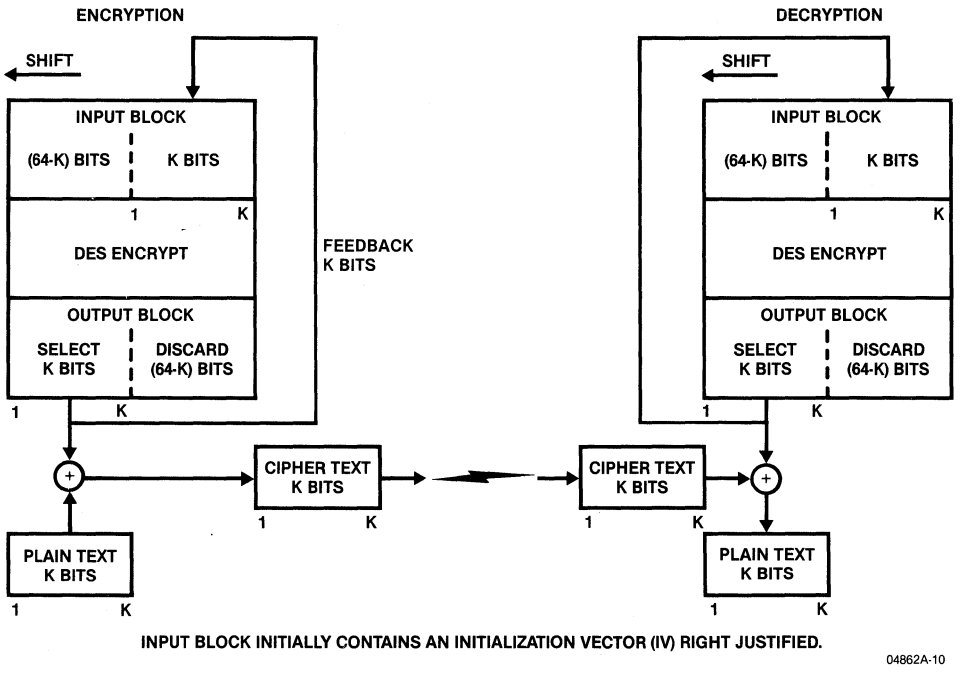


Figure 2.4. K-Bit Output Feedback (OFB) Mode

The ECB mode has the weakness that identical blocks of plain text generate identical blocks of ciphered text. This violates one of the basic laws of encryption security: Never encrypt information the same way twice because this makes it easier for the opponent to break the code. This problem is solved by the CBC mode.

CBC Chain Block Cipher (Figure 2.2) also operates on 64-bit data blocks. The input data block is EXORed with an 64-bit Initial Vector (IV) before being processed by the DES algorithm. The resulting ciphered-output block is loaded into the IV Register, to be ORed with the next input block. This chaining of cipher text blocks provides different outputs for identical input blocks. It also gives an error extention characteristic which protects against fraudulent data insertion, deletion or alteration in a block sequence. A one-bit error in the input text block, the key or the Initial Vector causes an average error rate of 50% in all subsequent output blocks. These features make CBC best suited for high-speed data communications.

CFB Cipher Feedback (Figure 2.3) operates on n-bit data blocks, "n" being any value from 1 to 64. The content of the IV Register is processed by the DES algorithm. The most significant n-bits of the result are EXORed with the n-bit input data block. The result is the n-bit ciphered output block. This output block is shifted into the "n" least significant bits of the IV Register.

The DCP supports 8-bit CFB. Character-based, low-speed to medium-speed data communications is best done by 8-bit CFB. In CFB Mode, the throughput of the DCP is lower than in CBC or ECB because each algorithm pass provides only 8 bits compared to 64 bits in the two high-speed modes.

The error extention characteristic is the same as in CBC.

OFB Under some circumstances, such as a noisy, narrowband digital signal in an encrypted speech application, it is best to use a data-independent stream cipher. Output Feedback (Figure 2.4) is the best technique in this environment. The advantage of OFB is that the output data is a function of only the input data and the number of preceding blocks. It is independent of the actual data contained in the blocks. An error in an input block causes a 50% bit error probability in its output block, but it does not influence subsequent outputs. There is no error extention.

OFB differs from CFB in that the feedback path is data-independent; a part of the output of the DES algorithm is fed back directly. The DES algorithm operates like a pseudo-random number generator.

The DCP does not support OFB directly, but with some external hardware 1-bit and 8-bit OFB can be implemented as shown in Chapter 3.8. No additional hardware is needed to perform 64-bit OFB.

2.2 PUBLIC VERSUS PRIVATE KEY CRYPTOSYSTEMS

The classical single-key cryptosystem, such as DES, operates on the premise that the sender and receiver of messages use the same key for the dual purpose of encryption and decryption. Although such a scheme is adequate for most purposes, it is deficient from the point of view of true "authentication". Authenticity assures that the message has not been tampered with during transmission, and also that the true identity of the sender (also called signature) can be extracted from the encrypted message. In schemes involving sharing of a secret key there is scope for "forgery" since the receiver of a message can generate authenticators that are indistinguishable from those generated by the sender. Furthermore, single-key systems require some form of key distribution prior to activation of the system.

Public key cryptosystems have evolved as an answer to the needs of digital signatures and also to overcome some of the shortcomings of DES. They were first introduced by Diffie and Hellman in 1976. In contrast to DES, these systems use a matched pair of keys (one private and the other public) for the sender and the receiver. Both pairs are generated independently. The private keys are retained by the individual users while their respective public keys are maintained in a common directory possibly managed by a network key server. This scheme separates the encryption and decryption keys. It can transmit encryption messages without prior exchange of keys and can implement digital signatures that are legally binding.

Public key cryptosystems are slow since they involve multiple-precision arithmetic on very large numbers (>100 digits). The functional advantages of a public key cryptosystem can, however, be combined with the advantages of a private key cryptosystem (speed and availability of dedicated VLSI circuits) to form a hybrid system (Figure 2.5).

To transmit a secret text, the sender (A) first generates a random key for encrypting the clear text by means of the fast DES algorithm. The random key is then encrypted using the complicated and slow public key method. Both the encrypted key and text are then transmitted to the receiver. The receiver first decrypts the key and then uses the decrypted key to decrypt the ciphered text. The authenticity of the text can be checked in a second pass.

Splitting the job between the public key and DES algorithm makes sense since the protection of a standard message requires many more DES encryptions than public key encryptions.

For more information on Public Key Systems see:

Burton, C. E. "RSA: A Public Key Cryptography System." Dr. Dobb's Journal, Mar 1984, 16-21.

Diffie, W. and Hellman, M. "New Directions in Cryptography." IEEE Transactions on Information Theory, IT-22(6), Nov 1976,

Gardner, M. "Mathematical Games." Scientific American, 237(2), Aug 1977, 120-124

Mueller-Schloer, Christian. "A Microprocessor-based Cryptoprocessor". IEEE Micro, Oct 1983, 5-15.

Rivest, R.L., A. Shamir and L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." Communications of the ACM, 21(2), Feb 1978, 120-126.

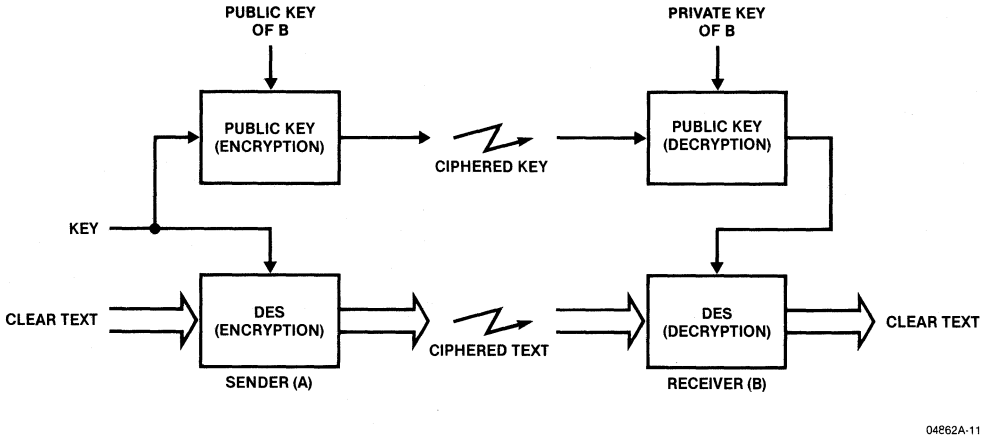


Figure 2.5. Hybrid System

2.3. THE DCP FAMILY

The DCP family consists of three devices:

Am9518	3-MHz version, Z8000*-type bus interface up to 1.3 Mbyte/s ciphering throughput
AmZ8068*	4-MHz version, Z8000-type bus interface up to 1.7 Mbyte/s ciphering throughput
Am9568	4-MHz version, 8086-type bus interface up to 1.5 Mbyte/s ciphering throughput

*Z8000 is a trademark of Zilog, Inc.
*AmZ8068 is a trademark of Advanced Micro Devices, Inc.

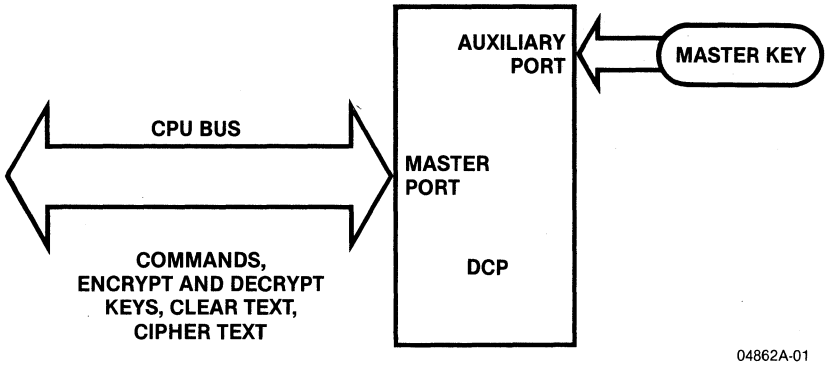


Figure 2.6. Data Flow for Single Port Configuration, Multiplexed Control Mode

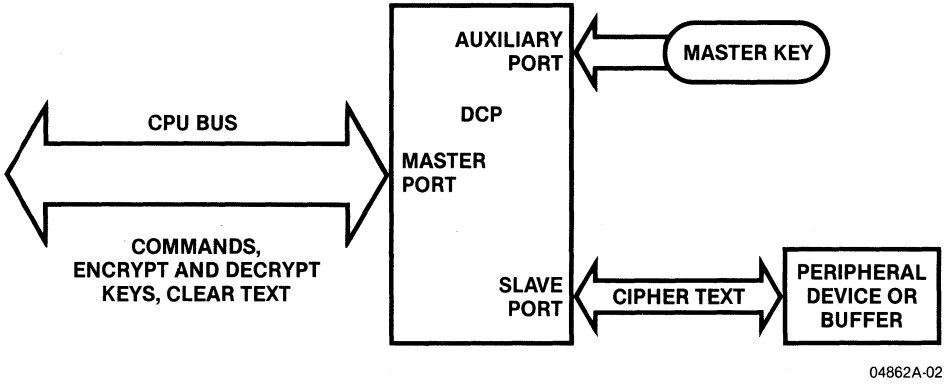


Figure 2.7. Data Flow for Dual Port Configuration, Multiplexed Control Mode

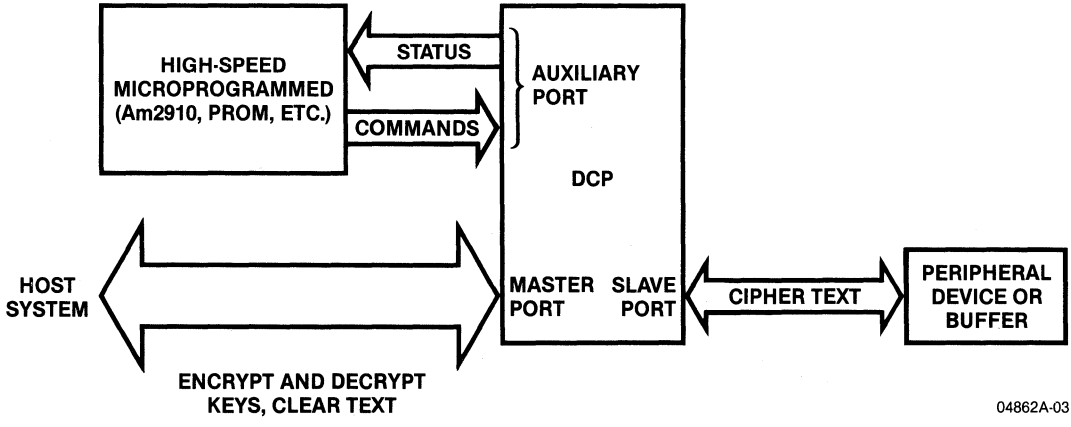


Figure 2.8. Data Flow for Dual Port Configuration, Direct Control Mode

General Description Applicable to All Three Devices

All three devices are designed to be used in a large variety of environments, including dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems. Usually the DCP will be controlled by a standard microprocessor. In this kind of environment, the DCP is interfaced similarly to other peripherals with a multiplexed address/data bus (e.g., AmZ8030, AmZ8036*, and AmZ8073). This mode is called Multiplexed Control Mode. In data storage applications, the data can be passed from the CPU bus through the DCP to the mass storage controller. Most of the tape or hard disk controllers are based on microprogrammed logic. The DCP can be programmed to provide a special microprogrammed interface. This mode is called Direct Control Mode.

The **Multiplexed Control Mode** provides a standard microprocessor interface. Chapters 4.1 to 4.11 show applications where the DCP operates in Multiplexed Control Mode. Figure 2.6 shows the most straightforward interface configuration; it is the single port configuration in Multiplexed Control Mode. In this configuration, all commands and data transferred between the CPU and DCP are passed through the Master Port. The keys for encryption and decryption may be entered through either the Master Port or the Auxiliary Port. The Master Key can only be entered through the Auxiliary Port. The Auxiliary Port is a separate port for key input only. It enhances the system security by separating the data path and the key path. In higher-speed data ciphering applications, the Master Port becomes the bottleneck of the system. Both the original text and the encrypted text have to be passed through this 8-bit port.

The dual port configuration (Figure 2.7) eliminates this bottleneck. The text now flows through the devices. The CPU passes the original text through the Master Port, while the peripheral device removes the encrypted text from the Slave Port. The internal architecture of the DCP is highly pipelined. The CPU may enter one block of data, while a previously entered block is ciphered and while a third previously ciphered block may be read out. This pipelining yields data ciphering rates between 10.6 and 14.2 Mbit/s.

The **Direct Control Mode** (Figure 2.8) provides a special microprogrammed logic interface. In Direct Control Mode the Auxiliary Port becomes a control port for the microprogrammed logic. Unlike Multiplexed Control Mode, where the DCP is now controlled by programming internal registers, the DCP is controlled by three pins of the Auxiliary Port. Two pins reflect the status of the device. In this mode, the DCP can execute only a subset of its data ciphering commands, such as loading encryption or decryption keys and initiating encryption or decryption versions.

*Z8030 and Z8036 are trademarks of Zilog, Inc.

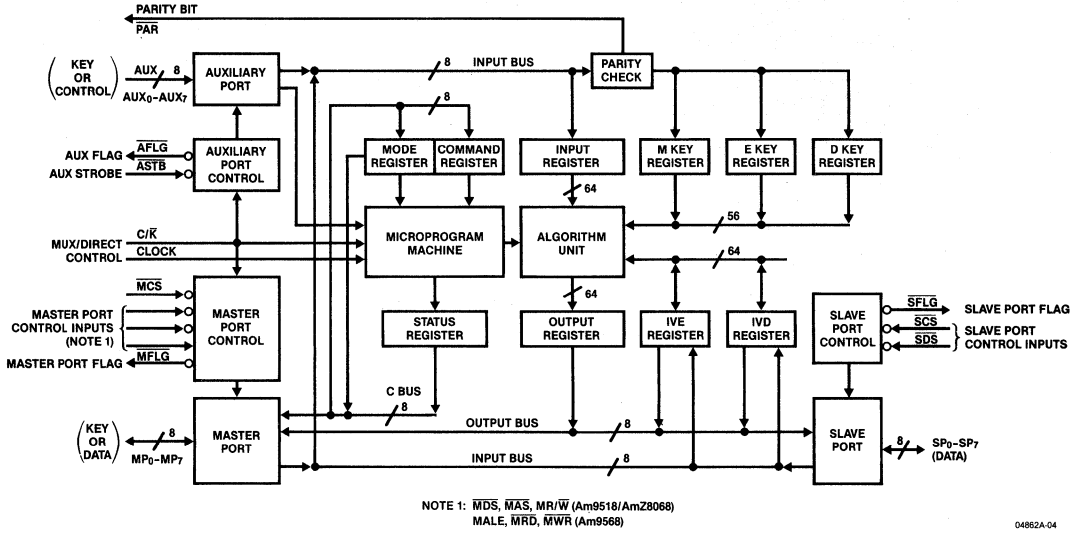


Figure 2.9. DCP Block Diagram

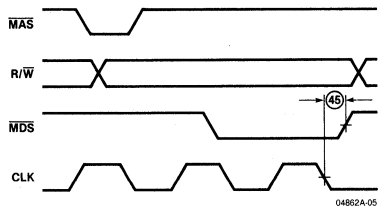


Figure 2.10. Z8000-Type Master Port Timing (Am9518, AmZ8068)

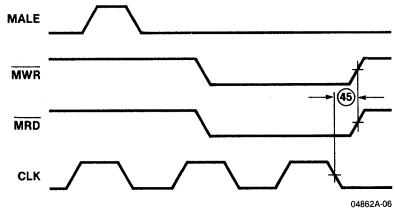


Figure 2.11. 8086-Type Master Port Timing (Am9568)

The Mode Register defines the basic operating parameters such as ciphering mode (ECB, CBC, and CFB) and port configuration. In Direct Control Mode this register cannot be programmed. However, a reset sets this register to its default value (see Chapter 3.5). To operate the DCP in modes different from the default mode, the DCP has to be switched to Multiplexed Control Mode to modify the Mode Register. Therefore, the C/$\overline{K}$ pin (selecting Multiplexed Control Mode or Direct Control Mode) should be mode programmable. Other operations such as loading the Master Key and the Initial Vector (IV) Registers require also that the DCP is switched to Multiplexed Control Mode. In Multiplexed Control Mode, the full data ciphering instruction set is provided.

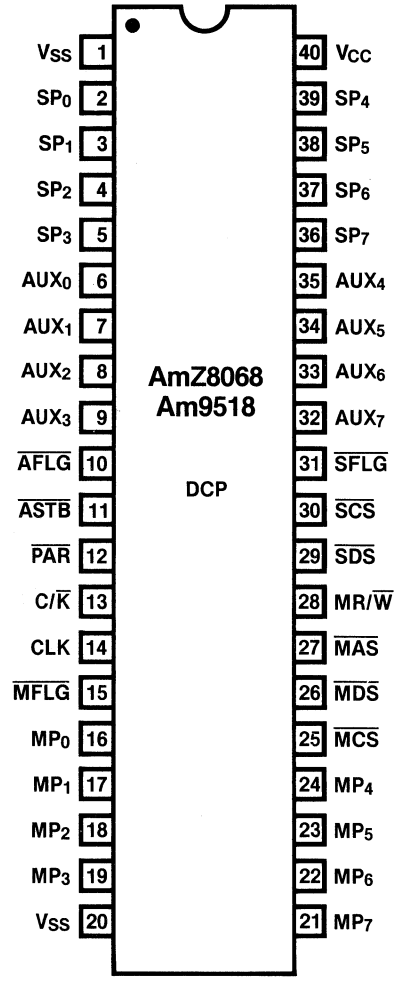
In Multiplexed Control Mode, the devices of the DCP family support two different types of microprocessor interfaces as shown below:

Am9518/AmZ8068

The Am9518 and AmZ8068 (Figure 2.10) support a Z8000-type interface. Figure 2.10 shows the basic timing. The Master Port Address Strobe ($\overline{MAS}$) is active Low. The rising (i.e. trailing) edge latches the level of Master Port Chip Select (MCS) and the 2-bit register address on MP_1 and MP_2. Master Port Data Strobe ($\overline{IMDS}$) provides the timing for the data transfer. The level on Read/Write (R/$\overline{W}$) defines the data transfer direction. Timing parameter 42 of the product specification defines the set-up time of R/$\overline{W}$ to $\overline{MDS}$. The rising edge of $\overline{MDS}$ must be synchronous to the falling edge of the clock. Most CPUs do not meet the specified narrow time range, so external synchronization logic must be added to satisfy this parameter. The interfaces in Chapter 4 show some approaches.

Am9568

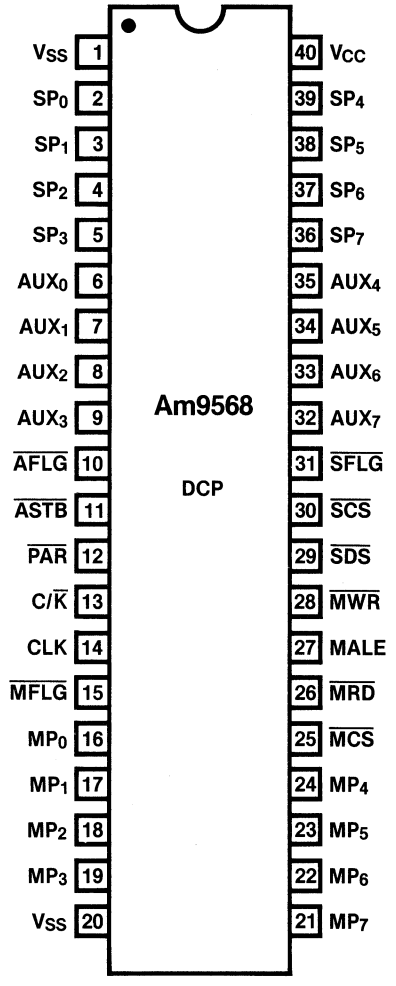
The Am9568 (Figure 2.11) has a host CPU interface which is optimized for the iAPX microprocessor family. Figure 2.11 shows the basic bus timing. Master Port Address Latch Enable (MALE) is active High. The falling (i.e. trailing) edge of ALE latches $\overline{MCS}$ and the register address on MP_1 and MP_2. Master Port Write ($\overline{MWR}$) provides timing for a data write transfer, Master Port Read ($\overline{MRD}$) provides timing for read transfers. Both strobes must be synchronous to the clock. The range is smaller than with the Am9518 or AmZ8068. The Am9568 has advantages in applications requiring narrow address strobes or where it is difficult to satisfy the set-up time of R/$\overline{W}$.



NOTE: PIN 1 IS MARKED FOR ORIENTATION

04862A-12

Figure 3.1 Am9518/AmZ8068 Connection Diagram



NOTE: PIN 1 IS MARKED FOR ORIENTATION.

04862A-13

Figure 3.2 Am9568 Connection Diagram

CHAPTER 3. FUNCTIONAL DESCRIPTION

The heart of the DCP is the Data Encryption Standard (DES) algorithm unit that encrypts 64-bit blocks of clear text into corresponding 64-bit blocks of cipher text using a 56-bit key. The DCP can hold three keys simultaneously: a Master Key to generate session keys, an Encryption Key, and a Decryption Key. A block diagram of the internal structure is shown in Figure 2.4.

The DCP has two 64-bit data registers: the Input and the Output Register. Transfers between these registers and the Master or Slave Port occur on the 8-bit input/output buses. The dual ports, separate internal buses and separate input and output registers compose a highly pipelined data path that maximizes the throughput by allowing simultaneous input, ciphering and output operation.

The 8-bit ports handle the 64-bit blocks of data one byte at a time. Each block is strobed into the Input Register with eight Data Strobes. The most significant byte is entered first. The result block can be read from the 64-bit Output Register, also one byte at a time with the most significant byte first.

3.1. PORTS

3.1.1. MASTER PORT

The Master Port is an 8-bit wide (MP_0-MP_7) bidirectional port. The Mode, Command, and Status Register can be accessed only through this port. The port operation is associated with four control lines, which are defined differently for the two groups of devices.

Am9518/AmZ8068:

- $\overline{MCS}$ Master Port Chip Select
- $\overline{MAS}$ Master Port Address Strobe
- $\overline{MDS}$ Master Port Data Strobe
- R/$\overline{W}$ Read/$\overline{Write}$

Am9568:

- $\overline{MCS}$ Master Port Chip Select
- MALE Master Port Address Latch Enable
- $\overline{MRD}$ Master Port Read
- $\overline{MWR}$ Master Port Write

The DCP executes a hardware reset when two specific control lines are pulled active Low simultaneously. Namely:

For the Am9518/AmZ8068 - $\overline{MAS}$ and $\overline{MDS}$

For the Am9568 - $\overline{MRD}$ and $\overline{MWR}$

In Direct Control Mode the address strobe ($\overline{MAS}$ or MALE) is a "don't care". To prevent hardware resets by mistake, tie $\overline{MAS}$ High for Am9518/AmZ8068 Direct Control Mode applications.

In Multiplexed Control Mode the address strobe latches the level of $\overline{MCS}$ and the two-bit pointer address into one of the five internal registers. In systems with a multiplexed address/data bus, this relieves the external address decode circuitry of the responsibility for latching Chip Select.

The Master Port Flag ($\overline{MFLG}$) shows the status of the device. It corresponds to the Master Port Flag bit of the Status Register. Figure 3.3 shows the association of the Master Port Flag with the Input and Output Register. In dual port configuration, the Flag reflects the status of the Master Port; it is active if data can be transferred to or from the Master Port. Input or Output Operation depends on the Mode (Encryption or Decryption) and clear or ciphered text, at the Master Port. In single port configuration, this flag always reflects the status of the Input Register, independent of the mode.

Master Port in Multiplexed Control Mode

The terminology of the "strobes" is defined below:

Address strobe	Am9518/AmZ8068	$\overline{MAS}$ is strobed Low
	Am9568	MALE is strobed High
Write strobe	Am9518/AmZ8068	$\overline{MDS}$ is strobed Low, while MR/$\overline{W}$ is Low
	Am9568	$\overline{MWR}$ is strobed Low
Read strobe	Am9518/AmZ8068	$\overline{MDS}$ is strobed Low, while MR/$\overline{W}$ is High
	Am9568	$\overline{MRD}$ is strobed Low

Entering encryption/decryption keys (clear or encrypted):

The key registers are loaded by a command/data sequence. The following sequence of operations must be performed:

- Provide $\overline{MCS}$, address the Command Register (MP_1=High, MP_2=High) and issue address strobe.

- Enter command code (see Figure 3.7) by presenting the appropriate one-byte command at the Master Port and issuing a write strobe.

- Provide $\overline{MCS}$, address the Input Register (MP_1=Low, MP_2=Low) and issue address strobe.

- Load eight bytes of key data, one byte at a time, through Master Port. Keys are loaded one byte per write strobe, the most significant byte first.

If the key is in encrypted form, the Master Key must be loaded first through the Auxiliary Port. Then the encrypted key can be loaded. The DCP decrypts this key internally using the Master Key and the ECB method. The clear session key is then stored in the appropriate key register. After loading the last byte of the encrypted key, no read/write to the internal registers is allowed for the subsequent 70 clock cycles.

A key can only be entered into the DCP; for security reasons it cannot be read back. Parity check logic in the DCP verifies that the key is entered correctly. The least significant bit of each byte of key is the parity check bit (odd parity). Flags in the Status Register are set if a parity error occurs during a key load sequence.

Entering/reading the Initial Vector for Encryption (IVE) or Decryption (IVD):

When using the Chain Block Cipher (CBC) or Cipher Feed Back (CFB) mode, the 64-bit IV Register must be initialized. The command/data sequence is similar to the sequence for entering keys.

Similar to the key, the IV can be loaded in either clear or encrypted form. The encrypted IV is decrypted using the Decryption Key (D Key) and ECB mode before loading the appropriate IV Register. The D Key must be loaded first.

When the IV should be read out in encrypted form, it is first encrypted using the E Key and ECB mode. It takes 70 clock cycles to encrypt or decrypt the IV.

Entering or removing data:

Depending on the Mode, either clear or encrypted data can be entered or removed from the Master Port. Data entered through the Master Port goes into the Input Register. Data removed from the Master Port comes from the Output Register. Data is transferred by the following sequence:

- Provide $\overline{MCS}$, address data register

- Transfer data bytes, one byte per write strobe or read strobe, starting with most significant byte. The data transfer is not limited to only one block. The device accepts data whenever the corresponding flag shows that the device is ready for a data transfer.

After entering one block of data, the input flag becomes inactive for 5 clock cycles if the data can be transferred to the algorithm unit. If the algorithm unit is still busy or if the device is blocked because the output data is not read out, the input flag stays inactive.

The output flag becomes active whenever data is in the Output Register. After removing one block, the output flag becomes inactive for 5 clocks if the algorithm unit can provide another block. If the algorithm unit is empty, the output flag stays inactive until data is ready again.

Master Port in Direct Control Mode

Master Port Chip Select ($\overline{MCS}$) is not latched internally. It is passed directly to the internal circuitry.

Enter clear E or D Key using the following sequence:

- Provide $\overline{MCS}$.
- Set up appropriate code at the Auxiliary Port for E/$\overline{D}$ Key load (see Auxiliary Port description).
- Strobe in eight bytes of the key, one byte per write strobe, most significant byte first.

Enter or remove data:

Depending on the configuration chosen by loading the Mode Register, the Master Port can be an input port, an output port or both. The mode determines the direction of data flow. The data access must agree with the mode. Thus data can only be read from the Master Port if the mode defines the Master Port as an Output Port, and data can only be written to the Master Port if it is defined as an Input Port.

- Provide $\overline{MCS}$.
- Provide appropriate code at the Auxiliary Port.
- Read or write one byte of data per read or write strobe starting with the most significant byte of a block.

3.1.2. SLAVE PORT

The Slave Port is an 8-bit-wide, bidirectional port controlled by the Slave Port Chip Select ($\overline{SCS}$) and the Slave Port Data Strobe ($\overline{SDS}$). The direction of the data flow is determined by control

bits in the Mode Register. In both Multiplexed and Direct Control Mode, the Slave Port may be used for either data input or output operation. The Slave Port is only active if the dual port configuration is chosen. In dual port configuration, the Slave Port Flag ($\overline{SFLG}$) reflects the status of the Slave Port (Figure 3.3). If the flag is active, data can be strobed in or removed depending on the programmed data flow direction. In single port configuration (Master Port only) the Slave Port Flag represents the status of the Output Register. The Slave Port Flag corresponds to one bit of the Status Register.

- Provide $\overline{SCS}$.

- Read or write one byte of data per strobe ($\overline{SDS}$) beginning with the most significant byte.

$\overline{SCS}$ is not latched internally, and may be tied permanently Low without impairing Slave Port operation.

3.1.3. AUXILIARY PORT

The Auxiliary Port has fundamentally different functions in Multiplexed Control Mode and in Direct Control Mode.

Auxiliary Port in Multiplexed Control Mode

The port is 8-bits wide and can be used for key input only. The status signal Auxiliary Port Flag ($\overline{AFLG}$) becomes active whenever key data can be entered. The rising edge of the control signal Auxiliary Port Strobe ($\overline{ASTB}$) strobes in the key data one byte at a time. $\overline{ASTB}$ is ignored unless $\overline{AFLG}$ and C/$\overline{K}$ are both Low. To use the Auxiliary Port for key entry, the following sequence can be performed:

- Enter an appropriate command through the Master Port into the Command Register that requires Auxiliary Port operation; e.g., "Load Encrypted E Key through Auxiliary Port".

- In response to these commands, the Auxiliary Flag ($\overline{AFLG}$) becomes active Low. Eight bytes of key can then be entered by strobing Auxiliary Strobe ($\overline{ASTB}$). $\overline{AFLG}$ becomes inactive shortly after the falling edge of the eighth strobe.

The Master Key, which is needed to generate session keys, can only be loaded through the Auxiliary Port. A key loaded in encrypted form is decrypted using the Master Key and ECB mode. To guarantee the system security, a key cannot be read back.

Auxiliary Port in Direct Control Mode

In this mode, the Auxiliary Port operates as a control port for the microprogrammed logic. A subset of the cipher processing commands can be executed. Three pins are control inputs, two pins are status outputs:

Encrypt/ Decrypt M_4	Port Configuration M_3	M_2	Input Register Flag	Output Register Flag
0	0	0	MFLG	SFLG
0	0	1	SFLG	MFLG
0	1	0	MFLG	SFLG
1	0	0	SFLG	MFLG
1	0	1	MFLG	SFLG
1	1	0	MFLG	SFLG

04862A-14

Figure 3.3. Association of Master Port Flag (MFLG) and Slave Port Flag (SFLG) with Input and Output Registers

Am9518/AmZ8068

C/$\overline{K}$	MP_2	MP_1	MR/$\overline{W}$	$\overline{MCS}$	Register Addressed
0	X	0	0	0	Input Register
0	X	0	1	0	Output Register
0	0	1	0	0	Command Register
0	0	1	1	0	Status Register
0	1	1	X	0	Mode Register
X	X	X	X	1	No Register Accessed
1	X	X	0	0	Input Register
1	X	X	1	0	Output Register

Am9568

C/$\overline{K}$	MP_2	MP_1	$\overline{MRD}$	$\overline{MWR}$	$\overline{MCS}$	Register Addressed
0	X	0	1	0	0	Input Register
0	X	0	0	1	0	Output Register
0	0	1	1	0	0	Command Register
0	0	1	0	1	0	Status Register
0	1	1	X	X	0	Mode Register
X	X	X	X	X	1	No Register Accessed
1	X	X	1	0	0	Input Register
1	X	X	0	1	0	Output Register

04862A-15

Figure 3.4. Master Port Register Addresses

AUX_7-K/$\overline{D}$ (Key/$\overline{Data}$, Input)

When this signal goes High, the DCP initiates a key-data input sequence as if a Load Clear E or D Key through Master Port command has been entered. The level on AUX_6-E/$\overline{D}$ determines whether the subsequently entered clear-key bytes are written into the E Key Register (E/$\overline{D}$ High) or into the D Key Register (E/$\overline{D}$ Low).

AUX_7-K/$\overline{D}$ and AUX_5-S/$\overline{S}$ are mutually exclusive control lines; when one goes active (High), the other must be and remain inactive (Low) until the first returns to an inactive state. In addition, both lines must be inactive (Low) whenever a transition occurs on C/$\overline{K}$ (entering or exiting Direct Control Mode).

AUX_6-E/$\overline{D}$ (Encrypt/$\overline{Decrypt}$, Input)

When AUX_5-S/$\overline{S}$ goes High, initiating a normal data ciphering operation, this input specifies whether the ciphering algorithm is to encrypt (E/$\overline{D}$ High) or decrypt (Low).

When AUX_7-K/$\overline{D}$ goes High, initiating entry of key bytes, the level on AUX_6-E/D specifies whether the bytes are to be written into the E Key Register (E/$\overline{D}$ High) or the D Key Register (E/$\overline{D}$ Low).

The AUX_6-E/$\overline{D}$ input is not latched internally, and must be held constant whenever one or more of AUX_5-S/$\overline{S}$, AUX_7-K/$\overline{D}$, AUX_2-BSY, or AUX_3-$\overline{CP}$ are active. Failure to maintain the proper level on AUX_6-E/$\overline{D}$ during loading or ciphering operations will result in scrambled data in the internal registers.

AUX_5-S/$\overline{S}$ (Start/$\overline{Stop}$, Input)

When this pin goes Low (Stop) the DCP will follow the sequence that would normally occur were a Stop command to be entered. Conversely, when this pin goes High, a sequence equivalent to a Start Encryption or Start Decryption command will be followed. At the time AUX_5-S/$\overline{S}$ goes High, the level on AUX_6-E/$\overline{D}$ (see above) selects either the Start Encryption or Start Decryption interpretation.

AUX_3-$\overline{CP}$ (Command Pending Output)

This active Low status output gives a hardware indication that the DCP is ready to accept input of key bytes following a Low-to-High transition on AUX_7-K/$\overline{D}$. AUX_3-$\overline{CP}$ is driven by the CP bit in the Status Register (see Register Description), such that when the CP bit is "1" (active), AUX_3-$\overline{CP}$ is Low.

AUX_2-$\overline{BSY}$ (Busy, Output)

This active Low status output gives a hardware indication that the ciphering algorithm is in operation. AUX_2-$\overline{BSY}$ is driven by the BSY bit in the Status Register, such that when the BSY bit is "1" (active), AUX_2-$\overline{BSY}$ is Low.

$AUX_{0,1,4}$ -Not used.

The Mode, Command, or Status Registers are not directly accessible in Direct Control Mode. A subset of commands can be executed by controlling pins of the Auxiliary Port as described above.

In most Direct Control Mode applications, the C/$\overline{K}$ input pin, which selects Multiplexed or Direct Control Mode, must be programmable. It allows the user to initialize the DCP in Multiplexed Control Mode, to choose a mode other than the default mode, to load the Master Key, to generate session keys, or to load the Initial Vectors. After doing this the device can be switched to Direct Control Mode by raising the level at the C/$\overline{K}$ input pin to High.

C/$\overline{K}$ can be tied High if the user wants the DCP to operate in the default mode (i.e. ECB, dual port configuration, Master Port handles clear text, and Slave Port handles encrypted text).

3.1.4. KEY AND DATA LOAD IN DIRECT CONTROL MODE

In Direct Control Mode, keys can only be entered through the Master Port. This is accomplished in the following manner:

- Hold AUX_6-E/$\overline{D}$ High when loading the encryption key or hold it Low when loading the decryption key.
- Keep AUX_5-S/$\overline{S}$ Low.
- Hold AUX_7-K/$\overline{D}$ High and issue eight write strobes at the Master Port as described in the Master Port section.

The levels of AUX_{5-7} should be held constant throughout the entire operation.

The data transfer is similar to the key load. AUX_6-E/$\overline{D}$ and the selected mode determine the data flow direction. In the default mode where the Master Port handles clear data while the Slave Port handles encrypted data, a High on AUX_6-E/$\overline{D}$ (encryption mode) defines the Master Port as an input port for the clear data and the Slave Port as an output port for the ciphered data. If AUX_6-E/$\overline{D}$ is switched to Low (decryption mode) the data flow direction is turned around. The Slave Port is now the input port for the encrypted data. The Master Port becomes the output port of the deciphered or clear data. A data ciphering session is set up as follows:

- Set AUX_6-E/$\overline{D}$ to the appropriate level.
- Keep AUX_7-K/$\overline{D}$ Low the entire session.
- Set AUX_5-S/$\overline{S}$ High to start the ciphering session.

3.2. REGISTERS

In Multiplexed Control Mode, five internal registers can be directly accessed:

- Command Register (write only)
- Status Register (read only)
- Mode Register (read/write)
- Input Register (write only)
- Output Register (read only)

In Direct Control Mode, only the Input and Output Registers are addressable through the Master Port. The register addresses are shown in Figure 3.4. The Input and Output Registers and the Command and Status Registers each have the same address. A read or write access determines which register is selected.

To gain access to any of these registers in Multiplexed Control Mode, execute the following sequence:

- Provide $\overline{MCS}$ and the register address.
- Provide address strobe.
- Read or write the addressed register by issuing a read or write strobe.

Command Register

Data written to the 8-bit, write-only Command Register through the Master Port is interpreted as an instruction. The commands and their hexadecimal representations are summarized in Figure 3.7. A detailed description of these commands is given in the section "Commands".

Status Register

The 8-bit, read-only Status Register (see Figure 3.5) has the same address as the Command Register. The status bits PAR, AFLG, SFLG, and MFLG indicate the status on the like-named output pins. Note, however, the status bits are active High, whereas the status pins are active Low. Additionally, in Direct Control Mode two pins of the Auxiliary Port reflect the flag bits CP and BUSY.

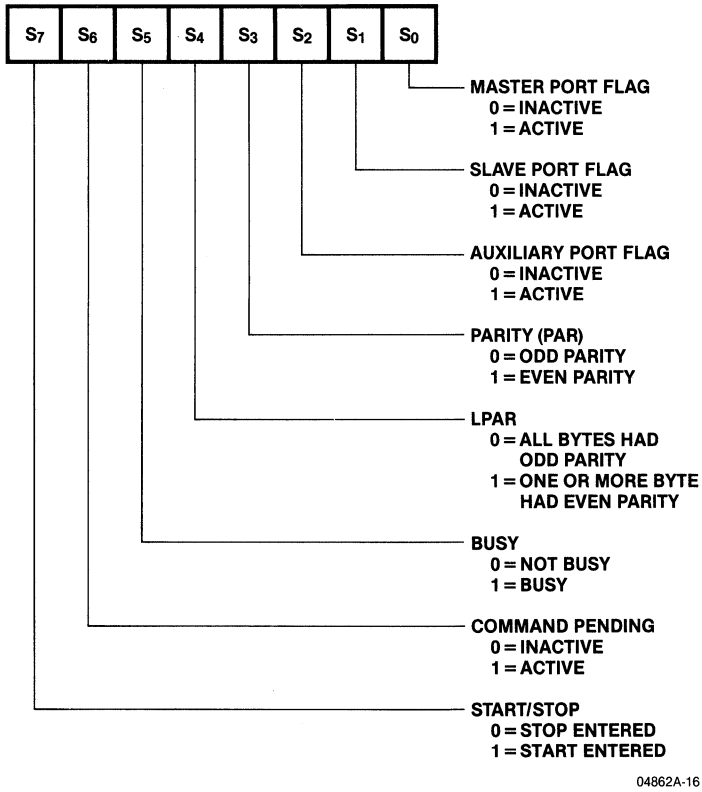


Figure 3.5. Status Register Bit Assignments

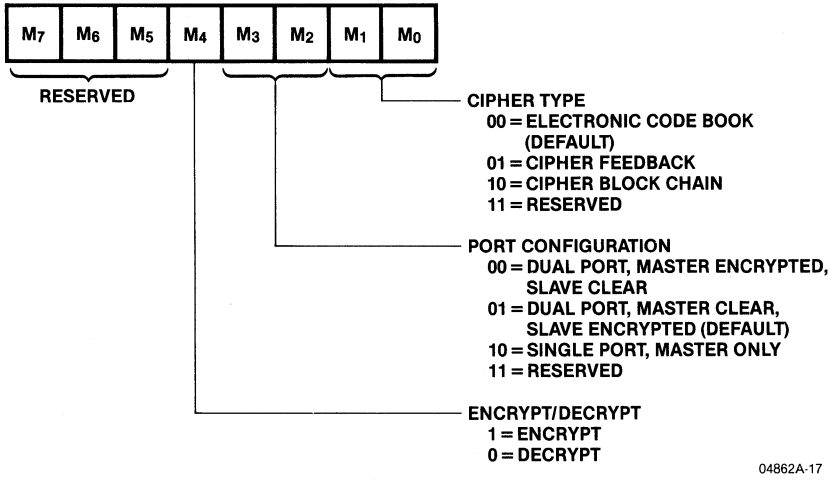


Figure 3.6. Mode Register Bit Assignments

The parity bit (PAR) indicates the parity of the most recently entered key byte. If this byte had even parity, the parity bit is set to signal a parity error.

The second parity bit (LPAR) stores the parity error. It is set if any one key byte had even parity since the last Reset or Load Key command.

The Busy bit will be a "1" whenever the ciphering algorithm unit is actively encrypting or decrypting data, either as a response to a command such as Load Encrypted Key (in which case the Command Pending bit will be a "1"), or in the ciphering of regular text (indicated by the Start/Stop bit being a "1"). The Busy bit will remain a "1", even after ciphering is complete, if the ciphered data cannot be transferred to the Output Register because it contains output from a previous ciphering cycle. Busy will be "Ø" at all other times, including the case where no ciphering is possible because no data has been written to the Input Register.

The Command Pending bit will be set to "1" by a command that requires the transfer of data to or from a non-addressable internal register, such as when writing key bytes to the E Key Register or reading bytes from the IVE Register. Thus, Command Pending will be set following all commands except the three Start commands, the Stop command and the Software Reset command. Command Pending will return to "Ø" after all eight bytes have been transferred following Load Clear, Read Clear or Read Encrypted commands; and after data has been transferred, decrypted and loaded into the desired register following Load Encrypted commands.

The Start/Stop bit is set to "1" when one of the Start commands is entered, and is reset to "Ø" whenever a reset occurs or when a command other than a Start is entered.

Mode Register

Bit Assignments in the 5-bit read/write register are shown in Figure 3.6. The Cipher Type bits (M_1, $M_Ø$) indicate to the DCP which ciphering algorithm is used. On reset, the Cipher Type defaults to Electronic Code Book.

Configuration bits (M_3, M_2) indicate which data ports are associated with the Input and Output Registers and flags. When these bits are set to the Single Port, Master Port-only configuration (M_3, $M_2=1Ø_B$) the Slave Port is disabled and no manipulation of Slave Port Chip Select ($\overline{SCS}$) or Data Strobe ($\overline{SDS}$) can result in data movement through the Slave Port; all data transfers are accomplished through the Master Port. Both $\overline{MFLG}$ and $\overline{SFLG}$ are used in this configuration; $\overline{MFLG}$ gives the status of the Input Register and $\overline{SFLG}$, the status of the Output Register.

When the Configuration Bits are set to one of the Dual Port configurations (M_3, $M_2=ØØ_B$ or $Ø1_B$), both the Master and Slave

Ports are available for input and output. When M_3, M_2=01_B (the default configuration), the Master Port handles clear data while the Slave Port handles encrypted data. Configuration M_3, M_2=00_B reverses this assignment. Actual data direction at any particular moment is controlled by the Encrypt/Decrypt bit.

The Encrypt/Decrypt bit (M_4) instructs the DCP algorithm processor to encrypt or decrypt the data from the Input Register using the ciphering method specified by the Cipher Type bits. The Encrypt/Decrypt bit also controls data flow within the DCP. For example, when the configuration bits are "01_B" (Dual Port, Master Clear, Slave encrypted) and the Encrypt/Decrypt bit is "1" (encrypt), clear data will flow into the DCP through the Master Port and encrypted data will flow out through the Slave Port. When the Encrypt/Decrypt bit is set to "0" (decrypt), data flow reverses.

Input Register

The 64-bit, write-only Input Register is organized to appear to the user as eight bytes of push-down storage. A status circuit monitors the number of bytes that have been stored. The register is considered empty when the data stored in it has been or is being processed; it is considered full when one byte of data has been entered in cipher feedback or when eight bytes of data have been entered in Electronic Code Book or Cipher Block Chain. If the user attempts to write data into the Input Register when it is full, the Input Register will disregard this attempt; no data in the register will be destroyed.

Output Register

The 64-bit, read-only Output Register is organized to appear to the user as eight bytes of pop-up storage. A status circuit detects the number of bytes stored in the Output Register. The register is considered empty when all the data stored in it has been read out. It is considered full if it contains one or more bytes of output data. If a user attempts to read data from the Output Register when it is empty, the buffers driving the output bus will remain in a three-state condition.

The following multibyte registers cannot be directly addressed, but are loaded or read in response to commands written to the Command Register. (See Commands.)

- Master Key Register (write only)
- Encryption Key Register (write only)
- Decryption Key Register (write only)
- Initial Vector for Encryption (read/write)
- Initial Vector for Decryption (read/write)

Master Key Register

The 56-bit Master Key Register can be loaded only with clear data through the Auxiliary Port. The load has to be preceded by the command "Load Clear M Key through Auxiliary Port". The Master Key is used to generate session keys. The correctness of entering the key can be verified by checking the LPAR bit of the Status Register.

Encryption and Decryption Key Register

The 56-bit Encryption Key or the 56-bit Decryption Key can be loaded through the Master Port or Auxiliary Port, in clear or in encrypted form. If the key is loaded in encrypted form, it is first routed to the Input Register, to be decrypted using the Master Key. It is then transferred to the appropriate key register.

Initial Vector Registers

Two 64-bit Initial Vector Registers are provided to store feedback values for CBC and CFB mode. Both registers can be loaded or read out through the Master Port in either clear or encrypted form. The E Key is used to decrypt the IV and the D Key to encrypt the IV utilizing the ECB mode. These registers have to be initialized only for CBC and CFB. The value is exclusive OR'ed with the first data block. Then the register is reloaded or modified. For detailed information refer to the section "Modes of Operation" in Chapter 2.2.

For test purposes these registers can be read out. Before reading the Initial Vectors, the Output Register must be flushed out by removing all data or by issuing a Reset. The IVs are eight bytes long and loaded one byte at a time with the most significant byte first. No parity check is done on these vectors.

3.3. COMMANDS

All operations of the DCP result from command inputs, which are entered in Multiplexed Control Mode by writing a command byte to the Command Register. Commands are entered in Direct Control Mode by raising and lowering the logic levels on the AUX_7-$K/\overline{D}$, AUX_6-E/D and AUX_5-$S/\overline{S}$ pins. Figure 3.7 shows all commands that may be given in Multiplexed Control Mode. Figure 3.8 shows that subset executable in Direct Control Mode.

Hex Code	Command
90	Load Clear M Key through Auxiliary Port
91	Load Clear E Key through Auxiliary Port
92	Load Clear D Key through Auxiliary Port
11	Load Clear E Key through Master Port
12	Load Clear D Key through Master Port
B1	Load Encrypted E Key through Auxiliary Port
B2	Load Encrypted D Key through Auxiliary Port
31	Load Encrypted E Key through Master Port
32	Load Encrypted D Key through Master Port
85	Load Clear IVE through Master Port
84	Load Clear IVD through Master Port
A5	Load Encrypted IVE through Master Port
A4	Load Encrypted IVD through Master Port
8D	Read Clear IVE through Master Port
8C	Read Clear IVD through Master Port
A9	Read Encrypted IVE through Master Port
A8	Read Encrypted IVD through Master Port
39	Encrypt with Master Key
41	Start Encryption
40	Start Decryption
C0	Start
E0	Stop
00	Software Reset

04862A-18

Figure 3.7. Command Codes in Multiplexed Control Mode

Pins				Command Initiated
C/$\overline{K}$	AUX_7-K/$\overline{D}$	AUX_6-E/$\overline{D}$	AUX_5-S/$\overline{S}$	
H	L	L	↑	Start Decryption
H	L	H	↑	Start Encryption
H	L	X	↓	Stop
H	↑	L	L	Load D Key Clear through Master Port
H	↑	H	L	Load E Key Clear through Master Port
H	↓	X	L	End Load Key Command
H	H	X	H	Not Allowed
L	Data	Data	Data	AUX Pins Become Key-Byte Inputs

04862A-19

Figure 3.8. Implicit Command Sequences in Direct Control Mode

Load Clear M Key Through Auxiliary Port (90_H)
Load Clear E Key Through Auxiliary Port (91_H)
Load Clear D Key Through Auxiliary Port (92_H)

These commands override the data flow specifications set in the Mode Register and cause the Master (M), Encrypt (E), or Decrypt (D) Key Register to be loaded with eight bytes written to the Auxiliary Port. After the Load command is written to the Command Register, the Auxiliary Port Flag ($\overline{AFLG}$) will go active (Low), and the corresponding bit in the Status Register (S_2) will go to "1", indicating that the device is able to accept key bytes at the Auxiliary Port pins. Additionally, the Command Pending bit (S_6) will go to "1" during the entire loading process.

Each byte is written by placing an active Low signal on the Auxiliary Port Strobe ($\overline{ASTB}$) once data has been set up on the Auxiliary Port pins. The actual write process occurs on the rising (trailing) edge of $\overline{ASTB}$.

The Auxiliary Port Flag ($\overline{AFLG}$) will go inactive immediately after the eighth strobe goes active (Low), but, the Command Pending bit (S_6) will remain "1" for several more clock cycles, until the key loading process is completed. All key bytes are checked for correct (odd) parity as they are entered (see Parity Checking).

Load Clear E Key Through Master Port (11_H)
Load Clear D Key Through Master Port (12_H)

These commands are available in both Multiplexed Control and Direct Control Modes. They override the data flow specifications set in the Mode Register and attach the Master Port inputs to the Encrypt (E) or Decrypt (D) Key Register, as appropriate, until eight key bytes have been written. In Multiplexed Control Mode, the command is initiated by writing the Load command to the Command Register. In Direct Control Mode, the command is initiated by raising the AUX_7-K/$\overline{D}$ control input while the AUX_5-S/$\overline{S}$ input is Low. In this later case, the level on AUX_6-E/$\overline{D}$ determines which key register is written (High=E-Key Register, Low=D-Key Register).

Once the command has been recognized, the Command Pending bit (S_6 in the Status Register) will go to "1", and in Direct Control Mode AUX_3-$\overline{CP}$ will go active (Low), indicating that key entry may proceed. The host system then writes exactly eight bytes to the Master Port (at the Input Register address in Multiplexed Control Mode). When the key register has been loaded, Command Pending will return to "Ø", and in Direct Control Mode the AUX_3-$\overline{CP}$ output will go inactive, indicating that the DCP can accept the next command.

Load Encrypted E Key Through Auxiliary Port ($B1_H$)
Load Encrypted D Key Through Auxiliary Port ($B2_H$)

Execution of these commands (in Multiplexed Control Mode only) is similar to the Load Clear E (D) Key Through Auxiliary Port, except that key bytes are first decrypted using the Electronic Code Book algorithm and the Master (M) key, and then loaded into the appropriate key register, after having passed through the parity check logic (see Parity Checking).

The Command Pending bit (S_6) will be "1" during the entire decrypt-and-load operation. In addition, the Busy bit (S_5) will be "1" during the actual decryption process.

Load Encrypted E Key Through Master Port (31_H)
Load Encrypted D Key Through Master Port (32_H)

These commands (in Multiplexed Control Mode only) are similar in effect to the Load Clear E (D) Key Through Master Port, except that key bytes are initially decrypted using the Electronic Code Book algorithm and the Master (M) Key, and then loaded byte-by-byte into the target key register, after having passed through the parity check logic (see Parity Checking).

The Command Pending bit (S_6) will be "1" during the entire decrypt-and-load operation. In addition, the Busy bit (S_5) will be "1" during the actual decryption process.

Load Clear IVE Register Through Master Port (85_H)
Load Clear IVD Register Through Master Port (84_H)

These commands (in Multiplexed Control Mode only) are almost identical to Load Clear E (or D) Key Through Master Port except that the data written to the Input Register address is routed to the Encryption Initial Vector (IVE) or Decryption Initial Vector (IVD) Register instead of a key register, and no parity checking occurs. Command Pending (S_6) is a "1" during the entire loading process.

Load Encrypted IVE Register Through Master Port ($A5_H$)
Load Encrypted IVD Register Through Master Port ($A4_H$)

These commands are analogous to the Load Encrypted E (or D) Key Through Master Port commands. The data flow specifications set in the Mode Register are overridden and the eight vector bytes are decrypted using the Decryption (D) Key and the Electronic Code Book algorithm. The resulting clear vector bytes are loaded into the target Initial Vector Register, and no parity checking occurs. The Busy bit (S_5) does not go to "1" during the decryption process, but Command Pending (S_6) will be "1" during the entire decryption-and-load operation.

Read Clear IVE Register Through Master Port ($8D_H$)
Read Clear IVD Register Through Master Port ($8C_H$)

The effect of these commands (in Multiplexed Control Mode only) is to override the data flow specifications set in the Mode Register and to connect the appropriate Initial Vector Register to the Master Port at the Output Register address. In this state, each IV Register appears as eight bytes of FIFO storage. The first byte of data will be available 6 clocks after the loading the Command Register. The Command Pending bit will be set to "1" and will remain a "1" until sometime after the eighth byte is read out. The host system has the responsibility to read out exactly eight bytes.

Read Encrypted IVE Register Through Master Port ($A9_H$)
Read Encrypted IVD Register Through Master Port ($A8_H$)

The effect of these commands (in Multiplexed Control Mode only) is to override the specifications set in the Mode Register and to encrypt the contents of the specified Initial Vector Register using the Electronic Code Book algorithm and the Encrypt (E) Key. The resulting cipher text is placed in the Output Register, from which it can be read out as eight bytes through the Master Port. During the actual encryption process the Busy bit (S_5) will be "1". When Busy goes to "Ø", the encrypted vector bytes are ready to be read out. Command Pending (S_6) will be "1" during the entire encryption-and-output process, and will go to "Ø" when the eighth byte is read out. The host system is responsible for reading out exactly eight bytes.

Encrypt with Master (M) Key (39_H)

This command, in Multiplexed Control Mode only, overrides the data flow specifications set in the Mode Register and causes the DCP to accept eight bytes from the Master Port, written to the Input Register. When eight bytes have been received, the DCP encrypts the input using the Master (M) Key. The encrypted data is loaded into the Output Register, where it may be read out through the Master Port. The Command Pending (S_6) and Busy (S_5) bits are used to sense the three phases of this operation. Command Pending goes to "1" as soon as the Input Register can accept data. When exactly eight bytes have been entered, the Busy bit will go to "1" until the encryption process is complete.

When Busy goes to "Ø", the encrypted data is available to be read out. Command Pending will return to "Ø" when the eighth byte has been read.

Start Encryption (41_H)
Start Decryption (40_H)
Start ($C0_H$)

The three "Start" commands begin normal data ciphering by setting the Start/Stop bit (S_7) in the Status Register to "1". The Start Encryption and Start Decryption commands explicitly specify the

ciphering direction by forcing the Encrypt/Decrypt bit (M_4) in the Mode Register to "1" or "Ø", respectively, whereas Start uses the current state of the Encrypt/Decrypt bit, as specified in a previous Mode Register load.

When a Start command has been entered, the Port Status Flag ($\overline{\text{MFLG}}$ or $\overline{\text{SFLG}}$) associated with the Input Register will become active (Low), indicating that data may be written to the Input Register to begin ciphering.

In Direct Control Mode, the Start command is issued by raising the level on the AUX_5-S/$\overline{S}$ input (see Figure 3.8). The ciphering direction is specified by the level on AUX_6-E/$\overline{D}$. If AUX_6-E/$\overline{D}$ is High when AUX_5-S/$\overline{S}$ goes High, the command is Start Encryption. If AUX_6-E/$\overline{D}$ is Low, it is Start Decryption.

Stop ($E\emptyset_H$)

The Stop command clears the Start/Stop bit (S_7) in the Status Register. This causes the input flag ($\overline{\text{MFLG}}$ or $\overline{\text{SFLG}}$) to become inactive and inhibits the loading of any further input into the algorithm unit. If ciphering is in progress (Busy bit (S_5) is "1" or AUX_2-$\overline{\text{BSY}}$ is active), the ciphering process is terminated. Any data in the Output Register will remain accessible (except in CFB mode). In CFB mode, the last byte of data must be read out before issuing the Stop command.

In Direct Control Mode, the Stop command is implied when the signal level on the AUX_5-S/$\overline{S}$ input goes from High to Low (see Figure 3.8).

Software Reset ($\emptyset\emptyset_H$)

This command has the same effect as a hardware reset; it forces the DCP back to its default configuration, and all processing flags go inactive. In the default configuration the Mode Register is set to Electronic Code Book cipher type, and Dual Port Configuration with Master Port clear, Slave Port encrypted.

3.4. PARITY CHECKING OF KEYS

To enhance system security, the DCP provides no way to read back the keys. A parity check on each byte of key input guarantees the user that the key is entered correctly.

Key bytes are considered to contain seven bits of key information and one parity bit. The parity checking circuit is enabled whenever a byte is written to one of three key registers. The output of the parity detection circuit is connected to pin $\overline{\text{PAR}}$ and the state of this pin is reflected in Status Register bit PAR (S_3). Status Register bit PAR goes to "1" whenever a byte with even parity (an even number of "1s") is detected. In addition to the PAR bit, the Status Register has a Latched Parity Bit (LPAR,

S_4) which is set to "1" whenever the Status Register PAR bit goes to "1". Once set, the LPAR bit is not cleared until a reset occurs or a new Load Key command is issued.

When an encrypted key is entered, the parity detect logic operates only after the decrypted key is available. The encrypted data is not checked for parity. The $\overline{PAR}$ signal will reflect the state of the decrypted bytes on a byte-to-byte basis, as they are clocked through the parity check logic on their way to the Key Register. Thus, the time $\overline{PAR}$ indicates the status of a byte of decrypted key data may be as short as four clock cycles. The LPAR bit in the Status Register will indicate if any erroneous bytes of key were entered.

3.5. INITIALIZATION

After power up the DCP must be reset in one of several possible ways. Under some conditions the DCP is reset automatically (e.g., aborting a command).

Hardware Reset:

Am9518/AmZ8068: $\overline{MAS}$ and $\overline{MDS}$ are Low simultaneously

Am9568: $\overline{MRD}$ and $\overline{MWR}$ are Low simultaneously

Figures 3.9 and 3.10 show the reset timings. Parameter 5 specifies the minimum strobe widths; parameter 6 the hold time to the rising edge of the clock. The strobe width may be wider than specified by parameter 5. In this case the strobe has to meet only the set-up time (parameter 5 minus parameter 6) and hold time (parameter 6) to at least one rising edge of the clock. This means, for strobes wider than one clock period, the trailing edge does not have to be synchronized to the rising edge of the clock.

Software Reset:

The DCP can be reset by software in three ways:

- Issue the Software Reset command (00_H).
- Load the Mode Register.
- The DCP is reset by aborting any command, i.e., by entering any command before the previous command is completely executed or terminated. The abort does not destroy the Mode Register; it only resets the flags.

A reset sets the Mode Register to the default value "14_H". It selects encryption, ECB mode, and dual port configuration with Master Port clear data and Slave Port encrypted data. The reserved bits of the Mode Register are read back as "1s".

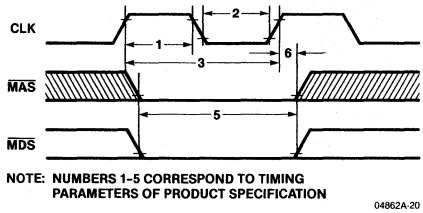


Figure 3.9. Am9518/AmZ8068 Clock and Reset

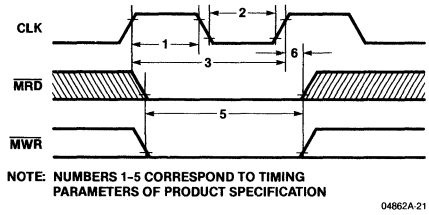


Figure 3.10. Am9568 Clock and Reset

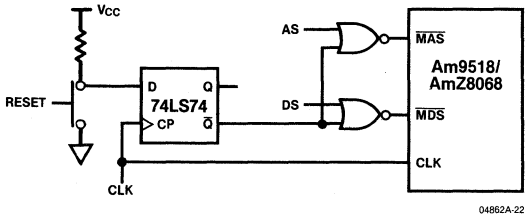


Figure 3.11. Am9518/AmZ8068 Reset Logic

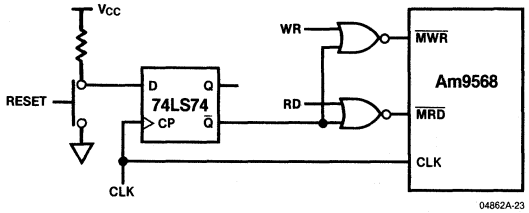


Figure 3.12. Am9568 Reset Logic

Figures 3.11 and 3.12 show hardware reset circuits which guarantee that the strobes are synchronous to the rising edge of the clock.

3.6. MULTIPLEXED CONTROL MODE

This chapter describes in detail which steps must be executed to operate the DCP using ECB, CBC, and CFB in Multiplexed Control Mode. All the program sequences are set up for a Master Port-only configuration. The device at the Master Port handles both input and output data. To set the DCP up for pipelined operation, strobe in additional data after initializing the device and before entering the data transfer loop (see Chapter 3.9).

For dual port configuration, the same basic program sequence can be executed, modifying only the data transfer session. Now the CPU handles either input or output data, so one transfer task must be removed from the command sequence. The high-speed peripheral connected to the Slave Port executes the remaining task. Data can be put in or read back concurrently.

3.6.1. ECB OPERATION

Figure 3.13 shows the program sequence.

Step 1: A hardware or software reset clears all Status Register flags and sets the Mode Register to the default condition.

Step 2: The Mode Register is loaded via the Master Port. The loaded value determines the port configuration, the mode of operation (ECB, CBC, or CFB) and encryption or decryption. For example, to enter clear data through the Slave Port and remove encrypted data from the Master Port using ECB mode for encryption, the Mode Register is loaded with 10_H (see Chapter 3.2, "Mode Register").

Step 3: The clear encryption or decryption key can be loaded through either the Master Port or the Auxiliary Port. After entering the appropriate command, the Command Pending bit of the Status Register becomes active (High) until the entire 8-byte key is entered with the most significant byte first.

Step 3A: Step 3A and 3B can be performed as an alternative to Step 3. In these two steps, the keys are loaded in encrypted form. The Master Key Register has to be loaded first for decrypting encrypted keys. The appropriate command is "Load M Key Through Auxiliary Port" (90_H). When this command is entered, the Auxiliary Flag in the Status Register goes active High and the $\overline{AFLG}$ output pin goes Low. The DCP expects data input through the

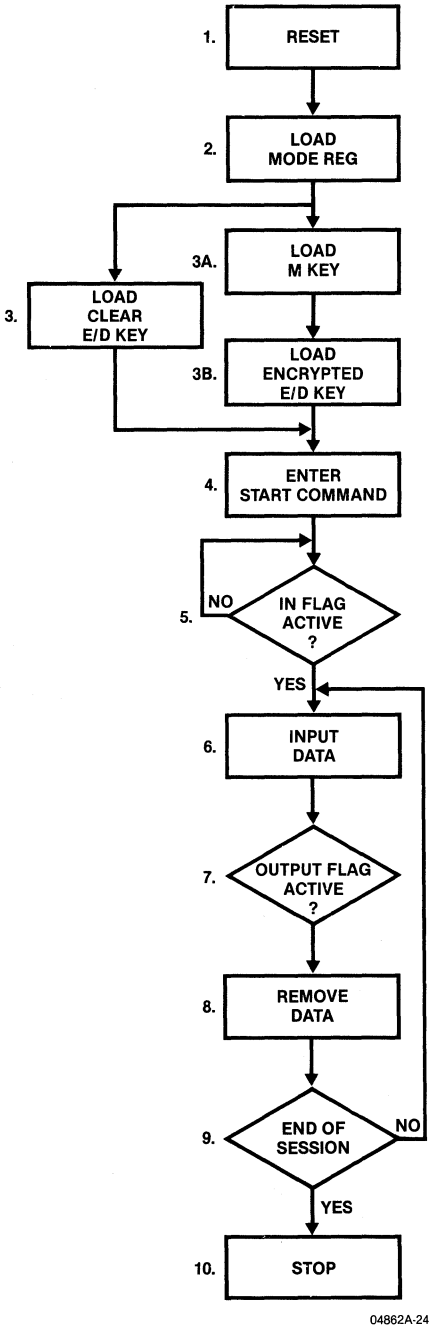


Figure 3.13. ECB Operation Flow Chart

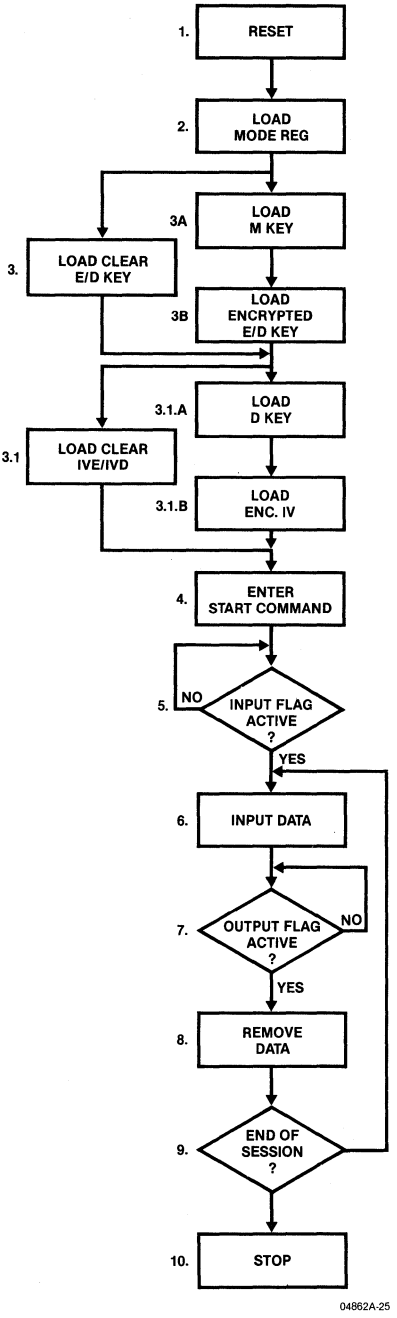


Figure 3.14. CBC Operation Flow Chart

Auxiliary Port. The Master Key is entered by strobing in eight bytes, one byte per Auxiliary Strobe (ASTB), most significant byte first.

Step 3B: The encrypted E or D Key can be loaded through the Master or Auxiliary Port. Chapter 3.3 lists the commands.

Step 4: The DCP recognizes three start commands: "Start Encryption", "Start Decryption" and "Start". The first commands set or reset the Encryption/Decryption bit of the Mode Register. If the "Start" command is issued, the Mode Register stays unchanged and the DCP is ready to process data according to the existing Mode Register bit configuration.

Step 5: After entering a Start command, the DCP indicates readiness for data input by activating the Input Register flag. Data then can be entered through the assigned input port.

NOTE: Ports are assigned on a Clear or Encrypted text basis. In dual port configuration, a change from encryption to decryption reverses the data flow direction. The ports are reassigned; the former input port becomes now an output port and vice versa. This reflects the logical situation in most applications. A good example is a hard disk application: For data security the data is stored in encrypted form on the hard disk. When writing to the disk, the data is encrypted by flowing through the DCP to the disk controller. When reading back, the DCP is programmed for decryption mode, and the data flows in the reverse direction from the disk controller to the system memory.

Two flags are associated with the data registers, the MFLG and the SFLG. For flag description see Chapter 3.1. These flags can be sensed by software or hardware. The CPU can monitor the bits of the Status Register by software; the two output pins can drive a Ready/Wait or DMA Request logic. Note that the Status Register bits are active High, whereas the flag output pins are active Low.

Step 6: Whenever the input flag is active, the DCP is ready to accept data. Data is transferred to the 64-bit Input Register one byte at a time, most significant byte first. When the Input Register is full (i.e., all eight bytes of data are entered) the input flag becomes inactive and the data is transferred via the internal bus to the algorithm unit.

Step 7: Whenever the output flag becomes active, data can be removed from the Output Register.

Step 8: Data is removed from the output port one byte at a time with the most significant byte first. The output flag becomes inactive when the eighth byte is removed, indicating that the transfer is complete.

Step 9: Loop through Step 5 through 8 until the ciphering session should be terminated.

Step 10: The session is terminated by issuing the "Stop" command. After termination, all remaining processed data will be available at the output port until the DCP is reset. Thus the "Stop" command can be issued after transferring the last input block. When all data is removed, all flag bits of the Status Register are inactive (00_H). To resume the ciphering session with the same parameters, issue a Start command as in Step 4 and proceed.

Before restart, any data from the previous session must be removed or it will be lost.

3.6.2. CBC OPERATION

A flow chart of CBC Operation in Multiplexed Control Mode is given in Figure 3.14. The flow chart of Cipher Block Chaining is very similar to ECB operation except that the IV Register must be loaded. The Initial Vector can be entered in clear (Step 3.1) or encrypted form (Step 3.1A and 3.1B). Listed below are those steps which differ from the ECB instruction sequence:

Step 3.1: Issue "Load Clear IV through Master Port" command and strobe in 8 bytes of IV, most significant byte first. The Initial Vector can only be loaded through the Master Port to the address of the Input Register. After the command is issued the Command Pending bit in the Status Register becomes active for the following IV transfer.

Step 3.1A: If the Initial Vector is entered in encrypted form, the vector is decrypted utilizing the D-Key before being loaded in the appropriate register. If the D-Key is not entered in Step 3, it must be entered now.

Step 3.1B: Issue "Load Encrypted IV through Master Port" command and strobe 8 bytes of encrypted IV into the Input Register, most significant byte first. The DCP then decrypts this Initial Vector using the D-Key in ECB mode, and loads it into the IV Register. The bits of the Mode Register are not affected. This sequence works for entering the IV for encryption (IVE) and decryption (IVD).

3.6.3. CFB OPERATION

The flow chart for the instruction sequence in CFB mode is very similar to the CBC mode. The DCP is programmable to execute

single-byte CFB Operation. In CFB, the Input and Output Registers can hold only one byte each.

The IV is ciphered by the algorithm unit. The result is then EXORed with the input byte which is treated as the most significant byte. The EXOR result is loaded into the Output Register to be read out by the CPU and is also shifted into the current IV Register. The lower seven bytes of the result block are discarded (see Chapter 2.2.).

The Output Register must be emptied in CFB mode before issuing a "Stop" command. The session can be resumed after stop by issuing "Start".

If the user has to stop in the middle of a data block input (ECB or CBC) operation in Multiplexed Control Mode, the following instruction sequence should be used to avoid erroneous data:

- Issue "Stop" command.
- Read all output data available.
- Reload the Mode Register.
- Issue "Start" command.
- Check for input flag active then resume data input.

3.7. DIRECT CONTROL MODE

The DCP operates in Direct Control Mode when the C/$\overline{K}$ input pin is High. The commands are issued by controlling the pins of the Auxiliary Port (see Chapter 3.1). The Mode Register cannot be accessed in Direct Control Mode.

The state of the E/$\overline{D}$ and K/$\overline{D}$ pins should be held constant throughout the entire loading process. The state of S/$\overline{S}$ must be held constant throughout the entire data ciphering session.

3.7.1. ECB OPERATION

A flowchart of ECB operation in Direct Control Mode is shown in Figure 3.15. An explanation of each step is given below:

Step 1: It is advisable to have the C/$\overline{K}$ pin programmable if the DCP is intended to operate in Direct Control Mode. C/$\overline{K}$ must be pulled Low (Multiplexed Control Mode) to access the Mode and Master Key Register in the initialization phase.

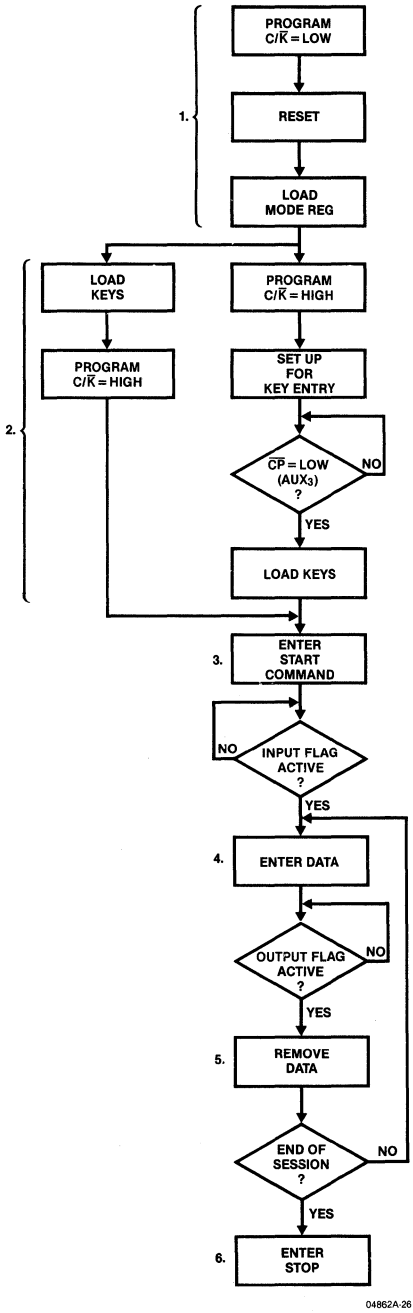


Figure 3.15. Direct Control Mode ECB Operation

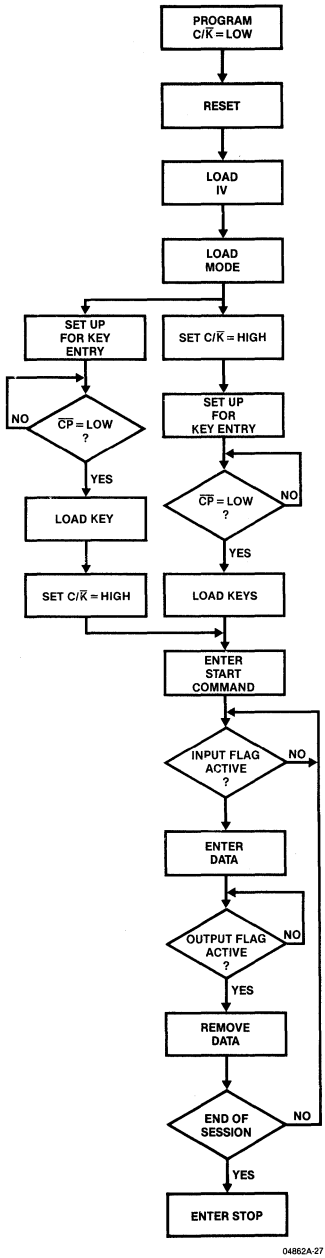


Figure 3.16. Direct Control Mode CBC/CFB Operation

C/$\overline{K}$ can be tied High permanently if the application requirement is the same as the default condition of the DCP. In the default condition, the Mode Register is set up for ECB encryption with Master Port assigned to clear data and Slave Port assigned to encrypted data. No session keys can be generated; only clear keys can be entered. The default condition may be achieved by a hardware reset (applying a Low to $\overline{MAS}$ and $\overline{MDS}$ (Am9518/AmZ8068) or $\overline{MRD}$ and $\overline{MWR}$ (Am9568) simultaneously).

If the default mode is not practical, switch to Multiplexed Control Mode and load the Mode Register. If necessary the Master Key Register can be loaded and session keys may be generated at this time. Then switch back to Direct Control Mode.

Step 2: A High on the K/$\overline{D}$ pin of the Auxiliary Port sets up the DCP for key entry. (S/$\overline{S}$ stays Low for the entire key load process. A High at the E/$\overline{D}$ pin selects the E-Key load, a Low selects the D-Key load. The DCP responds by activating the $\overline{CP}$ output pin. As soon as $\overline{CP}$ becomes active, keys can be strobed into the Master Port by providing data write strobes. $\overline{MCS}$ must be Low. The control lines of the Auxiliary Port should be held steady throughout the entire load process.

Step 3: A "Start" command is entered by raising the S/$\overline{S}$ line. The level at E/$\overline{D}$ selects encryption (High) or decryption mode (Low). K/$\overline{D}$ has to be Low throughout the ciphering session. The DCP responds to the start command by activating the input port flag. S/$\overline{S}$ must be held steady during the ciphering session. For flag assignment information refer to Chapter 3.1.

Step 4: Whenever the input flag is active, data can be entered through the Master or Slave Port depending on the selected mode. To achieve the highest throughput, follow the notes given in Chapter 3.9 (pipelining).

Step 5: When the DCP has processed the data, the output flag will become active. Data may be removed from the output port when the flag is active.

Step 6: At the end of the ciphering session, issue a "Stop" command by pulling S/$\overline{S}$ Low.

3.7.2. CBC AND CFB OPERATION

The instruction sequence to perform CBC or CFB operation in Direct Control Mode is similar to ECB operation. In these operation modes the C/$\overline{K}$ pin must be programmble, because the IV needed for CBC and CFB can only be loaded in Multiplexed Control Mode.

Loading the encryption and decryption keys can be performed when C/$\overline{K}$ is Low (Multiplexed Control Mode) or High (Direct Control Mode).

Figure 3.16 shows a flow chart. Do not issue a stop command if Busy (BSY) or Command Pending (CP) are active.

In CFB operation, all output data must be removed from the Output Register before a stop command is entered. In this mode the user is limited to one session at a time. The DCP must be reinitialized before resuming the ciphering session. The steps are shown below:

- Switch to Multiplexed Control Mode (C/$\overline{K}$=Low).
- Reload the Mode Register to previous configuration.
- Switch back to Direct Control Mode (C/$\overline{K}$=High).
- Issue Start command.
- Check for input flag active, then resume data input.

If the DCP is stopped in the middle of a data block input, the following steps must be performed to avoid erroneous data and to resume operation:

- Issue stop command.
- Read all available output data.
- Switch to Multiplexed Control Mode (C/$\overline{K}$=Low).
- Reload Mode Register.
- Switch back to Direct Control Mode (C/$\overline{K}$=High).
- Issue Start command.
- Check for input flag active, then resume data input.

If the data error is detected before input to the DCP, an error signal may be generated from the error detection logic to disable the input port data strobes. In this case the user does not need to switch out of Direct Control Mode. The input can be continued by enabling the input data strobes when correct data is available.

If the input data strobe is of the same frequency as the clock input and the user has to stop in the middle (less than 8 bytes) of an input block load, it is not possible to disable further data strobes by de-selecting the input port (Chip Select=High).

3.8. OUTPUT FEEDBACK (OFB) AND ONE-BIT CIPHER FEEDBACK (CFB)

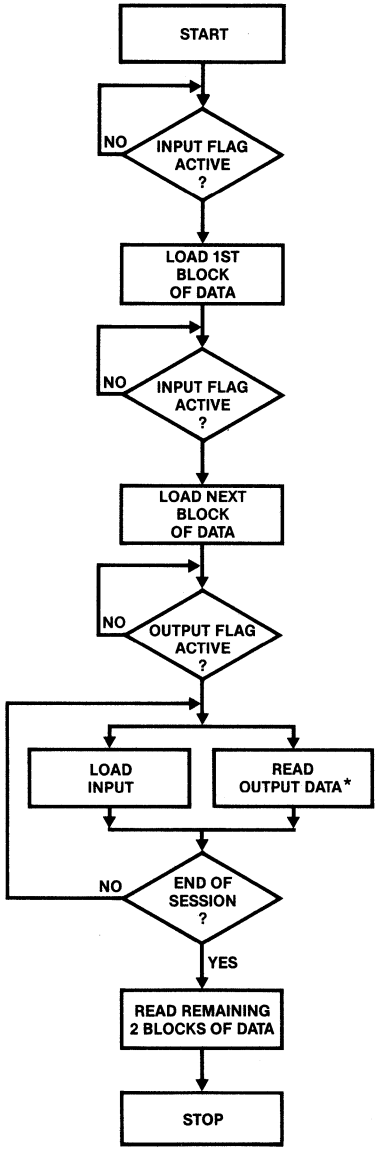
Only the three operation modes that are implemented in the DCP have been discussed in the preceding chapters. Two other types of data ciphering modes recommended by the National Bureau of Standards are OFB and one-bit CFB. These modes of operation are explained in Chapter 2.2.

The DCP can achieve 64-bit Output Feedback when the EXOR function is done by software. The DCP operates as a 64-bit pseudo random number generator. Figure 2.10 shows the data flow in this mode. The instruction sequence is:

- Set up DCP for CBC operation.
- Load Keys.
- Load IV with 64-bit initial value.
- Issue "Start Encryption".
- Load Input Register with zeros (00_H).
- Read Output Register.
- EXOR DCP result vector with 64-bit data block by software to get the 64-bit encrypted block (ciphered text).
- Jump to "Load Input Register" instruction.

One-bit CFB may be performed by the DCP with supporting software. Each 64-bit cipher process generates one bit output information. The user must be aware that this implementation of one-bit CFB can be used in fairly low-speed applications only. The DCP is set up for ECB mode. The EXOR and the SHIFT functions are executed in software. The instruction sequence is given below:

- Set up DCP for ECB.
- Load Keys.
- Issue "Start Encryption".
- Load 64-bit Input Register with Initial Vector.
- Read 64-bit output.



*DATA INPUT AND OUTPUT CONCURRENTLY

04862A-28

Figure 3.17. Operation Flow Diagram for Pipelining

- Take the most significant bit and EXOR it with the clear text. The output of the EXOR function is the ciphered text.

- Also left-shift this bit into the Initial Vector for the next cycle.

- Continue loading the Input Register with the Initial Vector.

3.9. THROUGHPUT

The highly pipelined architecture of the DCP allows simultaneous read, ciphering and write operation. For maximum throughput, the DCP must be programmed for dual port configuration. One port is the input port, the other is the output port. For single port configuration, the throughput is cut in half.

Figure 3.18 shows detailed timing of the ciphering of one 64-bit block in ECB or CBC. The input process starts at clock Ø. It takes 8 clock periods to strobe in the entire block. One data strobe is issued for each clock period. Five clock cycles are needed to update the flags and transfer the input block from the Input Register to the algorithm unit. The algorithm unit starts ciphering concurrently with the transfer. After updating the flags, another input block may be entered. The block is ciphered 18 clocks after loading the last byte. Transfer of the ciphered block to the Output Register and transfer of the next input block to the algorithm unit can be performed in parallel (see Pipelining Scheme A and B). The entire procedure of ciphering one block takes 39 clock periods. Because parts of this procedure can be overlayed, the DCP can process one block every 18 clocks.

Pipelining

Figure 3.17 shows a flow chart of the data entry and removal sequence for dual port configuration. After initialization, two data blocks are strobed into the device to fill the Output Register and the algorithm unit. Then blocks are strobed in and out concurrently. When terminating the session, the device must be emptied by reading out two more blocks.

The DCP can also be operated in pipelined mode when in single port configuration. After initialization, one block of data is strobed into the device. Then, in a loop, one block is strobed in and one block is read out. The block strobed in before entering the loop is ciphered concurrently with the input of the second block. This guarantees that the user need not wait for the algorithm to perform encryption. The Master Port can be switched between input and outputs without Waits.

Pipelining Scheme A (Figure 3.19) shows how to cipher a set of blocks in minimum time. The total time is (n + 1) * 18 + 3 clock periods where "n" is the number of blocks. Pipelining Scheme B (Figure 3.2Ø) is slightly modified compared to Scheme A. The

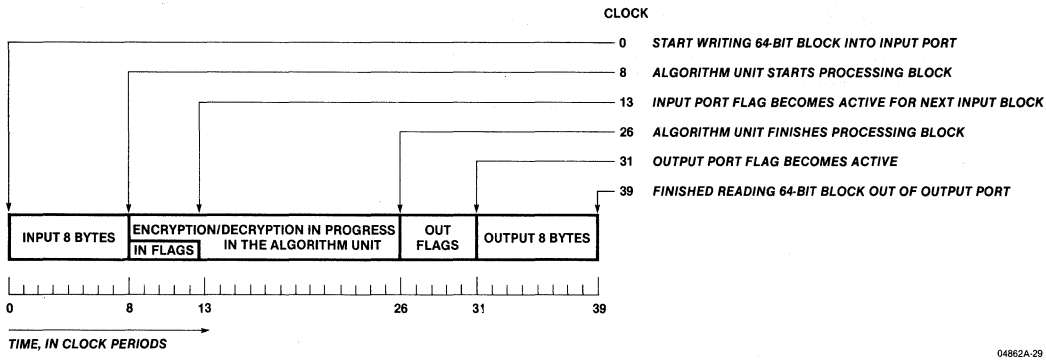


Figure 3.18. Detailed Timing of 1 Block

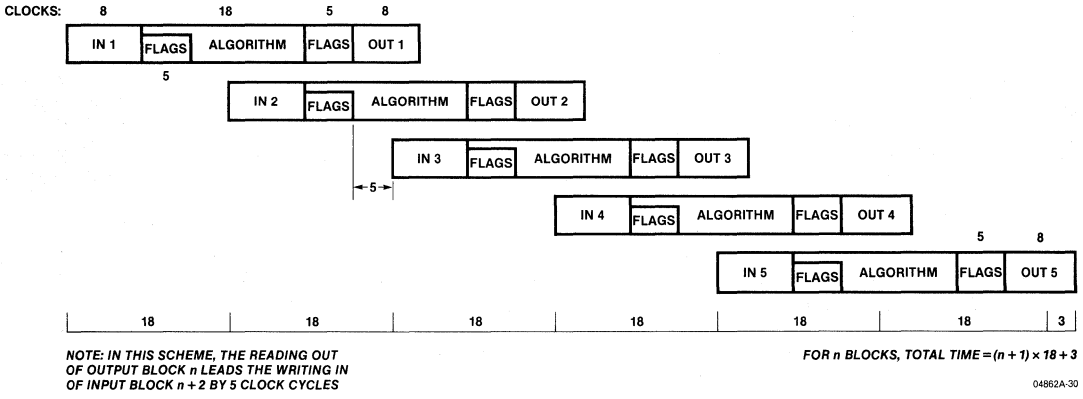


Figure 3.19. Pipelining Scheme A. Minimum Timing Operation

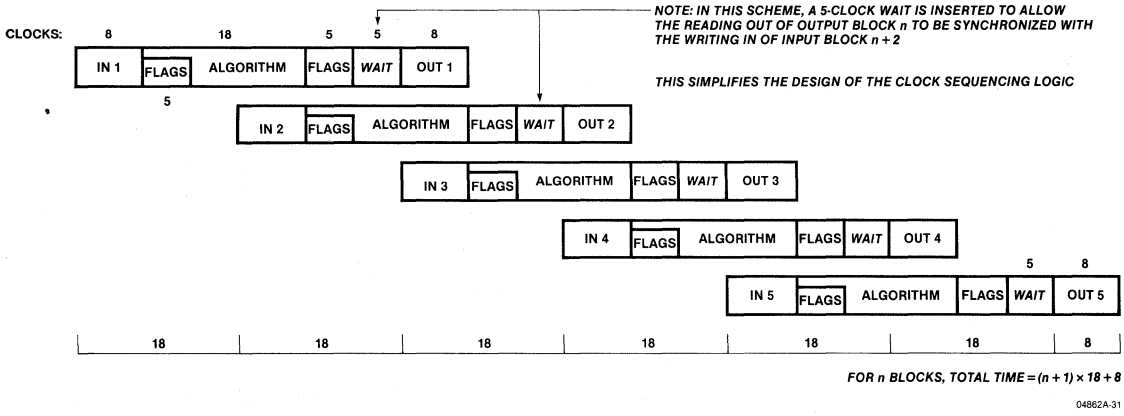


Figure 3.20 Pipeline Scheme B: Synchronized Port Operation

total time is slightly longer. It takes (n + 1) * 18 + 8 clock periods to cipher "n" blocks. But it has the advantage that data is put in and removed simultaneously. One signal may strobe data in and out. The interface hardware might be simpler.

To get the maximum throughput, block transfers must be executed in the 13-clock time slot between the update of flags. The examples in Figure 3.19 and 3.20 assume a transfer time of eight clock periods.

Only Direct Control Mode designs using high-speed control logic can satisfy this requirement. Chapter 4.12 "High Speed Serial Data Ciphering in Network Systems" shows such a design. All other application interfaces drive the DCP in Multiplexed Control Mode. The data transfer capabilities of most microprocessor systems are lower than required by the DCP. Even a design with high speed DMA controller is not able to transfer 8 bytes of data in 8 clock cycles.

When the system timing constrains the ciphering speed, this problem can be solved by putting a FIFO buffer between the system bus and the DCP. The system can thus operate asynchronously while the DCP operates at its optimum clock rate. The FIFO buffer also compensates for the time when no data can be transferred while the DCP updates flags.

Under ideal circumstances the throughput can be calculated as:

$$T = (f * 8) / 18$$

T = throughput
f = clock rate

Am9518:

$$T = (3\ MHz * 8) / 18 = 1.33\ MByte/s$$

AmZ8068:

$$T = (4\ MHz * 8) / 18 = 1.78\ MByte/s$$

Am9568:

To meet the minimum High and Low times of the read and write strobes, they cannot be issued every clock when operating at the maximum clock rate. The clock rate must be reduced to 3.33 MHz to have 300 ns strobe periods or strobes must be issued every other clock period. The throughput for both cases is determined below.

$$T = (4\ MHz * 8) / (2 * 8 + 5) = 1.52\ MHz$$

$$T = (3.33\ MHz * 8) / 18 = 1.48\ MHz$$

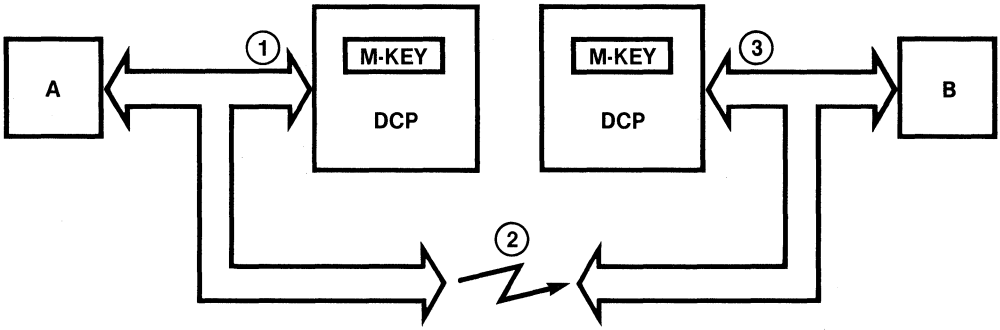


Figure 3.21 Key Transfer

3.10. KEY TRANSFER VIA THE COMMUNICATION LINK

The system security can be enhanced by changing keys frequently. These periodically changed keys are called session keys. In order to update the DCP with the new session keys the keys have to be distributed. A convenient way to distribute keys is to use an already existing communication link between the DCPs. The system designer has to make sure that no eavesdropper gets knowledge of the new session keys. Therefore keys should be transmitted in encrypted form.

The DCP has two commands and one special key to support key distribution.

Commands: Encrypt with Master Key
Load Encrypted Key

Key: Master Key

Figure 3.21 shows the operation sequence when distributing keys:

Step 1: "A" generates a 56-bit session key, splits the key into eight 7-bit groups and adds a parity bit to each group. The result (a 64-bit word) is encrypted with the Master Key. Therefore, "A" issues the command "Encrypt with Master Key" and strobes the 64-bit result through the Master Port into the Input Register. The DCP encrypts the 64-bit word with the Master Key and ECB mode. The encrypted key can be removed from the Output Register via the Master Port.

Step 2: "A" transmits the encrypted key via the communication link to "B".

Step 3: "B" issues the command "Load Encrypted Key". The received encrypted key is strobed through the Master Port into the Input Register and decrypted with the Master Key. The Master Key of "B" must be identical to the Master Key of "A". After decryption the parity is checked and the decrypted key is loaded into the appropriate register. To enhance the system security "B" cannot read the decrypted key.

CHAPTER 4. INTERFACING

This chapter contains interfaces between the DCP and the most common 8-bit and 16-bit microprocessors.

First, a look at the critical points in interfacing the DCP.

Demultiplexed Systems:

The DCP uses a multiplexed address/data bus which means that the system designer has to provide this kind of bus to the DCP. In a non-multiplexed system environment the address and data bus are separated and not time-multiplexed. There are two basic solutions for simulating a multiplexed address/data bus.

The interface logic multiplexes at least the two relevant lines (MP_1 and MP_2) addressing an internal DCP register. Multiplexing the other lines ($MP_Ø$, MP_3 to MP_7) is optional.

The second solution simulates a multiplexed address/data bus under software control. The CPU can access the DCP to latch an internal register address (Address Latch Cycle) or to transfer data (Data Read or Write Cycle). These two kinds of accesses usually are distinguished by the address line "$A_Ø$". In the Address Latch Cycle, only an address strobe is generated to strobe in the internal register address supplied via the CPU data bus. In the Data Transfer Cycle, only data strobes are generated to actually read a formerly addressed register or to write to it. So the Address Latch process and the Data Transfer are totally independent from each other.

The advantages of the second solution are that it usually takes less interface logic and that it is faster in most applications because there is no overhead in latching the address. The interfaces in Chapters 4.4, 4.6, 4.9 and 4.1Ø employ the second solution. A disadvantage of the second solution is a slight software overhead caused by the Address Latch Cycles. Once the DCP is initialized for a data ciphering session, there is no more need for Address Latch Cycles. During the high speed data ciphering session itself, only Data Transfer Cycles are executed.

The first approach has advantages where multiplexing the two above mentioned lines causes no overhead in hardware and timing. The iAPX286 to Am9568 interface is an example. The multiplexing logic can be integrated into the existing PAL* (Programmable Array Logic) interface, and the multiplexing does not extend the Data Transfer Cycle.

*PAL is a registered trademark of and is used under license from Monolithic Memories, Inc.

Synchronization:

One of the basic problems is to satisfy the required synchronization between the clock and data strobe.

The DCP requires that the rising edge of data strobe fall into a certain window after the falling edge of the clock. This window is specified in timing parameter 45 of the Product Specification as listed below:

Am9518: Ø - TWL - 1ØØ ns
Am9568: Ø - TWL - 85 ns
AmZ8Ø68: Ø - TWL - 65 ns

TWL is the actual clock width (Low) of the interface.

Several design techniques can guarantee this parameter.

Some CPU's, for example the 8Ø86 in Maximum Mode, have data strobe timing that inherently satisfies the DCP requirements. These interfaces do not need special synchronization logic.

In asynchronous systems, the interface control logic usually buffers the data strobe and can easily synchronize it to the clock. PAL devices with registered outputs clocked by the DCP clock simplify this task (Chapter 4.1Ø).

Another, sometimes simpler, approach is to make use of the clock Low width dependent specification by delaying the first rising edge of clock following data strobe (Chapter 4.4).

Address Strobe:

The three members of the DCP family have different specifications for the address strobe width:

Am9518 : 115 ns
AmZ8Ø68: 8Ø ns
Am9568 : 4Ø ns

The Am9568 should be used in systems with narrow address strobes (e.g., 8Ø86 CPU at 8 MHz).

Read/Write:

The Am9518 and AmZ8Ø68 require a set-up time of 1ØØ ns to data strobe. The Am9568 does not have this specification because of its functionally different bus interface. Read/Write and data strobe are replaced by write strobe and read strobe. The Am9568, therefore, has advantages in applications where it is difficult to satisfy the read/write set-up time.

PAL Devices:

Many of the following applications employ PAL devices to integrate the entire interface logic into one 20-pin device. Registered PAL devices like the AmPAL16R4 have registered and combinatorial outputs which enable the designer to build up small state machines for the interface handshake. An asynchronous bus, such as the iSBX* bus, can easily be adapted to the synchronous requirements of the DCP.

A PAL device is a semi-custom device that is supported by computer-aided-design tools like the PAL assembler. All interfaces described in this book that employ PAL devices have a complete listing of the PAL design specification program, the input of the PAL assembler. Each program consists of five sections as described below:

1) The first four lines of the PAL Design Specification list the PAL part number, the user's internal part number, the date, the designer's name, the device application name, and the company name and address.

2) The pin-list gives the symbolic names used for the inputs and outputs in the order of pin 1 to pin 20. Active Low signals are preceded by "/", a symbol used instead of a "bar".

3) The equations are the heart of the program. They define the conditions under which the outputs become active.

4) The function table is a powerful tool to test the correctness of the equations. The designer specifies the signals to be supplied to the inputs and to be seen at the outputs. In the simulation pass, the PAL assembler verifies whether the function table corresponds to the equations. This pass detects the most common errors (typing errors and signal inversions) and checks for logical errors. Each line of the function table represents a test vector containing inputs and outputs. The states are defined by characters as specified below:

Input:	L	Low
	H	High
	C	Clock registered outputs
	X	Don't care
Output:	L	Low expected
	H	High expected
	Z	High impedance expected
	X	Don't test

5) The description documents the operation of the device and its intended application.

*iSBX is a trademark of Intel Corporation.

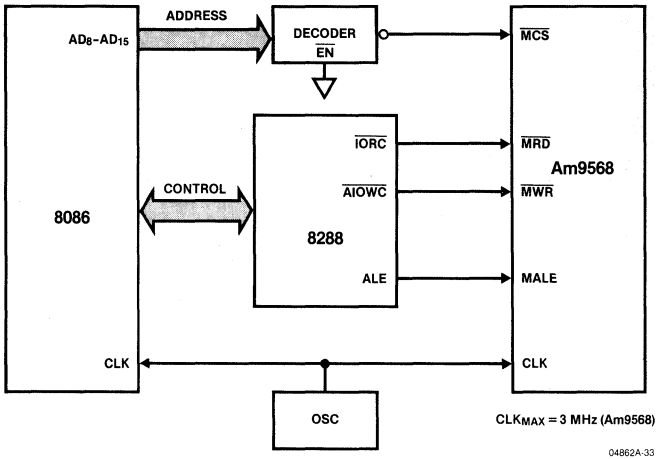


Figure 4.1. Direct Interface 8086-Am9568 (Maximum Mode)

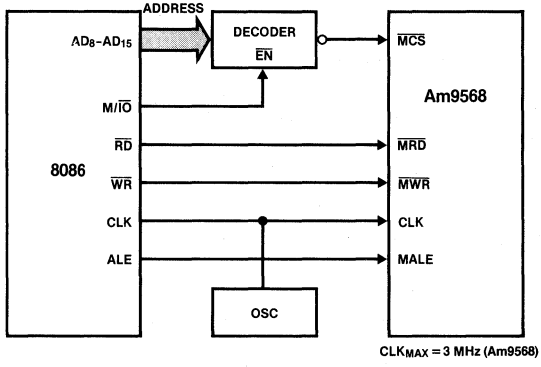


Figure 4.2. Direct Interface 8086-Am9568 (Minimum Mode)

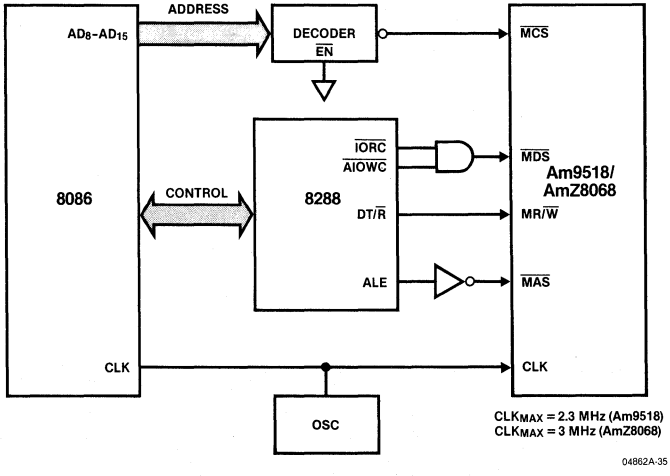


Figure 4.3. Direct Interface 8086-Am9518/AmZ8068 (Maximum Mode)

4.1. 8086/8088 - Am9518/AmZ8068/Am9568

Interfacing the DCP family to 8086 or its 8-bit bus equivalent, the 8088, is straightforward.

In systems with CPU clock rates up to 3 MHz, the Am9568 can be directly interfaced to the CPU (Figures 4.1 and 4.2). The clock rate is limited to 3 MHz because of the 33%/66% duty cycle (33% High, 66% Low) of the CPU clock and to satisfy the minimum clock High time of 115 ns of the Am9568. The second critical parameter is the relationship between the clock and data strobe. The Am9568 requires a delay of the rising edge of $\overline{MRD}$ or $\overline{MWR}$ to the falling edge of the clock of 0 - TWL - 85 ns. TWL is the clock Low width. In this interface the minimum clock Low width is 207 ns. This determines a maximum delay of up to 122 ns. The CPU is specified to have a "Control Active Delay" of 10 to 110 ns. With a margin of 12 ns, it is obviously impossible to increase the system clock by modifying its duty cycle.

Figures 4.3 and 4.4 show a similar interface using the Am9518 and the AmZ8068. This interface needs additional logic to convert the read or write strobes into a Read/Write (R/$\overline{W}$) and a Data Strobe ($\overline{MDS}$) and to invert the Address Latch Enable to generate a Master Port Address Strobe ($\overline{MAS}$). Similar to the interface discussed above, the clock rate is limited by the clock Low and High widths and the requirements of the DCP. The Am9518 needs a minimum clock High width of 150 ns determining a maximum clock rate of 2.3 MHz. The minimum clock Low width of 275 ns and the DCP specification of 0 - TWL - 100 ns provides a margin of 275 ns - 110 ns - 100 ns = 65 ns.

The AmZ8068 requires a minimum clock High width of 115 ns, resulting in the same maximum clock rate as in interfacing to the Am9568 (3 MHz). The specification about the synchronization of clock and data strobe is less critical in this interface (0 - TWL - 65 ns) so the margin becomes 32 ns.

An 8086/8088 system with clock rates larger than the rates mentioned above requires more sophisticated interface logic: the DCP clock must not exceed 4 MHz (3 MHz for the Am9518), the Address Strobe width has to be satisfied, and the data strobes must be synchronous to the clock. The case in which the DCP clock is divided down by two from the CPU clock is discussed below.

An application where the DCP runs asynchronously from the 8086 clock is not discussed here. Ideas can be taken from the Chapter 4.10 iSBX Bus to Am9568 interface.

8086/8088 - Am9518/AmZ8068 (Figures 4.5 and 4.6)

The Control/Key Mode input (C/$\overline{K}$) is wired Low to select the Multiplexed Control Mode. In this mode the address to the internal registers of the DCP, MP_1 and MP_2, is multiplexed with the data byte on the eight bidirectional lines of the Master

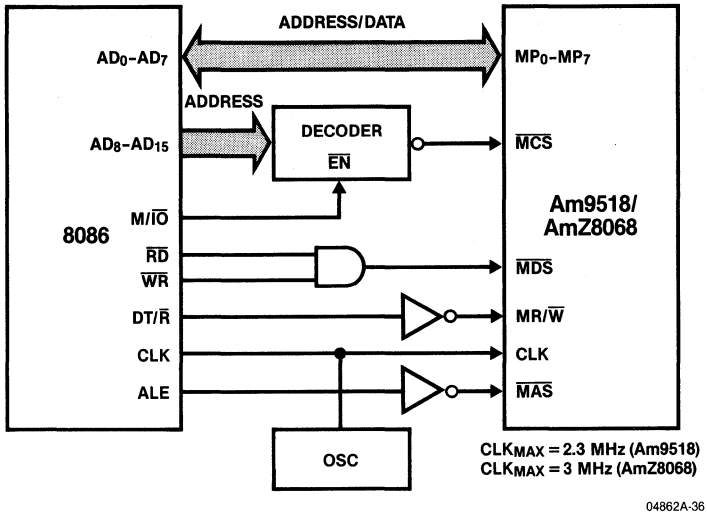


Figure 4.4. Direct Interface 8086-Am9518/AmZ8068 (Minimum Mode)

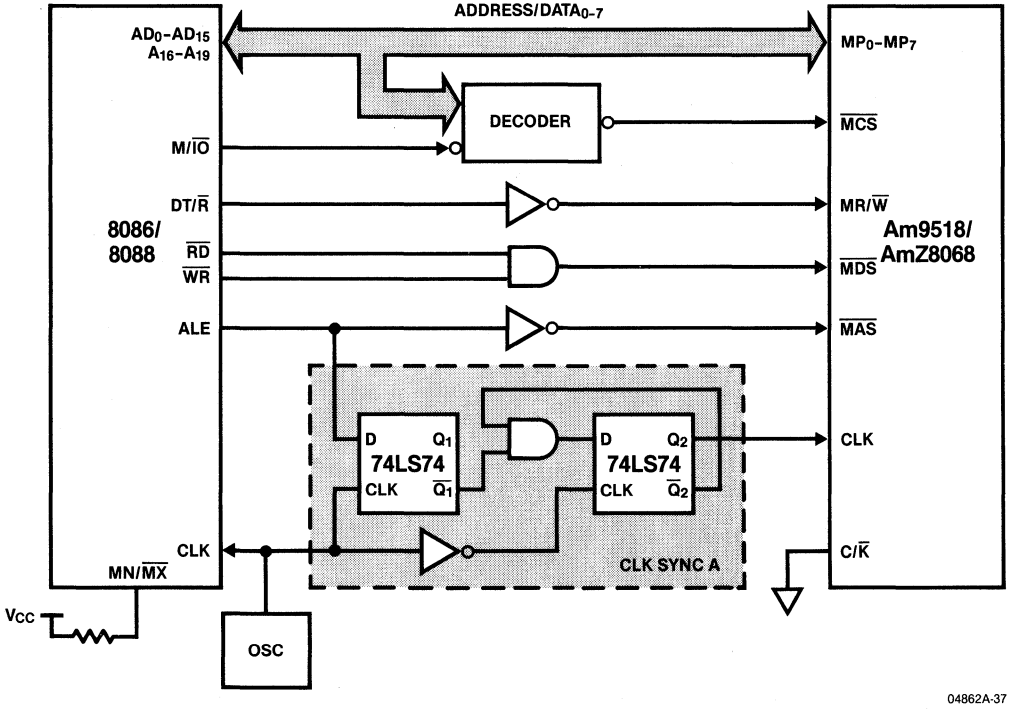


Figure 4.5. 8086/8088-Am9518/AmZ8068 Interface (Minimum Mode)

Port bus. MP_1 and MP_2 are latched on the rising edge of $\overline{MAS}$ (Master Port Address Strobe), to select the internal register for subsequent data transfer cycles.

$\overline{MAS}$ is the inverted Address Latch Enable of the 8086 bus. The state of $\overline{MCS}$ (Master Port Chip Select) is also latched at the rising edge of $\overline{MAS}$. In the Minimum Mode of the 8086 (MN/$\overline{MX}$=High) $\overline{MCS}$ may only go Low during Input/Output cycles (M/$\overline{IO}$=Low); therefore, M/$\overline{IO}$ enables the address decoder in Minimum Mode.

The Read/Write input (MR/$\overline{W}$) is connected to Data Transmit/Receive (DT/$\overline{R}$). DT/$\overline{R}$ satisfies the set-up and hold time requirements of MR/$\overline{W}$.

Master Port Data Strobe ($\overline{MDS}$) is active if either Input/Output Read Control ($\overline{IORC}$) or Advanced Input/Output Write Control ($\overline{AIOWC}$) are active. The $\overline{AIOWC}$ has a wider Low width than $\overline{IOWC}$ (Input/Output Write Control) and so gives a wider margin in interfacing.

In Minimum Mode (Figure 4.5), $\overline{RD}$ and $\overline{WR}$ are logical ORed to generate $\overline{MDS}$. The timing is the same as in Maximum Mode.

8086/8088 - Am9568 (Figure 4.7)

CPU clock rates above 4.44 MHz (above 5.8 MHz for the AmZ8068) require use of the Am9568 instead of the Am9518, because TWA (Master Port Address Strobe width) becomes critical with increased clock rate, as shown below:

```
Am9518  :   TWA  = 115 ns
AmZ8068 :   TWA  =  80 ns
Am9568  :   TWA  =  40 ns

8086/8088 :   TLHLL = 115 ns at 4.44 MHz
8086/8088 :   TLHLL =  80 ns at 5.80 MHz
8086/8088 :   TLHLL =  48 ns at 8.00 MHz
```

TLHLL is the Address Latch Enable width (ALE) of the 8086.

For CPU clock rates above 7 MHz, one Wait state has to be inserted during Control Register Reads (timing parameter 44).

Note: In the interfaces shown, the number of Wait states must be the same for all read or write accesses to the DCP, because the Clock Synchronizer is designed for either an even or an odd number of Wait states.

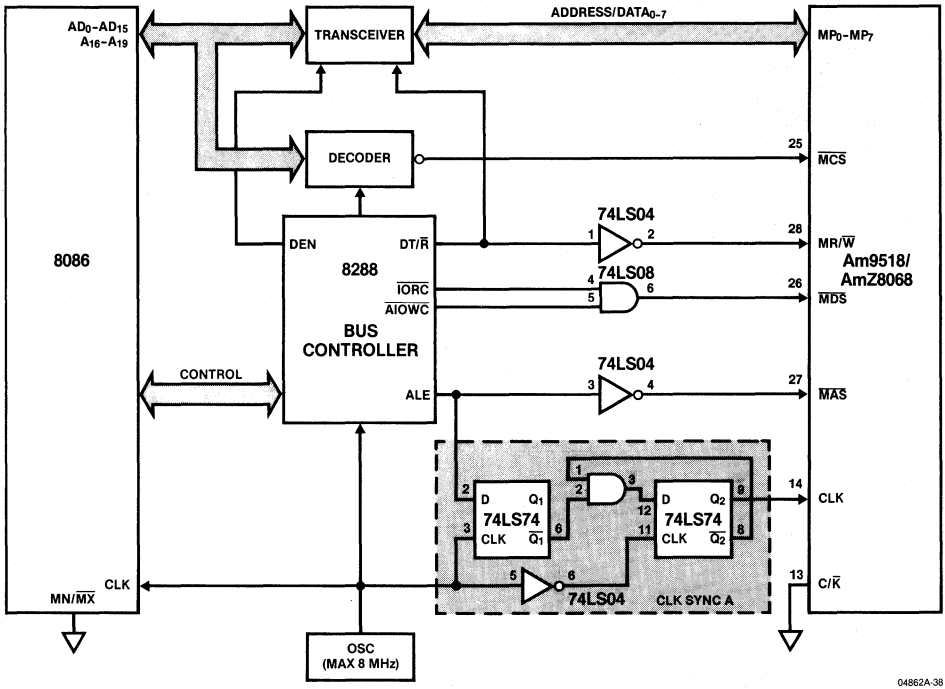


Figure 4.6. 8086/8088-Am9518/AmZ8068 Interface (No Wait State)

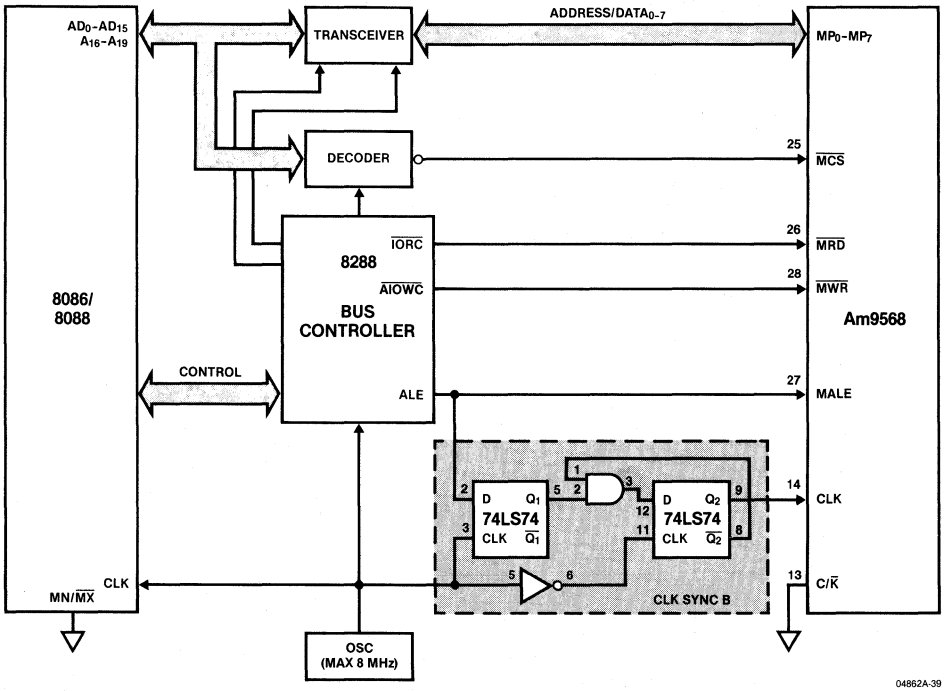


Figure 4.7. 8086/8088-Am9568 Interface (1 Wait State)

Clock Synchronization

A very important factor in designing the interface to the 8086 is that the rising edge of MDS must be synchronous to the falling edge of the DCP clock (timing parameter 45).

In a system where the DCP runs at a divided system clock, a clock synchronizer is required. Without a synchronizer the rising edge of the Data Strobes (MDS, MRD and MWR) would be synchronous to either the falling or rising edge of the divided clock. Two simple Clock Synchronizers are used in these interfaces; one is designed for an even number, the other is designed for an odd number of Wait states. The DCP clock is synchronized to the Data Strobes at the falling edge of the CPU clock at the end of the CPU cycle T1 (Figures 4.8 and 4.9). At this edge, the state of the DCP clock is forced to a Low (CLK SYNC A in Figure 4.8) or to a High (CLK SYNC B in Figure 4.9), depending on the number of Wait states inserted. DCP CLK 1 and 2 show the two possible phases of the DCP clock and how the Clock Synchronizer adjusts the phase.

Data Ciphering Speed

The data ciphering speed of the DCP is limited by the byte transfer capability of the 8086 bus. A high-performance DMA like the AM9516 increases the throughput as shown in the following table:

8086 clock	DMA clock	DCP clock	N	T
8 MHz	4 MHz	4 MHz	36	0.78 MByte/s
6 MHz	6 MHz	3 MHz	18	1.05 MByte/s
8 MHz	no DMA	4 MHz	70	0.42 MByte/s

The formula for calculating the throughput is:

$$T = (8 * f) / (N + 5) \text{ MByte/s}$$

T = Throughput in MByte/s

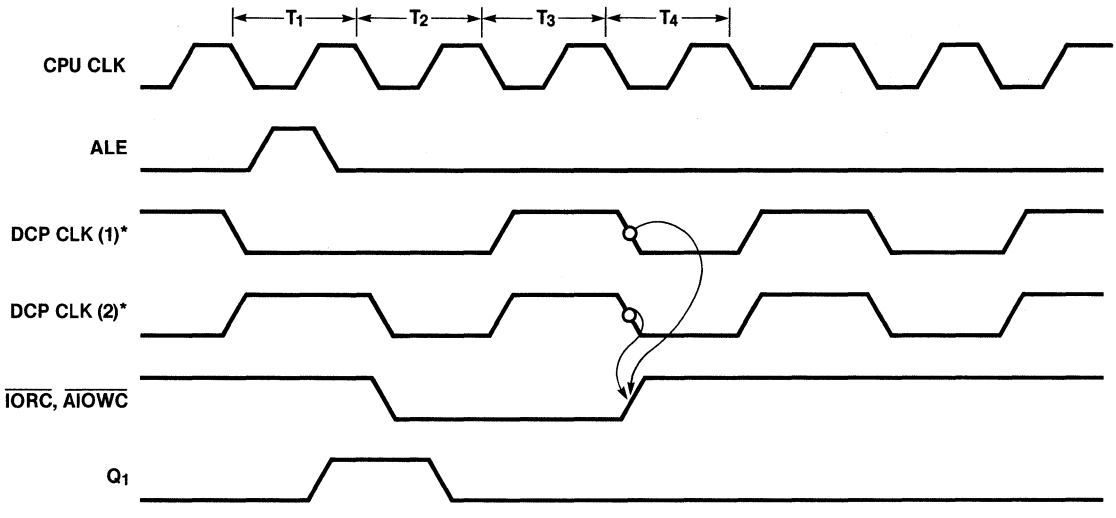
N = Number of clock cycles per 8 byte transfer

5 = Internal operation time (5 clocks per block)

f = DCP clock in MHz

8 = 8 data bytes per block

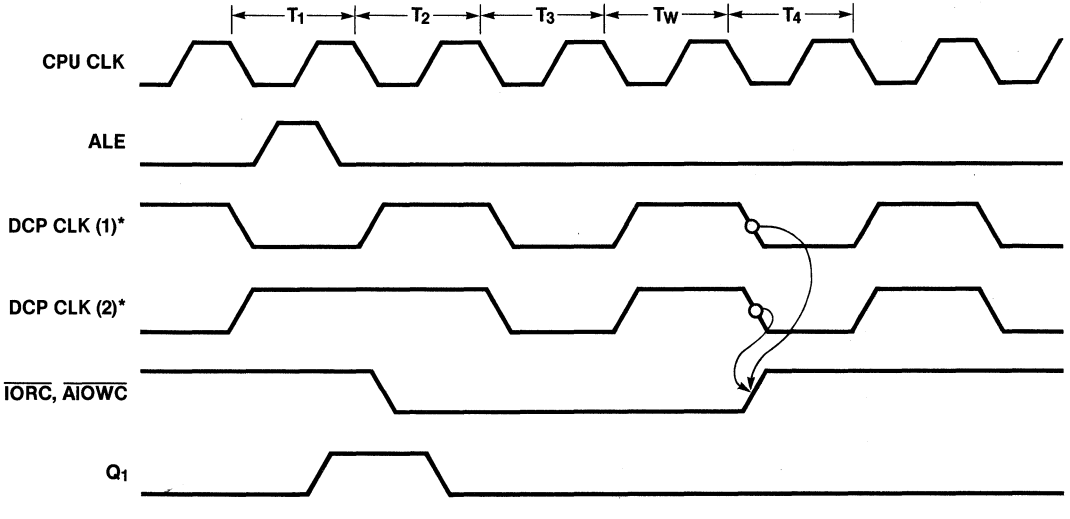
The first two cases in the table above are fast enough to encrypt and decrypt the data transferred to or from a 5 1/4-inch Winchester Disk Controller "on the fly" (5 MBit/s=0.625 MByte/s).



* DCP CLK (1) AND (2) SHOW TWO PHASES OF DCP CLK

04862A-40

Figure 4.8. DCP CLK Synchronization Timing (No Wait States, CLK SYNC A)



* DCP CLK (1) AND (2) SHOW TWO PHASES OF DCP CLK

04862A-41

Figure 4.9. DCP CLK Synchronization Timing (1 Wait State, CLK SYNC B)

Testing

The interface of Figures 4.6 and 4.7 and both Clock Synchronizers were built and tested using the software described below.

- The DCP is reset by software writing "00_H" to the Command Register.

- The ciphering mode is selected by writing "18_H" into the Mode Register. Here the mode is: Master Port-only configuration, Electronic Code Book (ECB) and Encryption.

- The Clear Encryption key is loaded through the Master Port by issuing the command "11_H". After the command is entered, the Status Register content is read out. Only the Command Pending bit should be set (40_H). If other bits are set, the program sets the error flag "CODE" to FF_H and terminates. If the status is correct, eight bytes of key are strobed in through the Master Port in eight output instructions. The Key is "8001010101010101_H". The most significant byte is loaded first.

- The status of the DCP is checked, the Command Pending bit and the parity error bits should be reset (00_H).

- The encryption is started by entering the command "Start Encryption" (41_H).

- One block of data (8 bytes) is strobed into the Master Port. The source is the byte string "PLAIN". In this example, the plain text is: "0000000000000000_H".

- Loop3 is executed until the Busy bit of the Status Register shows the encryption is done.

- One block of ciphered data is read out of the Master Port and transferred to the program location "CIPHER". The ciphered text should be: "95A8D72813DAA94D$_H$".

- The Status Register is checked; only the Start Entered bit should be set (80_H).

- The encryption session is stopped by issuing the command "Stop Encryption" ($E0_H$).

- After that the status should be 00_H; all flags are reset.

The program can be used to decrypt data, if two program locations are changed:

- The "Enter Key" command of location 0110_H has to be changed to 12_H ("Load Clear D-Key Through Master Port").

- The Start Command of location 0131_H has to be changed to 40_H ("Start Decryption").

After running the program, the error flag in "CODE" should be reset (00_H).

This test was performed to verify the communication between the 8086 and the DCP. By providing clear and encrypted data for the key shown, users should be able to verify operation of any variation to the design. The software was kept simple to avoid dependence on other hardware in the system.

```
        ASM86 VER 1.0    SOURCE: APPL8068.ASM

+-------------------------------------------------
;                                          JUERGEN STELBRINK  4-12-83
;                                          ADVANCED MICRO DEVICES
;
;         8086 TO AM9518 (AMZ8068) INTERFACE TEST PROGRAM
;
;-------------------------------------------------------------------------
;
;
; ADDRESSES OF THE DCP (EVEN ADDRESSES)
;
FC00                            MPSEL   EQU     0FC00H          ; BASE ADDRESS OF MASTER PORT
FC02                            MPCOM   EQU     MPSEL+2         ; COMMAND REGISTER (WRITE ONLY)
FC02                            MPSTAT  EQU     MPSEL+2         ; STATUS REGISTER (READ ONLY)
FC06                            MPMODE  EQU     MPSEL+6         ; MODE REGISTER (READ AND WRITE)
FC00                            MPINP   EQU     MPSEL           ; INPUT REGISTER (WRITE ONLY)
FC00                            MPOUT   EQU     MPSEL           ; OUTPUT REGISTER (READ ONLY)
0018                            ECB     EQU     18H             ; ENCRYPT, MP ONLY, ECB
0080                            KEY1    EQU     80H             ; KEY: 80010101010101010H
0001                            KEY2    EQU     01H
                        ;
                                ORG     100H
                        ;
0100 BA 02 FC             BEGIN:  MOV     DX,MPCOM                ; DX: POINTER TO PORT ADDRESS
0103 B0 00                        MOV     AL,0                    ; SOFTWARE RESET
0105 EE                           OUT     DX,AL
                        ;
0106 BA 06 FC                     MOV     DX,MPMODE
0109 B0 18                        MOV     AL,ECB                  ; SELECT MODE
010B EE                           OUT     DX,AL
                        ;
010C BA 02 FC                     MOV     DX,MPCOM
010F B0 11                        MOV     AL,11H                  ; LOAD CLEAR E-KEY THROUGH MP
0111 EE                           OUT     DX,AL
                        ;
0112 BA 02 FC                     MOV     DX,MPSTAT
0115 EC                           IN      AL,DX                   ; READ STATUS
0116 3C 40                        CMP     AL,40H                  ; 40= CP SET
0118 75 5F                        JNE     ERROR
                        ;
011A B0 80                        MOV     AL,KEY1                 ; LOAD 1. KEY BYTE
011C BA 00 FC                     MOV     DX,MPINP
011F EE                           OUT     DX,AL                   ; OUTPUT 1. KEY BYTE
0120 B9 07 00                     MOV     CX,7                    ; LOAD COUNTER FOR NEXT 7 BYTES KEY
0123 B0 01                        MOV     AL,KEY2                 ; FOLLOWING KEY DATA
0125 EE                   LOOP1:  OUT     DX,AL
0126 E2 FD                        LOOP    LOOP1
                        ;
0128 BA 02 FC                     MOV     DX,MPSTAT
012B EC                           IN      AL,DX                   ; READ STATUS
012C 3C 00                        CMP     AL,0                    ; FLAGS RESET?
012E 75 49                        JNE     ERROR
                        ;
0130 BA 02 FC                     MOV     DX,MPCOM
0133 B0 41                        MOV     AL,41H                  ; START ENCRIPTION
0135 EE                           OUT     DX,AL
                        ;
0136 B9 08 00                     MOV     CX,8                    ; 8 BYTES (1 BLOCK) OUTPUT
```

```
ASM86 VER 1.0   SOURCE: APPL8068.ASM

  0139 BB 00 00                 MOV     BX,0                    ; INITIALIZE POINTER
  013C BA 00 FC                 MOV     DX,MPINP
  013F 2E 8A 87 81 01   LOOP2:  MOV     AL,PLAIN[BX]            ; LOAD DATA
  0144 43                       INC     BX                      ; INCREMENT POINTER
  0145 EE                       OUT     DX,AL                   ; WRITE PLAIN DATA
  0146 E2 F7                    LOOP    LOOP2
                        ;
  0148 BA 02 FC                 MOV     DX,MPSTAT
  014B EC               LOOP3:  IN      AL,DX                   ; WAIT UNTIL ENCRYPTION IS DONE
  014C 24 20                    AND     AL,20H                  ; TEST BUSY BIT
  014E 75 FB                    JNZ     LOOP3
                        ;
  0150 B9 08 00                 MOV     CX,8                    ; 8 BYTES (1 BLOCK) INPUT
  0153 BB 00 00                 MOV     BX,0                    ; INITIALIZE POINTER
  0156 BA 00 FC                 MOV     DX,MPOUT
  0159 EC               LOOP4:  IN      AL,DX                   ; READ ENCRYPTED DATA
  015A 2E 88 87 89 01           MOV     CIPHER[BX],AL           ; STORE DATA
  015F 43                       INC     BX                      ; INCREMENT POINTER
  0160 E2 F7                    LOOP    LOOP4
                        ;
  0162 BA 02 FC                 MOV     DX,MPSTAT
  0165 EC                       IN      AL,DX                   ; TEST STATUS REGISTER
  0166 3C 80                    CMP     AL,80H                  ; 80= START ENTERED
  0168 75 0F                    JNE     ERROR
                        ;
  016A BA 02 FC                 MOV     DX,MPCOM
  016D B0 E0                    MOV     AL,0E0H                 ; STOP ENCRYPTION
  016F EE                       OUT     DX,AL
                        ;
  0170 BA 02 FC                 MOV     DX,MPSTAT
  0173 EC                       IN      AL,DX                   ; TEST STATUS REGISTER
  0174 3C 00                    CMP     AL,0                    ; ALL BITS MUST BE RESET
  0176 75 01                    JNE     ERROR
  0178 C3                       RET
                        ;
  0179 B0 FF            ERROR:  MOV     AL,0FFH                 ; LOAD ERROR CODE
  017B 2E A2 80 01              MOV     CODE,AL
  017F C3                       RET
                        ;
  0180 00               CODE    DB      00H                                     ; ERROR CODE
  0181 00 00 00 00 00 00 PLAIN  DB      00H,00H,00H,00H,00H,00H,00H,00H         ; PLAIN TEXT
       00 00
  0189 12 23 34 45 56 67 CIPHER DB      12H,23H,34H,45H,56H,67H,78H,89H         ; CIPHER TEXT
       78 89
                        ;
                                END

END OF ASSEMBLY. NUMBER OF ERRORS:   0
```

4.2. iAPX186 - AmZ8068

The iAPX186 can operate in two basic modes: Minimum Mode or Maximum Mode. In Maximum Mode the 8288 Bus Controller provides command and control timing. Refer to Chapter 4.1 for examples of this type of interface.

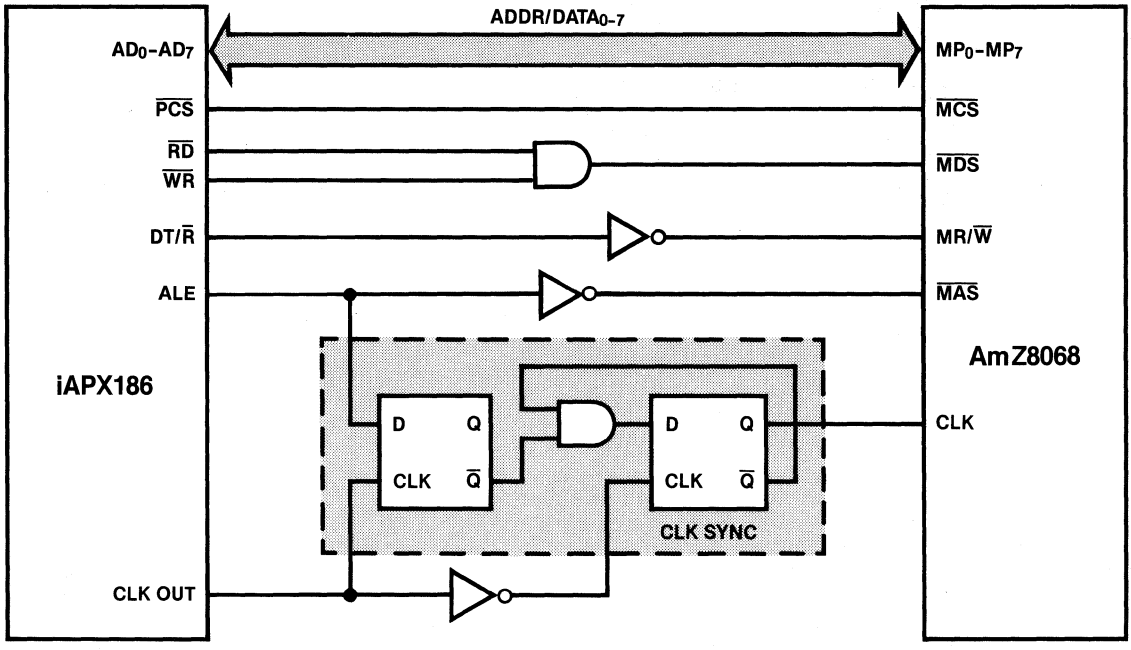
In Minimum Mode the bus timing of the iAPX186 is slightly different from the 8086 bus timing. Figure 4.10 shows the interface logic. The maximum clock rate for the DCP is 4 MHz, resulting in a maximum CPU clock rate of 8 MHz. No Wait states are required.

An AmZ8068 must be used in this application because of the wider range in delay time from clock to the read or write control signal delay with respect to the clock. This parameter is specified for the iAPX186 as 10 to 55 ns. The AmZ8068 requires a delay of 0 to 50 ns at 4 MHz, the Am9568 0 to 30 ns at 4 MHz. Because of two delays in the clock path (Inverter and D-Flip-Flop) and only one delay in the control signal path (AND gate), the timing tolerance of these signals at the DCP is decreased to 0 to 45 ns.

At lower CPU clock rates the timing is less critical because the specified time relationship between clock and data strobe becomes wider (timing parameter 45 of the data sheet).

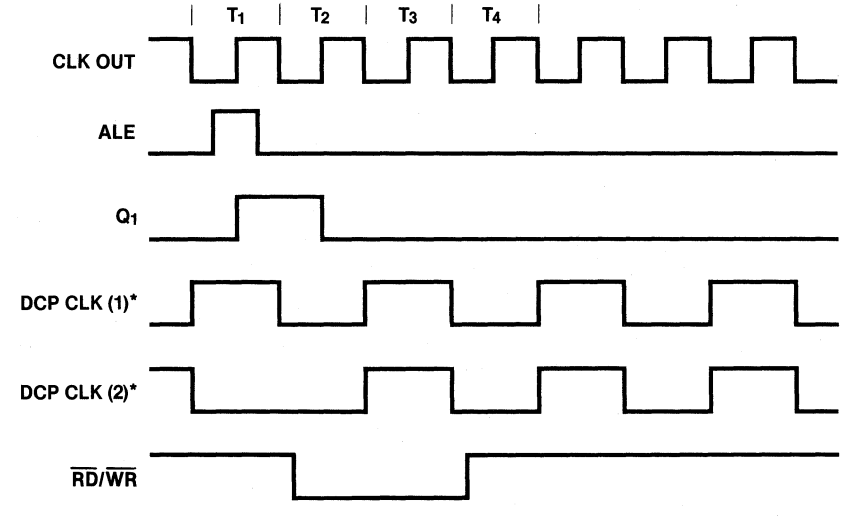
The maximum clock for operating without a Wait state can be calculated like this: The $\overline{RD}$ width is specified as 2 * TCLCL - 50 ns for the iAPX186. The $\overline{WR}$ width is 2 * TCLCL - 40 ns. The smaller $\overline{RD}$ width is used for the calculation. At an 8-MHz clock, the 186 generates an $\overline{RD}$ signal 200 ns wide. The AmZ8068 requires a minimum data strobe width of 200 ns for a Status Register access. The system can, therefore, operate up to this clock rate without a Wait state.

The Clock Synchronizer in Figure 4.10 is the same as Clock Synchronizer A in Figure 4.5. Figure 4.11 illustrates how this logic synchronizes the data strobe to the clock. DCP CLK(1) and DCP CLK(2) show the possible phases of the CPU clock before synchronization. At the end of cycle T1 the clock is synchronized. No Wait state is allowed when accessing the DCP. (An odd number of Wait states would synchronize the data strobe to the wrong edge of the clock.)



04862A-42

Figure 4.10. iAPX186-DCP Interface (Minimum Mode)



*DCP CLK (1) AND (2) SHOW TWO PHASES OF DCP CLK

04862A-43

Figure 4.11. DCP CLK Synchronization Timing (No Wait States)

4.3. iAPX286 - Am9568

This chapter shows an iAPX286 (80286) to Am9568 interface (Figure 4.12). The Am9568 is chosen because of the narrower width of address strobe. The address strobe width of a 8-MHz CPU is about 60 ns. This interface is designed for an 8-MHz CPU where the DCP is synchronously operating at the maximum clock rate of 4 MHz.

The Interface

The Multibus* Mode Select input of the Bus Controller 82288 is tied Low to optimize the command and control signals for short bus cycles. The Command Delay (CMDLY) becomes active High for one 16-MHz clock cycle whenever the DCP is selected to delay the Read and Write strobes by 125 ns. This satisfies the timing requirement of the minimum delay between ALE inactive and Read or Write strobe active of the DCP. An open collector gate must be added to allow other peripherals to drive this input.

The ALE, $\overline{IORC}$ and $\overline{IOWC}$ outputs of the 82288 are wired directly to the DCP. ALE strobes a D-Flip-Flop to store the state of Chip Select for the whole cycle.

$\overline{Q}_3$ and the latched Chip Select CSL are ANDed externally to generate the Synchronous Ready for the 82284. The 82284 samples the line at the falling edge of the clock. The registered output $\overline{Q}_3$ is clocked with the rising edge of the same clock, thus satisfying the set-up and hold time requirements of the 82284. Two Wait States are inserted.

Half of the PAL device operates as a bidirectional Address/Data Multiplexer. During the Address Latch Enable active phase, the state of A_1 and A_2 is transferred to the AD_1 and AD_2 pin of the PAL device. The DCP latches this two-bit address with the falling edge of ALE.

When $\overline{IORC}$ and CSL are active, the states of AD_1 and AD_2 are passed to D_1 and D_2 respectively. The DCP Register can be read. If $\overline{IOWC}$ and CSL are active, the data path is turned around; D_1 and D_2 are inputs, AD_1 and AD_2 are outputs.

The address hold time of the PAL device is sufficient, because the address information is passed to AD_1 and AD_2 whenever IORC*CSL or IOWC*CSL are not true, i.e. whenever data is not transferred between the CPU and the DCP.

The read data hold time requirement of 5 ns of the Am9568 is satisfied by the propagation delay of the PAL device.

The read data hold time requirement of 5 ns of the iAPX286 is also satisfied by the PAL device.

*MULTIBUS is a registered trademark of Intel Corporation.

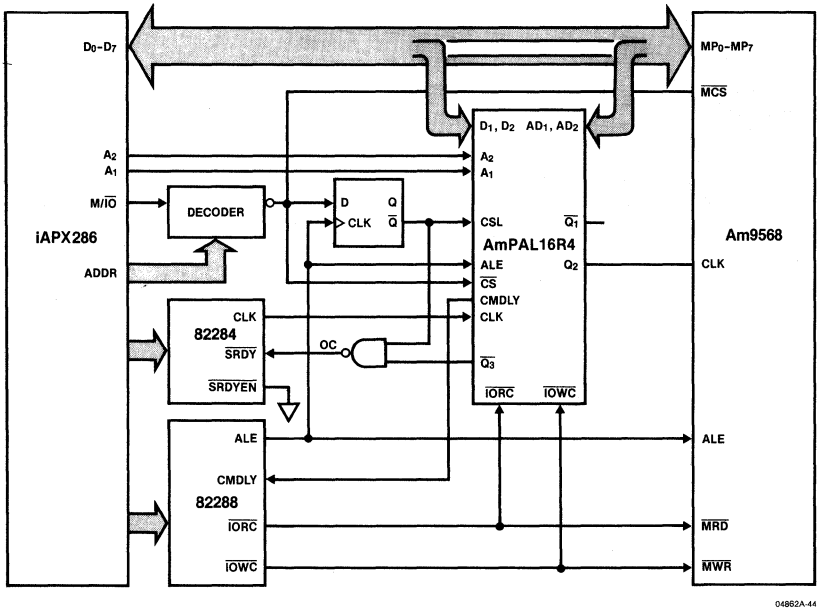


Figure 4.12. iAPX286-Am9568 Interface

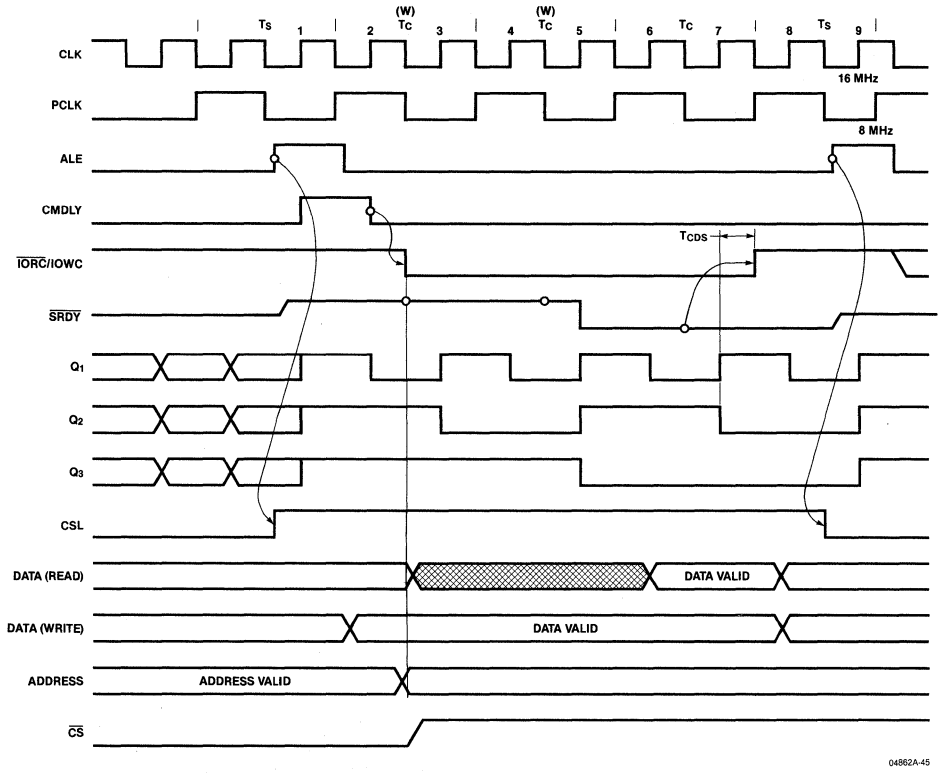


Figure 4.13. Timing Diagram

The Master Port Chip Select ($\overline{MCS}$) input of the DCP is connected to the unlatched address decoder output.

The DCP Clock

The PAL device synchronizes the DCP clock to the data strobes $\overline{IORC}$ and $\overline{IOWC}$ (Figure 4.13). It also divides the 16-MHz system clock (8-MHz CPU clock) down to the maximum DCP clock rate of 4 MHz. At this clock rate the data strobe delay to the DCP clock must be Ø to 3Ø ns. The Bus Controller is specified to generate a data strobe timing of 3 to 15 ns to the falling edge of CLK (16 MHz). Because of the higher propagation delay of a standard PAL device, the registered outputs are toggled at the rising edge of CLK before the data strobes become inactive. This gives additional 32.5 ns for the DCP clock signal path.

Q_1 to Q_3 are three outputs of the PAL state machine. The registered output are clocked with the rising edge of the 16-MHz 82284 clock. Whenever ALE and $\overline{CS}$ are active, Q_1 to Q_3 are set to the initial state. Q_1 to Q_3 are outputs of a 3-bit down counter, with Q_3 as the most significant bit.

Q_3 is used to generate the $\overline{SRDY}$ signal for the 82284 as mentioned above.

Q_2 is the DCP clock. This design must guarantee that the minimum DCP clock High or Low time is at least 115 ns or two 16-MHz clock cycles. This is done by toggling Q_2 only during phase 2 cycles of the CPU. The CPU design guarantees that there is always a phase 1 cycle between two phase 2 cycles.

Assuming a typical PAL propagation delay of 25 ns, timing parameter TCDS (Time Clock Data Strobe) is 1Ø.5 to 22.5 ns (3 + 32.5 - 25 ns to 15 + 32.5 - 25 ns). It satisfied the required Ø to 3Ø ns.

The AmPAL16R4 has active Low outputs. But one output, Q_2, should be active High. The equation for Q_2 was derived to be

$$Q_2 = ALE * CS + Q_1 * Q_2 + \overline{Q}_1 * \overline{Q}_2$$

To compensate for the inversion in the PAL device either de Morgan Theorem or Karnaugh-Veitch diagrams can be used to convert it to the form shown in the PAL Design Specification.

Improvements

The DCP needs two Wait states only when the Control Registers are read. Data Register read or writes and Control Register writes can be executed with only one Wait state, which improves the Data Ciphering speed of this interface. The more sophisticated Wait control logic and the two external TTL gates can be integrated into one AmPAL22V1Ø device.

```
PAL16R4                                          PAL DESIGN SPECIFICATION
DCP043                                           JUERGEN STELBRINK  8-23-83
iAPX286 - Am9568 (DCP) INTERFACE DEVICE
ADVANCED MICRO DEVICES

CLK   /CS   CSL   ALE   /IORC   /IOWC   A1      A2    NC    GND
/OE   D1    D2    /Q1   Q2      /Q3     CMDLY   AD1   AD2   VCC

Q1       :=  ALE*CS + /Q1

/Q2      :=  Q1*/Q2*/ALE + Q1*/Q2*/CS + /Q1*Q2*/ALE + /Q1*Q2*/CS

Q3       :=  ALE*CS + Q1*Q2*Q3 + /Q1*Q2*Q3 + Q1*/Q2*Q3 + /Q1*/Q2*/Q3

/CMDLY   :=  /ALE+/CS

IF(CSL*IORC) /D1  =  /AD1

IF(CSL*IORC) /D2  =  /AD2

IF(CSL*/IORC) /AD1 =  /A1*ALE + /D1*/ALE

IF(CSL*/IORC) /AD2 =  /A2*ALE + /D2*/ALE


FUNCTION TABLE

CLK /CS CSL ALE /IORC A1 A2 D1 D2 AD1 AD2 /Q1 /Q2 /Q3 CMDLY

;              /                                      C
;              I                                      M
; C  /  C  A  O                    A   A    /  /  /  D
; L  C  S  L  R    A   A   D   D   D   D    Q  Q  Q  L
; K  S  L  E  C    1   2   1   2   1   2    1  2  3  Y     COMMENT
-------------------------------------------------------------------
;
  C  L  H  H  H    L   L   Z   Z   L   L    L  L  L  H ; 1 (/CS ACTIVE)
  X  L  H  H  H    L   H   Z   Z   L   H    L  L  L  H
  X  L  H  H  H    H   H   Z   Z   H   H    L  L  L  H
  C  L  H  L  H    H   L   L   H   L   H    H  L  L  L ; 2 (WRITE CYCLE)
  X  H  H  L  L    L   L   L   H   L   H    H  L  L  L ;   (READ CYCLE)
  C  H  H  L  L    H   L   L   H   L   H    L  H  L  L ; 3
  C  H  H  L  L    H   L   L   L   L   L    H  H  L  L ; 4
  C  H  H  L  L    H   L   H   H   H   H    L  L  H  L ; 5
  C  H  H  L  H    H   L   H   H   H   H    H  L  H  L ; 6
  C  H  H  L  H    H   L   L   L   L   L    L  H  H  L ; 7
  C  H  H  L  H    H   L   H   L   H   L    H  H  H  L ; 8
;
  C  H  L  H  H    X   X   Z   Z   Z   Z    L  L  L  L ; 1 (NO /CS)
;
-------------------------------------------------------------------
```

DESCRIPTION:

INPUT SIGNALS:

CLK 16 MHZ SYSTEM CLOCK OF THE 82284 SYSTEM TIMING CONTROLLER. THIS CLOCKS TRIGGERS THE D-FLIP-FLOPS OF FOUR PAL OUTPUTS

/CS ACTIVE LOW UNLATCHED CHIP SELECT OF THE ADDRESS DECODER

CSL ACTIVE HIGH LATCHED CHIP SELECT. IT HAS TO BE ACTIVE TO THE RISING EDGE OF ALE OF THE NEXT CYCLE

ALE ADDRESS LATCH ENABLE OF THE 82288 BUS CONTROLLER

A1,A2 DEMULTIPLEXED ADDRESS INPUTS. THEY CARRY THE 2-BIT REGISTER ADDRESS FOR THE DCP

/IORC INPUT/OUTPUT READ CONTROL OF THE 82288

/IOWC INPUT/OUTPUT WRITE CONTROL OF THE 82288

OUTPUT SIGNALS:

/Q1 INTERNAL STATE SIGNAL. IT IS DIVIDED BY TWO FROM CLK AND SYNCHRONIZED TO ALE

/Q2 INTERNAL STATE SIGNAL. IT IS DIVIDED BY TWO FROM /Q1 AND SYNCHRONIZED TO ALE. IT IS THE INVERTED DCP CLOCK (4MHZ). THE RIGHT EDGE OF Q2 IS SYNCHRONOUS TO THE DATA STROBES /IORC AND /IOWC, IF TWO WAIT STATES ARE INSERTED.

/Q3 INTERNAL STATE SIGNAL. IT IS DIVIDED BY TWO FROM /Q2 AND SYNCHRONIZED TO ALE. IT IS USED TO GENERATE THE SYNCHRONOUS READY (/SRDY) FOR THE 82284. EXTERNALLY IT HAS TO BE LOGICALLY AND'ED WITH THE THE LATCHED CHIP SELECT (CSL).

CMDLY COMMAND DELAY GOES ACTIVE FOR ONE CLOCK WIDTH TO DELAY THE DATA STROBES. THE AM9568 REQUIRES A DELAY BETWEEN ALE INACTIVE AND DATA STROBE ACTIVE.

BIDIRECTIONAL SIGNALS:

D1,D2 DEMULTIPLEXED DATA BUS LINES TO 8Ø86 CPU

AD1,AD2 MULTIPLEXED ADDRESS/DATA BUS LINES FOR THE DCP

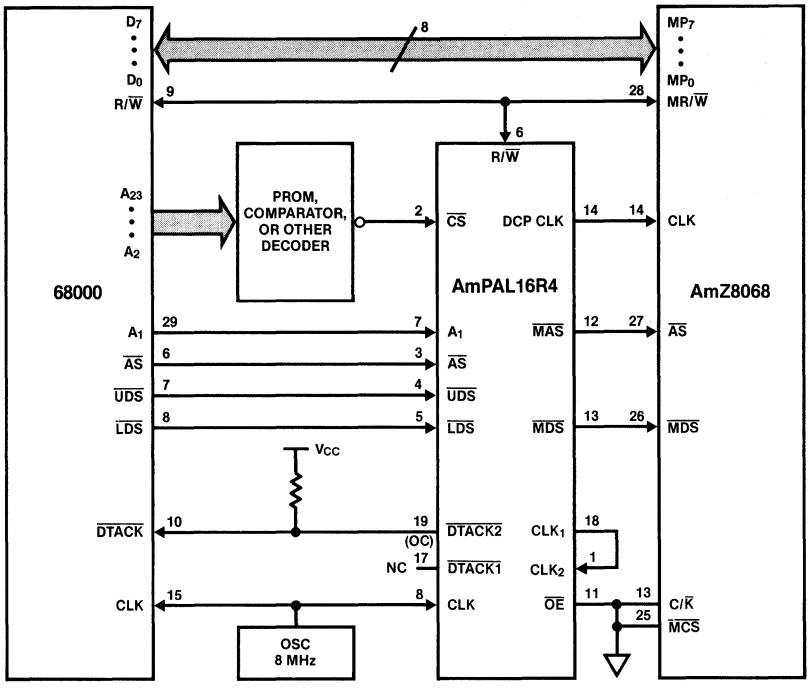


Figure 4.14. AmZ8068 to 68000 Connection Using a PAL

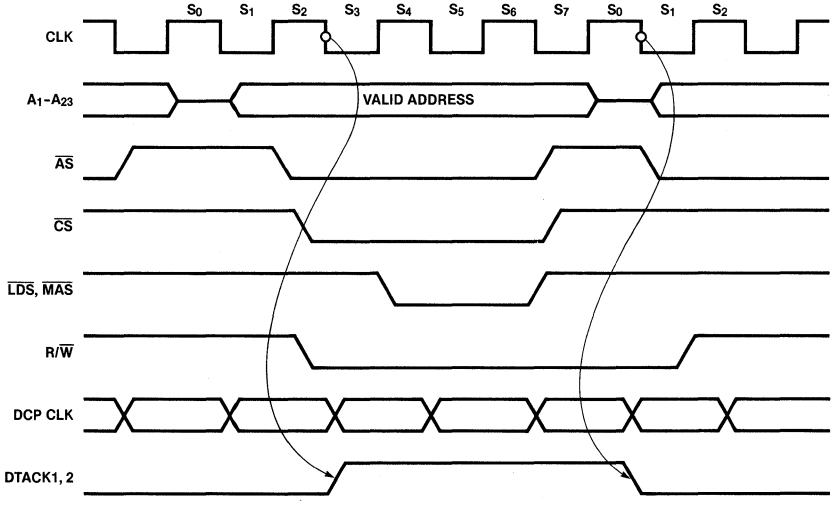


Figure 4.15. 68000—AmZ8068 Address Latch Cycle (A_1 = Low)

4.4. 68000 - AmZ8068

This two-chip solution adds high-speed data ciphering to a 68000-based system. About 500 kByte/s are possible in a CPU-controlled transfer. The ciphering rate can be increased with a sophisticated DMA controller or with several DCPs operating in parallel.

In the application described below, the CPU operates at 8 MHz and the DCP operates synchronously at 4 MHz. The interface controller, a PAL device, generates the Address and Data Strobes for the DCP and the Data Acknowledge for the CPU. It also divides the CPU clock by two and synchronizes it to the Data Strobes.

Programming

Data transfers between the CPU and the DCP are accomplished by a two-cycle operation. First the address of an internal register is latched in, then the data is transferred. This causes a small overhead in the initialization phase, but improves the ciphering rate in a high-speed data ciphering session. The rate of 500 kByte/s can be reached only if a high-speed peripheral device is connected to the Slave Port and the DCP is programmed for dual-port configuration.

The I/O Addresses

The PAL device is programmed to allow only CPU transfers to the DCP. A_0 must be odd to make the CPU transfer the data on the Low byte of the data bus.

A "0" on A_1 indicates an Address Latch Cycle, whereas a "1" on A_1 indicates a Data Transfer Cycle. A_0 must be "1" in both cycles.

Interface Descriptions

Figure 4.14 shows the 68000-DCP interface. Figures 4.15, 4.16, and 4.17 show the interface timing.

An address decoder generates the Chip Select for the DCP. The Address Strobe indicates a valid address. The PAL device is only activated if the Lower Data Strobe becomes active while the Upper Data Strobe stays inactive. This means that data is transferred in MOVE.B instructions with an odd peripheral address.

The PAL device provides two Data Acknowledge outputs. $\overline{DTACK_1}$ is an active Low TTL output. $DTACK_2$ has the same timing as $\overline{DTACK_1}$, but is an Open Collector output. (The Open Collector output is realized by a three-state output which assumes only two states, Low or Floating.)

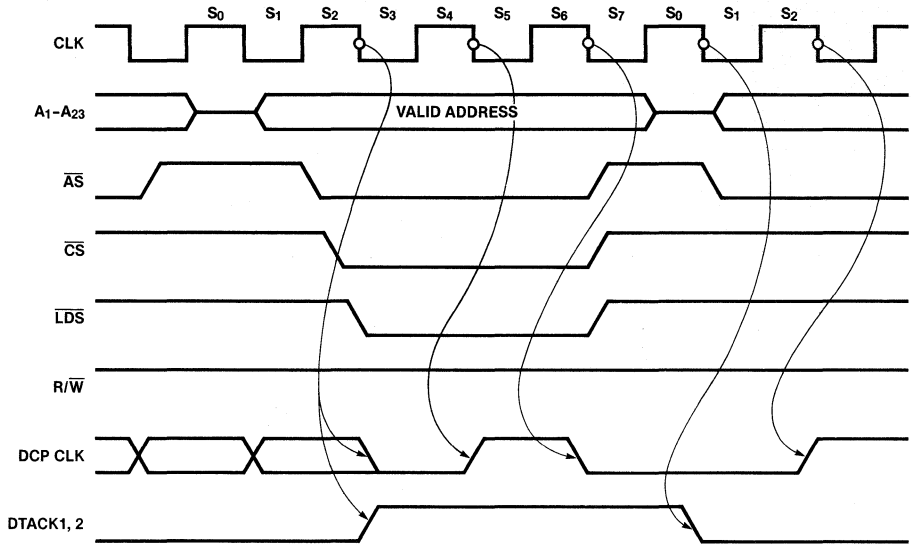


Figure 4.16. 68000—AmZ8068 Data Read Cycle (A_1 = High)

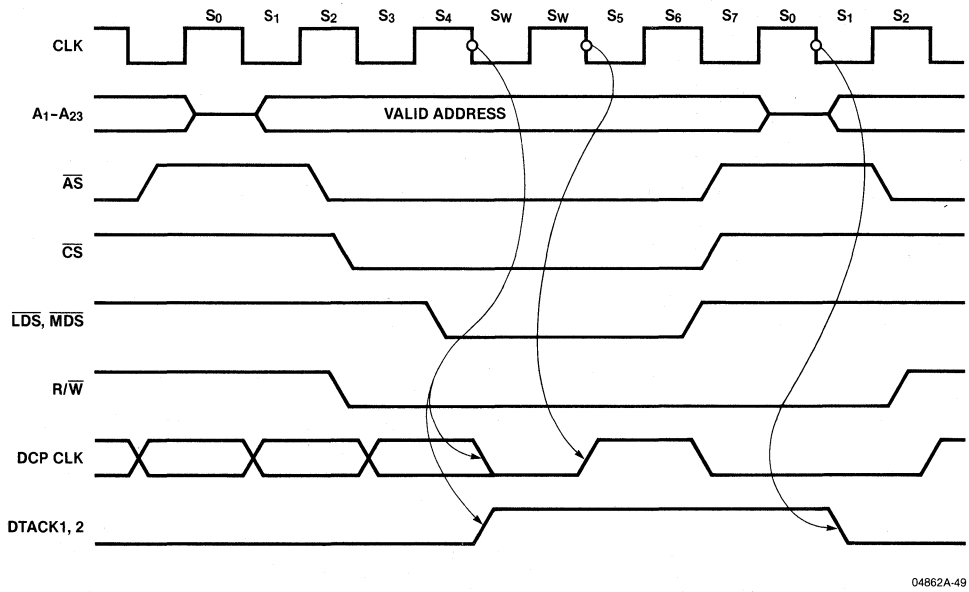


Figure 4.17. 68000—AmZ8068 Data Write Cycle (A_1 = High)

Address Latch Cycle

In this cycle only a Master Port Address Strobe ($\overline{MAS}$) is generated. Master Port Chip Select ($\overline{MCS}$) is tied to Low. $\overline{LDS}$ is sent to the $\overline{MAS}$ output. The minimum pulse width of $\overline{LDS}$ is 115 ns; 80 ns are required for the AmZ8068.

$\overline{DTACK}$ is activated with the falling edge of the CPU clock after cycle S_2. The CPU inserts no Wait states. $\overline{DTACK}$ is deactivated with the first edge of CLK after $\overline{AS}$ becomes inactive.

Data Write Cycle

A Data Write Cycle is performed when A_0 is High, $\overline{AS}$, $\overline{CS}$ and $\overline{LDS}$ are Low. The minimum pulse width of $\overline{LDS}$ is not sufficient for the DCP which requires at least 125 ns. One Wait state or a slower system clock will satisfy this parameter. One Wait State is inserted by activating $\overline{DTACK}$ at the end of S_4.

The DCP clock is synchronized in Data Read or Write Cycles by forcing it Low when $\overline{DTACK}$ becomes active. This guarantees that the DCP clock has a falling edge just before $\overline{LDS}$ ($\overline{MDS}$) rises. The delay of the DCP clock to CLK is typically 8 ns for a normal-speed PAL device. The delay of $\overline{LDS}$ to $\overline{MDS}$ is typically 12 ns. The delay of $\overline{LDS}$ to the system clock is 0 to 70 ns for the 8-MHz version. This results in a delay of 4 to 74 ns of $\overline{MDS}$ to the DCP clock. The DCP requires 0 to 50 ns when operating at the maximum clock rate.

This problem is solved by stretching the clock one cycle. The DCP clock stays Low for two cycles in the end of a transfer cycle. This is done automatically by the PAL device (see Figure 4.17).

Data Read Cycle

The generation of $\overline{MDS}$ in a Data Read Cycle is similar to the Data Write Cycle. Because the CPU activates $\overline{LDS}$ one cycle earlier, there is no need for a Wait State. The minimum pulse width of $\overline{LDS}$ is 240 ns; the DCP requires 200 ns for a Status Register read. DTACK is activated using the same logical condition as in the Data Write Cycle. Because of the earlier activation of $\overline{LDS}$, DTACK becomes active earlier and the CPU inserts no Wait States.

```
PAL16R4                                          PAL DESIGN SPECIFICATION
DCP044                                           JUERGEN STELBRINK  8-24-83
68000 - AmZ8068 (DCP) INTERFACE DEVICE
ADVANCED MICRO DEVICES

CLK2  /CS   /AS   /UDS    /LDS  RW   A1       CLK   NC       GND
/OE   /MAS  /MDS  DCPCLK  NC    NC   /DTACK1  CLK1  /DTACK2  VCC

/CLK1   =  CLK           ; INVERT CLOCK TO TRIGGER THE REGISTERED
                         ; OUTPUTS WITH THE FALLING EDGE OF CLK

MAS     =  AS*LDS*/UDS*/RW*/A1*CS

MDS     =  AS*LDS*/UDS*A1*CS

/DCPCLK := DCPCLK +                      ; DIVIDE BY TWO
           /DTACK1*CS*AS*LDS*/UDS +
           DTACK1*/AS*/LDS*/UDS          ; TWO CLOCKS LOW IN
                                         ; THE END OF A DATA CYCLE

DTACK1  := AS*LDS*/UDS*A1*CS +           ; DATA TRANSFER CYCLE
           AS*/RW*/A1*CS                 ; ADDRESS LATCH CYCLE

IF (DTACK1*AS*CS) DTACK2 = DTACK1


FUNCTION TABLE

CLK2 CLK CLK1 /CS /AS /LDS /UDS RW A1
DCPCLK /MAS /MDS /DTACK1 /DTACK2

;                                                  /   /
;                                        D         D   D
;                                        C         T   T
; C     C         /   /          P   /   /   A   A
; L  C  L  /  /  L  U          C   M   M   C   C
; K  L  K  C  A  D  D  R  A    L   A   D   K   K
; 2  K  1  S  S  S  S  W  1    K   S   S   1   2     COMMENT
-------------------------------------------------------------------
; CLOCK INVERT
  X  L  H  X  X  X  X  X  X    X   X   X   X   X
  X  H  L  X  X  X  X  X  X    X   X   X   X   X
; DATA WRITE CYCLE
  C  X  X  L  H  H  H  H  H    X   H   H   H   Z  ; S0
  C  X  X  L  L  H  H  L  H    X   H   H   H   Z  ; S2
  C  X  X  L  L  L  H  L  H    L   H   L   L   L  ; S4
  C  X  X  L  L  L  H  L  H    H   H   L   L   L  ; SW (1 WAIT STATE)
  C  X  X  L  L  L  H  L  H    L   H   L   L   L  ; S6
  X  X  X  L  H  H  H  L  H    L   H   H   L   Z  ; S7
  C  X  X  H  H  H  H  L  X    L   H   H   H   Z  ; S0
  C  X  X  H  L  X  H  X  X    H   H   H   H   Z  ; S2
; DATA READ CYCLE
  C  X  X  H  H  H  H  H  H    X   H   H   H   Z  ; S0
  C  X  X  L  L  L  H  H  H    L   H   L   L   L  ; S2
  C  X  X  L  L  L  H  H  H    H   H   L   L   L  ; S4
```

```
  C  X  X  L  L  L  H  H  H    L  H  L  L  L ; S6
  X  X  X  L  H  H  H  H  H    L  H  H  L  Z ; S7
  C  X  X  L  H  H  H  H  H    L  H  H  H  Z ; SØ
  C  X  X  X  H  H  H  H  H    H  H  H  H  Z ; S2
; ADDRESS LATCH CYCLE
  C  X  X  L  L  H  H  L  L    X  H  H  L  L ; S2
  C  X  X  L  L  L  H  L  L    X  L  H  L  L ; S4
  C  X  X  L  L  L  H  L  L    X  L  H  L  L ; S6
  X  X  X  L  H  H  H  L  L    X  H  H  L  Z ; S7
  C  X  X  X  H  H  H  L  L    X  H  H  H  Z ; SØ
;
```

DESCRIPTION:

INPUT SIGNALS:

CLK2 CLOCK FOR THE REGISTERED OUTPUTS OF THE PAL. IT IS CONNECTED TO CLK1

CLK 8 MHZ 68ØØØ SYSTEM CLOCK

/CS CHIP SELECT FOR DCP (A2-A23 ARE RELEVANT)

/AS ADDRESS STROBE

/LDS LOWER DATA STROBE USED TO TIME THE MASTER PORT DATA STROBE

/UDS UPPER DATA STROBE HAS TO BE INACTIVE DURING ALL TRANSFERS

A1 ADDRESS BIT 1 DISTINGUISHES BETWEEN ADDRESS LATCH AND DATA TRANSFER CYCLES

A1=LOW ADDRESS LATCH
A1=HIGH DATA TRANSFER

RW READ/ WRITE CONTROL

OUTPUT SIGNALS:

/MAS MASTER PORT ADDRESS STROBE

/MDS MASTER PORT DATA STROBE

CLK1 INVERTED CLOCK CLK

/DTACK1 LOW ACTIVE DATA ACKNOWLEDGE FOR 68ØØØ
ONE WAIT STATE IS INSERTED IN A DATA WRITE CYCLE

/DTACK2 LOW ACTIVE DATA ACKNOWLEDGE FOR 68ØØØ (OPEN COLLECTOR)

DCPCLK 4 MHZ DCP CLOCK, IT IS SYNCHRONIZED TO THE MASTER PORT DATA STROBE. IN A DATA TRANSFER CYCLE DCPCLK STAYS TWO CLK CYCLES LOW TO DELAY THE FIRST RISING EDGE OF THE DCPCLK TO TH DATA STROBES. IT IS DONE TO SATISFY TIMING PARAMETER 45 OF THE DCP PRODUCT SPECIFICATION.

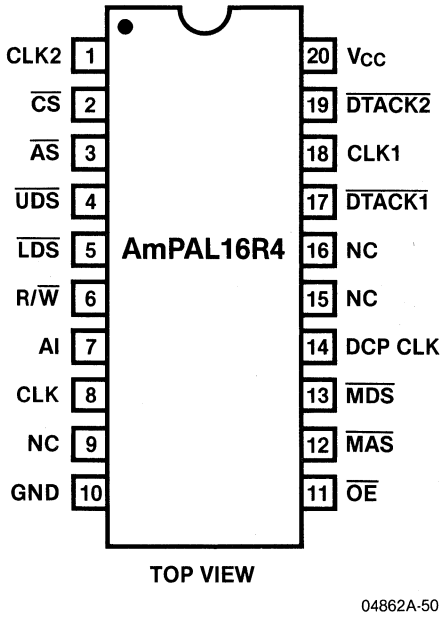


Figure 4.18. AmPAL16R4 Connection Diagram

4.5. Z8000 - AmZ8068

Figure 4.19 shows an interface between a 4-MHz Z8001/2* microprocessor and the AmZ8068. The CPU and the DCP can operate synchronously at a clock rate up to 3.5 MHz. All control and strobe signals can be connected directly to the DCP.

The clock rate is reduced to 3.5 MHz to satisfy timing parameter 45. The delay time from clock falling to Data Strobe (DS) rising is specified at 0 to 70 ns; the DCP requires 0 to 50 ns at 4 MHz. By reducing the clock rate, this parameter becomes 0 to 70 ns at 3.5 MHz.

The system can operate at 4 MHz, if a 10-MHz Z8001/2 is used. This faster version is specified for 0 to 45 ns.

A Sample Program

A universal program for testing the DCP is included at the end of this chapter. The program is written in Z8002 (nonsegmented) assembly language. The DCP must be initialized for Multiplexed Control Mode and "Master Port only" configuration. The ciphering mode can be ECB or CBC. The mode is defined by the variable "MODE". A one-cycle operation of the interface is assumed. For a two-cycle operation interface, instructions to latch the register address must be added.

Structure of the Program

Some variable fields are located in the beginning of the program:

DCP-OUT 32-byte buffer for the ciphered text

DCP-IN 32-byte buffer for the clear input text; the information to be ciphered must be loaded here before starting the program

CIVE 8-byte buffer for the CBC Initial Vector (IV) for encryption

CE-KEY 8-byte buffer for the encryption key (for ECB and CBC)

MODE defines mode of operation (18_H = ECB, $1A_H$ = CBC)

DATAREG address of Data Register (AD_1=0, AD_2=0)

CSREG address of Command/Status Register (AD_1=1, AD_2=0)

MODEREG address of Mode Register (AD_1=1, AD_2=1)

*Z8001/2 are trademarks of Zilog, Inc.

First, the DCP is reset by loading the Mode Register. The IVE Register is loaded by issuing command "85_H", "Load Clear IVE through Master Port", and strobing in eight bytes of data. The E Key Register is loaded in a similar way. The command is "11_H", "Load Clear E Key through Master Port". Loading of the IVE Register is not required for ECB. After entering these load commands, the Command pending bit of the Status Register becomes active until the eighth byte is strobed in.

The data ciphering session is started by writing "41_H", "Start Encryption" to the Command Register. The Command Pending bit becomes active and stays active until a stop command is entered or the DCP is reset. The Master Port Flag ($\overline{MFLG}$) and the Slave Port Flag ($\overline{SFLG}$) can be monitored to see whether the DCP is ready for input or output of data. In this sample program, these flags are not monitored because the structure of the program and the speed of the CPU guarantee that there are at least 5 DCP clocks between input or output of succeeding blocks.

This program operates the DCP in pipelined mode. First, two blocks of clear data are loaded into the chip, then the first block is read out. During input of the second block, the algorithm unit ciphers the first block. When the eight bytes of the second block are loaded, the first block is ready to be read out. The CPU can put data in and read data out without having to wait for the algorithm unit to cipher the data.

After ciphering four blocks, a stop command is entered. The result is stored in the field "DCP-OUT".

Improvements

If the DCP should be interfaced to a faster Z8000, the designer must take particular care that:

- the Address Strobe width does not become too narrow,

- the Data Strobe width does not become too narrow for Status Register read operations (a Wait State might be inserted),

- $\overline{MDS}$ is synchronous to the DCP clock.

Three approaches are discussed in more detail below. The interface logic of these interfaces may be integrated into one PAL device. Ideas of realization can be found in the other chapters.

8-MHz Z8000 - AmZ8068

- Use two-cycle operation.

- Divide clock by two.

- Synchronize clock to $\overline{DS}$.

- For Status Register reads, one additional Wait state must be inserted.

8-MHz Z8000 - Am9568

- Use multiplexed address/data bus of CPU; the Am9568 accepts the narrow Address Strobe directly.
- Transform R/$\overline{W}$ and $\overline{DS}$ into $\overline{MRD}$ and $\overline{MWR}$.
- Divide system clock by two and synchronize it to $\overline{MRD}$ or $\overline{MWR}$.
- Keep the DCP clock Low for two clock cycles at the end of the transfer cycle to satisfy the critical timing parameter 45 (0 to 30 ns) (see 68000-DCP interface).
- Insert Wait State for Status Register read operations.

Z8000 - AmZ8068

- DCP and CPU operate asynchronously with separate clocks.
- Design interface analogous to "iSBX Bus - DCP".
- Use two-cycle transfer mode.
- Less efficient CPU-DCP transfer, but no restrictions for system clock rate.

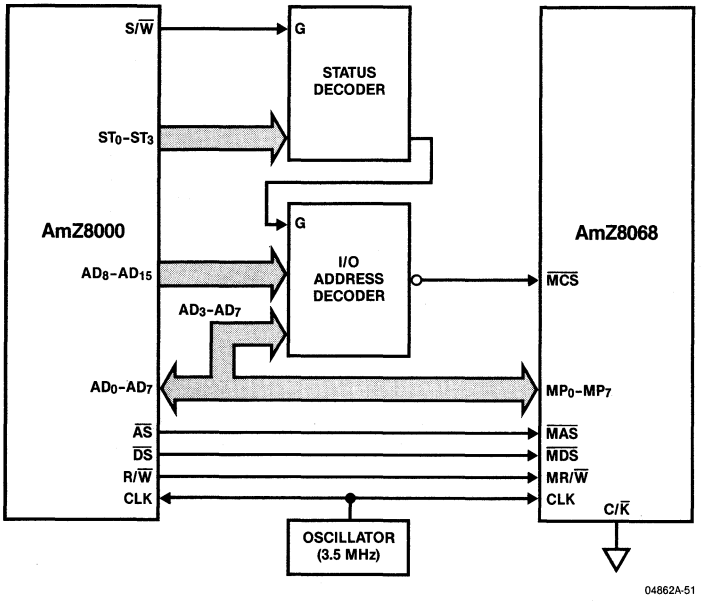


Figure 4.19. Z8000-AmZ8068 Interface

```
0000                    %*********************************************************************
0000                    %*                                                          JS  3/12/84 *
0000                    %*        ENCRYPTION EXAMPLE FOR Z8000                                 *
0000                    %*                                                                     *
0000                    %*********************************************************************
0000
0000                         PROGRAM DCP_SHOW;
0000                         ORIGIN  #1000;
1000
1000                    DCP_OUT:        BYTE (32);      % DCP OUTPUT STORAGE AREA
1020                    DCP_IN:         BYTE (32);      % DCP INPUT STORAGE AREA
1040                    CIVE:           BYTE (8);       % CLEAR IV STORAGE FOR CBC/CFB ENCRYPTION
1048                    CE_KEY:         BYTE (8);       % CLEAR ENCRYPTION KEY
1050                    MODE:           BYTE (1);       % MODE VALUE
1051   00
1052                    DATAREG:        WORD (1);       % DATA REGISTER ADDRESS (MASTER PORT)
1054                    CSREG:          WORD (1);       % COMMAND/STATUS REGISTER ADDRESS
1056                    MODEREG:        WORD (1);       % MODE REGISTER ADDRESS
1058
1058                    DCP_SHOW:
1058   6103 1052                LD      R3,DATAREG;     % LOAD DATA REGISTER ADDRESS
105C   6101 1054                LD      R1,CSREG;       % LOAD COMMAND/STATUS REGISTER ADDRESS
1060   6102 1056                LD      R2,MODEREG;     % LOAD MODE REGISTER ADDRESS
1064   600F 1050                LDB     RL7,MODE;       % LOAD MODE VALUE
1068   3E2F                     OUTB    R2,RL7;         % SET MODE (INCLUDES SOFTWARE RESET)
106A
106A                    % LOAD IVE REGISTER
106A   CFA5                     LDB     RL7,#A5;        %IVE LOAD COMMAND
106C   3E1F                     OUTB    R1,RL7;
106E   2108 0008                LD      R8,#8;          % BYTE COUNTER
1072   2109 1040                LD      R9, CIVE;       % ADDRESS OF CLEAR IVE FIELD
1076   3A92 0830                OTIRB   R3,R9 ,R8;      % STROBE 8 BYTE IV IN
107A
107A                    % LOAD E KEY REGISTER
107A   CF11                     LDB     RL7,#11;        % LOAD E KEY COMMAND
107C   3E1F                     OUTB    R1,RL7;
107E   2108 0008                LD      R8,#8;          % BYTE COUNTER
1082   2109 1048                LD      R9, CE_KEY;     % ADDRESS OF CLEAR E KEY FIELD
1086   3A92 0830                OTIRB   R3,R9 ,R8;      % STROBE 8 BYTES KEY IN
108A
108A                    % ENCRYPTION SESSION
108A   CF41                     LDB     RL7,#41;        % START ENCRYPTION COMMAND
108C   3E1F                     OUTB    R1,RL7;
108E   2108 0008                LD      R8,#8;          % BYTE COUNTER
1092   2109 1020                LD      R9, DCP_IN;     % DATA INPUT FIELD
1096   3A92 0830                OTIRB   R3,R9 ,R8;      % TRANSFER FIRST BLOCK
109A   2108 0008                LD      R8,#8;          % BYTE COUNTER
109E   3A92 0830                OTIRB   R3,R9 ,R8;      % TRANSFER SECOND BLOCK
10A2   2108 0008                LD      R8,#8;          % BYTE COUNTER
10A6   210A 1000                LD      R10, DCP_OUT;   % DATA OUTPUT FIELD
10AA   3A30 08A0                INIRB   R10 ,R3,R8;     % READ FIRST CIPHERED BLOCK BACK
10AE   2108 0008                LD      R8,#8;          % BYTE COUNTER
10B2   3A92 0830                OTIRB   R3,R9 ,R8;      % TRANSFER THIRD BLOCK
10B6   2108 0008                LD      R8,#8;          % BYTE COUNTER
10BA   3A30 08A0                INIRB   R10 ,R3,R8;     % READ SECOND CIPHERED BLOCK BACK
10BE   2108 0008                LD      R8,#8;          % BYTE COUNTER
10C2   3A92 0830                OTIRB   R3,R9 ,R8;      % TRANSFER FOURTH BLOCK
10C6   2108 0008                LD      R8,#8;          % BYTE COUNTER
10CA   3A30 08A0                INIRB   R10 ,R3,R8;     % READ THIRD CIPHERED BLOCK BACK
10CE   2108 0008                LD      R8,#8;          % BYTE COUNTER
10D2   3A30 08A0                INIRB   R10 ,R3,R8;     % READ FOURTH CIPHERED BLOCK BACK
10D6
10D6                    % TERMINATE CIPHERING SESSION
10D6   CFE0                     LDB     RL7,#E0;        % LOAD STOP COMMAND
10D8   3E1F                     OUTB    R1,RL7;         % ISSUE STOP COMMAND
10DA
10DA                            END.
```

4.6. Z8Ø* - Am9518/AmZ8Ø68

This chapter shows in two examples how the Data Ciphering Processor (DCP) can be interfaced to a Z8Ø (Z8ØA, Z8ØB) CPU. All interface control signals are generated by one PAL device.

In CPU transfer mode a ciphering speed up to 28Ø kByte/s can be reached. A Z8ØA DMA controller can double this value. Chapter 4.8 (Z8Ø-DMA-DCP) shows how to increase the speed to 1.1 MByte/s.

The multiplexed address/data bus of the DCP is simulated using a two-cycle operation mode. An output instruction to an even address ($A_Ø$=Low) selects one of the internal registers of the DCP. In all subsequent I/O operations with $A_Ø$=High, the CPU can transfer data to or from DCP registers. The register address stays latched in the chip until the next Address Strobe latches in a new address. The Address Latch Cycle does not represent significant overhead in an encryption or decryption session because, once the DCP is initialized and the data register is selected, no further Address Latch Cycle is needed.

I/O addresses:	XXXX XXXØ	-	Address Latch Cycle
	XXXX XXX1	-	Data Transfer Cycle

X - user definable

The AmPAL16R4 device controls the interface timing. It generates the synchronized strobe signals for the DCP and the Wait for the CPU to extend the cycles.

The PAL device is programmed to allow two operation modes. In Mode A the DCP works with the same clock rate as the CPU. Mode B increases the ciphering speed by allowing higher than 4-MHz system clock rates for the CPU. In this mode, the PAL device provides half the system clock rate for the DCP.

A system with a Z8ØB at 6 MHz and an AmZ8Ø68 at 3 MHz increases the ciphering speed compared to a system where both the CPU and the DCP clock are 4 MHz; the limiting factor is the data transfer capability of the CPU.

The key requirement in interfacing the DCP to a Z8Ø CPU is to meet the timing relationship between the Master Port Data Strobe ($\overline{MDS}$) and the DCP clock. The rising edge of $\overline{MDS}$ must be synchronous to the falling edge of the clock.

The Operation Modes

Mode A: Both the Z8Ø CPU and the DCP are operating synchronously at the same frequency. The DCP clock is inverted. This mode can be used with system clocks up to 4 MHz. No extra Wait states are inserted.

*Z8Ø is a trademark of Zilog, Inc.

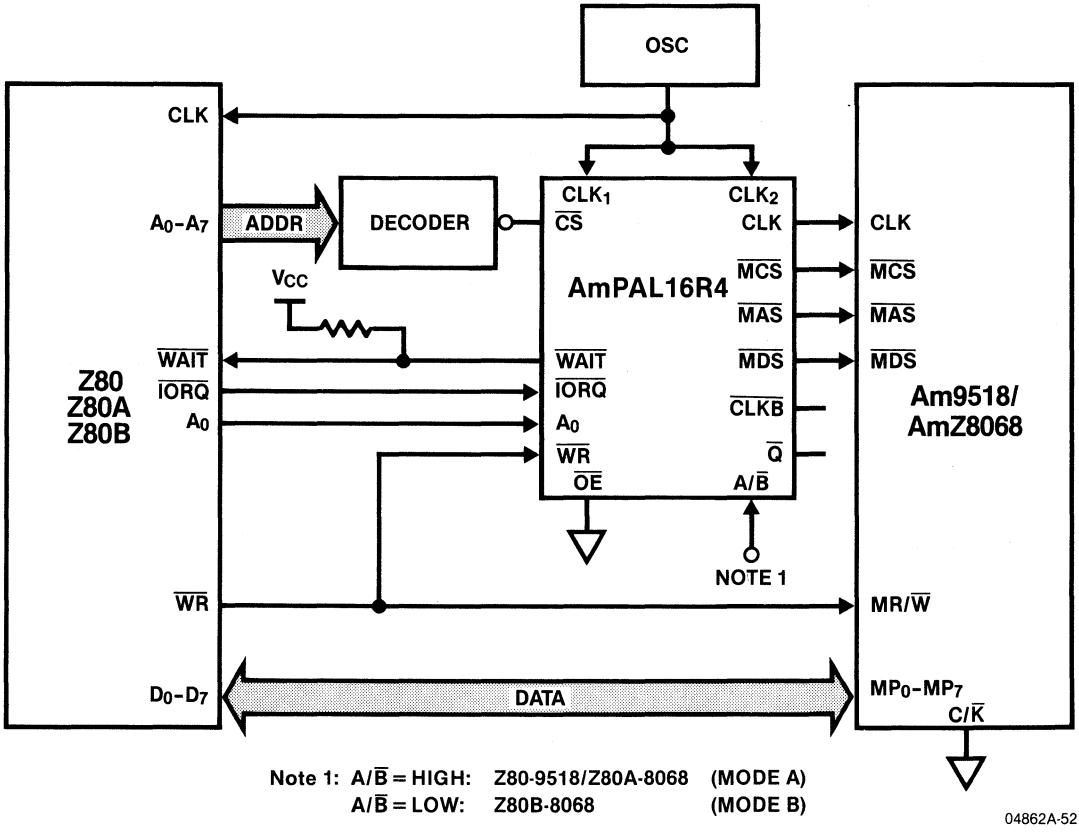


Figure 4.20. Z80-DCP Interface

Mode B: To get higher ciphering throughput, the data transfer speed of the Z8Ø bus should be increased by using a higher system clock rate. In Mode B the PAL device divides the system clock by two to generate the DCP clock. The DCP clock is synchronized to the $\overline{MDS}$ by delaying the clock one half cycle if they are not in phase (Figures 4.23 and 4.24). During a Data Write Cycle, one extra Wait state is inserted. An AmZ8Ø68 must be used in this mode even at a DCP clock rate of 3 MHz because of its faster register access time.

Figure 4.2Ø shows the interface. The A/$\overline{B}$ input of the PAL device is wired High to select Mode A or Low to select Mode B.

The Interface Timing

Address Latch Cycle: (Figures 4.21 and 4.22)

Master Port Chip Select ($\overline{MCS}$) is active when $\overline{IORQ}$ and $\overline{CS}$ are active Low and $A_Ø$=Low (even address). Master Port Address Strobe ($\overline{MAS}$) is strobed Low for one system clock cycle during the automatically inserted Wait cycle T_W to meet the hold time requirement of $\overline{MAS}$ High to $\overline{MCS}$ High (parameter 35).

Data Read Cycle: (Figures 4.21 and 4.22)

A Data Read Cycle reads the register whose address was latched in the previous Address Latch Cycle. $\overline{MCS}$ and $\overline{MAS}$ are inactive the whole cycle. $\overline{MDS}$ is active during the last two clock cycles, T_W and T_3. In both A and B Modes, no Wait state is inserted. $\overline{WR}$ and $A_Ø$ must be High. In Mode B the DCP clock is set High in the beginning of T_3 using an internal signal $\overline{Q}$ to synchronize the falling edge of the DCP clock to the rising edge of $\overline{MDS}$. $\overline{Q}$ is only active in Mode B during Wait state T_W. This interface meets the data hold time of the Z8Ø, because the data is stable to the beginning of T_1 of the next machine cycle.

Data Write Cycle:

In this cycle, the CPU can write one byte into the addressed register. $\overline{MCS}$ and $\overline{MAS}$ are inactive. $\overline{WR}$ is active and $A_Ø$ is High.

Mode A (Figure 4.2Ø)

$\overline{MDS}$ is strobed Low for T_W. The DCP reads the data in at the beginning of T_3. No Wait state is inserted.

Mode B (Figure 4.23)

$\overline{MDS}$ is strobed Low for the Wait cycle T_W and the additional Wait cycle T_W, to meet the minimum data strobe active time (parameter 44) of the DCP. The DCP reads the data in at the begin of T_3.

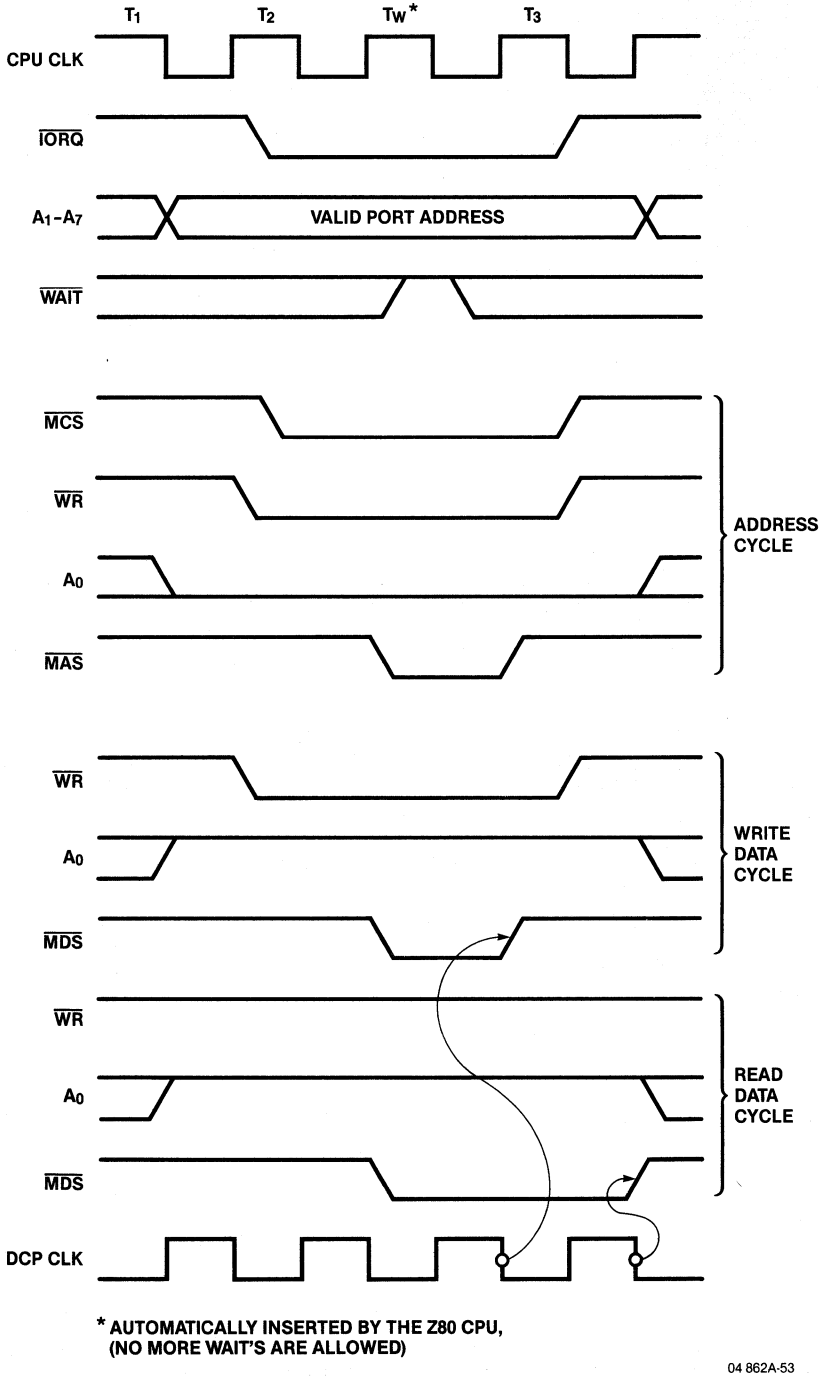


Figure 4.21. Z80-Am9518/Z80A-AmZ8068 Timing Diagram (Mode A)

Data Ciphering Speed

The byte transfer capability of the Z8Ø system bus limits the data ciphering throughput of the DCP. A Z8Ø DMA controller doubles the maximum throughput compared to a CPU-controlled transfer as indicated in the following table:

System Clk	DCP Clk	CPU	DCP	Mode	N	T
6 MHz	3 MHz	Z8ØB	AmZ8Ø68	B	168/176	Ø.28/Ø.27
4 MHz	4 MHz	Z8ØA	AmZ8Ø68	A	168	Ø.19
2.5 MHz	2.5 MHz	Z8Ø	Am9518	A	168	Ø.14

N = Number of DCP clock cycles to transfer and cipher 8 bytes of data. In CPU-controlled modes the use of the Z8Ø block transfer commands like INIR, INDR, OTIR or OTDR is assumed.

T = Throughput in MByte/s

The formula for calculating the throughput is:

T = (8 * f) / (N + m) MByte/s

f = DCP clock in MHz

8 = 8 bytes per block

m = Number of extra DCP clock cycles to get a minimum delay time of five clocks between transferring the last byte of one block and the first byte of the next block. In CPU controlled transfers m=Ø can be assumed, because the CPU has to evaluate instruction fetches and memory data transfers between two I/O accesses. MFLG indicates if the DCP accepts data transfer.

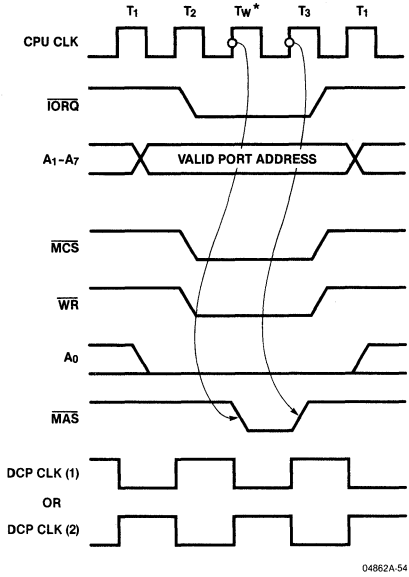


Figure 4.22. Address Latch Cycle (Mode B) (No Clock Synchronization)

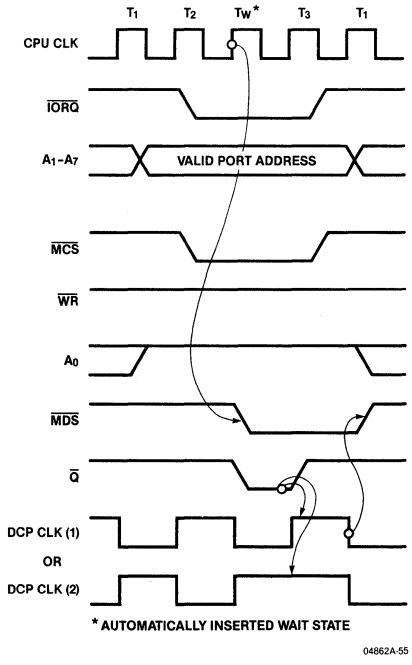


Figure 4.23. Data Read Cycle (Mode B)

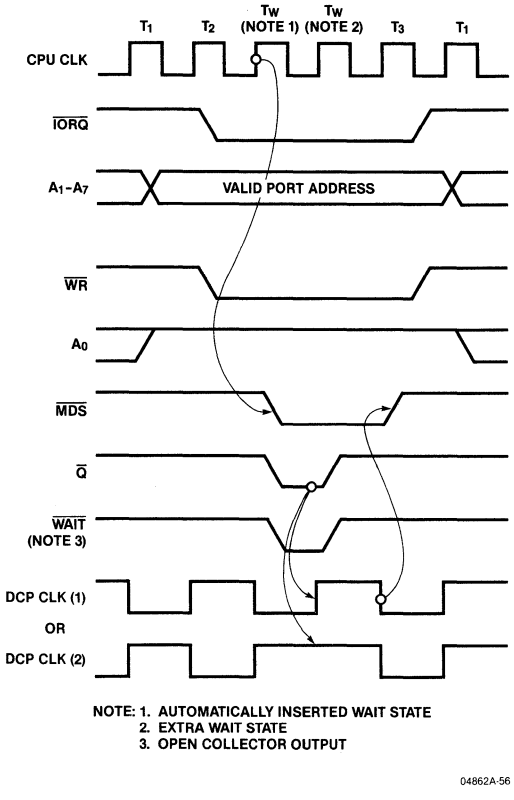


Figure 4.24. Data Write Cycle (Mode B)

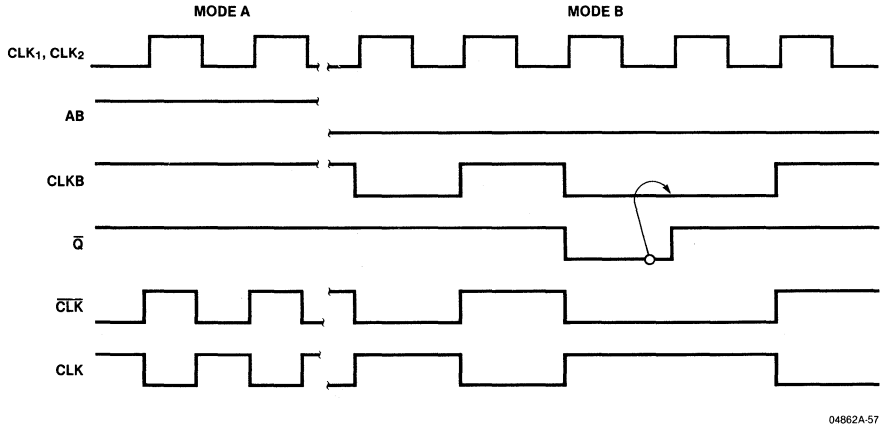


Figure 4.25. Clock Timing Diagram (Mode A and B)

```
PAL16R4                                          PAL DESIGN SPECIFICATION
DCP046                                           JUERGEN STELBRINK    5/2/83
Z8Ø- AM9518/AMZ8Ø68  INTERFACE CONTROLLER
ADVANCED MICRO DEVICES

CLK1  CLK2  /CS    /IORQ  AØ  /WR   AB    NC    NC   GND
/OE   NC    /WAIT  /CLKB  /Q  /MDS  /MAS  /MCS  CLK  VCC

MCS  =   IORQ*CS*/AØ                        ; MASTER PORT CHIP SELECT

MAS  :=  IORQ*CS*/AØ*WR*/MAS                ; MASTER PORT ADDRESS STROBE

MDS  :=  IORQ*CS*WR*/MDS*AØ*AB          +   ; WRITE DATA STROBE (MODE A)
         IORQ*CS*WR*AØ*/MDS*/Q*/AB      +   ; WRITE DATA STROBE (MODE B)
         IORQ*CS*WR*AØ*MDS*Q*/AB        +   ; WRITE DATA STROBE (MODE B)
         IORQ*CS*/WR*AØ                     ; READ  DATA STROBE (MODE A+B)

CLKB :=  /CLKB*/Q*/AB                       ; CLOCK FOR MODE B

/CLK =   CLK2*AB  +                         ; (MODE A)
         CLKB                               ; (MODE B)

Q    :=  IORQ*CS*/MDS*/Q*AØ*/AB             ; USED TO GENERATE MDS AND WAIT

IF (Q*WR) WAIT =   Q*WR                     ; WAIT TO Z8Ø

FUNCTION TABLE

 CLK1 CLK2 AB  /CS /IORQ  AØ  /WR       CLK /MCS /MAS /MDS /WAIT  /Q /CLKB

;                /                          /
; C  C          I                  /  /  /  W      C
; L  L      /   O      /     C  M  M  M  A      L
; K  K  A   C   R  A   W     L  C  A  D  I   /  K
; 1  2  B   S   Q  Ø   R     K  S  S  S  T   Q  B      COMMENT
------------------------------------------------------------------------------
;
; MODE A:  Z8Ø- AM9518 OR Z8ØA- AMZ8Ø68 INTERFACE
;          (DCP CLOCK = CPU CLOCK)
;
; CLOCK GENERATION
;
  X  L  H  X  X  X  X     H  X  X  X  Z  H  H
  X  H  H  X  X  X  X     L  X  X  X  Z  H  H
;
; ADDRESS LATCH
;
  H  X  H  H  H  X  H     X  H  H  H  Z  H  H    ; MACHINE CYCLE T1
  L  X  H  L  H  L  H     X  H  H  H  Z  H  H
  C  X  H  L  H  L  H     X  H  H  H  Z  H  H
  H  X  H  L  L  L  L     X  L  H  H  Z  H  H    ; CYCLE T2
  C  X  H  L  L  L  L     X  L  L  H  Z  H  H
  C  X  H  L  L  L  L     X  L  H  H  Z  H  H    ; CYCLE TW
  H  X  H  L  H  L  H     X  H  H  H  Z  H  H    ; CYCLE T3
  C  X  H  L  H  L  H     X  H  H  H  Z  H  H
```

```
;
; WRITE DATA OPERATION
;
  C  X  H  L  H  H  H     X  H  H  H  Z  H  H    ; CYCLE T1
  C  X  H  L  L  H  L     X  H  H  L  Z  H  H    ; CYCLE T2
  C  X  H  L  L  H  L     X  H  H  H  Z  H  H    ; CYCLE TW
  C  X  H  L  H  H  H     X  H  H  H  Z  H  H    ; CYCLE T3
;
; READ DATA OPERATION
;
  C  X  H  L  H  H  H     X  H  H  H  Z  H  H    ; CYCLE T1
  C  X  H  L  L  H  H     X  H  H  L  Z  H  H    ; CYCLE T2
  C  X  H  L  L  H  H     X  H  H  L  Z  H  H    ; CYCLE TW
  C  X  H  L  H  H  H     X  H  H  H  Z  H  H    ; CYCLE T3
;
; INVALID OPERATION (READ IN ADDRESS LATCH)
;
  C  X  H  L  L  L  H     X  L  H  H  Z  H  H    ; NO /MAS !
;
;
;              /                       /
; C  C         I           /  /  /     W     C
; L  L     /   O     /     M  M  M     A     L
; K  K  A  C  R  A  W     L  C  A  D  I  /  K
; 1  2  B  S  Q  Ø  R     K  S  S  S  T  Q  B     COMMENT
;-------------------------------------------------------------------------
;
; MODE B: Z8ØB- AMZ8Ø68 INTERFACE
;         (DCP CLOCK = CPU CLOCK/2)
;
; WRITE DATA OPERATION
;
  C  X  L  H  H  H  H     L  H  H  H  Z  H  L    ; CYCLE T1
  C  X  L  H  H  H  H     H  H  H  H  Z  H  H    ; CYCLE T2
  C  X  L  L  L  H  L     L  H  H  L  L  L  L    ; FIRST WAIT CYCLE (CLK=L)
  C  X  L  L  L  H  L     H  H  H  L  Z  H  H    ; SECOND WAIT CYCLE
  C  X  L  L  L  H  L     L  H  H  H  Z  H  L    ; CYCLE T3
;
  C  X  L  L  L  H  L     H  H  H  L  L  L  H    ; FIRST WAIT CYCLE (CLK=H)
  C  X  L  L  L  H  L     H  H  H  L  Z  H  H    ; SECOND WAIT CYCLE (SYNC !)
  C  X  L  L  L  H  L     L  H  H  H  Z  H  L    ; CYCLE T3
;
; READ DATA OPERATION
;
  C  X  L  H  H  H  H     H  H  H  H  Z  H  H    ; CYCLE T1
  C  X  L  H  H  H  H     L  H  H  H  Z  H  L    ; CYCLE T2
  C  X  L  L  L  H  H     H  H  H  L  Z  L  H    ; WAIT CYCLE
  C  X  L  L  L  H  H     H  H  H  L  Z  H  H    ; CYCLE T3 (SYNC!)
  C  X  L  L  H  H  H     L  H  H  H  Z  H  L    ; NEXT CYCLE
;
-------------------------------------------------------------------------
```

DESCRIPTION:

THIS PAL GENERATES ALL NECESSARY BUS CONTROL SIGNALS, TO INTERFACE THE AM9518 OR AMZ8Ø68 TO THE Z8Ø CPU WITH A SYSTEM CLOCK UP TO 6 MHZ.

2 INPUT AND 1 INPUT/ OUTPUT PINS ARE NOT USED, SO THAT FOR EXAMPLE A DATA BUS TRANSCEIVER CONTROL LOGIC CAN BE ADDED.

IN SYSTEMS WITH A CLOCK UP TO 4 MHZ, THE DCP RUNS DIRECTLY AT THIS FREQUENCY (MODE A, INPUT AB = HIGH).
IF THE FREQUENCY IS HIGHER, THE DCP IS DIVIDED BY TWO FROM THE SYSTEM CLOCK (MODE B, AB = LOW).

INPUT PINS:

CLK1, CLK2 — CLK1 IS THE CLOCK INPUT FOR THE FOUR INTERNAL D-FLIP-FLOPS. THEY ARE CLOCKED BY THE RISING EDGE OF CLK1. THE DCP DATA STROBE MUST BE SYNCHRONOUS TO THE FALLING EDGE OF THE CLOCK; THE INVERTED CLK2 IS THEREFORE SENT TO THE OUTPUT CLK. IN MODE B CLK2 IS SYNCHRONIZED BEFORE IT APPEARS ON THE CLK OUTPUT. BOTH INPUTS ARE CONNECTED TO THE Z8Ø SYSTEM CLOCK.

/CS — CHIP SELECT GENERATED BY AN ADDRESS DECODER LOGIC (ACTIVE LOW). IF /CS IS ONLY ACTIVE IN I/O CYCLES, THE /IORQ INPUT CAN BE WIRED LOW.

/IORQ — INPUT/ OUTPUT REQUEST OF THE Z8Ø (LOW ACTIVE)

AØ — LEAST SIGNIFICANT BIT OF THE Z8Ø ADDRESS BUS TO SELECT TYPE OF OPERATION:
- AØ= LOW — SELECT REGISTER FOR NEXT DATA CYCLES (ADDRESS LATCH)
- AØ= HIGH — READ OR WRITE INTERNAL REGISTER (DATA TRANSFER TO CONTROL, MODE, INPUT OR OUTPUT REGISTER)

/WR — WRITE SIGNAL OF THE Z8Ø, DEFINES DATA TRANSFER DIRECTION

AB
- AB= HIGH — MODE A
- AB= LOW — MODE B

OUTPUT SIGNALS:

/WAIT — ACTIVE LOW DURING FIRST WAIT CYCLE IN WRITE DATA OPERATION IN MODE B, TO GENERATE AN EXTRA WAIT STATE. THE OTHER TIME /WAIT IS IN THREE STATE.

/MCS — MASTER PORT CHIP SELECT, ONLY ACTIVE IN ADDRESS LATCH CYCLES

/MAS — MASTER PORT ADDRESS STROBE, ACTIVE IN ADDRESS CYCLES TO LATCH THE REGISTER ADDRESS AND /MCS IN. THE DCP STORES INTERNALLY THE ADDRESS AND THE CHIP SELECT TO THE NEXT ADDRESS LATCH CYCLE

/MDS	MASTER PORT DATA STROBE TO ENABLE DATA TRANSFER TO THE INTERNAL REGISTERS OF THE DCP
CLK	DCP CLOCK, IN MODE B SYNCHRONIZED TO THE MASTER PORT DATA STROBE (/MDS)
/CLKB	DCP CLOCK OUTPUT INTERNALLY USED FOR MODE B (NOT CONNECT)
/Q	INTERNAL STATUS SIGNAL (NOT CONNECT)

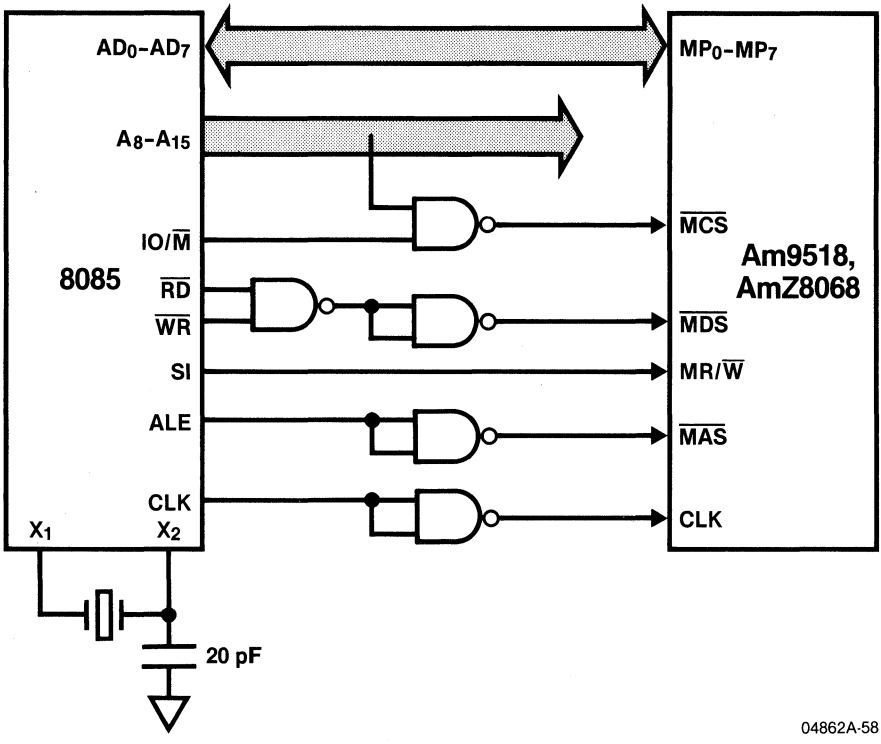


Figure 4.26. 8085-DCP Interface

4.7. 8085A - Am9518

Figure 4.26 shows the interface diagram between the 8085 microprocessor and the Am9518 Data Encryption device. The DCP and the CPU operate synchronously at a maximum clock rate of 2.2 MHz, considerably simplifying the interface requirements.

Interface Description

The 8-bit address/data bus of the CPU is directly connected to the Master Port of the DCP. The Master Port Data Strobe is driven by $\overline{RD}$ or $\overline{WR}$. The MR/$\overline{W}$ input of the DCP is connected to the status line S1 of the 8085. This line is High whenever the CPU executes a read instruction. The Master Port Address Strobe ($\overline{MAS}$) is the inverted Address Latch Enable (ALE). A decoded address and M/$\overline{IO}$=Low produces an active Low Master Port Chip Select. It is latched by $\overline{MAS}$.

The Clock

The DCP can operate with the inverted CPU clock if the clock is slowed down to satisfy the minimum High time requirement of the DCP. The 8085A data sheet gives a formula to determine the minimum clock High and Low times for slower clocks.

Minimum High time: 0.5 * T - 80 ns (T=clock cycle width)

This time must be at least 150 ns for a Am9518 and 115 ns for a AmZ8068, resulting in a maximum clock rate of 2.2 MHz and 2.5 MHz respectively.

Minimum Low time: 0.5 * T - 40 ns

It is 190 ns at 2.2 MHz.

The DCP requires that the $\overline{MDS}$ is synchronous to the clock. The range is 0 - TWL - 100 ns for the Am9518. TWL is the real Low time of the clock.

The 8085 timing specification does not specify a timing relationship between the clock and $\overline{RD}$ or $\overline{WR}$; the designer must verify.

Improvements

A more sophisticated interface avoids the missing timing specification and allows interfacing to a faster CPU. Ideas can be found in the iSBX Bus Interface (Chapter 4.10) or 68000 Interface (Chapter 4.4). The first shows a totally asynchronous operation of the DCP and the CPU; the second shows how to delay the rising edge of the clock following $\overline{MDS}$.

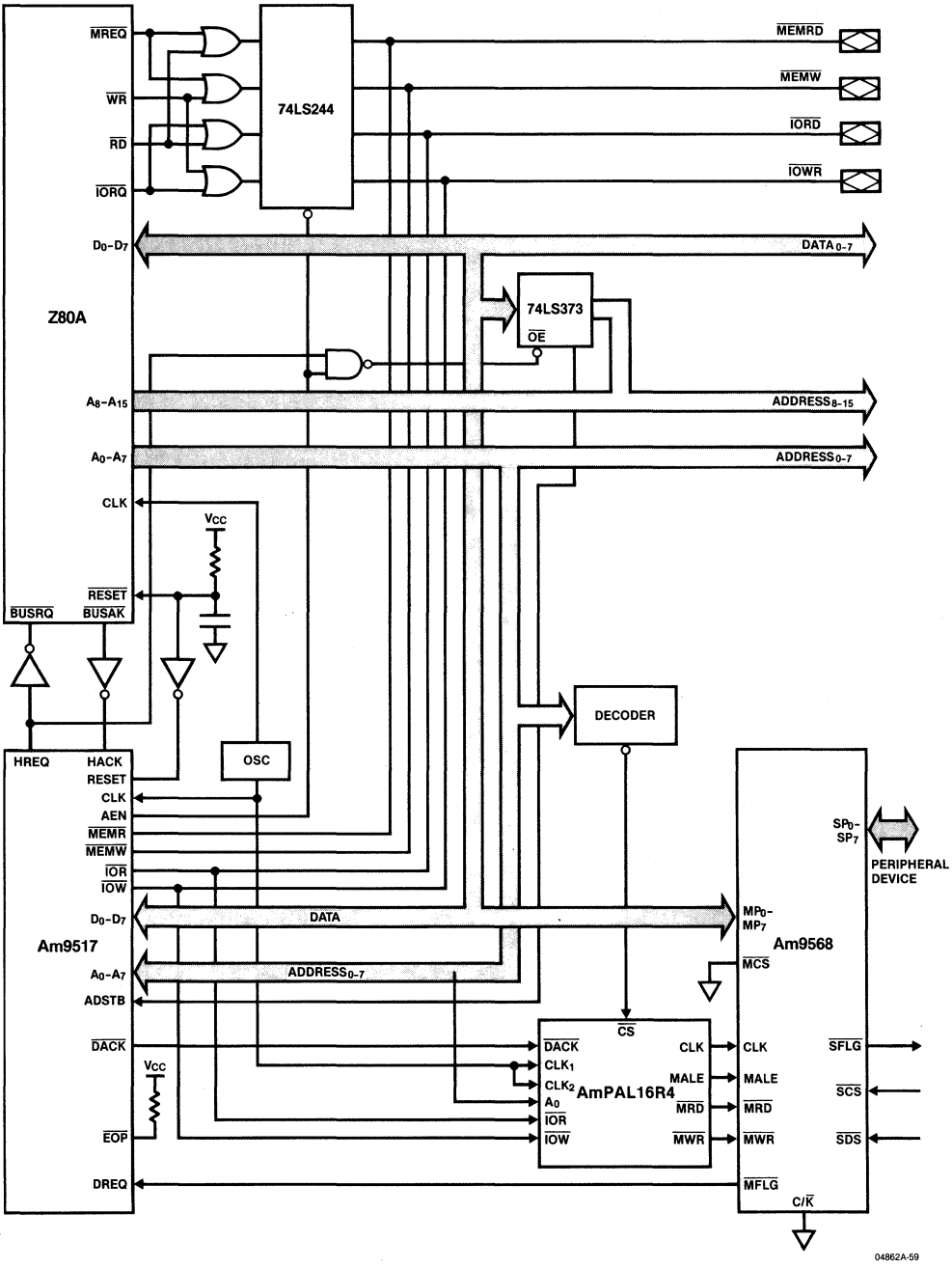


Figure 4.27. Logic Diagram

4.8. Z8Ø - DMA - Am9568

This application design shows how to increase the ciphering throughput to 89Ø kByte/s using the advanced 8-bit DMA Controller Am9517A-5 (also called the 8237-5). The host CPU is a Z8ØA (Figure 4.27).

The CPU sets up a data block in memory and programs the DMA controller to transfer this data block to the DCP via the Master Port. The DCP encrypts the data. A high-speed peripheral device can read out the ciphered data from the Slave Port. This dual-port configuration allows data input and output simultaneously and increases the throughput compared to a single-port configuration by a factor two. In the single-port configuration, only the Master Port is used for data transfer; it handles both the clear and ciphered data.

The multiplexed address/data bus of the DCP is simulated in a two-cycle operation. For output operation to an even address the PAL interface timing controller generates a Master Port Address Strobe ($\overline{MAS}$) to select one of the internal registers. Subsequent I/O operations to an odd address ($A_Ø$=High) transfer data to or from the preselected DCP register. During I/O operations to an odd address, the PAL device generates Master Port Data Strobes ($\overline{MRD}$ or $\overline{MWR}$). Before the DMA block transfer is started, the CPU must preselect the DCP data register. The register address of the data register is $ØØ_H$.

The DMA controller operates in "flyby" mode. Data is transferred on the system data bus one byte at a time from memory to the DCP or vice versa without going through a DMA register. An I/O Read ($\overline{IOR}$) and Memory Write ($\overline{MEMW}$) or I/O Write ($\overline{IOW}$) and Memory Read ($\overline{MEMR}$) are active at the same time. The DCP is selected by DMA Acknowledge ($\overline{DACK}$). The PAL device treats $\overline{DACK}$ as $\overline{CS}$ active and $A_Ø$=High. In this design the DMA controller can only execute data transfer cycles; it is not able to change the internal register address of the DCP.

The DMA controller is set up for Demand Transfer Mode. It releases the bus when the data request input goes inactive. The Master Port Flag ($\overline{MFLG}$) is wired to the data request input. The flag output goes active when the DCP is ready to accept data or the output data is ready to be read out. After transferring one block of data (8 bytes), this flag goes inactive until a new block can be put in or read out. The inactive time depends on the response time of the peripheral logic at the Slave Port. This flag is inactive a minimum of five clocks.

Speed

The DMA controller needs three clock cycles to transfer one byte. After each block transfer (8 bytes) the DMA controller releases the bus and requests it back if $\overline{MFLG}$ goes active again. This time is assumed to be 12 clocks. The ciphering of one block is

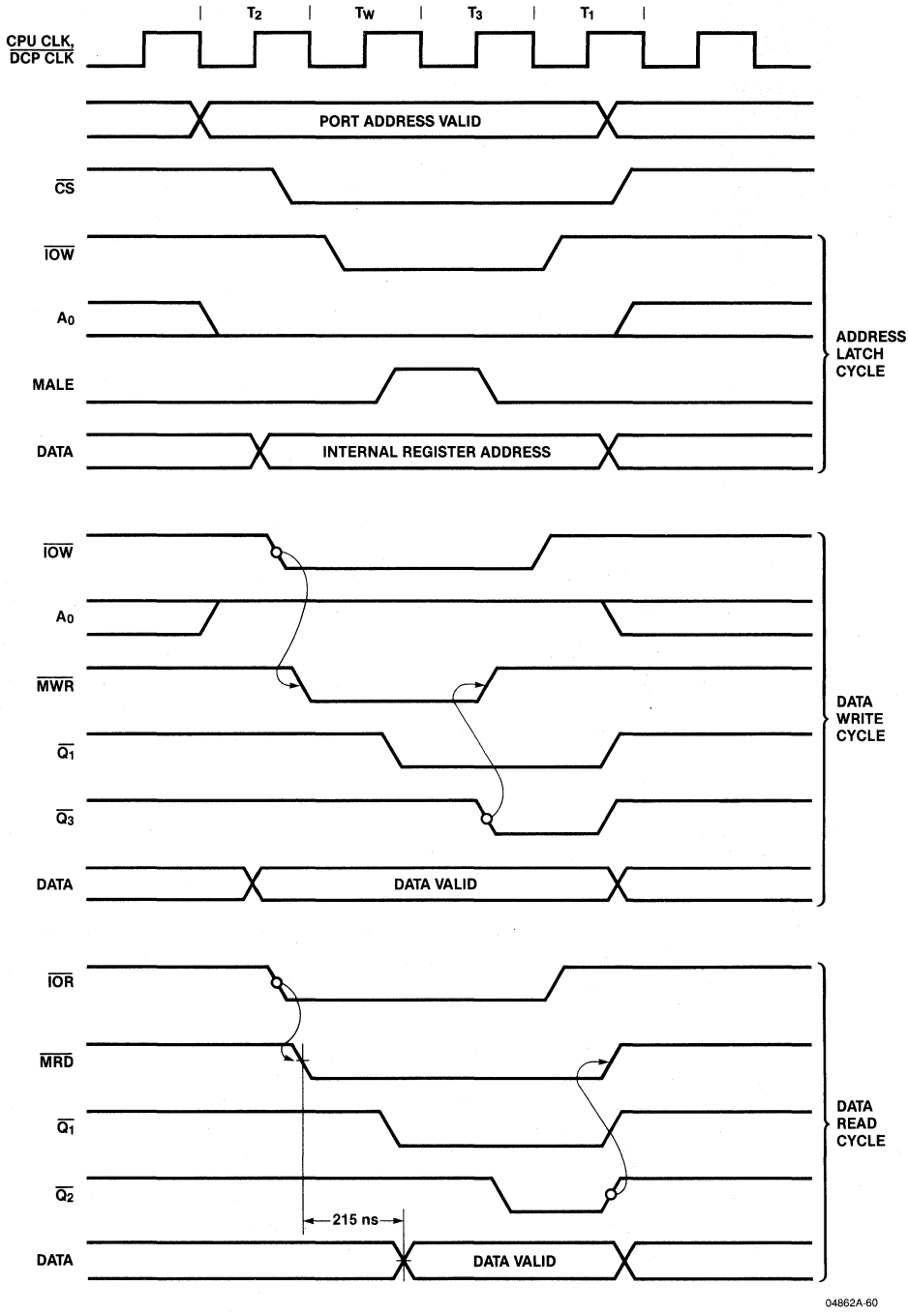


Figure 4.28. CPU-DCP Timing Diagram

done concurrently with the input of the next block; the internal operation is pipelined. The maximum throughput can be calculated as:

$$T = 8 / (8 * 3 + 12) * 4 \text{ MHz} = 0.89 \text{ MByte/s}$$

The Compressed Transfer mode of the DMA controller cannot be used, because the PAL synchronization logic needs normal timing to synchronize the Data Strobes to the DCP clock.

Initialization

The Multiplexed Control Mode (C/$\overline{K}$=Low) of the DCP is selected to enable access to the internal registers. The CPU first programs the Mode Register to reset the DCP and to set up the port configuration and ciphering mode. After that, the keys and initial vectors can be loaded. To initialize the DCP for DMA transfer, the CPU executes one Address Latch Cycle, to pre-select the data register.

The DMA controller must be programmed such that $\overline{DREQ}$ and $\overline{DACK}$ are active Low.

Timing

The PAL device simulates the multiplexed address/data bus of the DCP assuming a two-cycle operation mode. In the first cycle the CPU latches the address of the internal register into the DCP; subsequent cycles transfer data to or from the selected register. Address A_0 distinguishes the two cycles (Figure 4.28). An I/O instruction with A_0=Low generates an address latch cycle; an I/O instruction with A_0=High generates a data transfer cycle.

The DMA controller must be initialized for "extended" I/O write in order to have a similar I/O bus timing to the Z80A CPU. A "late" I/O write delays the Master Port Write Strobe ($\overline{MWR}$) to the DCP by one clock cycle. If a late write is used, the data bus will not be valid at the time data is latched.

To execute a DCP-to-memory transfer, the DMA does an I/O read and memory write. The DMA controller can be programmed for an "extended" or "late" write, depending on the memory design.

In "flyby" mode the DMA controller generates no I/O address, so the CPU has to preselect the data Input or Output Register. A DMA Acknowledge ($\overline{DACK}$) enables $\overline{MRD}$ or $\overline{MWR}$ to control the data transfer.

Figure 4.29 shows the DMA-DCP data transfer timing. When the DMA Controller has transferred one block of data, the data transfer has to be stopped until the DCP is ready for the next block transfer. The DCP makes the DMA Controller stop the transfer by deactivating $\overline{MFLG}$. If $\overline{MFLG}$ is Low, data may be transferred; if $\overline{MFLG}$ is High, the DCP does not accept data transferred. The timing of the $\overline{MFLG}$ to DREQ path is the most critical in this

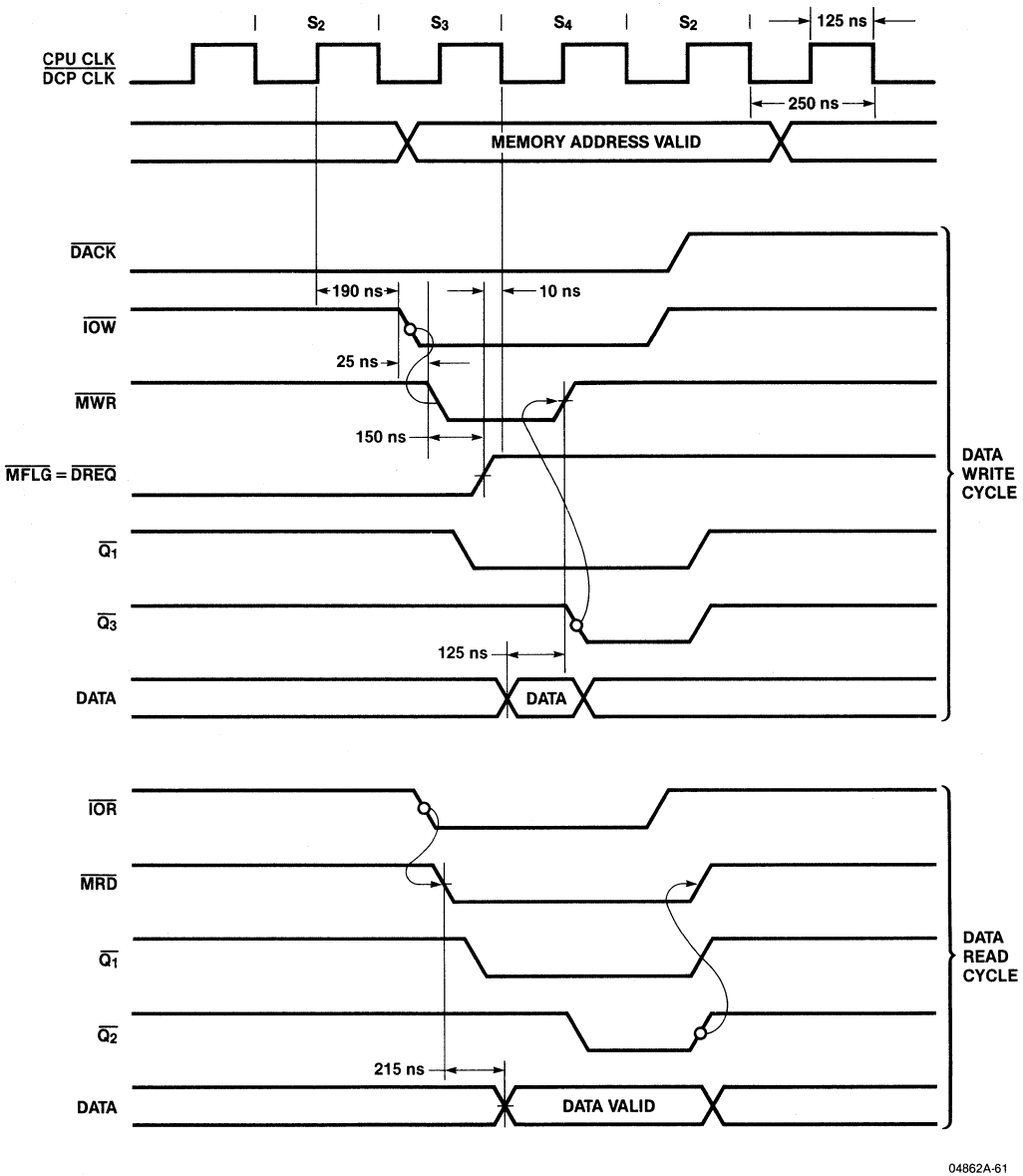


Figure 4.29. DMA-DCP Timing Diagram

application. If $\overline{\text{MFLG}}$ is deactivated too late, the DMA Controller will issue another data transfer which will be disregarded by the DCP. The critical signal path will be analyzed below.

To prevent the DMA from issuing another cycle the Data Request input has to go inactive by the falling edge of the DMA clock at the end of cycle S3. The DMA controller samples the input at this time and instigates another cycle if the request is still active. The set-up time of DREQ is Ø ns. The Master Port Flag which is connected to the DREQ input goes inactive in the eighth cycle with a maximum delay time of 15Ø ns after the Data Strobes. The Data Strobe itself has a maximum delay time of 19Ø ns (Am9517A-5) after the rising edge of the clock in cycle S_2. That gives a time window of 375 ns of which 34Ø ns are already used for the two delays (19Ø ns + 15Ø ns). The propagation delay of a fast PAL device is 25 ns. This leaves 1Ø ns for other delays in the signal path.

The PAL design assumes that the system memory needs no Wait states.

The peripheral logic at the Slave Port can use the Slave Port Flag ($\overline{\text{SFLG}}$) to time the transfer. If $\overline{\text{SFLG}}$ is active Low, data can be written to or read from the data register.

```
PAL16R4                                        PAL DESIGN SPECIFICATION
DCP048                                         JUERGEN STELBRINK  8-9-83
Z80A- AM9517(DMA)- AM9568(DCP) INTERFACE DEVICE
ADVANCED MICRO DEVICES

CLK1   CLK2   /CS    /IOR   /IOW   A0      /MFLG   /DACK   NC     GND
/OE    /MWR   /MRD   /Q1    /Q2    /Q3     MALE    NC      CLK    VCC

/MALE :=   /IOW+IOR+/CS+A0+MALE          ; MASTER PORT ADDRESS STROBE

Q1     :=  CS*A0*IOR*/IOW*/Q2     +
           CS*A0*IOW*/IOR*/Q3     +
           DACK*IOR*/IOW*/Q2      +
           DACK*IOW*/IOR*/Q3

Q2     :=  CS*A0*IOR*/IOW*Q1      +
           CS*A0*IOR*/IOW*Q2      +
           DACK*IOR*/IOW*Q1       +
           DACK*IOR*/IOW*Q2

Q3     :=  CS*A0*IOW*/IOR*Q1      +
           CS*A0*IOW*/IOR*Q2      +
           DACK*IOW*/IOR*Q1       +
           DACK*IOW*/IOR*Q2

MRD     =  CS*A0*IOR*/IOW         +      ; MASTER PORT READ
           DACK*IOR*/IOW          +
           Q2

MWR     =  CS*A0*IOW*/IOR*/Q3     +      ; MASTER PORT WRITE
           DACK*IOW*/IOR*/Q3

/CLK   =   CLK2                          ; DCP CLOCK


FUNCTION TABLE

CLK1 CLK2 /CS /IOR /IOW /DACK A0   CLK MALE /MRD /MWR /Q1 /Q2 /Q3

;                  /
; C  C     /  /   D          M  /  /
; L  L  /  I  I   A       C  A  M  M  /  /  /
; K  K  C  O  O   C  A    L  L  R  W  Q  Q  Q
; 1  2  S  R  W   K  0    K  E  D  R  1  2  3      COMMENT
-------------------------------------------------------------
; CLOCK GENERATION
  X  L  X  X  X   X  X    H  X  X  X  X  X  X
  X  H  X  X  X   X  X    L  X  X  X  X  X  X
;  ADDRESS LATCH
  C  X  H  H  H   H  L    X  L  H  H  H  H  H   ; CYCLE T2 (CPU)
  C  X  L  H  L   H  L    X  H  H  H  H  H  H   ; CYCLE TW
  C  X  L  H  L   H  L    X  L  H  H  H  H  H   ; CYCLE T3
  C  X  H  H  H   H  L    X  L  H  H  H  H  H   ; CYCLE T1
; READ DATA
  X  X  H  H  H   H  H    X  L  H  H  H  H  H   ; CYCLE TW (CPU)
```

```
  X  X  L  L  H  H  H    X  L  L  H  H  H  H
  C  X  L  L  H  H  H    X  L  L  H  L  H  H
  C  X  L  L  H  H  H    X  L  L  H  L  L  H   ; CYCLE TW (EXTRA WAIT STATE)
  C  X  L  L  H  H  H    X  L  L  H  H  L  H   ; CYCLE T3
  C  X  H  H  H  H  H    X  L  H  H  H  H  H   ; CYCLE T1
  X  X  H  L  H  L  X    X  L  L  H  H  H  H   ; CYCLE S3 (DMA)
  C  X  H  L  H  L  X    X  L  L  H  L  H  H
  C  X  H  L  H  L  X    X  L  L  H  L  L  H   ; CYCLE S4
  C  X  H  H  H  H  X    X  L  H  H  H  H  H   ; CYCLE S2
; WRITE DATA
  X  X  L  H  L  H  H    X  L  H  L  H  H  H   ; CYCLE TW (CPU)
  C  X  L  H  L  H  H    X  L  H  L  L  H  H
  C  X  L  H  L  H  H    X  L  H  H  L  H  L   ; CYCLE T3
  C  X  H  H  H  H  H    X  L  H  H  H  H  H   ; CYCLE T1
  X  X  H  H  L  L  H    X  L  H  L  H  H  H   ; CYCLE S3 (DMA)
  C  X  H  H  L  L  H    X  L  H  L  L  H  H
  C  X  H  H  L  L  H    X  L  H  H  L  H  L   ; CYCLE S4
  C  X  H  H  H  H  H    X  L  H  H  H  H  H   ; CYCLE S2
;
```

DESCRIPTION:

THIS PAL GENERATES ALL NECESSARY BUS CONTROL SIGNALS, TO INTERFACE A Z80A CPU AND A AM9517 DMA CONTROLLER TO THE AM9568 DATA CIPHERING PROCESSOR. THE MAXIMUM SYSTEM CLOCK FOR ALL PARTS IS 4 MHZ.

1 INPUT AND 3 INPUT/ OUTPUT PINS ARE NOT USED.

INPUT SIGNALS:

CLK1, CLK2 — Z80 SYSTEM CLOCK

/CS — CHIP SELECT FOR THE DCP, GENERATED BY A DECODER LOGIC

/IOR — INPUT/OUTPUT READ

/IOW — INPUT/OUTPUT WRITE

A0 — LEAST SIGNIFICANT BIT OF THE Z80 ADDRESS BUS TO SELECT THE TYPE OF OPERATION:
- A0 = LOW — SELECT DCP REGISTER FOR NEXT DATA CYCLES (ADDRESS LATCH)
- A0 = HIGH — READ OR WRITE INTERNAL REGISTER (DATA TRANSFER TO CONTROL, MODE, INPUT OR OUTPUT REGISTER)

/DACK — DMA ACKNOWLEDGE FROM DMA CONTROLLER, TREATED AS /CS=LOW AND A0=HIGH

OUTPUT SIGNALS:

CLK INVERTED SYSTEM CLOCK FOR THE DCP

MALE MASTER PORT ADDRESS LATCH ENABLE, ACTIVE DURING ADDRESS LATCH CYCLES TO LATCH THE REGISTER ADDRESS ON MP1 AND MP2 (2 LINES OF THE MASTER PORT BUS) AND THE STATE OF /MCS IN. THE DCP STORES INTERNALLY THE ADDRESS AND CHIP SELECT TO THE NEXT ADDRESS LATCH CYCLE

/MRD MASTER PORT READ, TO ENABLE REGISTER READ OPERATIONS

/MWR MASTER PORT WRITE, TO ENABLE REGISTER WRITE OPERATIONS

/Q1, /Q2, /Q3 INTERNAL USED STATE SIGNALS (DO NOT CONNECT). Q1 IS ACTIVE 2 CLOCK CYCLES IN EACH DATA TRANSFER OR DMA ACKNOWLEDGE CYCLE. IT IS USED TO GENERATE THE DELAYED Q2 AND Q3. Q2 IS USED TO HOLD /MRD ACTIVE UNTIL /IOR IS GONE INACTIVE. Q3 MASKS /MWR OFF.

4.9. 8088 - DMA - AmZ8068

This interface design is similar to that of the previous chapter. The differences are that the Am9568 is replaced by the AmZ8068 and the PAL device is reprogrammed for the 8088 CPU bus timing (READY). In this chapter, only the differences in the Z80-DMA-DCP interface are discussed. For additional information refer to Chapter 4.8.

Figure 4.30 shows the CPU-DMA interface. The CPU is operating in Maximum Mode. The bus arbitration handshake of the DMA controller (HREQ and HACK) must be translated into the Bus Request/Grant handshake of the 8088 CPU, as described in the application note, "A Tested Design for the Evaluation of the Am9516 UDC in an 8086 Environment" published in the Am9516/AmZ8016* Technical Manual.

If the CPU is programmed to operate in Minimum Mode, both devices have the same bus arbitration handshake. The HREQ and HACK of the DMA controller can be connected directly to the corresponding pins of the CPU (HREQ to HACK).

The central part of this interface is a PAL device. The Chip Select 2 ($\overline{CS_2}$) input of the PAL device must be stable during the entire I/O transfer. This is guaranteed by decoding $\overline{CS_2}$ from the latched address/data bus of the 8088 (A_0 to A_{15} in Figure 4.30).

Master Port Read/Write is latched in the D-Flip-Flop. It is clocked in an output operation with $\overline{CS_3}$ active. One of the data lines is latched in to define the status on the MR/$\overline{W}$ input. This is necessary because the DCP requires a set-up time of 100 ns of MR/$\overline{W}$ to the Data Strobe. Generation of MR/$\overline{W}$ for each cycle of a high-speed data transfer session of the DMA controller would extend each cycle and slow down the maximum throughput. This logic cannot be integrated into the PAL device because of the flip-flop's asynchronous clock.

Before executing an access to the DCP the CPU must latch the MR/$\overline{W}$. The transfer itself is evaluated in a two-cycle operation.

Master Port Address Strobe ($\overline{MAS}$) is only generated if the CPU executes an output instruction to a specific I/O address ($\overline{CS_2}$ active, A_0=Low) (Figure 4.31). Address Latch Enable of the CPU (ALE) cannot be used for the generation of $\overline{MAS}$ because the CPU must set up the DCP for data transfer before a DMA transfer session is started. The DCP is set up by putting out a 00_H (data register address) to the I/O address mentioned above.

Figures 4.32 and 4.33 show data read and write cycles. Figure 4.34 shows DMA data read and writes cycles.

*AmZ8016 is a trademark of Advanced Micro Devices, Inc.

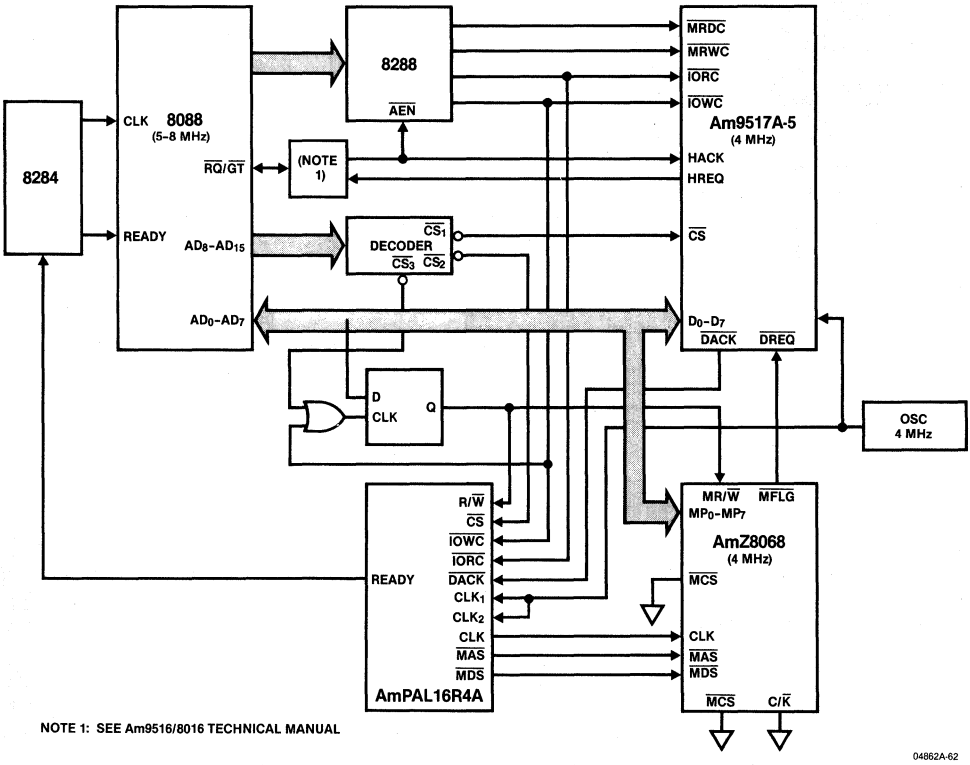


Figure 4.30. 8088-Am9517-AmZ8068 Interface

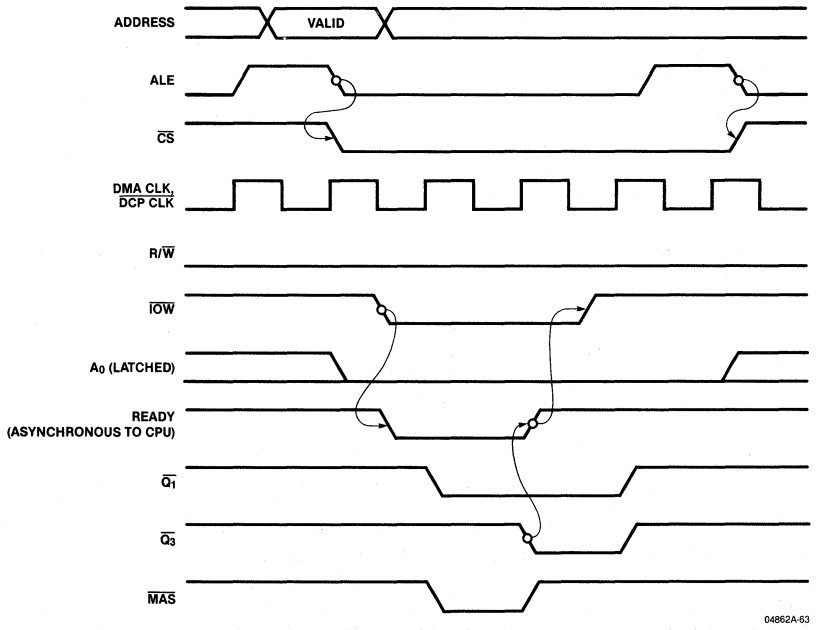


Figure 4.31. Address Latch Cycle Timing (CPU-DCP)

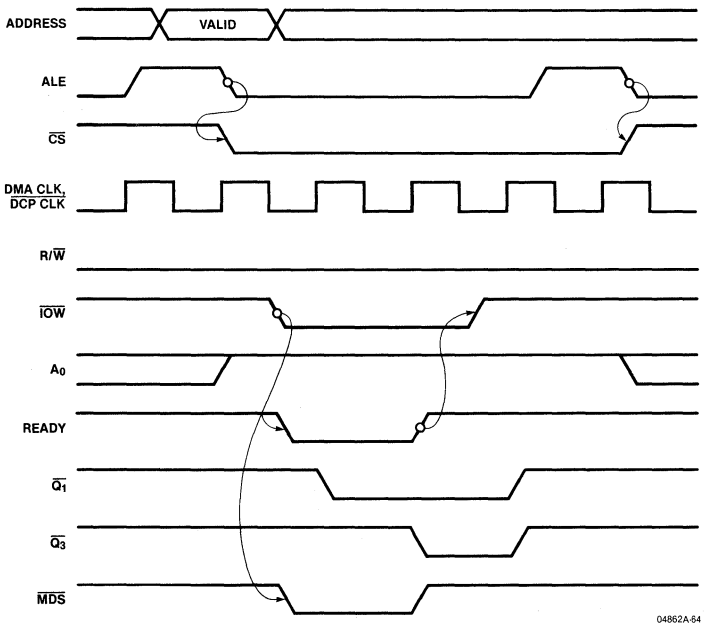


Figure 4.32. Data Write Cycle Timing (CPU-DCP)

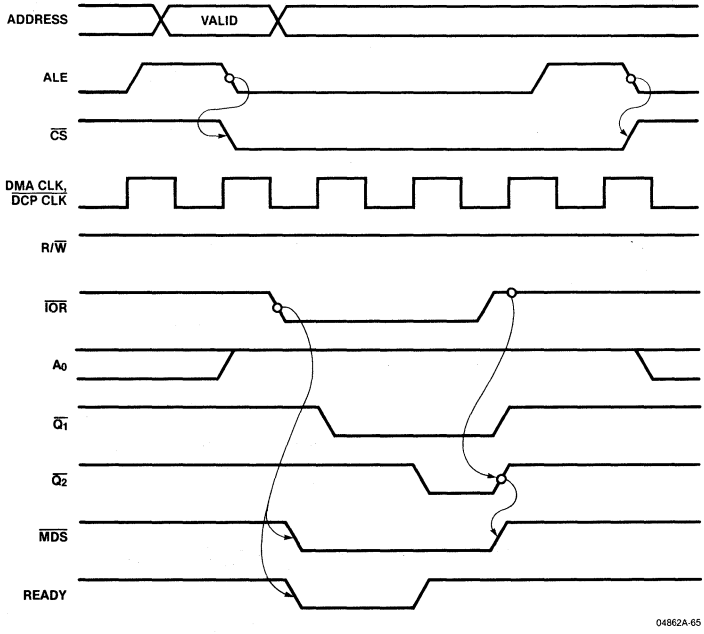


Figure 4.33. Data Read Cycle Timing

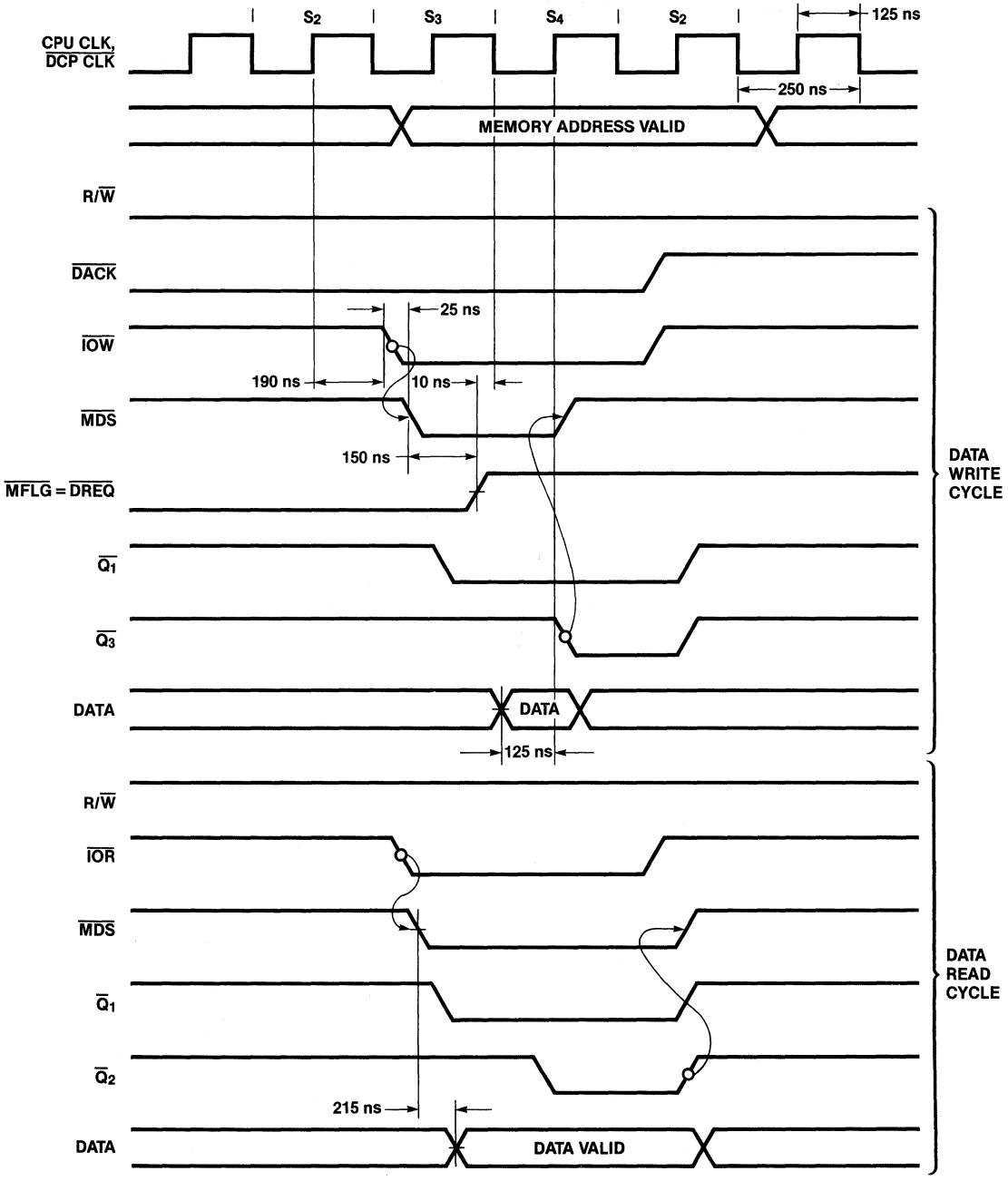


Figure 4.34. DMA-DCP Timing Diagram

04862A-66

```
PAL16R4                                        PAL DESIGN SPECIFICATION
DCP049                                         JUERGEN STELBRINK  8-12-83
8088- AM9517(DMA)- AMZ8068(DCP) INTERFACE DEVICE
ADVANCED MICRO DEVICES

CLK1  CLK2  /CS   /IOR  /IOW  A0    RW     /DACK  NC   GND
/OE   /MDS  READY /Q1   /Q2   /Q3   /MAS   NC     CLK  VCC

MAS   :=  IOW*/IOR*CS*/A0*/Q3*/MAS   ; MASTER PORT ADDRESS STROBE

Q1    :=  CS*IOR*/IOW*RW*/Q2    +
          CS*IOW*/IOR*/RW*/Q3   +
          DACK*IOR*/IOW*RW*/Q2  +
          DACK*IOW*/IOR*/RW*/Q3

Q2    :=  CS*IOR*/IOW*RW*Q1     +
          CS*IOR*/IOW*RW*Q2     +
          DACK*IOR*/IOW*RW*Q1   +
          DACK*IOR*/IOW*RW*Q2

Q3    :=  CS*IOW*/IOR*/RW*Q1    +
          CS*IOW*/IOR*/RW*Q2    +
          DACK*IOW*/IOR*/RW*Q1  +
          DACK*IOW*/IOR*/RW*Q2

MDS    =  CS*A0*IOR*/IOW*RW     +    ; MASTER PORT READ
          DACK*IOR*/IOW*RW      +
          Q2*A0                 +
          CS*A0*IOW*/IOR*/RW*/Q3+    ; MASTER PORT WRITE
          DACK*IOW*/IOR*/RW*/Q3

/READY =  CS*/A0*IOW*/IOR*/RW*/Q3+   ; ADDRESS LATCH CYCLE
          CS*A0*IOW*/IOR*/RW*/Q3 +   ; DATA WRITE CYCLE
          CS*A0*IOR*/IOW*RW*/Q2

/CLK   =  CLK2                       ; DCP CLOCK


FUNCTION TABLE

CLK1 CLK2 /CS /IOR /IOW /DACK A0 RW   CLK /MAS /MDS READY /Q1 /Q2 /Q3

;                  /                    R
; C  C     /  /  D                 /  /  E
; L  L  /  I  I  A           C  M  M  A  /  /  /
; K  K  C  O  O  C  A  R     L  A  D  D  Q  Q  Q
; 1  2  S  R  W  K  0  W     K  S  S  Y  1  2  3      COMMENT
-------------------------------------------------------------
; CLOCK GENERATION
  X  L  X  X  X  X  X  X     H  X  X  X  X  X  X
  X  H  X  X  X  X  X  X     L  X  X  X  X  X  X
;  ADDRESS LATCH
  C  X  H  H  H  H  L  L     X  H  H  H  H  H  H    ; CPU
  X  X  L  H  L  H  L  L     X  H  H  L  H  H  H
  C  X  L  H  L  H  L  L     X  L  H  L  L  H  H
```

```
  C  X  L  H  L  H  L  L     X  H  H  H  L  H  L
  C  X  H  H  H  H  L  L     X  H  H  H  H  H  H
; READ DATA
  X  X  H  H  H  H  H  H     X  H  H  H  H  H  H    ; CPU
  X  X  L  L  H  H  H  H     X  H  L  L  H  H  H
  C  X  L  L  H  H  H  H     X  H  L  L  L  H  H
  C  X  L  L  H  H  H  H     X  H  L  H  L  L  H
  C  X  L  L  H  H  H  H     X  H  L  H  H  L  H
  C  X  H  H  H  H  H  H     X  H  H  H  H  H  H
  X  X  H  L  H  L  X  H     X  H  L  H  H  H  H    ; CYCLE S3 (DMA)
  C  X  H  L  H  L  X  H     X  H  L  H  L  H  H
  C  X  H  L  H  L  X  H     X  H  L  H  L  L  H    ; CYCLE S4
  C  X  H  H  H  H  X  H     X  H  H  H  H  H  H    ; CYCLE S2
; WRITE DATA
  X  X  L  H  L  H  H  L     X  H  L  L  H  H  H    ; CPU
  C  X  L  H  L  H  H  L     X  H  L  L  L  H  H
  C  X  L  H  L  H  H  L     X  H  H  H  L  H  L
  C  X  H  H  H  H  H  L     X  H  H  H  H  H  H
  X  X  H  H  L  L  H  L     X  H  L  H  H  H  H    ; CYCLE S3 (DMA)
  C  X  H  H  L  L  H  L     X  H  L  H  L  H  H
  C  X  H  H  L  L  H  L     X  H  H  H  L  H  L    ; CYCLE S4
  C  X  H  H  H  H  H  L     X  H  H  H  H  H  H    ; CYCLE S2
; INVALID CYCLES
  X  X  L  L  L  H  H  H     X  H  H  H  H  H  H
  X  X  L  L  H  H  H  L     X  H  H  H  H  H  H
  X  X  L  H  L  H  H  H     X  H  H  H  H  H  H
-----------------------------------------------------------------
```

DESCRIPTION:

THIS PAL GENERATES ALL NECESSARY BUS CONTROL SIGNALS, TO INTERFACE A 8088 CPU AND A AM9517 DMA CONTROLLER TO THE AMZ8068 DATA CIPHERING PROCESSOR. THE MAXIMUM SYSTEM CLOCK FOR THE DMA CONTROLLER AND THE DCP IS 4 MHZ, THE SYSTEM CLOCK OF THE CPU CAN BE UP TO 8 MHZ. THE DEVICES ARE WORKING ASYNCHRONOUSLY.

INPUT SIGNALS:

CLK1, CLK2 — DMA CLOCK

/CS — CHIP SELECT FOR THE DCP, GENERATED BY A DECODER LOGIC

/IOR — INPUT/ OUTPUT READ

/IOW — INPUT/ OUTPUT WRITE

A0 — LEAST SIGNIFICANT BIT OF THE Z80 ADDRESS BUS TO SELECT THE TYPE OF OPERATION:
- A0 = LOW — SELECT DCP REGISTER FOR NEXT DATA CYCLES (ADDRESS LATCH)
- A0 = HIGH — READ OR WRITE INTERNAL REGISTER (DATA TRANSFER TO CONTROL, MODE, INPUT OR OUTPUT REGISTER)

/DACK DMA ACKNOWLEDGE FROM DMA CONTROLLER, TREATED AS /CS=LOW AND AØ=HIGH

RW READ/ WRITE SIGNAL STORED IN A EXTERNAL LATCH, TO ALLOW A DMA OPERATION WITHOUT WAIT STATES. THIS SOLVES THE PROBLEM OF THE SETUP TIME OF MR/W OF THE MASTER PORT TO MDS GOING ACTIVE. THE STATUS OF THIS SIGNAL MUST AGREE WITH /IOR OR /IOW OR THE PAL GENERATES NO STROBES.

OUTPUT SIGNALS:

CLK INVERTED DMA CLOCK FOR THE DCP

/MAS MASTER PORT ADDRESS LATCH ENABLE, ACTIVE DURING ADDRESS LATCH CYCLES TO LATCH THE REGISTER ADDRESS ON MP1 AND MP2 (2 LINES OF THE MASTER PORT BUS) AND THE STATE OF /MCS IN. THE DCP STORES INTERNALLY THE ADDRESS AND CHIP SELECT TO THE NEXT ADDRESS LATCH CYCLE

/MDS MASTER PORT DATA STROBE, TO TIME DCP DATA TRANSFERS

/Q1, /Q2, /Q3 INTERNAL USED STATE SIGNALS (DO NOT CONNECT). Q1 IS ACTIVE 2 CLOCK CYCLES IN ALL CYCLES. IT IS USED TO GENERATE THE DELAYED Q2 AND Q3. Q2 IS ACTIVE IN A DATA READ CYCLE. IT ALLOWS /MDS TO BE ACTIVE UNTIL /IOR HAS GONE INACTIVE. Q3 IS ACTIVE IN AN ADDRESS LATCH OR DATA WRITE CYCLE. Q3 DISABLES READY AND /MDS IN THE SECOND HALF OF THE CYCLE.

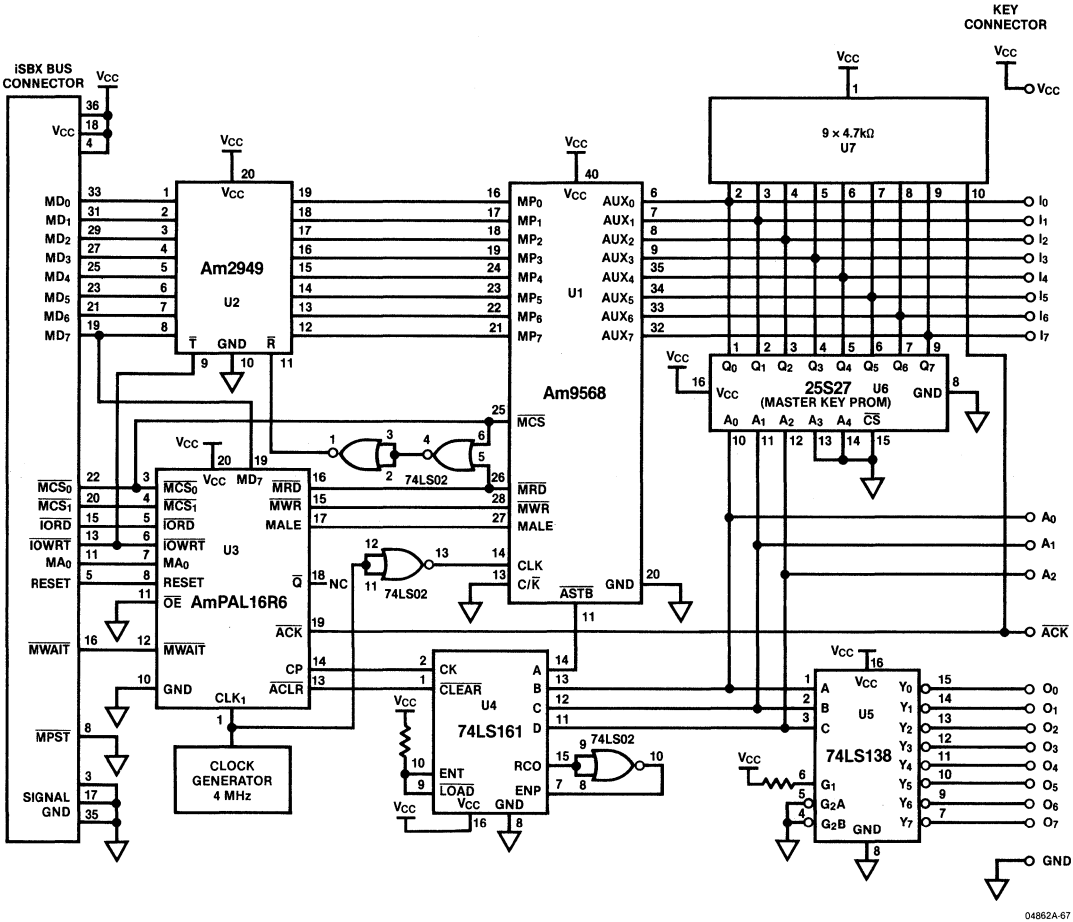


Figure 4.35. iSBX Bus—Am9568 Interface

4.1Ø. iSBX BUS - Am9568

The iSBX board described below adds high-speed data ciphering capability to a Multibus-based system . This iSBX board can be plugged into any Multibus board with an iSBX connector. The iSBX bus timing and bus signals are described in the "iSBX Bus Specification" (see Literature List).

The Master Port of the DCP is interfaced to the iSBX bus. The multiplexed address/data bus of the DCP is simulated in a two-cycle operation.

The interface timing controller, a PAL device, generates the address and data strobes for the DCP and the Wait signal for the host CPU.

The Auxiliary Port enhances the security of the system by preventing a CPU access to the keys. The keys can be loaded from a small bipolar PROM or from a device connected to the Key Connector. This device can be an optical or magnetic key reader.

The Key Connector provides two power supply lines for the external device, Ground and +5 V. Two address buses (a 3-bit encoded bus ($A_Ø$ to A_2) and an 8-bit decoded bus ($O_Ø$ to O_7)) select one of the eight key bytes (Figure 4.35). The user can choose one of these two address buses. At any time, only one of the eight lines of the decoded bus ($O_Ø$ to O_7) is active Low. Eight input lines ($I_Ø$ to I_7) carry the key byte to the Auxiliary Port. Pull-up resistors force the data lines High if no device is connected to the Key Connector.

The ciphering throughput of this particular design is limited by the iSBX bus byte transfer capability. In the single-port operation mode chosen, the maximum throughput is about 2ØØ kBytes/s, high enough even for speech ciphering applications. The throughput can be doubled if the interface design is changed to allow dual-port operation.

The two-cycle operation mode is chosen in this interface design because it allows a faster ciphering speed and needs less interface logic. The whole interface logic fits into one PAL device. The disadvantage of this approach is software overhead for initializing the device. Under software control two types of cycles are generated, an Address Latch Cycle and a data transfer cycle.

The address latch cycle is started by an output operation of the CPU to an even I/O address which selects this iSBX board. The internal DCP register address to be accessed by the CPU is transferred via the Master Port data bus. MP_1 and MP_2 carry the relevant address information. In this cycle only MALE is generated.

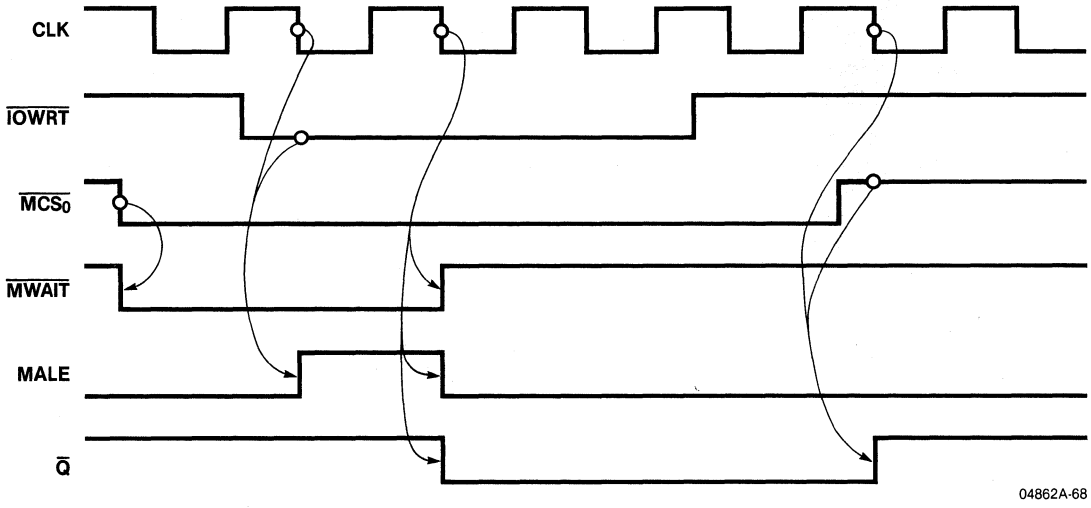


Figure 4.36. Address Latch Cycle Timing (MA_0 = Low)

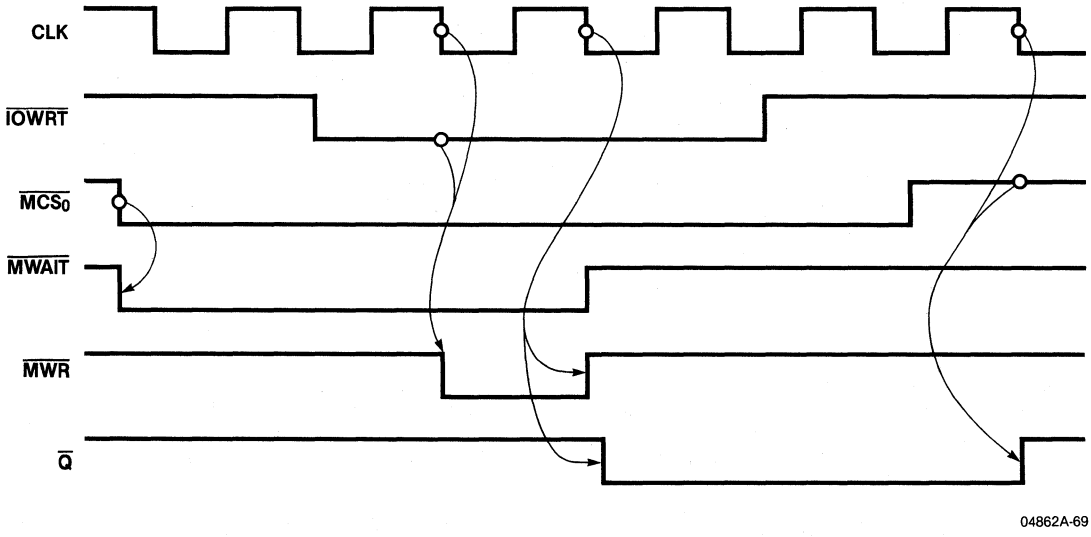


Figure 4.37. Data Write Cycle (MA_0 = High)

A data transfer cycle is executed in an output operation to an odd address. The transfer is made from or to the register that was selected in the previous Address Latch Cycle.

This approach is faster than simulating a multiplexed bus because a Master Port Address Latch Enable (MALE) need not be generated in a high-speed data transfer session. The data register address is latched in the chip by an Address Latch Cycle at the beginning of the session. The data session itself has no address latch overhead.

Address Latch Cycle

The Master Port Address Latch Enable (MALE) latches the state of Master Port Chip Select ($\overline{MCS}$) and the internal register address on MP_1 and MP_2. Subsequent data cycles use this 2-bit address.

The PAL device starts generating an Address Latch Cycle if the iSBX signals indicate a CPU output operation to an even port address. $\overline{IOWRT}$ (I/O write command) and $\overline{MCS_0}$ (M Chip Select Ø) are active, MA_0 (M Address Ø) is Low and $\overline{MCS_1}$ is inactive.

The portion of the PAL device generating MALE operates as a state machine. MALE is set at the first falling edge of CLK, when $\overline{MCS_0}$ and $\overline{IOWRT}$ are active. The next falling edge resets MALE and sets the internal state variable $\overline{Q}$ which inhibits MALE from being set again.

$\overline{MWAIT}$ inserts CPU Wait states until the register address is latched on the falling edge of MALE. The rest of the cycle is unavoidable overhead because the iSBX bus timing specifies no minimum delay time between $\overline{MWAIT}$ inactive and the end of the I/O cycle. If $\overline{MCS}$ glitches, $\overline{MWAIT}$ also glitches. The delay is less than 35 ns, which meets the iSBX timing specification. $\overline{Q}$ removes $\overline{MWAIT}$, after MALE became inactive.

Figure 4.36 illustrates an Address Latch Cycle.

Data Write Cycle

The CPU can write commands, data or keys to the previously selected internal register. Data is latched with the rising edge of Master Port Write ($\overline{MWR}$).

The generation of $\overline{MWR}$ is similar to that of MALE. The difference is that an output operation to an even address (MA_0=High) initiates the state machine of the PAL device. The pulse width of $\overline{MWR}$ is one clock cycle.

$\overline{MWR}$ is synchronous to the falling edge of the clock (CLK) to meet the critical timing parameter 45 of the Am9568 product specification.

Figure 4.37 illustrates a Data Write Cycle.

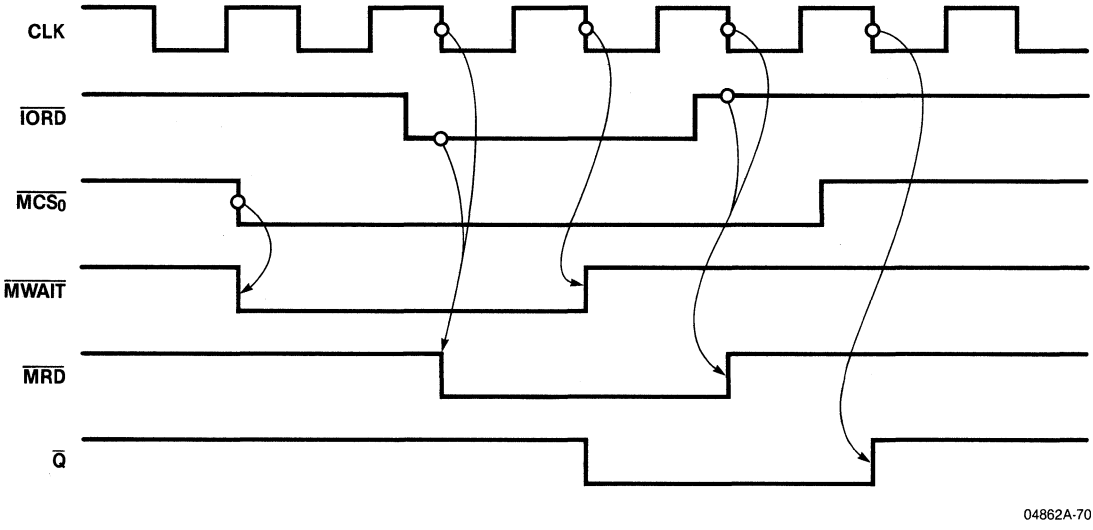


Figure 4.38. Data Read Cycle (MA_0 = High)

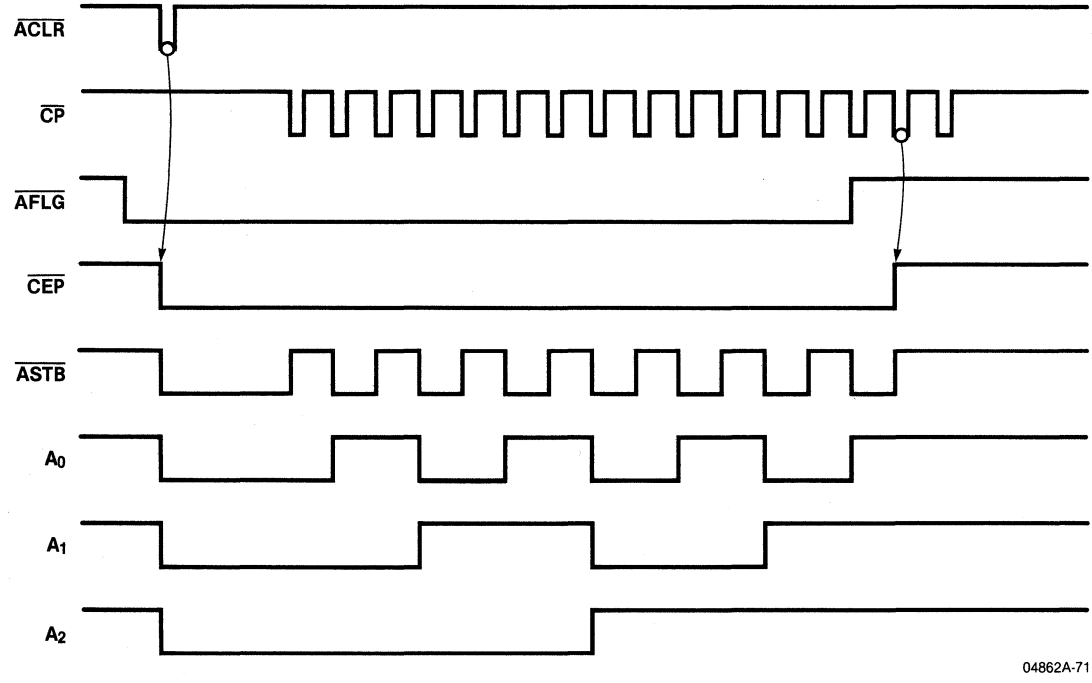


Figure 4.39. Auxiliary Port Key Load Timing

Data Read Cycle

A data read cycle is initiated when $\overline{MCS}_0$ and $\overline{IORD}$ are active, MA_0 is High and $\overline{MCS}_1$ is inactive. The CPU then can read the addressed internal register.

$\overline{MCS}_0$ causes $\overline{MWAIT}$ to be asserted Low in order to extend the cycle. $\overline{MWAIT}$ guarantees a minimum of one clock access time to the DCP register (min. 250 ns at 4-MHz DCP clock). This satisfies timing parameter 49 (200 ns minimum). The CPU can latch the data bus any time between $\overline{MWAIT}$ and $\overline{IORD}$ becoming inactive. The data on the DCP data bus is valid until the first falling edge of CLK after $\overline{IORD}$ becomes inactive. $\overline{MRD}$ changes to High synchronous with that edge to satisfy timing parameter 45 (0 to 30 ns).

The iSBX bus timing specifies that the data bus has to be floating within 150 ns after $\overline{MCS}$ inactive. To satisfy this parameter and to prevent data bus contention in the end of a data read cycle, the data bus transceiver U2 in Figure 4.35 disconnects the DCP data bus from the CPU data bus. Two NOR gates (74LS02) combine $\overline{MCS}$ and $\overline{MRD}$, to generate the receive control signal for U2.

Figure 4.38 illustrates a data read cycle.

Key Load Logic

The DCP has three keys stored on the chip: one key for encryption, one key for decryption, and a Master Key. Each of these 56-bit keys can be loaded through either the Master Port or the Auxiliary Port. The keys are transferred in eight cycles, one byte at a time. Note that the least significant bit of each byte is a parity bit for odd parity ((8 - 1) * 8 = 56).

This application note offers two methods of loading the keys through the Auxiliary Port:

- A 32 * 8-bit PROM can hold one key, either the Master Key or one key used for both encryption and decryption.

- A wide variety of devices from a simple 8 by 8 jumper matrix to an advanced card reader can be plugged into the Key Connector. Software compensates the speed of the device.

Sequencer U4, a 74LS161 4-bit up counter, generates a 3-bit address sequence for the Master Key PROM U6 and the Key Connector. The least significant bit of the sequencer is wired to the Auxiliary Port Strobe input $\overline{ASTB}$ of the DCP.

The two sequencer control signals, $\overline{ACLR}$ and CP, are controlled by software.

The Asynchronous Clear input $\overline{CLEAR}$ initializes U4 with outputs A to D Low. The first key byte is addressed. Ripple Carry output $\overline{RCO}$ is inactive High.

The first pulse on the clock input CK produces a rising edge at $\overline{ASTB}$ to strobe in the first key byte. The rising edge of $\overline{ASTB}$ is synchronous to the clock CLK to satisfy timing parameter 62 (Ø to 5Ø ns). The software controlled delay time between $\overline{ACLR}$ and CP or between the following CPs allows interfacing to any external key device. In the case of reading from the Master Key PROM, no software Wait loop is required because the access time of this PROM meets any CP sequence.

The acknowledge input $\overline{ACK}$ can be pulled Low by the Key Load Device to signal the CPU that the key byte at the Auxiliary Port is valid. The PAL device transfers the state of this input to the iSBX data bus line Ø during an I/O read operation with $\overline{MCS}_1$ active.

The second pulse on CP increments the address output of the sequencer. The delay time between the first and second pulse satisfies the data hold time requirement of 8Ø ns (timing parameter 65) of the Auxiliary Port.

A sequence of 15 pulses on CP transfers all 8 bytes of the key into the DCP. After the 15th pulse RCO becomes active to disable further key strobes ($\overline{ASTB}$).

The 3 to 8 line decoder U5 creates a decoded address for the Key Connector.

Figure 4.39 illustrates the key load sequence.

The PAL Device

The interface timing circuit, a PAL device, is programmed to generate: Four control signals for the DCP (CLK, MALE, $\overline{MRD}$ and $\overline{MWR}$), the Wait signal for the CPU, and the $\overline{ACLR}$ and CP to control the key load logic.

The PAL device used in this application note is an AmPAL16R6 device. It has eight inputs and eight outputs. Two outputs are combinatorial, six are registered. The input Output Enable $\overline{OE}$ is wired Low to enable all outputs.

CLK and $\overline{MWAIT}$ are combinatorial outputs of the PAL device. $\overline{MWAIT}$ must be a combinatorial output to meet the timing relationship to $\overline{MCS}$ as specified in the iSBX specification (see the paragraph "Address Latch Cycle").

The other outputs -- $\overline{MAS}$, $\overline{MRD}$, $\overline{MWR}$, $\overline{Q}$, CP and $\overline{ACLR}$ -- are registered outputs. They are synchronous to the rising edge of the CLK1 input and, therefore, to the falling edge of the CLK output.

The $\overline{\text{ACLR}}$ is strobed Low when executing an output operation to an even I/O address with $\overline{\text{MCS}}_1$ active.

The CP is strobed low when executing an output operation to an odd address with $\overline{\text{MCS}}_1$ active. The loading of keys is software-controlled so that a wide variety of devices can be plugged into the Key Connector.

```
PAL16R6                                    PAL DESIGN SPECIFICATION
DCPØ41Ø                                    JUERGEN STELBRINK  6/28/83
ISBX- BUS TO AM9568 INTERFACE DEVICE
ADVANCED MICRO DEVICES

CLK   /ACK    /MCSØ  /MCS1  /IORD  /IOWRT  MAØ   NC   NC   GND
/OE   /MWAIT  /ACLR  CP     /MWR   /MRD    MALE  /Q   MD7  VCC

MWAIT  =  MCSØ*/MCS1*/Q

Q      :=  MCSØ*/MCS1*/MAØ*IOWRT*MALE*/Q  +       ; ADDRESS LATCH
           MCSØ*/MCS1*MAØ*IOWRT*MWR*/Q   +       ; WRITE DATA
           MCSØ*/MCS1*MAØ*IORD*MRD*/Q    +       ; READ DATA
           MCSØ*/MCS1*Q

/MALE :=  /MCSØ+MCS1+MAØ+/IOWRT+MALE+Q

MWR    :=  MCSØ*/MCS1*MAØ*IOWRT*/MWR*/Q

MRD    :=  MCSØ*/MCS1*MAØ*IORD

IF (/MCSØ*MCS1*IORD) /MD7  =  ACK

/CP    :=  /MCSØ*MCS1*MAØ*IOWRT

ACLR   :=  /MCSØ*MCS1*/MAØ*IOWRT

FUNCTION TABLE

CLK /ACK /MCSØ /MCS1 /IORD /IOWRT MAØ
MD7 MALE /MRD /MWR /MWAIT /Q CP /ACLR

;                    /                              /
;             /  /   /   I                          M           /
;        /    M  M   I   O         M   /   /   W           A
; C   A   C  C   O   W   M    M   A   M   M   A           C
; L   C   S  S   R   R   A    D   L   R   W   I   /   C   L
; K   K   Ø  1   D   T   Ø    7   E   D   R   T   Q   P   R    COMMENT
----------------------------------------------------------------------
  C   X   H  H   H   H   L    Z   L   H   H   H   H   H   H   ; ADDRESS LATCH
  H   X   L  H   H   H   L    Z   L   H   H   L   H   H   H
  C   X   L  H   H   L   L    Z   H   H   H   L   H   H   H
  C   X   L  H   H   L   L    Z   L   H   H   H   L   H   H
  C   X   L  H   H   L   L    Z   L   H   H   H   L   H   H
  C   X   H  H   H   H   L    Z   L   H   H   H   H   H   H
;
  C   X   L  H   H   L   H    Z   L   H   L   L   H   H   H   ; DATA WRITE
  C   X   L  H   H   L   H    Z   L   H   H   H   L   H   H
  C   X   L  H   H   L   H    Z   L   H   H   H   L   H   H
  C   X   H  H   H   H   H    Z   L   H   H   H   H   H   H
;
  C   X   L  H   L   H   H    Z   L   L   H   L   H   H   H   ; DATA READ
  C   X   L  H   L   H   H    Z   L   L   H   H   L   H   H
  C   X   L  H   L   H   H    Z   L   L   H   H   L   H   H
  C   X   H  H   H   H   H    Z   L   H   H   H   H   H   H
```

```
;
  C  X  H  L  H  L  L    Z  L  H  H  H  H  H  L  ; RESET COUNTER
  C  X  H  L  H  H  L    Z  L  H  H  H  H  H  H
  C  X  H  L  H  L  H    Z  L  H  H  H  H  L  H  ; CLOCK COUNTER
  C  X  H  L  H  H  H    Z  L  H  H  H  H  H  H
;
  X  L  H  L  L  H  X    L  L  H  H  H  H  H  H  ; ACKNOWLEDGE READ
  X  H  H  L  L  H  X    H  L  H  H  H  H  H  H
```

DESCRIPTION:

GENERATION OF ALL NECESSARY BUS CONTROL SIGNALS, TO INTERFACE THE AM9568 (DCP) TO ISBX- BUS.

INPUTS:

CLK	4 MHZ DCP CLOCK
/MCSØ	DCP CHIP SELECT MAØ = LOW ADDRESS LATCH CYCLE MAØ = HIGH DATA TRANSFER CYCLE
/MCS1	KEY COUNTER SELECT WRITE: MAØ = LOW COUNTER RESET MAØ = HIGH COUNTER STROBE (8 TIMES 2 STROBES, TO LOAD THE 8 KEY- BYTES READ: PUT STATE OF ACKNOWLEDGE INPUT TO MD7
/IORD	INPUT/ OUTPUT READ
/IOWRT	INPUT/ OUTPUT WRITE
MAØ	ADDRESS LINE Ø
/ACK	ACKNOWLEDGE SIGNAL FROM EXTERNAL KEY LOAD DEVICE

OUTPUTS:

/MWAIT	WAIT SIGNAL TO THE CPU, TO EXPAND THE IO TRANSFER
/MRD	MASTER PORT READ
/MWR	MASTER PORT WRITE
MALE	MASTER PORT ADDRESS LATCH ENABLE
MD7	MASTER PORT DATA LINE 7
CP	CLOCK PULSE FOR THE KEY ADDRESS COUNTER
/ACLR	RESET KEY ADDRESS COUNTER

Testing

The DCP iSBX board was tested in a CP/M 86 system. It was hooked up to the Module 2 connector of an AMD iSBX Motherboard (PWA 009520014). This Motherboard has to be configured for byte mode with the Module 2 addresses from 90 to $9F_H$ in order to run the test program without any changes. Therefore, jumper HDR1 is removed and HDR2 is installed. Jumpers 1-2 and 11-12 are installed.

The test program is written in 8086 Assembly Language. The structure of the program is described below.

It programs the DCP for ECB (Electronic Code Book) encryption mode and single-port operation by loading 18_H into the Mode Register. Then 8 bytes of encryption key are put in and one block is ciphered. The 8 result bytes are stored at location "CIPHER".

The result should be: 95_H,$A8_H$,$D7_H$,28_H,13_H,DA_H,$A9_H$ and $4D_H$.

Writing a 91_H to the Command Register sets the DCP up for key input through the Auxiliary Port. A following Status Register read should show a 44_H: Command Pending and Auxiliary Port Flag (AFLG) are active.

The instruction "OUT ACLR,AL" initializes the key load logic. The loop LAB1 sends 16 strobes to the sequencer to strobe in encryption key (Figure 4.39). If all the key bytes do not have odd parity, the LPAR flag in the Status Register is set. If everything is correct after strobing the key in, the Status Register will contain 00_H.

The start command $C0_H$ sets the Start/Stop bit of the Status Register and sets the device up for a data encryption session. Loop2 loads 8 bytes of plain data into the Input Register. When this block is loaded, a Status Register read will show 83_H: Start/Stop is active, the input flag is active to indicate that more blocks of data can be put in, and the output flag is active to indicate that data can be read out.

Loop3 reads one block of cipher data out of the Output Register and transfers it to the memory location "CIPHER".

A following status read shows that the output flag is inactive indicating the Output Register is empty.

The Stop command $E0_H$ terminates the ciphering session; all bits of the Status Register are reset.

```
ASM86 VER 1.0   SOURCE: TESTISBX.A86

;------------------------------------------------------------------------
;                                       JUERGEN STELBRINK   6/13/83
;                                       ADVANCED MICRO DEVICES
;
;            9568 INTERFACE TO THE ISBX-BUS TEST PROGRAM
;
;            (KEY LOAD THROUGH AUXILLARY PORT)
;
;------------------------------------------------------------------------

CSEG
        ORG     100H

0090            ASTROBE EQU     90H             ; ADDRESS STROBE (EVEN ADDRESS)
0091            DSTROBE EQU     91H             ; DATA STROBE (ODD ADDRESS)
0098            ACLR    EQU     98H             ; RESET LOAD KEY LOGIC
0099            CP      EQU     99H             ; 1.OUTPUT: LOAD KEY
                                                ; 2. OUTPUT: INCREMENT ADDRESS
0000            DATA    EQU     00H
0002            CONTROL EQU     02H
0006            MODE    EQU     06H

0100 B0 06              MOV     AL,MODE
0102 E6 90              OUT     ASTROBE,AL
0104 B0 18              MOV     AL,18H          ; DEFINE MODE: MASTER ONLY, ECB, ENCRYPTION
0106 E6 91              OUT     DSTROBE,AL

0108 B0 02              MOV     AL,CONTROL
010A E6 90              OUT     ASTROBE,AL
010C B0 91              MOV     AL,91H          ; LOAD CLEAR E KEY THROUGH AUX PORT
010E E6 91              OUT     DSTROBE,AL

0110 E4 91              IN      AL,DSTROBE      ; READ STATUS REGISTER (AL=44H)

0112 E6 98              OUT     ACLR,AL         ; DUMMY OUTPUT, TO RESET KEY LOAD LOGIC
0114 B9 10 00           MOV     CX,16           ; 16 CLOCKS
0117 E6 99      LAB1:   OUT     CP,AL           ; DUMMY OUTPUT
0119 E0 FC              LOOPNZ  LAB1

011B E4 91              IN      AL,DSTROBE      ; READ STATUS REGISTER (AL=00H)

011D B0 02              MOV     AL,CONTROL      ; LATCH CONTROL REGISTER ADDRESS
011F E6 90              OUT     ASTROBE,AL
0121 E4 91              IN      AL,DSTROBE      ; READ STATUS REGISTER (AL=81H)

0123 B0 C0              MOV     AL,0C0H         ; ENTER START COMMAND
0125 E6 91              OUT     DSTROBE,AL

0127 BB 00 00           MOV     BX,0
012A B9 08 00           MOV     CX,8
012D B0 00              MOV     AL,DATA         ; LATCH DATA REGISTER ADDRESS
012F E6 90              OUT     ASTROBE,AL
0131 2E 8A 87 6A 01 LAB2:   MOV     AL,CS:CLEAR[BX] ; WRITE 1 BLOCK DATA TO INPUT REGISTER
0136 E6 91              OUT     DSTROBE,AL
0138 43                 INC     BX
0139 E0 F6              LOOPNZ  LAB2

013B B0 02              MOV     AL,CONTROL      ; LATCH CONTROL REGISTER ADDRESS
013D E6 90              OUT     ASTROBE,AL
```

```
ASM86 VER 1.0   SOURCE: TESTISBX.A86

  013F E4 91                        IN      AL,DSTROBE      ; READ STATUS REGISTER (AL=83H)

  0141 BB 00 00                     MOV     BX,0
  0144 B9 08 00                     MOV     CX,8
  0147 B0 00                        MOV     AL,DATA         ; LATCH DATA REGISTER ADDRESS
  0149 E6 90                        OUT     ASTROBE,AL
  014B E4 91                LAB3:   IN      AL,DSTROBE              ; READ 1 BLOCK DATA FROM OUTPUT REGISTEF
  014D 2E 88 87 72 01               MOV     CS:CIPHER[BX],AL
  0152 43                           INC     BX
  0153 E0 F6                        LOOPNZ  LAB3

  0155 B0 02                        MOV     AL,CONTROL      ; LATCH CONTROL REGISTER ADDRESS
  0157 E6 90                        OUT     ASTROBE,AL
  0159 E4 91                        IN      AL,DSTROBE      ; READ STATUS REGISTER (AL=81H)

  015B B0 E0                        MOV     AL,0E0H
  015D E6 91                        OUT     DSTROBE,AL      ; ENTER STOP COMMAND

  015F E4 91                        IN      AL,DSTROBE      ; READ STATUS REGISTER (AL=00H)

  0161 CB                           RETF                    ; INTERSEGMENT RETURN

  0162 80 01 01 01 01 01 KEY        DB      80H,1,1,1,1,1,1,1
       01 01
  016A 00 00 00 00 00 00 CLEAR      DB      0,0,0,0,0,0,0,0
       00 00
  0172                   CIPHER     RB      8

                                    END

END OF ASSEMBLY. NUMBER OF ERRORS:   0
```

4.11. 8051 - Am9518/AmZ8068

The 8031/8051/8751 Single-Component 8-Bit Microcomputer family can easily be interfaced to the DCP. Both devices together with TTL logic can form a stand-alone data ciphering system for low- to medium-speed data communication networks. Clear and ciphered data is handled serially with a programmable handshake protocol.

Using the Am9568 eliminates the need of Port 1.x to control Master Port Read/Write. $\overline{RD}$ and $\overline{WR}$ can directly be connected to the corresponding inputs of the DCP ($\overline{MRD}$ and $\overline{MWR}$). ALE does not have to be inverted when connected to MALE.

Figure 4.40 shows the 8051-DCP interface. The 8051 must be programmed so that Port 0 provides a multiplexed address/data bus. Port 0 is connected to the Master Port of the DCP.

$\overline{RD}$ and $\overline{WR}$ are logically ORed to generate the Master Port Data Strobe. Port 1.x controls the Master Port Read/Write input (MR/$\overline{W}$). This satisfies the set-up time requirement of MR/$\overline{W}$ to $\overline{MDS}$.

Master Port Chip Select can be tied Low if it is guaranteed that $\overline{RD}$ or $\overline{WR}$ only become active in a DCP access cycle. Otherwise it must be generated by an address decoder.

Clock Divider

The DCP clock divider logic as shown in Figure 4.40 divides the CPU clock by four or six depending on the type of instruction the CPU executes (See the timing diagram in Figure 4.41). If the CPU generates an ALE every sixth clock, the CPU clock is divided by six. This is the normal case. The speed calculation of the DCP should be done for this clock rate. If the CPU executes "MOVX" instructions, every second ALE is left out and the divide factor is four. For both cases the minimum DCP clock High or Low width is two CPU clock periods which guarantees that even a CPU clock of 12 MHz satisfies the minimum clock requirement for the Am9518 as well as the AmZ8068.

The AmZ8068 gives a wider range for the Data Strobe to $\overline{RD}$ or $\overline{WR}$ delay. The typical value for the 8051 at room temperature with a full load at these outputs is 50 ns.

At a CPU clock rate of 10 MHz, this timing requirement is 0 to 100 ns (two clocks minus 100 ns) for the Am9518 and 0 to 135 ns (two clocks minus 65 ns) for the AmZ8068 at a CPU clock rate of 10 MHz.

Programming

Port 1.x must be High for a read access and Low for a write access. Data is transferred using a "MOVX @Ri,A" or "MOVX A,@Ri" instruction. Ri is register R_0 or R_1. Only this

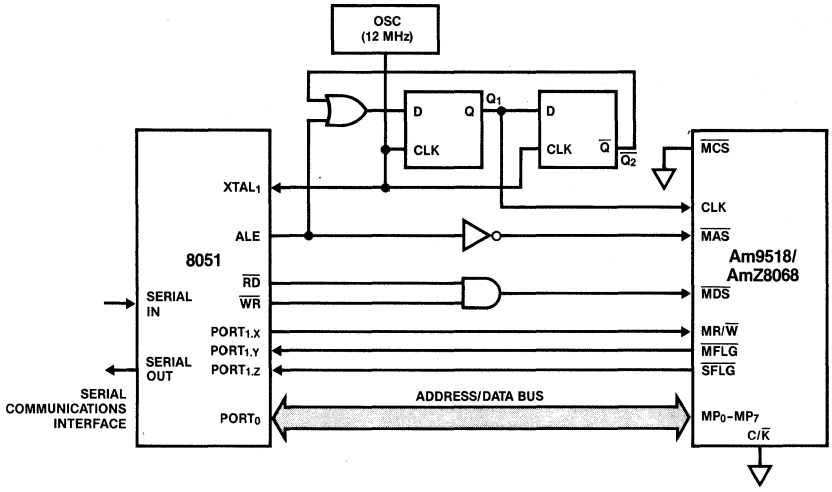


Figure 4.40. 8051-DCP Interface

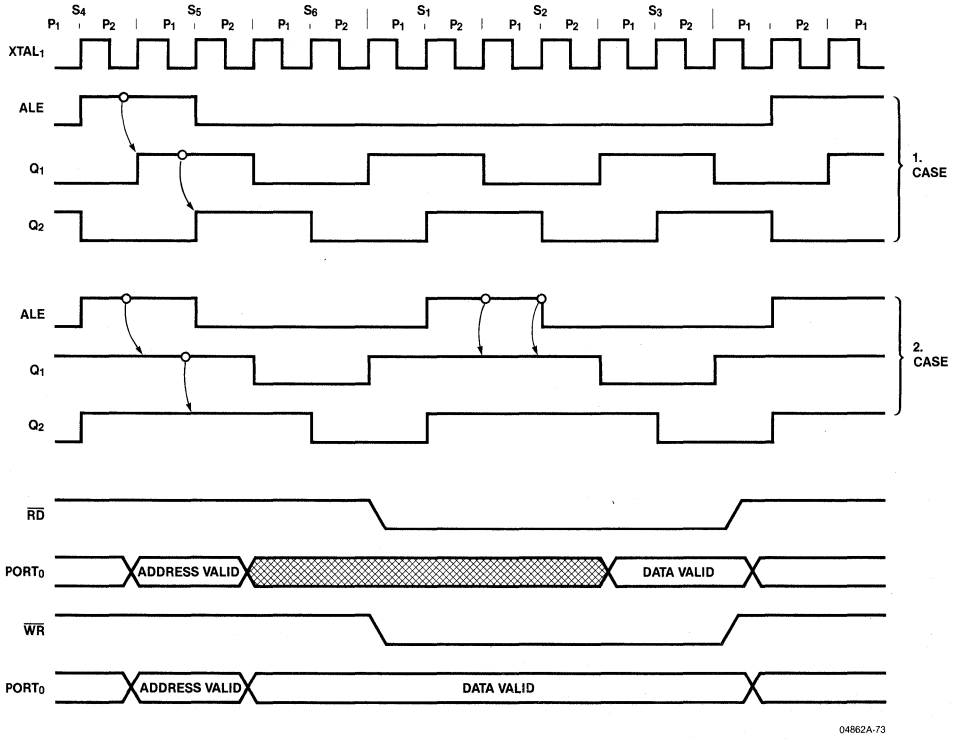


Figure 4.41. 8051-DCP Timing Diagram

instruction generates the interface timing needed for the DCP. The internal register address is loaded into Rn before executing this instruction.

ØØ - Data Input or Output Register

Ø2 - Command or Status Register

Ø6 - Mode Register

The Flags can be monitored by two input pins of the CPU, Port 1.y and 1.z. One Flag corresponds to the status of the Input Register, the other one to the status of the Output Register. They become active Low if the CPU can perform a data transfer. For details refer to Chapter 3.1.

In high-speed data ciphering applications, it might be too time consuming to toggle Port 1.x (MR/$\overline{W}$). The toggling can be avoided by choosing the dual port configuration of the DCP. Both the Master and Slave Port are connected to Port Ø of the CPU. During the data ciphering session, one port operates as the data input port, the other port operates as the data output port. This means that during the whole session, the data flow direction does not have to be turned around; MR/$\overline{W}$ can stay Low or High for the whole session. $\overline{MCS}$ and $\overline{SCS}$ select the appropriate port.

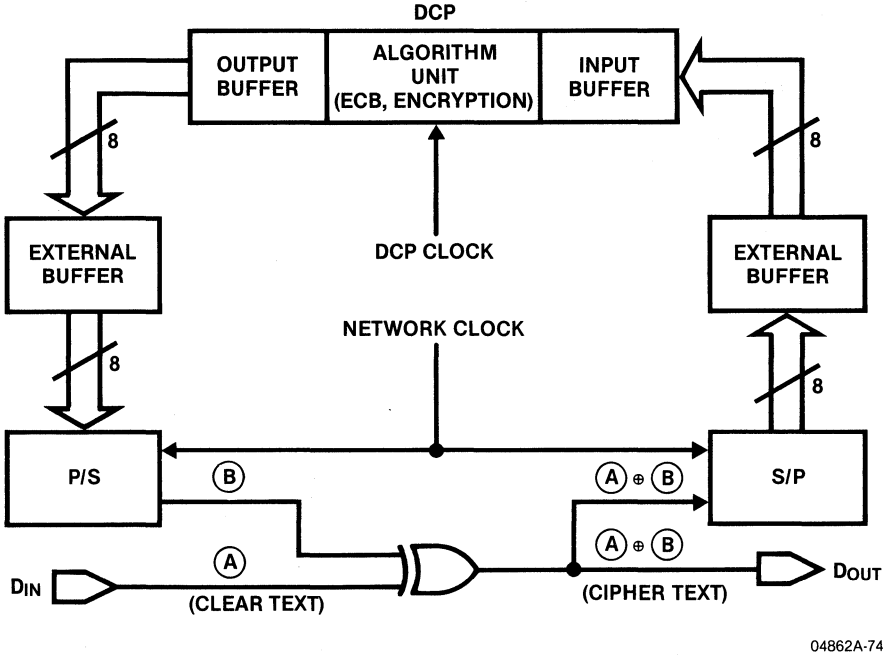


Figure 4.42. Network Transmitter

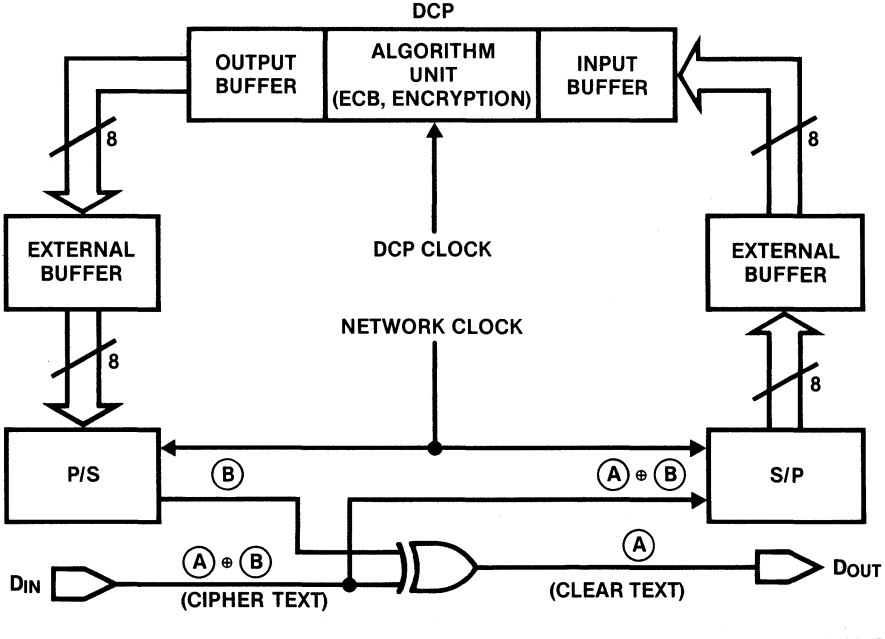


Figure 4.43. Network Receiver

4.12. HIGH SPEED SERIAL DATA CIPHERING IN NETWORK SYSTEMS

This chapter discusses the use of the data encryption chip (Am9518/AmZ8Ø68) in local area networks. In some of these applications, it is desirable to use encryption as an option to an existing system. When this happens, the option board may have to take serial data from the former network driver and reprocess the data to transmit and receive cipher test. The following discussion should shed some light on a practical approach to this problem.

First, the system must meet the required level of security. This is a system philosophy problem related to the handling of keys, CRC generation, and system partitioning. Secondly, data must meet transmission requirements such as continuous transmission of data, non-block size packet length, and transparency. The second requirement, which is the concern of this note, is a hardware configuration problem.

The DCP (Am9518/AmZ8Ø86) can be configured to cipher data at up to 14.2 Mbits/s. This can be accomplished by using the device in Direct Control Mode with a feedback path between the output port of the unit and its input port. The DCP may be looked upon as a three stage system: the input buffer, the output buffer and the algorithm unit. The DCP handles data in 64-bit (ECB and CBC) or 8-bit (CFB) blocks. Between block transfers the system has to provide a recovery time of five clocks to allow the DCP to update its internal flags. External Buffers smooth this discontinuous data flow to provide a continuous data flow onto the network (see Figures 4.42 and 4.43).

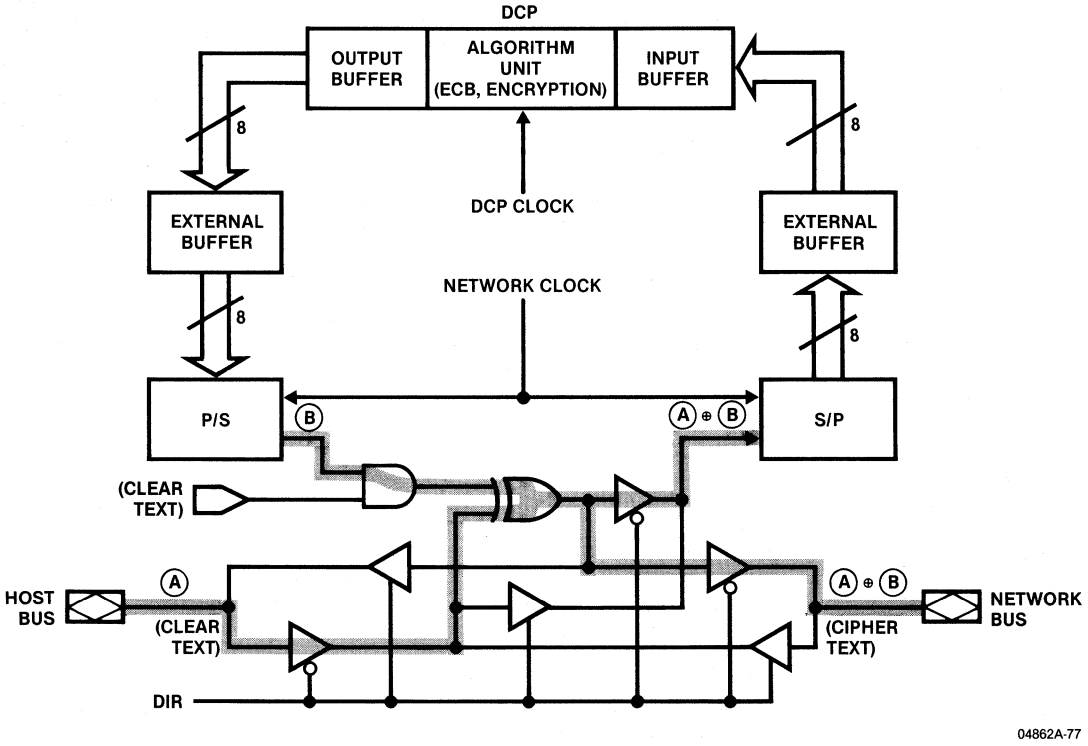
The system may be looked upon as a closed system in which the number of bytes in the system remain constant. Therefore, if nine bytes are rotated, the system would be initialized with eight bytes in the output buffer and one in the input buffer. At some time there would be eight bytes in the input buffer ready to move into the algorithm unit and one byte in the output buffer ready to be loaded into the P/S-XOR-S/P feedback circuit. Operation on the data will take eight network clocks. The data moving through the algorithm unit will take 23 DCP clocks (5.75 microseconds for the 4-MHz 8Ø86). This would allow a frequency of 1.39 MHz for the network clock. If 1Ø bytes were allowed to circulate in the system, one byte would still be available in the output buffer while one was being shifted through the feedback circuit, and a block was being processed in the algorithm unit. This would allow 16 network clocks to transpire during the 5.75 microseconds that data moved through the algorithm unit. This would allow a network clock of 2.78 MHz.

This reasoning holds until the data must be stored in an external buffer during the flag inactive period of the input and output DCP buffers. The inactive period is five DCP clocks of 1.25 microseconds for the 4-MHz AmZ8Ø68. This happens when the network clock is 6.4 MHz. At this rate additional buffering,

Number of Initialization Bytes	Number of Bits in Circulation	Minimum Period in μsec ($5.75\ \mu s$/# bits)	Maximum Network Clock in MHz
9	8	0.718	1.39
10	16	0.359	2.78
11	24	0.220	4.17
12	32	0.180	5.75
13	40	0.144	6.9
14	48	0.112	8.33
15	56	0.103	9.74
16	64	0.0899	11.13

04862A-76

Figure 4.44. Maximum Network Clock as a Function of the Number of Bits in Circulation



04862A-77

Figure 4.45. Bidirectional Interface, Transmit Mode

external to the DCP, is required. This would allow data to be stored in the external buffer while data is transferred from the algorithm unit to the output buffer on the output port, or from the external input buffer to the input buffer on the input port, while data from the input buffer is being transferred to the algorithm unit. The foregoing analysis holds up to 11 MHz (See Figure 4.44).

To operate at the maximum frequency of 1.78 Mbytes/s, or 14.2 Mbits/s, three additional initialization bytes must be added to the system, making a total of 19 bytes. This scheme is based on pipelining scheme A: minimum timing operation. The idea is to have enough data in the system to allow transfers through the algorithm unit in 18 DCP clocks. During the time data is being moved to or from the algorithm unit (1.25 microseconds) the external buffers must store 18 bits. This would require two registers in addition to the feedback circuit.

The maximum number of bytes that can be used to initialize the DCP results from the need to minimize buffering while providing continuous data to the network. During the period when the DCP is in a lockout phase, there are 16 bytes in the DCP and the remaining number of bytes reside in the external buffers. This would correspond to a condition in which the output buffer has just been emptied and the algorithm unit and input buffer are full. The lockout period takes five DCP clocks or 1.25 microseconds. During this time, 18 bits must be transferred in order to meet network requirements. This requires that three buffer locations be available. Since there are six to begin, only 3 bytes can be stored externally; therefore, the maximum number of initialization bytes allowed would be 19.

Figures 4.45, 4.46, and 4.47 show a block diagram of a system that will handle data from the bus or network side of the board. The controller must be able to handle some of the link functions. In particular, it must be able to respond to clear text or cipher text on a real-time basis. It must synchronize data transfers between the DCP, the buffers and the host or network buses, and initialize the DCP. Data is most rapidly transferred in Direct Control Mode; however, the DCP must also be able to manipulate keys and Initial Vectors. This requires switching to Multiplexed Control Mode, as these functions are not supported in Direct Control Mode. It must also be able to set the DCP to ECB, CBC, or CFB encrypt or decrypt modes. Because the cipher text may inadvertently contain control characters, it must be deciphered before it is decoded or the system must be operated in Transparent Mode. In addition to the normal transmission characters, it is usually desirable to add a message number or date stamp to the front of the encrypted data and include the destination address.

The initialization time required would be at least 31 clocks x Ø.25 microseconds/clock or 7.75 microseconds. This could be done during the clock time when the network is recovering from the previous transmission.

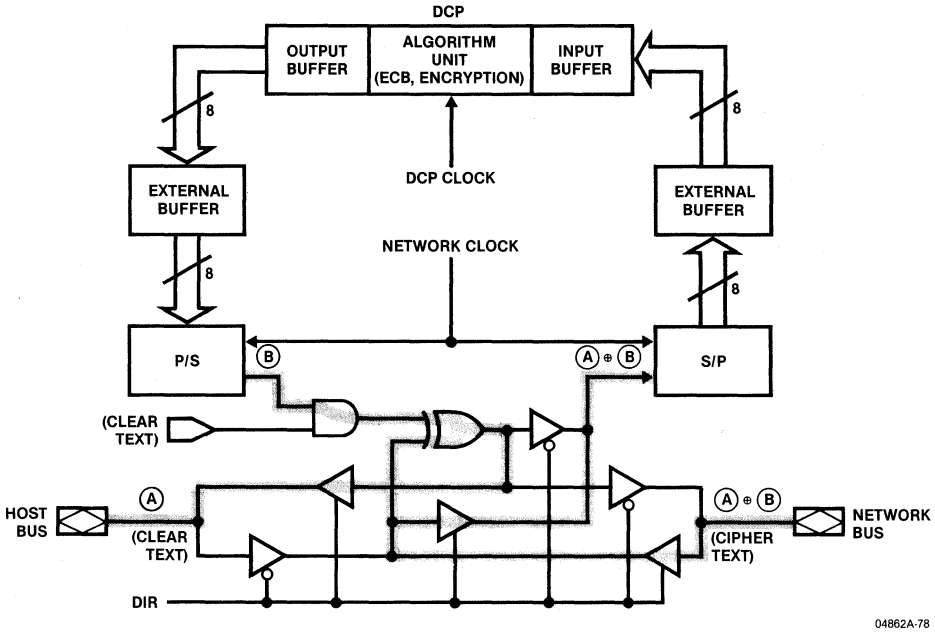


Figure 4.46. Bidirectional Interface, Receive Mode

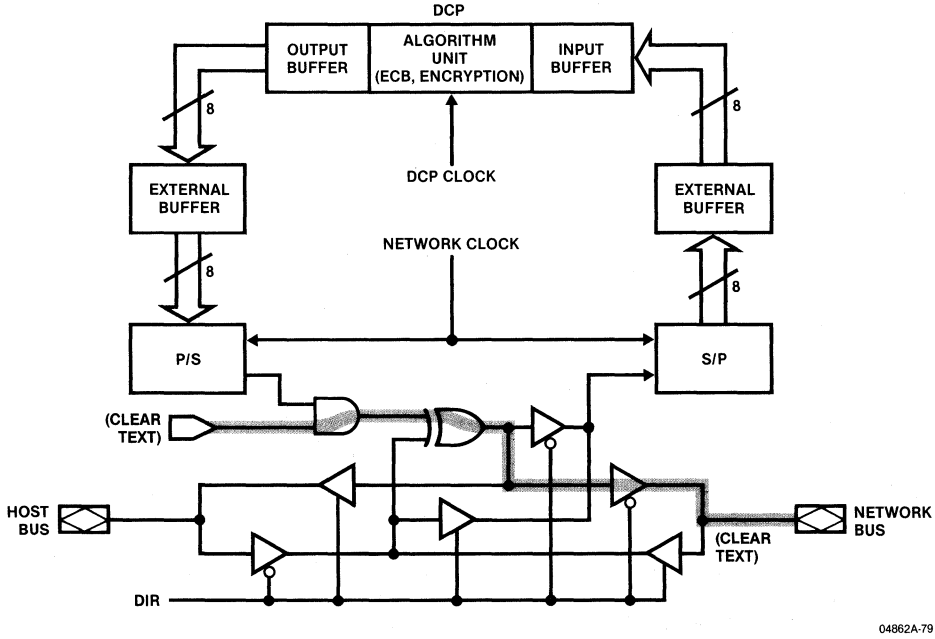


Figure 4.47. Bidirectional Interface, Transparent Mode

The previous information has discussed the possibility of using the DCP in a link application in which only serial data is transferred between the host and network. We have found that the DCP can run at its maximum transfer rate by adjusting the initialization data and the amount of external buffering. We have also looked at some of the requirements for the controller in a secure network environment. We can conclude that the DCP may be used effectively in a link application at rates up to 14.2 MHz.

APPENDIX A. Electronic Codebook (ECB) Test Data

E-Key = D-Key = Ø123456789ABCDEF

Encryption:

Time	Plain Text	Cipher Text
1	4E6F772Ø69732Ø74	3FA4ØE8A984D4815
2	68652Ø74696B652Ø	6A271787AB8883F9
3	666F7220616C6C2Ø	893D51EC4B563B53

Decryption:

Time	Cipher Text	Plain Text
1	3FA4ØE8A984D4815	4E6F772Ø69732Ø74
2	6A271787AB8883F9	68652Ø74696B652Ø
3	893D51EC4B563B53	666F7220616C6C2Ø

The plain text is the ASCII code for "Now is the Time for all..." These seven-bit characters are written in the hexadecimal notation (Ø,b6,b5,b4,b3,b2,b1,bØ).

APPENDIX B. Cipher Block Chaining (CBC) Test Data

E-Key = D-Key = Ø123456789ABCDEF
IVE = IVD = Ø123456789ABCDEF

Encryption:

Time	Plain Text	Cipher Text
1	4E6F772Ø69732Ø74	E5C7CDDE872BF27C
2	68652Ø74696D652Ø	43E934ØØ8C389CØF
3	666F7220616C6C2Ø	683788499A7CØ5F6

Decryption:

Time	Cipher Text	Plain Text
1	E5C7CDDE872BF27C	4E6F772Ø69732Ø74
2	43E934ØØ8C389CØF	68652Ø74696D652Ø
3	683788499A7CØ5F6	666F7220616C6C2Ø

The plain text is the ASCII code for "Now is the Time for all..." These seven-bit characters are written in the hexadecimal notation (Ø,b6,b5,b4,b3,b2,b1,bØ).

APPENDIX C. Eight-bit Cipher Feedback (CFB) Test Data

E-Key = D-Key = Ø123456789ABCDEF
IVE = IVD = Ø123456789ABCDEF

Encryption:

Time	Plain Text	DES Input (IVE)	DES Output	Cipher Text
1	4E	1234567890ABCDEF	BD661569AE874E25	4E+BD = F3
2	6F	34567890ABCDEFF3	7039546F9A0F6330	6F+70 = 1F
3	77	567890ABCDEFF31F	AD1B78B0BB371BE7	77+AD = DA

Decryption:

Time	Cipher Text	DES Input (IVD)	DES Output	Plain Text
1	F3	1234567890ABCDEF	BD661569AE874E25	F3+BD = 4E
2	1F	34567890ABCDEFF3	7039546F9A0F6330	1F+70 = 6F
3	DA	567890ABCDEFF31F	AD1B78B0BB371BE7	DA+AD = 77

The plain text is the ASCII code for "Now is the Time for all..." These seven-bit characters are written in the hexadecimal notation (Ø,b6,b5,b4,b3,b2,b1,bØ). The "+" represents the EXOR-function.

APPENDIX D. Certification by National Bureau of Standards

National Bureau of Standards

DATA ENCRYPTION STANDARD (DES) VALIDATION CERTIFICATE

The National Bureau of Standards has tested the encryption device identified as AmZ8068 (also known as Am9518) and manufactured by Advanced Micro Devices, Inc. in accordance with the specifications of the Data Encryption Standard (Federal Information Processing Standard 46) and in accordance with the procedures specified in NBS Special Publication 500-20.

The device has passed the DES test and in addition has passed a Monte Carlo test that lasted four million iterations. For the Monte Carlo test the initial value of the key was 6B1038B367D980E5 and the initial value of the input was 073F292FB9BC2DDE . The final value of the key was B9234507D31A07AD and the final value of the output was was B599E74AF567496A .

Devices bearing the same identification and manufactured to the same design specifications may be labeled as complying with the Data Encryption Standard. No reliability test has been performed and no warranty of the devices by the National Bureau of Standards is either expressed or implied.

28 Jan 1981
Date

S. Jeffery, Director
Center for Programming
Science and Technology
Institute for Computer Sciences
and Technology

Appendix D

National Bureau of Standards

DATA ENCRYPTION STANDARD (DES) VALIDATION CERTIFICATE

The National Bureau of Standards has tested the encryption device identified as AM 9568 and manufactured by Advanced Micro Devices Inc. in accordance with the specifications of the Data Encryption Standard (Federal Information Processing Standard 46) and in accordance with the procedures specified in NBS Special Publication 500-20.

The device has passed the DES test and in addition has passed a Monte Carlo test that lasted four million iterations. For the Monte Carlo test the initial value of the key was 9DFE6DD3457A9DB9 and the initial value of the input was 3F98477A85B300FD . The final value of the key was FB8929CE83A2737C and the final value of the output was was 404CB50060AE6C04 .

Devices bearing the same identification and manufactured to the same design specifications may be labeled as complying with the Data Encryption Standard. No reliability test has been performed and no warranty of the devices by the National Bureau of Standards is either expressed or implied.

February 28, 1984
Date

Dennis K. Branstad

```
/*********************************************************************
*       Am9568 Certification Program                        2/22/84     *
*********************************************************************/

#include <bdscio.h>
char buffer[BUFSIZ];
char file[12];

mode(value)                         /* initialize mode register of DCP */
int value;
{
outp(0x80,0x06);                    /* address mode register */
outp(0x81,value);                   /* ECB, master port only */
}

command(value)                      /* issue command "value" to the DCP */
int value;
{
outp(0x80,0x02);                    /* address command register */
outp(0x81,value);                   /* load command */
}

write_block(text)                   /* write one block */
int text[];
{
int i;
outp(0x80,0x00);                    /* address data register */
for(i=0;i<=7;i++)                   /* load 8 bytes */
  outp(0x81,text[i]);
}

read_block(text)                    /* read a block */
int text[];
{
int i;
outp(0x80,0x00);                    /* address data register */
for(i=0;i<=7;i++)                   /* read 8 bytes */
  text[i]=inp(0x81);
}

encrypt(clear,cipher)               /* encrypt one block */
int clear[],cipher[];
{
command(0x41);                      /* start encryption */
write_block(clear);
read_block(cipher);
command(0xe0);                      /* stop */
}

decrypt(cipher,clear)                   /* decrypt one block */
int clear[],cipher[];
{
command(0x40);                          /* start decryption */
write_block(cipher);
read_block(clear);
```

```
command(0xe0);                          /* stop */
}

key_load(value,key)                     /* load 56-bit key into DCP */
int value,key[];
{
command(value);
write_block(key);
}

show(n,text1,text2)                     /* write one line to the file */
int n,text1[],text2[];
{
printf("This is pass %d\n",n);
fprintf(buffer," KEY(%4d) = %02x%02x%02x%02x%02x%02x%02x%02x%02x    ",n,
        text1[0],text1[1],text1[2],text1[3],text1[4],text1[5],text1[6],
        text1[7]);
fprintf(buffer,"DATA(%4d) = %02x%02x%02x%02x%02x%02x%02x%02x%02x\n",n,
        text2[0],text2[1],text2[2],text2[3],text2[4],text2[5],
        text2[6],text2[7]);
}

error(keys,rounds)                      /* print error message */
int keys,rounds;
{
fprintf(buffer,"Comparison error for keys = %d and rounds = %d\n",keys,round
exit();
}

odd_parity(text)                        /* generate odd parity of array */
int text[];
{
int i,j,n;
for(i=0;i<=7;i++)
   {
   n=text[i]&1;
   for(j=1;j<=7;j++) n=n^((text[i]>>j)&1);
   n=n^1;
   text[i]=text[i]^n;
   }
}

main()
{
#define keys 400
#define rounds 10000
int i,j,ic,pln1[8],pln2[8],pln3[8];
strcpy(file,"CERT.DAT");                /* define filename */
if(fcreat(file,buffer)==ERROR)
  {
  printf("File already exists\n");
  exit();
  }
fprintf(buffer,"AMD#2 Am9568 Certification Data:              Feb-23-84\n\n"
```

```
ode(0x18);
ln1[0]=0x3f;pln1[1]=0x98;pln1[2]=0x47;pln1[3]=0x7a;   /* init plain text */
ln1[4]=0x85;pln1[5]=0xb3;pln1[6]=0x00;pln1[7]=0xfd;
ln2[0]=0x9d;pln2[1]=0xfe;pln2[2]=0x6d;pln2[3]=0xd3;   /* init key */
ln2[4]=0x45;pln2[5]=0x7a;pln2[6]=0x9d;pln2[7]=0xb9;
=0;
how(i,pln2,pln1);
or(i=1;i<=keys;i++)
  {
  key_load(0x11,pln2);              /* load encryption key */
  key_load(0x12,pln2);              /* load decryption key */
  for(j=1;j<=rounds;j++)
     {
     encrypt(pln1,pln2);            /* encrypt twice */
     encrypt(pln2,pln1);
     decrypt(pln1,pln3);            /* decrypt block to verify */
     for(ic=0;ic<=7;ic++)           /* operation of DCP */
        if(pln2[ic]!=pln3[ic]) error(i,j);
     }
  odd_parity(pln2);                 /* modify new key for odd parity */
  show(i,pln2,pln1);                /* load result into file */
  }
utc(CPMEOF,buffer);                 /* put EOF mark into file */
flush(buffer);                      /* flush buffer to disk */
close(buffer);                      /* close file */
```

Appendix D

AMD#2 Am9568 Certification Data: Feb-23-84

KEY(0) =	9DFE6DD3457A9DB9	DATA(0) =	3F98477A85B300FD
KEY(1) =	51AD1391CDBF7AAD	DATA(1) =	10B447B6B53242C8
KEY(2) =	B9A2FB298AC18C67	DATA(2) =	8947274835DE2B10
KEY(3) =	DA58E08A3B7CD9D9	DATA(3) =	F07774A0985A1426
KEY(4) =	38B310161CBCA2A8	DATA(4) =	45CC342BF898B00A
KEY(5) =	6ECEF4756BDCF49D	DATA(5) =	F2B6375FAB01839E
KEY(6) =	EA45B394683B9DFE	DATA(6) =	6AE3FEEBA7EBB8C9
KEY(7) =	B0E023736B89FD83	DATA(7) =	852ACBF25D8A57AE
KEY(8) =	6D989415073DFE04	DATA(8) =	4B3586841CBCDC2D
KEY(9) =	1C6E9D4ABA37B35D	DATA(9) =	643B492C10E33EAB
KEY(10) =	C797618526CBC49B	DATA(10) =	9D17D98CDE6BEAEE
KEY(11) =	F4C42CCECDF2ADDC	DATA(11) =	D0129C0487D56EA3
KEY(12) =	43080157EFAE04B0	DATA(12) =	F37C3BEF5496184F
KEY(13) =	DF5EBFFB5E204C64	DATA(13) =	38B1D27307F5B1EA
KEY(14) =	32648F9BA8798A5B	DATA(14) =	73B27722687B44D3
KEY(15) =	B531450719343454	DATA(15) =	880ACED367B543B2
KEY(16) =	6B4CF27A68CBC8C1	DATA(16) =	309CB7900E3B61C0
KEY(17) =	20C84C91F7344351	DATA(17) =	E22871C470836511
KEY(18) =	7C02A79E2C7C38CE	DATA(18) =	A5ADC80285F43777
KEY(19) =	58760DA8A8E3B089	DATA(19) =	61693B23CA9AA67E
KEY(20) =	85A298020D8A6D86	DATA(20) =	C1F946029706DC2D
KEY(21) =	9BFD753D3BDAEA98	DATA(21) =	4C33767B6E1A4E4C
KEY(22) =	626B850431F8B58A	DATA(22) =	03D98B090B901063
KEY(23) =	FB10F8B985E9B3B0	DATA(23) =	85504CF4072BC45F
KEY(24) =	257FFB70CBADE094	DATA(24) =	0C3DFEFBB364657F
KEY(25) =	C8F70276D0942AD6	DATA(25) =	CF014ADDDD418668
KEY(26) =	C17038E0FE6B4A94	DATA(26) =	7BB8C2A0B4CD2900
KEY(27) =	9829AB75DA5E9401	DATA(27) =	B4A7D98CB0AEBC58
KEY(28) =	4AEC01737AE3C767	DATA(28) =	68028F9B1FDF151B
KEY(29) =	86264C265DD6EC31	DATA(29) =	41A32D0221E37265
KEY(30) =	6B1038B367D980E5	DATA(30) =	073F292FB9BC2DDE
KEY(31) =	70496DCDEC155261	DATA(31) =	56FFA102DE7A2156
KEY(32) =	F480E0380B94C45E	DATA(32) =	E48E2D08DA845585
KEY(33) =	40BF5BA1A264F237	DATA(33) =	1399742A091D7C06
KEY(34) =	5ED0898A68BF455D	DATA(34) =	68FA2A0CCAA01464
KEY(35) =	4FC1B0BFDA0BE554	DATA(35) =	25D1F75FFDE14A93
KEY(36) =	4A43616EC89E86D3	DATA(36) =	5006CE31CC7BA3D9
KEY(37) =	F151F8DF1F583DD0	DATA(37) =	87DBE71F4B35583F
KEY(38) =	D0BAF42A375DCDD0	DATA(38) =	B4AE933196D30A59
KEY(39) =	02EADACDA7A70861	DATA(39) =	D97446565310401E
KEY(40) =	7075D337EF345D15	DATA(40) =	AB2164B792E066C4
KEY(41) =	013DE98A46D93E83	DATA(41) =	DED9E86480E9BF55
KEY(42) =	45CE3D2ABA2076EC	DATA(42) =	A3F26B7B30C86AC6
KEY(43) =	850D23E661D552BA	DATA(43) =	BABE64BC8B1EA6A9
KEY(44) =	4361A4A7C76B62E9	DATA(44) =	FC596A1EACFD21B4
KEY(45) =	1A5DA26E6B3D4AA4	DATA(45) =	F07EB7D219C56CED
KEY(46) =	B07FF88A290B3B08	DATA(46) =	F62C61D5EE647AA3
KEY(47) =	072079C740947002	DATA(47) =	CF99A25984AC6454
KEY(48) =	A8EFDAE654BFBCD0	DATA(48) =	4BB53BC42CE91E5F
KEY(49) =	8A675B9EC19204AE	DATA(49) =	321646D5733BFD67
KEY(50) =	D9E9F2F7E39858B0	DATA(50) =	6F67FBC3A6A3EDF2
KEY(51) =	46F46E91FB7CCE8F	DATA(51) =	2E2965810068EEDB
KEY(52) =	BCADF40B94B5204F	DATA(52) =	3BAEE2156A0B2CD5

```
KEY(  53) = EC4929F42C024F62    DATA(  53) = ED6E45612F350959
KEY(  54) = B5D51A629B252CC8    DATA(  54) = AA76DD7DEBB37402
KEY(  55) = 0879731F0470D6EC    DATA(  55) = 355933E36FC8A565
KEY(  56) = 4FDFEC267FFEBCD6    DATA(  56) = AFFE7C400A8651AD
KEY(  57) = 68318A3D86649EE9    DATA(  57) = 29C08984FD68F4C4
KEY(  58) = 0E1C040875436BAB    DATA(  58) = CA5DED97C80C73F0
KEY(  59) = 9B58EC345DD620B3    DATA(  59) = 71E883DEC868470 5
KEY(  60) = 2994E6510E20E5E5    DATA(  60) = C5B0F6CF2E5464AD
KEY(  61) = 75F49B25B3AE0DDA    DATA(  61) = 7E246CF8F714E459
KEY(  62) = CB31BC3DCB61F245    DATA(  62) = 238CABABA58606EA
KEY(  63) = 4A9D2A4C6B5B4AFD    DATA(  63) = C1730770A7FDC5C3
KEY(  64) = BFBCD69898D32C8A    DATA(  64) = 4C24233FF8234017
KEY(  65) = 83BAF27026C745A1    DATA(  65) = 6EBE02E3E4F2E396
KEY(  66) = 6EEABF68EF683EF1    DATA(  66) = DA169E824285756D
KEY(  67) = 4FBC5E7FE607E39E    DATA(  67) = 3771FEC4F271325B
KEY(  68) = A8494C5E732CF17A    DATA(  68) = 029DA71E13D9A38D
KEY(  69) = F8FDDF5E2F97D092    DATA(  69) = 094F361B173D25AA
KEY(  70) = 618F9132CB640B07    DATA(  70) = 0CBD4112B9F15D4C
KEY(  71) = 437C7C34B3FB4F61    DATA(  71) = 66AD98CD4C65344D
KEY(  72) = F2E60BBCB6AD2CB5    DATA(  72) = CB6F597AAC228AAF
KEY(  73) = 948A29B54AA854F2    DATA(  73) = A0CDB91B41FD8EF2
KEY(  74) = 75BC3104ABA468AD    DATA(  74) = 117BF060B11ABB12
KEY(  75) = A2802554077F832A    DATA(  75) = D3E825FF1BF6A175
KEY(  76) = E9A11AE57C01CD83    DATA(  76) = 84F534E60CC1CEB8
KEY(  77) = 51B5DA7FEC389D6B    DATA(  77) = DE75D30EA5DEF075
KEY(  78) = 52B0976EC1B5310E    DATA(  78) = 6A1850A098E24B08
KEY(  79) = 37A42F3DF8C75B4A    DATA(  79) = 338364A073CA6EF5
KEY(  80) = 2C890E89162C7515    DATA(  80) = CE19B1FFD282C78D
KEY(  81) = C4C7080BC115B043    DATA(  81) = 1374CCDB7A167ACA
KEY(  82) = AD540DA8648394A7    DATA(  82) = C235720039454D1F
KEY(  83) = 0EB3D5A4AD106D92    DATA(  83) = EFE10687C6603191
KEY(  84) = 6D325EB3C8526D73    DATA(  84) = BE16A9316648E836
KEY(  85) = 581932DAA74C29CB    DATA(  85) = A58B70893D2E6B4A
KEY(  86) = C2DC8FC1E0853E4A    DATA(  86) = 62DAEE9BE5AB2C14
KEY(  87) = AE299BA19280139B    DATA(  87) = E23ADE1A17B568F2
KEY(  88) = 1576DC52EAE0A162    DATA(  88) = 9AB844FC293A8A5A
KEY(  89) = 94C2B568E5018F13    DATA(  89) = 4F1F2F7C183C8B7A
KEY(  90) = 761C7526254CFE4C    DATA(  90) = F334F6BD1B282D61
KEY(  91) = 83CD29D0FB9B2AC2    DATA(  91) = 016184E731297F4D
KEY(  92) = 58571A83CE791A3E    DATA(  92) = FA8279C2C91B5343
KEY(  93) = 6B3B3B3E7C973D91    DATA(  93) = 9984E4E8EF4D6F5A
KEY(  94) = FE469BBCEF79136E    DATA(  94) = 0E7E16D8A375304D
KEY(  95) = 264C5191C2A29EB3    DATA(  95) = 969E778016097595
KEY(  96) = AD6BAED367BF6140    DATA(  96) = 0162DB0A30101D1D
KEY(  97) = 43DC6B316E2F2302    DATA(  97) = C3AE3D98BE39DF0E
KEY(  98) = BAE39423A16BAB13    DATA(  98) = FF0120F0CFB99A44
KEY(  99) = 573E267F048AF7DF    DATA(  99) = E1D9FA30CEC6DA1F
KEY( 100) = 01831AA71A4C0E0B    DATA( 100) = 15F4898C2B414582
KEY( 101) = 165243F24C349B19    DATA( 101) = 273816D72C9667B5
KEY( 102) = AB6DC8FB3BBA0B13    DATA( 102) = E295882E9C608F5F
KEY( 103) = 58FE51327F6849D5    DATA( 103) = D331D15BFDE66AF0
KEY( 104) = F26152EC89A451D6    DATA( 104) = 3C9B49A5DA25E4F1
KEY( 105) = 9413A41F2FCE8F37    DATA( 105) = CE193A8372D2059A
KEY( 106) = F1ADB38AD6EAAED0    DATA( 106) = 4B5CDE71D09C96F7
KEY( 107) = 61EAC8191C61CBAD    DATA( 107) = 64880B6D104BFAD6
```

Appendix D

KEY(108) = E37F8A73FDABF41A	DATA(108) = 4AEF220C883C0B25
KEY(109) = EAA710529EBABAC7	DATA(109) = 3506609DC298CD44
KEY(110) = DF52E385B64A02FE	DATA(110) = 45961A64255773F2
KEY(111) = DF296DAE312976B6	DATA(111) = D526C0B5899B3519
KEY(112) = 7608DAD67F40765E	DATA(112) = DB833F7D802AF4FB
KEY(113) = DAA4E0C49440F1EC	DATA(113) = 03BD84A3A61B3C89
KEY(114) = 85B3EA6DD6269467	DATA(114) = A7EE146CA8DDF1BB
KEY(115) = 23D094D31A571FAE	DATA(115) = F1949D2CBECA8910
KEY(116) = CDE6FB49CB7A9BDC	DATA(116) = E8696A082ED64BE9
KEY(117) = D068627A017398AB	DATA(117) = 5ACBFC953A6F5064
KEY(118) = 4552676E0426B9E0	DATA(118) = A3936766BE3C44E2
KEY(119) = 852A026419381FD3	DATA(119) = DA24F20A113845E4
KEY(120) = 5DFD2F9BF20D7A1A	DATA(120) = BBFB892A1C9705DC
KEY(121) = 948C3B91D362F7B6	DATA(121) = 987B508E5F9DAC22
KEY(122) = 54BA08ECB9AEDCEC	DATA(122) = 97C36AF7B1140A15
KEY(123) = 4AF154A26E405783	DATA(123) = 49DBD96B35360264
KEY(124) = E0BC9794C80BBF1A	DATA(124) = 2002EA1433AE1488
KEY(125) = 02A757F7615BAE31	DATA(125) = 208F2B28F47032C3
KEY(126) = 62707CD554453E54	DATA(126) = 72EA5B23504650F4
KEY(127) = 15C4A49BB3082A85	DATA(127) = BD809AD497E54A43
KEY(128) = FB98A42AC7AE8FF7	DATA(128) = B53ADD87458C17F4
KEY(129) = B9D0CB49CB191F92	DATA(129) = 8B5C7A96CFEE40B5
KEY(130) = 253B52455446A843	DATA(130) = 1E77DFB27C704EEC
KEY(131) = A2C175D5B06BF862	DATA(131) = DFB52005F5CAF5EC
KEY(132) = 8F25FEAED62AC8D5	DATA(132) = 373822390C2E3BB9
KEY(133) = 75CD04E619346D43	DATA(133) = C4416EF6236B4B71
KEY(134) = 7A62400D3E3E624A	DATA(134) = CE0FB5A7B106B1E7
KEY(135) = C7D93DA134204AA8	DATA(135) = A0748C70DAD49ADA
KEY(136) = 942F913DC7341079	DATA(136) = AFD66D112036E8FA
KEY(137) = D04C5208F7B97632	DATA(137) = D1464F19AA431C44
KEY(138) = B61A31252CAE345D	DATA(138) = 49056658733A87AC
KEY(139) = B9E55761040898CD	DATA(139) = E45249DBB0386669
KEY(140) = A48013AE891C0EEC	DATA(140) = BC103391D32DC120
KEY(141) = 327C4C7A467358C7	DATA(141) = 9E57FF52B6D73862
KEY(142) = 4CBF491FABFB5131	DATA(142) = 114E1C6AB4FA348F
KEY(143) = A467A71FE979BAF8	DATA(143) = 8AE70F2F306E7819
KEY(144) = B0A7D661B504CD04	DATA(144) = 0BEF21FBE4FDA9EC
KEY(145) = 8F074F6240E59B97	DATA(145) = 448ACDB5725CD63E
KEY(146) = 8F15ECCDBC1F0D20	DATA(146) = B72E1690609E6009
KEY(147) = D0018FCE343149A8	DATA(147) = 17AAD80F1207C524
KEY(148) = 8A7540E345676D15	DATA(148) = DA8DCB4D7DFE4FA2
KEY(149) = CE624C34C185EA26	DATA(149) = 40870A18FB515AF3
KEY(150) = 04E63BB01C1C75C7	DATA(150) = 89BF1DFE12E1A227
KEY(151) = 68833489151F015E	DATA(151) = 307A153565AD45BE
KEY(152) = 7552B3515775EAB5	DATA(152) = BB53A4248831FC16
KEY(153) = 01F4F80149D6B957	DATA(153) = 9022A940C966CEED
KEY(154) = 67294632675EEA5E	DATA(154) = 3D7E64DAC80EA48E
KEY(155) = A23E92DFEFCE9D0B	DATA(155) = 3142CB1751A6092D
KEY(156) = BFD3FE51C26823D3	DATA(156) = 074789A18A036567
KEY(157) = B91AF73B5ED03416	DATA(157) = 69B4D4A5D1B4FA3B
KEY(158) = E98504253B97C4EC	DATA(158) = 775A0E9F61F4DB47
KEY(159) = 807AC8D958A1DCB3	DATA(159) = 209E90BBC6F13F4E
KEY(160) = 3245D9E349466489	DATA(160) = 915592B2669DF526
KEY(161) = C8E002E6D06BAB1F	DATA(161) = 2995DAB0F349E12A
KEY(162) = 940ED320151C45DA	DATA(162) = 72CB2DF78A2F7FB3

KEY(163) = AD20B991C12FDF6E DATA(163) = 421691BDE2150lA5
KEY(164) = DC708A3280163D2C DATA(164) = 26EC847D00E4B3C0
KEY(165) = D5A783A11586236B DATA(165) = 7E8F28F6DC9A46E9
KEY(166) = 7A260ED6A2F4E315 DATA(166) = CE4CEFEE1CC54BE1
KEY(167) = 5757709DA8155EA8 DATA(167) = 8F31AC9EB64EF458
KEY(168) = CD7CA2F72C324FCD DATA(168) = 2DCA8F7A7B076C23
KEY(169) = 045D25FD2C5E2A02 DATA(169) = 1D923FDE7AABADDE
KEY(170) = B39119A73D2C6B5E DATA(170) = 1E29F3026E1AFFEB
KEY(171) = 85F25EDF91F7CBA1 DATA(171) = 0D1927E668022411
KEY(172) = 491F2CECEFB9BA52 DATA(172) = 484DF1EEC700CF8F
KEY(173) = 5B681F0457D60BDF DATA(173) = D8702322FCE81358
KEY(174) = 67DAEC7F4C755726 DATA(174) = 3FB456D8CAB5FAA8
KEY(175) = 52BC2507F88A5B43 DATA(175) = 19A1813E26890530
KEY(176) = F804BA79BA236710 DATA(176) = 8FA60ED5D9539F9E
KEY(177) = E39162D079A47F8F DATA(177) = 2C3C0A6FBCA01B82
KEY(178) = 19629E808FFBC22F DATA(178) = A4865E2759311EA7
KEY(179) = 13CB5D97620279C2 DATA(179) = 00DA6E9C33F7EE88
KEY(180) = 8F02FB7308453157 DATA(180) = B21A824B157C6096
KEY(181) = 3702928F047567D0 DATA(181) = 46CAE83950D9A66D
KEY(182) = 01893410B52A1F3B DATA(182) = 3F520F31C1E89337
KEY(183) = 4613BF0D64B0077A DATA(183) = CECCE5FBB20D91B4
KEY(184) = 6DFB5849B0A8ECBC DATA(184) = CECE6EC0D1C4F636
KEY(185) = 0146CE320E46433B DATA(185) = 0ECC8C2E93652446
KEY(186) = 26C8E5C194CEBAF1 DATA(186) = 991D5C0EBA481E1A
KEY(187) = DF799234CB1FF1D5 DATA(187) = 9CA5E5CBA4E8A6B3
KEY(188) = B316B64A5D9B3D32 DATA(188) = A30B955D83E308B3
KEY(189) = E0BA0BD62CAEDFA1 DATA(189) = F51273B33D6BD2FE
KEY(190) = 979D6E671C255BC4 DATA(190) = 1055F2917A2711E3
KEY(191) = AD9870AEC854080E DATA(191) = 2C9D09D281636347
KEY(192) = E094A21CB6E06D46 DATA(192) = 688F251E2376AA24
KEY(193) = FD2A2C37AEC4FDE6 DATA(193) = F425F3785853FF6C
KEY(194) = 2083F10870F1CEF7 DATA(194) = 68169137EA09DB32
KEY(195) = 68C2E0A7A157E52A DATA(195) = 0E7B2ACE7E28D472
KEY(196) = 8F32F1FE8319E31C DATA(196) = 8E8A2F01C3EB2855
KEY(197) = 4A62EC75A75D31BA DATA(197) = F8F9143392D10F1F
KEY(198) = A4B6DC267370EA0B DATA(198) = FAE43BFD1A277297
KEY(199) = 2A708CBC9D43B64F DATA(199) = F52836C43A9D2EB3
KEY(200) = 34864A08DCCE07D6 DATA(200) = 728FF595A625C446
KEY(201) = 07A834FB498F61C4 DATA(201) = 2A6FA5EF886489E9
KEY(202) = E3ECC70101D60834 DATA(202) = 230F7857721131A2
KEY(203) = 1C928061168640F7 DATA(203) = 27E87FF540511F17
KEY(204) = E6E37CBFB3541A0B DATA(204) = FD673DB3D3B856AC
KEY(205) = 623BD39B0BE5D62C DATA(205) = 1D98E730BB33EE98
KEY(206) = 799D796BD00443AE DATA(206) = 7DFD7EABF5C3F24C
KEY(207) = 080DCD070446ADA4 DATA(207) = BC1B6040AF158C43
KEY(208) = 67BA6B58E9ECFBD3 DATA(208) = EB358ABAB2ECE312
KEY(209) = 3DEFCEADFEE9B07F DATA(209) = 367DCF2C7FC6C854
KEY(210) = CEF15B077AE3E0FB DATA(210) = 36811A2B5B9B71C8
KEY(211) = 0252C80E20347976 DATA(211) = 907F21B77D3C797B
KEY(212) = D6D9254F977529C4 DATA(212) = C3DFA06F077F8531
KEY(213) = 7A45E0F2A1DFFBA2 DATA(213) = C31E2A8641E85A6A
KEY(214) = 23B6A891BF4C54C2 DATA(214) = 5820ECC4D6A33176
KEY(215) = D9FBCBC791D66D73 DATA(215) = 620653D59C1E3B3A
KEY(216) = F27F3810491F0273 DATA(216) = 1512912307D43B7D
KEY(217) = 5DB68AEA3723FDCE DATA(217) = 7449877B128D6E68

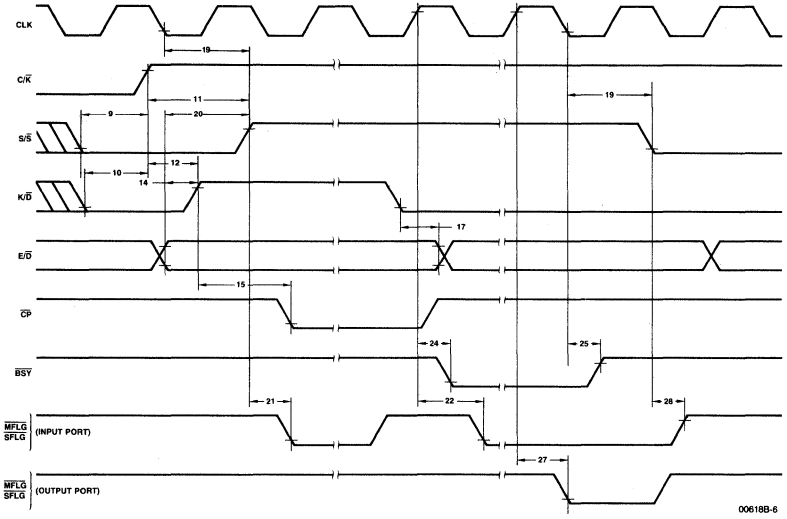
KEY(218) = 76C8F773D9E5FD9D	DATA(218) = 308DC9D55A207705
KEY(219) = 686DB57CCB79B5F8	DATA(219) = 19B32CBEBCE9223D
KEY(220) = 2A7543385EEF49B5	DATA(220) = 9374E66C4BED559D
KEY(221) = 9B6B4A0292FB6E3B	DATA(221) = CB18D8B36F0FA900
KEY(222) = 80750762E67613EA	DATA(222) = 57C32D3F732E5981
KEY(223) = 7954C8C19ED0C891	DATA(223) = 61B8FF52C66C1E5E
KEY(224) = 646EE02AE9AD4A75	DATA(224) = BCBA700F07F3666A
KEY(225) = 7F54293764073 78A	DATA(225) = A026398B32227D7F
KEY(226) = F491D93185EC4FFB	DATA(226) = 7E36C7184AAA8316
KEY(227) = 32BA1F8A0BCD9E92	DATA(227) = 095209BF4EEDE25B
KEY(228) = 2385A80885831AD5	DATA(228) = FC9CEF4EC9519AAD
KEY(229) = B929F7917516B540	DATA(229) = 87799D1264E5997D
KEY(230) = AEC4837AABAE19D6	DATA(230) = 86BE9818D8D73595
KEY(231) = DCFB8C1A20154916	DATA(231) = 72040AE5D007AD54
KEY(232) = 9E61D5383B2008EF	DATA(232) = 750EFEA52AD57666
KEY(233) = 9D89BAE376FBA7D3	DATA(233) = 80B27F2850C114C0
KEY(234) = E0236E1AA1EFD90D	DATA(234) = C5696D324621A59C
KEY(235) = 319B4334A229198A	DATA(235) = 9A69B18E9C79EA91
KEY(236) = FDFE7AF2D36E683E	DATA(236) = DFC23F2C37A23F42
KEY(237) = 07319D26F87F4FE5	DATA(237) = 06076B9E2FE26FB3
KEY(238) = F410F1B320583425	DATA(238) = ACEABD6A0F45ED8D
KEY(239) = 372392EC4A6BEFEF	DATA(239) = 765A045DBB8D7CA1
KEY(240) = 405DD33D94252A5D	DATA(240) = F4B9A0FD827BD835
KEY(241) = 73D0BC8979E59132	DATA(241) = 473A2ECD2361EBD4
KEY(242) = 2A986E91E9C16E61	DATA(242) = 94864ECC36512772
KEY(243) = 5ED973C794D0313D	DATA(243) = DFE254163E4A8A3B
KEY(244) = 52572F854C348CEF	DATA(244) = 020A4BF31DB1DC52
KEY(245) = 5BADC7C2854613EC	DATA(245) = 8728405B82A02D7E
KEY(246) = 253B1064F4EF9BB6	DATA(246) = 01DE63A81C31BE66
KEY(247) = A72AB6D63D922F91	DATA(247) = E329179E3DEAD31B
KEY(248) = CE9467CE1CE0F44F	DATA(248) = 478471F4702103D2
KEY(249) = AEDF019125CE3852	DATA(249) = D81896BC09F2DAD9
KEY(250) = D364D9D5587C3794	DATA(250) = DB01C717A6ABC0C8
KEY(251) = 209E8CD91C94AEC2	DATA(251) = 1D868151504AB4C5
KEY(252) = DFB3BCBAF4C852C2	DATA(252) = 1C2DC142E6010D7D
KEY(253) = C1B97A62F489D07F	DATA(253) = 255FC314E71796BB
KEY(254) = 851615073D1FDCCE	DATA(254) = AB8A3EF50853B151
KEY(255) = B552AB8025B552A7	DATA(255) = A1DFDA861DF8100F
KEY(256) = 94949B86B5E91580	DATA(256) = 3D08892CC1E63497
KEY(257) = D9E3CB1589FBA889	DATA(257) = 7CFE0497DB7530C4
KEY(258) = FD58D5A7586D32EC	DATA(258) = E726184715160E4D
KEY(259) = 2C491A4F6857B0C4	DATA(259) = 2783EE784B3DDA62
KEY(260) = 042A4C54F183E319	DATA(260) = 3D636C1CDDDFF904
KEY(261) = 1625E38F343B3B0B	DATA(261) = 9A1B2B8B49A5192D
KEY(262) = 942937D68F7C9813	DATA(262) = 02AE62E5DDA6523C
KEY(263) = 585D64E5A41634D0	DATA(263) = 4A1D0947EFFE29F6
KEY(264) = D9C49B1F021361A1	DATA(264) = DD30D58F355EF42C
KEY(265) = 016B8A61EAD0F220	DATA(265) = E304520CCA141EE6
KEY(266) = 4F75075EFB83629E	DATA(266) = 48E708E362CF23A6
KEY(267) = 918AA4E6CD85514A	DATA(267) = B048E10456F7E55D
KEY(268) = 45468AE6ABB00D02	DATA(268) = 1EB842800BC665EB
KEY(269) = 4FAE9EC16DDF5891	DATA(269) = 167C043457448E6C
KEY(270) = C26E6B43A792B385	DATA(270) = BE2470A9A3DAD69B
KEY(271) = F82397AB3473C2C4	DATA(271) = 6336E6EE66B00642
KEY(272) = EA94EF04169BA76E	DATA(272) = 77B4C9B7A8187F59

KEY(273) = 7A86C110B5CD23F4	DATA(273) = 62093068EBBAD9DC
KEY(274) = FE43FD7C80A43731	DATA(274) = B695948CB58FDB34
KEY(275) = CD7CBF0EA4C210CE	DATA(275) = 5746E10BE003B0C6
KEY(276) = 3EE60479D575086B	DATA(276) = BC351F7FDF599619
KEY(277) = 7C5204FEB0FB0710	DATA(277) = 5A4AAD9E0C705137
KEY(278) = 5E8F155BA8A2ABC1	DATA(278) = E532862BE2C93207
KEY(279) = BA9210E3E96186CB	DATA(279) = 9F7E8DF79E5ABBC6
KEY(280) = 67A1F47CC83B1CA1	DATA(280) = 6762FD1C198DD070
KEY(281) = 23624983DCFD85CB	DATA(281) = 8AC7FEB112D2C89B
KEY(282) = 49A2A8373D10C446	DATA(282) = E556340D2607D221
KEY(283) = 31CB9407F1C8A8E3	DATA(283) = 79F3121E1C57C0BF
KEY(284) = 37B35B163D158A73	DATA(284) = 0B68E7F2DC60937F
KEY(285) = 989DE69BCD73E6C7	DATA(285) = D0F7E382E3427329
KEY(286) = 9E3EE3372A138A97	DATA(286) = 5B8094D8A6EB8364
KEY(287) = 4F1007C145F4CBF1	DATA(287) = BA215A97B5A24FD3
KEY(288) = FE2957B56E2A57C7	DATA(288) = 6E07F51B761D848F
KEY(289) = DAB976B0EAADC7A1	DATA(289) = 10020DD698EFCB1D
KEY(290) = EC37EAF89B611C92	DATA(290) = D86CAEC881F8058D
KEY(291) = ADC76B688A1C9443	DATA(291) = F1FEA11421C3255B
KEY(292) = 1626167FE39BEA40	DATA(292) = 7D4C4B3DA933E9F7
KEY(293) = A183D90DCE9B8062	DATA(293) = 993E29D1570F656C
KEY(294) = 26BAD0AD864994AE	DATA(294) = 4DCFA7997190511A
KEY(295) = C7E985754A83FB1C	DATA(295) = 27A65E1556FAD8AD
KEY(296) = 1F519140FBEF8AF7	DATA(296) = 91798BBD3428F192
KEY(297) = 6E7086CBBCA19829	DATA(297) = 29BE686B78E19D13
KEY(298) = 155BEA157604074F	DATA(298) = A375D2077A40E52C
KEY(299) = E99E5897F149C485	DATA(299) = E165E74B60E9D0F1
KEY(300) = E9D5A8DF689B9D0D	DATA(300) = E57B95BC18A6A114
KEY(301) = DFF1084034794945	DATA(301) = 0E92A3B1E30D1793
KEY(302) = BABAB06B2F2986E3	DATA(302) = 3DBF573A415650EA
KEY(303) = 455BBF641CFD4A5D	DATA(303) = 8B85F05C74087F1D
KEY(304) = AD541F3232F4079B	DATA(304) = C3F1B903EAE22BE4
KEY(305) = 9DFE29E51F430E83	DATA(305) = EBF5D593E0DEE457
KEY(306) = 73EF2A856DB5BA5D	DATA(306) = 08EFCCDD78EBC1D6
KEY(307) = DFA423E3B91CA4A4	DATA(307) = 22EE7DA4A47D14C4
KEY(308) = 1658C43BF10B8A94	DATA(308) = F19417EEF5577301
KEY(309) = 7938D076158673CB	DATA(309) = A3E3B4D7FCBCCA7A
KEY(310) = D6CBDA5EADBFB025	DATA(310) = D8C6A163C3F8632C
KEY(311) = CDF43B51ADD043D3	DATA(311) = 88D60BA668F84A9D
KEY(312) = 7A02D01CE6132A58	DATA(312) = 7DE0D6024410F097
KEY(313) = E37A98136D08BC38	DATA(313) = 9B02F91CB56FC6EF
KEY(314) = E62634F8D58992A7	DATA(314) = D856700C09777605
KEY(315) = DCE6EC32DC31ADA8	DATA(315) = AADF07DC34AEA3F2
KEY(316) = 1C80CB681A26CD6B	DATA(316) = 519F2143BD45325D
KEY(317) = 29D67FC798856804	DATA(317) = 284A2756F05D6EBE
KEY(318) = FE7A1A3298A13DEA	DATA(318) = C4E646E854335698
KEY(319) = 9768C71ACBC72FA8	DATA(319) = 35C4C390F46BCA9A
KEY(320) = A137A8F770FD76A8	DATA(320) = EE1418955988B4BD
KEY(321) = 401A1551CD854383	DATA(321) = F297A55B06BDEC57
KEY(322) = 865E7CEF2ADC6BBC	DATA(322) = BAB2CAAB7F0FD816
KEY(323) = 8FB05D322602F1B9	DATA(323) = E67B466382 0B3D8A
KEY(324) = 575E76A11CAD254C	DATA(324) = FCF121962D2EEE6C
KEY(325) = C22FEA1ABC85B60D	DATA(325) = 4A7951D248A8BCD1
KEY(326) = B6E5B5FD80460746	DATA(326) = E51B08274D8A66A8
KEY(327) = 6E157AB62F08166E	DATA(327) = 2A5463F7DE58FB3B

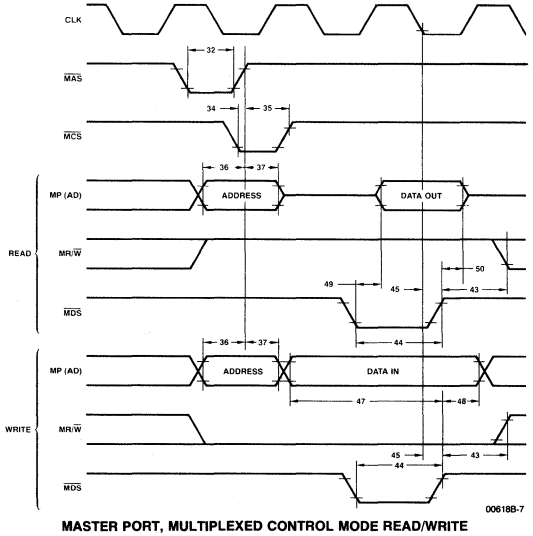
KEY(328) = 6152A743B5CE7337	DATA(328) = CA2740CEAB9FD243
KEY(329) = 38971FD3645DEA97	DATA(329) = 3E12F40F53FD97A8
KEY(330) = 5E971A5D75B5D6FE	DATA(330) = 3314157B27FE886C
KEY(331) = 58F4511AA1DC529D	DATA(331) = C45DA83F3867FCA2
KEY(332) = 9B455DDCC1458C40	DATA(332) = E009E0BE4BF045A1
KEY(333) = BF83A2642A9EBABC	DATA(333) = BAD9B8A6D2646632
KEY(334) = C883F2A7A8DAB604	DATA(334) = 7379D06752FCD161
KEY(335) = 1CD34C626BC7C2AB	DATA(335) = 223E612E8F7CF09F
KEY(336) = 75B51C3BE5C85B7C	DATA(336) = 0C166A5CB8C21C0C
KEY(337) = 4A574AE6CB837380	DATA(337) = FED19F785A5B46A6
KEY(338) = 46BCFEEAFB54A480	DATA(338) = A8FF821640BA1A27
KEY(339) = 6D196473F2018A97	DATA(339) = 2614631CB6301859
KEY(340) = 91C78C807915FE5B	DATA(340) = B476D87AC727F69B
KEY(341) = 832698F876D9A167	DATA(341) = FCC0D2C25D746947
KEY(342) = 38D96B13B9CDBAA2	DATA(342) = D6FAF1F46C33C51C
KEY(343) = 4C86625289074C1C	DATA(343) = 49C1169172F9B9C8
KEY(344) = 62A1FB2A08160DBC	DATA(344) = B2F1A13FF25BBA05
KEY(345) = DC9ECBAE10466779	DATA(345) = 3EDDFB3E0FF3D34E
KEY(346) = 92F8B573B0A825BF	DATA(346) = 78EA7688E6C4D128
KEY(347) = 73E579737A01E580	DATA(347) = 605EF6CAD8EE2C06
KEY(348) = 01BA617608921345	DATA(348) = 95D2039A13D2E688
KEY(349) = 8A574C5D8A197067	DATA(349) = 029FE7ADA291A861
KEY(350) = FE400BB3AB5EBFD5	DATA(350) = 6B6B7FA804AB62C6
KEY(351) = 64EF25FD58A15185	DATA(351) = CBD52B393D04A27A
KEY(352) = 6D2F4FBCE9805440	DATA(352) = 3D79EF2EBB226654
KEY(353) = DFB03BAE04295EF4	DATA(353) = 5D86923988CDDFD6
KEY(354) = A46707DFF285B3FD	DATA(354) = 146FAECE771F0EEE
KEY(355) = 16EAE3686B9EA238	DATA(355) = 4B6F29540693E99D
KEY(356) = 4CBFBC91A758EA75	DATA(356) = 8782B09EF09767D3
KEY(357) = 7532BCCB3E0E9E13	DATA(357) = D978892B04C803BA
KEY(358) = 67B5CECE40808520	DATA(358) = D783A15F95F60CE4
KEY(359) = 1A5454E6B94A6425	DATA(359) = 47B6D56768E56CC8
KEY(360) = 4F91CE625DA10792	DATA(360) = 2B9A3E9645E4AF1E
KEY(361) = BABA674C807C3B3E	DATA(361) = 050E700748723FEA
KEY(362) = 648AE5B08F860804	DATA(362) = E94825AA1605A1AC
KEY(363) = 0283C4DF1A57BAA7	DATA(363) = AB71BA670509E0BC
KEY(364) = D943757CFE1AA445	DATA(364) = CE8AA4C3D363BFF1
KEY(365) = C1AB58E01AEF7089	DATA(365) = 2B9E8E4E8671D468
KEY(366) = 7338D01A9D0D7562	DATA(366) = 7F010D3D11C13A08
KEY(367) = 83C8041CE020A8E5	DATA(367) = E98CBE8D367D36F7
KEY(368) = 9115F889F2BFE0BA	DATA(368) = E36DD26D12D27FCC
KEY(369) = 571C01436E68CD29	DATA(369) = 62EDA377B9DA2589
KEY(370) = 43437C31970E1AF1	DATA(370) = 90E5CFC245A878DE
KEY(371) = 92C1AE4326314A6E	DATA(371) = 294FF405C824665B
KEY(372) = F4894629A40DBCFB	DATA(372) = 9F4EB55EC8F3F30A
KEY(373) = A4C8FDF298F4382A	DATA(373) = A44BC9D454418FAE
KEY(374) = 6E5231EA7CA20BC2	DATA(374) = 0A93CE7BCFBC8455
KEY(375) = 757613E99BDC5BAD	DATA(375) = B7523D8B2FEF331A
KEY(376) = 080D75944A86A876	DATA(376) = 02F4403D86D5ADD1
KEY(377) = 4CBFC42A6B026EA7	DATA(377) = 3F11954EC2848277
KEY(378) = 5143AE073D8ADCB5	DATA(378) = A01A1A977EAAA109
KEY(379) = 86E00E1543108A49	DATA(379) = 0881CEACA47E7661
KEY(380) = E53ED9C2B631C70B	DATA(380) = D98449A36BC04DB4
KEY(381) = 7FD61F3B1070A1D5	DATA(381) = 7AFA04C6C0537859
KEY(382) = 2F98B5924364978A	DATA(382) = BBA514DD4D189133

KEY(383) = 2A6DD3F7BAB69186	DATA(383) = 50F0A4849AE82024
KEY(384) = 4954BAF85489AB8A	DATA(384) = 449016A145CA83BE
KEY(385) = CE980EAD1513947A	DATA(385) = 9E04B063661AD974
KEY(386) = 6258FB15F83D9868	DATA(386) = 1030770076332729
KEY(387) = B5A86D2585F8492A	DATA(387) = 22161BA03E98801D
KEY(388) = 8AFB299EADB6526E	DATA(388) = 8A3BC96A9163DA27
KEY(389) = B9494683F1518FB5	DATA(389) = 792E72347FA526CC
KEY(390) = 4A265ECB041CD383	DATA(390) = 7AEFB5211B40A208
KEY(391) = 83EF834998C49D6B	DATA(391) = DBF80F308DE9B048
KEY(392) = A1F213989B76E976	DATA(392) = 6A082DFCCDDFEEE3
KEY(393) = FB38D60E5401CBA4	DATA(393) = 0CE92FD8EC40EF4D
KEY(394) = 6468BAFDEAB5E989	DATA(394) = 21CBA8759C1CCA05
KEY(395) = D0CB8538F49D9E9D	DATA(395) = 2138A4F6C106E236
KEY(396) = C74989A831D6B69B	DATA(396) = 98C45E19F6D6FF31
KEY(397) = E6F8CEC8D0C7F12F	DATA(397) = 5CDB0A695686139E
KEY(398) = B361FE800D623B3E	DATA(398) = 06E0D9924B7060DD
KEY(399) = 045B6758A89B5732	DATA(399) = 648920D62CC02BFF
KEY(400) = FB8929CE83A2737C	DATA(400) = 404CB50060AE6C04

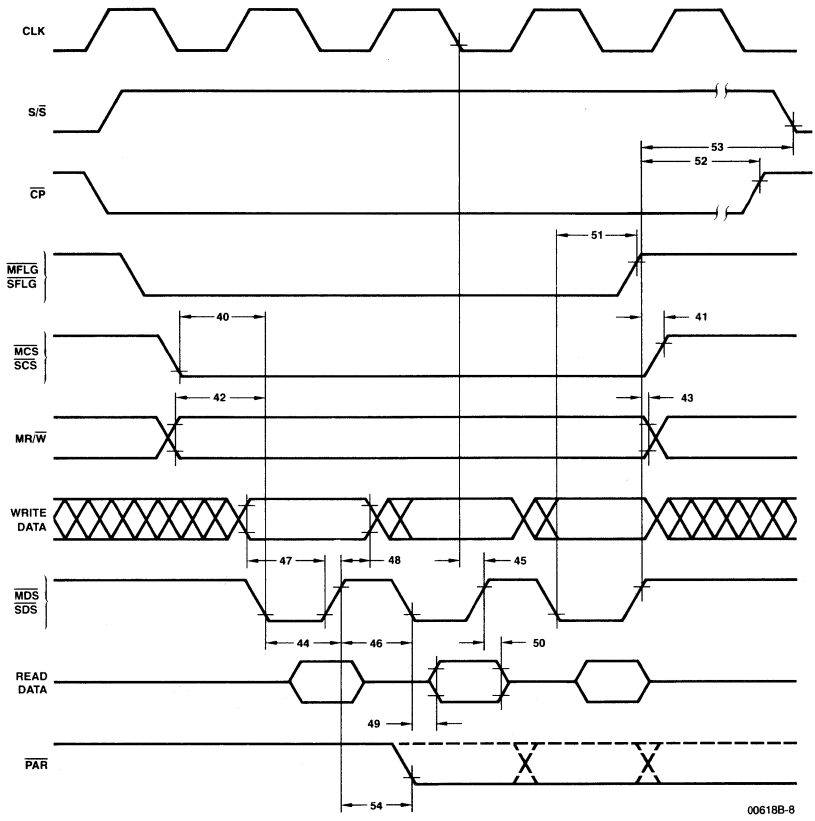
APPENDIX E. Timing Diagrams



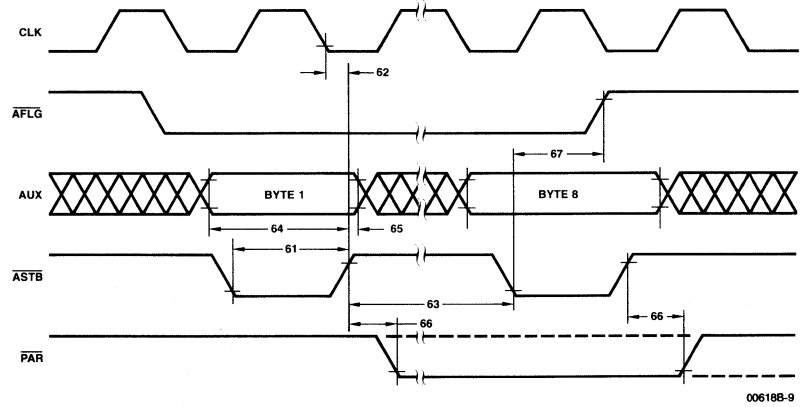
CONTROL AND STATUS SIGNALS (DIRECT CONTROL MODE)



MASTER PORT, MULTIPLEXED CONTROL MODE READ/WRITE



MASTER (SLAVE) PORT READ/WRITE



AUXILIARY-PORT KEY ENTRY

APPENDIX F. Literature

(1) Federal Information Processing Standards Publication 81
DES MODES OF OPERATION

Standards Information Office
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

(2) NBS Special Publication 500-20
VALIDATING THE CORRECTNESS OF HARDWARE IMPLEMENTATIONS OF THE NBS DATA ENCRYPTION STANDARD

National Bureau of Standards
U. S. Department of Commerce
Washington, D.C. 20234

(3) Federal Information Processing Standards Publication 46
DATA ENCRYPTION STANDARD

National Bureau of Standards

(4) Product Specifications: 8086/8086-1/8086-2
AmZ8068
Am9518
Am9568
AMD 20-Pin PAL Family

Advanced Micro Devices, Inc.
901 Thompson Place
Sunnyvale, CA 94086

(5) iSBX BUS SPECIFICATION
Manual Order Number: 142686-002

INTEL Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

ADVANCED MICRO DEVICES, INC.
901 Thompson Place
P.O. Box 3453
Sunnyvale,
California 94088
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306
TOLL FREE
(800) 538-8450

04862A

IM-CPI-5M-9/84-1