Am95C85

Content Addressable Data Manager

FINAL

DISTINCTIVE CHARACTERISTICS

- High-performance sorting, searching, and updating
- 1K byte software-reconfigurable memory array
- Programmable record size
- Cascadable up to 256 devices
- Content-addressable operation, independent of record size
- Intelligent peripheral with sixteen powerful instructions
- Stack mode allows inserting of data without resorting
- Up to 16-MHz operation
- CMOS technology
- 44 lead PLCC

GENERAL DESCRIPTION

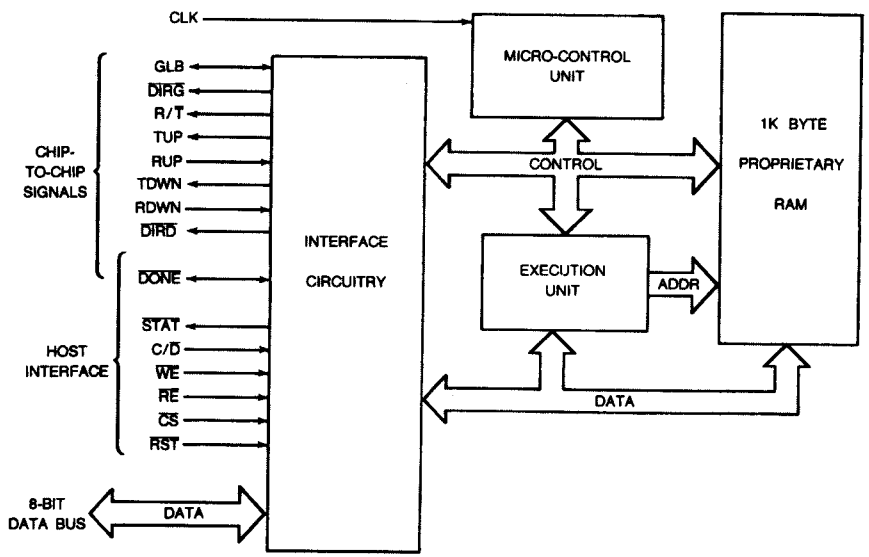
The Am95C85 Content Addressable Data Manager (CADM) is an intelligent CMOS peripheral device designed to enhance the performance of applications involving sorting, searching, and insertion or deletion. Orders of magnitude performance improvement can be seen when compared to the implementation using software algorithms.

The CADM uses an on-chip proprietary 1K byte memory for data manipulation. This specially designed memory can be easily reconfigured to meet different application requirements. The data stored in the CADM are collated into records that consist of a key field and a pointer field. The length of these two fields are software programmable. The sorting and searching of records are based on the values of the key fields. A mask register is also provided to selectively mask out unwanted bits in the key field for comparison. For applications that require large storage area for data manipulation, the CADM can be easily cascaded up to 256 devices.

Content-addressable operation allows the host to retrieve data without having to do extensive searching. Address generation for memory access is done internally, relieving the host from the burden of physical address calculation. Stack-mode operation allows the user to delete records simply by popping the records out of memory, and to insert records by pushing the records into the memory.

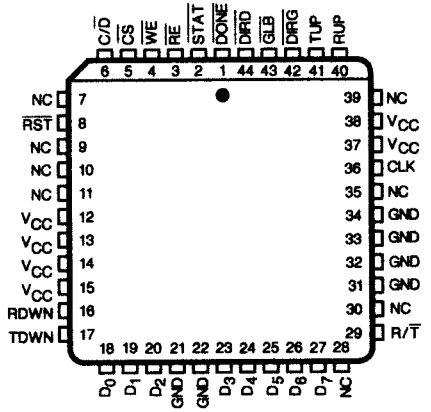
By providing content-addressable searching, automatic sorting, programmable record length, and address-independent operation, the CADM allows the host to off-load repetitive, time-consuming data manipulation. For applications that require substantial sorting, searching, and updating operations, the CADM offers significant improvement in overall performance.

BLOCK DIAGRAM



Publication # 06956 Rev. C Amendment /0
Issue Date: August 1989

CONNECTION DIAGRAM
Top View

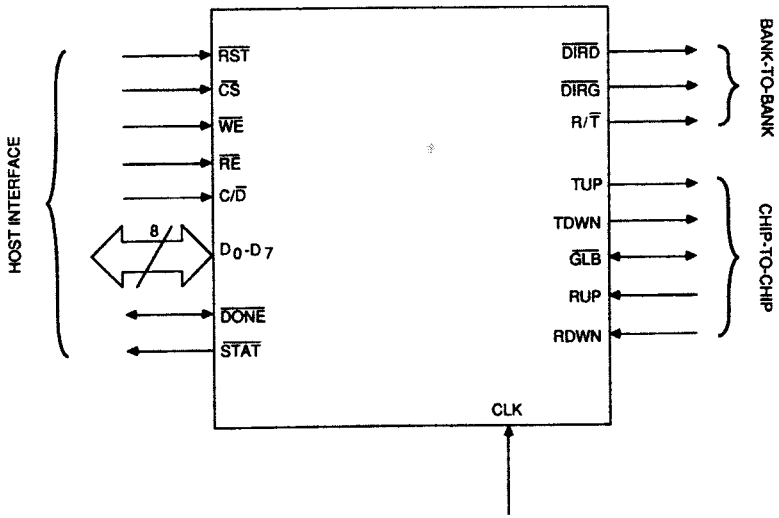


NC = No Connection

CD010442

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

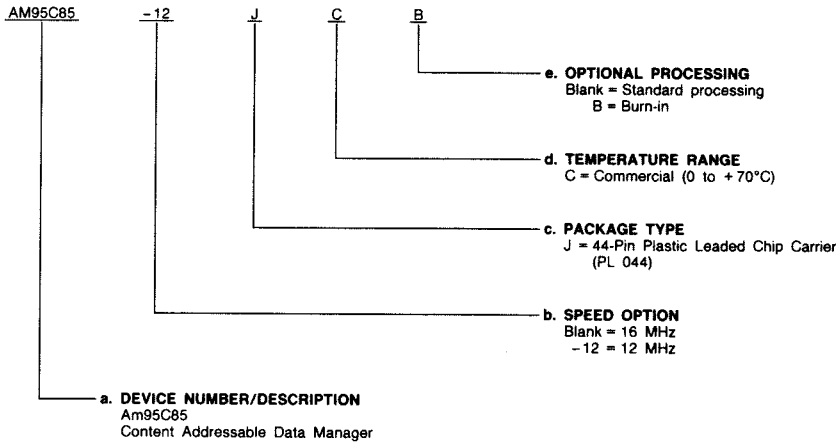


LS002891

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM95C85	JC
AM95C85-12	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Host Interface

C/$\overline{D}$ Command/$\overline{Data}$ (Input)
A HIGH on this input allows the command register to be loaded with the information on the data bus. A LOW on this input allows the data to be read from, or written into, the internal RAM.

$\overline{CS}$ $\overline{Chip\ Select}$ (Input; Active LOW)
The $\overline{CS}$ input enables the host CPU to perform read or write operations with the Am95C85 devices. The read and write inputs are ignored when $\overline{CS}$ is HIGH.

$D_0 - D_7$ Data Bus (Input/Output; Three State)
The eight bidirectional data pins are used for information exchanges between the Am95C85 (CADM) and the host processor, and between CADM parts themselves. A HIGH on a data line corresponds to a Logic "1," and a LOW corresponds to a Logic "0." These lines act as inputs when $\overline{WE}$ and $\overline{CS}$ are active, and as outputs when $\overline{RE}$ and $\overline{CS}$ are active. D_0 is the least significant bit and D_7 the most significant bit.

$\overline{DONE}$ $\overline{Done}$ (Input/Output; Active LOW, Three State)
This signal indicates the termination of an operation, and is precharged to HIGH at the beginning of a new command, data writes, or data reads. A LOW on this output indicates the device is ready for the next command or data transfer.

$\overline{RE}$ $\overline{Read\ Enable}$ (Input; Active LOW)
The $\overline{RE}$ input, together with $\overline{CS}$ and C/$\overline{D}$ inputs, are used to control data transfer from the Am95C85 to the host. The Am95C85 will put the data onto the data bus when $\overline{RE}$, $\overline{CS}$, and C/$\overline{D}$ inputs are LOW.

$\overline{RST}$ $\overline{Reset}$ (Input; Active LOW)
A LOW on this input will reset the Am95C85. Any command under execution is terminated.

$\overline{STAT}$ $\overline{Status}$ (Output; Active LOW, Three State)
When LOW, the $\overline{STAT}$ output indicates that an exception condition has occurred following the execution of an instruction or data transfer. This pin is precharged to HIGH at the beginning of a new command, or when a write or read is initiated.

$\overline{WE}$ $\overline{Write\ Enable}$ (Input; Active LOW)
The simultaneous occurrence of $\overline{WE}$ and $\overline{CS}$ indicates that information from the data bus is to be transferred to the Am95C85. The C/$\overline{D}$ input determines whether the data will be loaded into the command register or internal RAM.

Chip-to-Chip Communication

These pins are used in chip-to-chip communications in multiple Am95C85 memory configurations. They do not affect the system interface.

$\overline{GLB}$ $\overline{Global}$ (Input/Output; Active LOW, Three State)
This signal is used for part-to-part synchronization during instruction execution. All CADM devices in the same bank should have this pin connected together and pulled up through a resistor to the power supply. This pin is precharged to HIGH at the beginning of a new command, or when a Write or Read is initiated.

RDWN Receive from Downward (Input; Active HIGH)
This pin should be connected to TUP of the next lower order CADM in cascade. The last chip in the daisy chain should have this pin pulled up to the power supply through a resistor.

RUP Receive from Upward (Input; Active HIGH)
This pin should be connected to TDWN of the next higher order CADM in cascade. The first chip in the daisy chain should have this pin pulled up to the power supply through a resistor.

TDWN Transmit Downward (Output; Active HIGH)
This signal is issued by the higher order CADM to the next lower order CADM, in cascade, to synchronize the chip-to-chip data transfer. It should be connected to RUP of the next lower order CADM.

TUP Transmit Upward (Output; Active HIGH)
This handshaking signal is issued by the lower order CADM to the next higher order CADM, in cascade, during chip-to-chip data transfer. It should be connected to RDWN of the next higher order CADM.

Bank-to-Bank Control

Bank-to-bank communication is needed when multiple banks of Am95C85 devices are used in a system. The CADM array can be grouped into multiple banks and separated by buffers. The following signals are used to control the direction of buffer signals that separate the banks. They can be left unconnected if only one bank is used.

$\overline{DIRD}$ $\overline{Direction\ of\ Done\ Signal}$ (Output; Active LOW, Three State)
This signal is used to control the direction of the circuit that is buffering the $\overline{DONE}$ signal. All CADMs in the same bank should have this pin connected together and pulled up through a resistor to the power supply. When driven LOW, this signal indicates that the CADM is driving the $\overline{DONE}$ signal. This pin is precharged to HIGH at the beginning of a new command, or when a Write or Read is initiated.

$\overline{DIRG}$ $\overline{Direction\ of\ Global\ Signal}$ (Output; Active LOW, Three State)
This signal is used to control the direction of the circuit that is buffering the $\overline{GLB}$ signal. All CADMs in the same bank should have this pin connected together and pulled up through a resistor to the power supply. When driven LOW, this signal indicates that the CADM is driving the $\overline{GLB}$ signal. This pin is precharged to HIGH at the beginning of a new command, or when a Write or Read is initiated.

R/$\overline{T}$ Receive/$\overline{Transmit}$ (Output)
This output is driven LOW when the Am95C85 is driving the data bus. It should be used to control the direction of buffers which isolate the data bus from specific Am95C85 banks.

FUNCTIONAL DESCRIPTION

Introduction

The Am95C85 Content Addressable Data Manager (CADM) is an intelligent peripheral device intended to relieve the host CPU of many of the time-consuming tasks associated with data-list manipulation. Sorting and finding data are tasks implemented by both applications software and operating systems. By providing these functions in hardware, which were previously the responsibility of software, execution time is reduced. This performance improvement can be 100 to 500 times, depending upon the application.

The 44-pin Am95C85 contains 1K byte of RAM whose organization is programmable. It contains a micro-engine, registers, pointers, and an instruction decoder. Most of these functions are transparent to the user.

The Programmer's View

Hardware

The Am95C85 CADM interacts with the host system through the use of a command port, data port, and two status pins called $\overline{STAT}$ and $\overline{DONE}$. Both the command and data ports are accessed through the single 8-bit data bus. The two ports are differentiated by the use of a Command/$\overline{Data}$ pin (C/$\overline{D}$). The familiar signals, $\overline{RE}$, $\overline{WE}$, and $\overline{CS}$ are used to write and read data or commands.

Data Array

The CADM contains 1K byte of internal RAM. It consists of a mask area, a record area, an unused area, and an input buffer area as shown in Figure 1. The mask area exists only if an SMB command is issued. The length of the mask area is the same length as the key, as specified by the KPL command. The data stored in this area is used to select the desired bits in the key field for comparison during the sorting and searching process. Those mask bits with "0" will cause the associated bits in the key field to be ignored during the comparison. The record space stores data as records in the CADM. This area starts from address "K" if the masking option is chosen, or "0" if the masking option is not chosen. It ends at the last address as programmed by the KPL command. The length of this area should be a multiple of (K + P) bytes. The last (K + P) bytes are designated as input buffer area. They are reserved to temporarily store the incoming record. The remaining area between the record area and the input buffer is unused area and is not accessible by the user. This area should be kept as small as possible to optimize the performance of the CADM.

The internal RAM structure, a patented AMD design, is unique in that the record width is controlled by the CPU, using the KPL command. Each record is comprised of two fields, referred to as a key field (K) and a pointer field (P). The KPL command sets the width of these two fields, then partitions the entire array into records, each with a length of K + P bytes. Figure 2 shows the logical model of the CADM data array. The length of K may vary from 1 to 255 bytes, and P may be set between 0 and 255 bytes. The variable record width provides significant flexibility that is very useful for general-purpose data manipulation. It allows complex operations, such as sort and search, to be performed on virtually any type of data. For example, the Am95C85 devices can be used to search a file-allocation table for a particular file address. It may then be reprogrammed to manipulate a disk-directory table. The Am95C85 can sort a database index file and is versatile enough to handle each of the tasks described above, even though each has a differernt record width.

The maximum number of records stored in each CADM depends on the record width (K + P) and the value of Last Address (LA). To efficiently use the memory space of the CADM, the LA should be programmed with the following value:

if mask bytes are used,

$$LA = \{INT[(1024 - 2*K - P)/(K + P)]\} * (K + P) + K - 1$$

if mask bytes are not used,

$$LA = \{INT[(1024 - K - P)/(K + P)]\} * (K + P) - 1$$

The Am95C85 array can be easily expanded if the application requires more record storage; up to 256 CADMs can be cascaded to meet the application requirements. The addition of hardware is transparent to software. The programmer still sees one command port, one data port, one $\overline{STAT}$ pin and one $\overline{DONE}$ pin. The only difference is that there is more record space for data manipulation. The number of CADM devices in cascade can be easily determined by reading the data port after a hardware or software reset.

Addressing Flexibility

To take advantage of the flexibility of the unique memory array, the Am95C85 allows several different addressing modes:

1) Auto-Increment Access
2) Stack Access
3) Indirect Random Access
4) Content-Addressable Access

The programmer will first issue a command that either directly, or by implication, places the Am95C85 CADM in a particular addressing mode. For example, the command AIM allows the host to read or write the currently addressed location, while subsequent reads and writes will be to the next byte (i.e., the CADM auto-increments the address pointer after each data access). Alternatively, STK sets the Stack-Access mode, which means that any subsequent data access physically moves all data below the current location for a read or write. A data read pops the byte at the current location, and moves all the data below up. A data write pushes a byte on the array at the address pointer moving all the data below down. The Stack-Access mode allows for immediate insertion or deletion of records (in previously sorted data), without the need for re-sorting.

The pointer into the memory array, the address pointer, is maintained by the Am95C85, although the programmer can load the address pointer through the use of the LAL (Load Address Long) and LAS (Load Address Short) commands.

The Find (FND) instruction implies a Content-Addressable Access mode. The description of the FND command is "set the address pointer to the key whose value is equal to the following bytes. If not present, point to the next higher value key." Following this instruction, the Am95C85 may be read to acquire the key plus pointer that was found. Since the FND instruction relies on the Am95C85 CADM data being in sorted order, the next section describes how a sort can be accomplished.

If more records matching a particular key value are to be located, additional FND commands without a key following the command can be issued. In this case, the value of the key contained in the input buffer space from the previous FND is used. The Address Pointer is incremented and the key comparisons are performed. This continues with each subsequent FND. To terminate this mode of operation, for instance to allow a new record to be sought, a command other than FND or RRB should be issued. The CADMs will then expect a subsequent FND command to be followed by a new key for which to search.

Host-Independent Sorting

Sorting may be accomplished on data which is in the form of a relational database index file. The programmer sets the length of the key and pointer fields by the KPL command and sets up K bytes of mask if the masking option is used. The data list may be loaded into the Am95C85 devices via DMA or slower programmed I/O. Two methods of sorting are possible:

1) Load data by DMA or I/O and then issue a Sort-Off-Line (SOF) command. This method loads all the data first and then performs the sort. The CPU can be performing other functions during the SOF execution. DMA completion must be detected by software before the SOF command is issued. $\overline{DONE}$ must be detected after the SOF command to signal that the sort has been completed.

2) Sort-On-Line (SON) command, followed by I/O or DMA load, allows each record to be placed in sorted order as it is loaded. If DMA is utilized, the CPU is free toperform other non-CADM tasks during the entire operation. DMA complete, followed by $\overline{DONE}$, defines the end of the sort.

The $\overline{DONE}$ pin signals the acceptance of each byte of data and indicates the device is ready for the next byte. It also signals the completion of the active sort for the SOF command. In the case of SON, after the last byte of each record is received, $\overline{DONE}$ is asserted after the record is merged with existing records. After the last record is sent to the CADM array, the final $\overline{DONE}$ signal represents the end of Sort On Line.

The Hardware Designer's View

Reset

The CADM will go into the reset cycle after the hardware reset is asserted or a software-reset command is issued. Each device in an array will number itself and determine its chip address. The first device with RUP tied to HIGH assumes it has a chip address of 0, the next chip assumes an address of 1, and so on, until the last device with RDWN tied to HIGH numbers itself. Completion of reset is signaled by $\overline{DONE}$ going LOW. After reset, the address pointer is set to the first byte location in the last chip. The key length, K, is set to 1; the pointer length, P, is set to 0, and the last address is set to 1023. Masking is disabled. A hardware reset is required after power-up to bring the internal logic into a known state.

System Interface

All system interface signals are designed to be standard TTL compatible.

The system-control signals, $\overline{RE}$, $\overline{WE}$, $\overline{CS}$, and C/$\overline{D}$ are used to control the interface between the host and the CADM array. The command port access, with $\overline{CS}$ = "LOW" and C/$\overline{D}$ = "HIGH," is used to send commands to the device and is write-only. The data port access, with $\overline{CS}$ = "LOW" and C/$\overline{D}$ = "LOW," is used to transfer data between the host and the CADM array when reading and writing. These control signals should be connected to all the CADM devices in cascade.

The CADM data bus is used for host interface and chip-to-chip data transfer. Because of this, the CADM should not be directly connected to the host data bus. A transceiver is required to isolate the CADM data bus from the host data bus to avoid possible contention.

Two pins indicate the status of the Am95C85. $\overline{DONE}$ is used to indicate the completion of a command execution or data transfer. $\overline{STAT}$ going active indicates an exception condition following the execution of command or data transfer. The host should not drive the CADM data bus when $\overline{DONE}$ is inactive; otherwise, an unexpected outcome may occur. $\overline{DONE}$ may stay inactive forever if an invalid command sequence is issued. In this case, a reset is required to bring $\overline{DONE}$ back to LOW. If there is more than one CADM in cascade, the $\overline{DONE}$ pin from each CADM should be connected together and pulled up through a resistor to the power supply. Similarly, the $\overline{STAT}$ pin from each CADM should be connected together and also tied to the power supply through a pull-up resistor.

The CLK signal should reside between 1 MHz and its maximum rating.

Chip-To-Chip Communications

During the execution of some commands, it may be necessary to transfer data from one chip to another. These signals, TUP, RUP, TDWN, RDWN, and $\overline{GLB}$ are used to perform handshaking between the devices involved in the transfer. RDWN should be connected to the TUP of the next lower order chip, and TDWN should be connected to the RUP of the next lower order chip. The first device should have RUP pulled HIGH through a resistor to the power supply as should RDWN of the last device. Figure 3 shows the signal connections for cascading multiple devices.

Bank-To-Bank Control

As the number of CADMs used in the system increases, the capacitive load seen by each CADM device will increase. Depending on the system environment, up to 16 devices may be cascaded. If the effective load exceeds the specified test load, the designer will have two choices:

1) Reduce the clock frequency to the CADM array.

2) Insert a buffer circuit between banks of CADMs to increase driving capabilities.

If Option 1 is chosen, $\overline{DIRD}$, $\overline{DIRG}$ and $\overline{T}$/R are not used and can be left unconnected. If Option 2 is chosen, $\overline{DIRD}$, $\overline{DIRG}$ and $\overline{T}$/R signals are used to control the direction of buffering circuitry between banks. Even if the designer chooses Option 2, the clock frequency still has to be slowed down from its maximum rating because of buffer delay. The designer must decide which option is best suited to the system.

Command Summary

There are 16 commands to control the operation of the CADM. These commands are used to initialize the CADM, to control the internal pointers, to load the data, and perform sorting and searching. A command is loaded into the command register by writing an operation code into the command port. The command port is used to load the operation code only. For commands that require parameters following the command operation code, the parameters should be loaded through the data port. Commands requiring literal data are: LAS, FND, KPL, SMB, SON, LUD, and LAL. Table 1 summarizes the operation code, mnemonic, and functional description for each command.

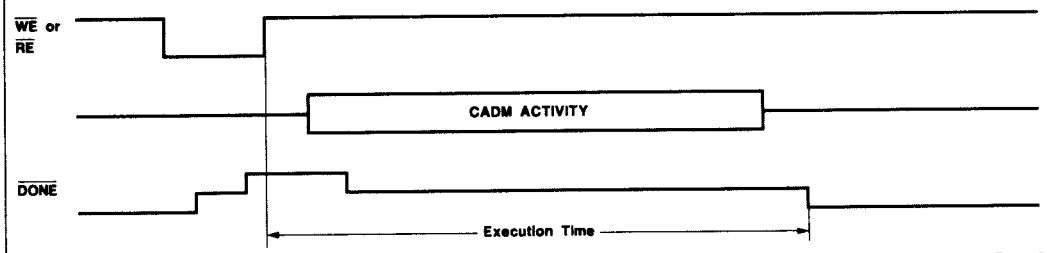
TABLE 1. COMMAND DESCRIPTIONS

OpCode	Mnemonic	Operands	Functional Description
00H	RST		Software reset command.
01H	LAS	Addr (LSB), Addr (MSB)	Load Address Short. Load the following two bytes of data at the address pointer of the currently active device.
02H	DEC		Decrement the address pointer by one.
03H	FND	Key (MSB), ... , Key (LSB) (Note 1)	Find the key specified following this command. Asserted $\overline{STAT}$ LOW if the key is not found.
04H	NXT		Set the address pointer to the first byte of the next record.
05H	RRB		Restore the address pointer to the first byte of the current record.
06H	AIM		Set Auto-Increment Mode. Address pointer is incremented by one after each data read/write.
07H	STK		Set Stack Mode. In Stack-Access mode, a read will pop data out of the data array at the address pointer and a write will push data into data array at the address pointer. The address pointer remains unchanged.
08H	KPL	K, P, LA (LSB), LA (MSB)	Load length of key and pointer fields and set the last address pointer.
09H	SMB	Mask (MSB), ... , Mask (LSB) (Note 1)	Set Mask Byte. The following K bytes of data will be used as mask during sorting and searching.
0AH	SON	Data (MSB), ... , Data (LSB) (Note 2)	Sort On Line. The CADM will insert the record into the data array in sorted order after the last byte of the record is loaded.
0BH	LUD	Data (MSB), ... , Data (LSB), ... (Note 3)	Load Unsorted Data. Data loaded following this command will be placed in the locations after existing meaningful data, if there is any.
0CH	SOF		Sort Off Line. Sort the existing data in the CADM in ascending order.
0DH	LAL	Addr (LSB), Addr (MSB), Chip Addr	Load Address Long. Load the following two bytes at the address pointer of the chip whose number is specified by the third byte.
0EH	PRE		Set the address pointer to the first byte of the previous record.
0FH	GSF		Get Status Full. Asserted $\overline{STAT}$ LOW if the CADM record space is full.

Notes: 1. Requires K Bytes following Opcode.
2. Requires integer multiples of (K + P) Bytes. Execution begins after each (K + P) Bytes are written.
3. Requires integer multiples of (K + P) Bytes.

Command Execution Time

The execution time of each command is expressed in clock cycles per byte of transfer. The execution times are measured from $\overline{WE}$ or $\overline{RE}$ to $\overline{DONE}$ as shown below.



WF024170

Command	Clock Cycles per Byte	Conditions
LUD	6	Command
	6	Per byte within a chip
	7	If crosses chip boundary
SMB	8	Command: first occurrence
	7	Command: all other occurrences
	6	Per mask byte for first k-1 bytes
	8	For last byte
SON	8	Command
	6	For first K + P – 1 bytes
	16 + S	For last byte (where S = binary search time; see FND performance equation)
AIM (Read/Write in AIM mode)	6	Command
	6	If on same chip
	9	If crosses chip boundary
KPL	9	Command; (+ 1 if only one chip in system)
	6	K: (+ 1 if user erroneously sets K = 0)
	7 + 2 * (9 – x)	P: where x = number of lower order zeros in K + P (binary)
	5	LA: (lsb)
	7	LA: (msb)
SOF	(See Sort-Off-Line performance equation)	
RST	8 + 4 * N	Where N = number of chips in system (Note: this applies to hardware and software RESETs).
LAL	4	Command
	5	lsb
	4	msb
	7	chip
LAS	4	Command
	5	lsb
	7	msb
DEC	8	If on same chip
	10	If crosses chip boundary
PRE	10	If on same chip
	13	If crosses chip boundary
RRB	7	Command
GSF	6	Command
FND	(See FIND performance equation)	
NXT	11	If on same chip
	14	If crosses chip boundary
STK	6	Command
–Push– (data write in Stack mode)	8	If on same chip. Add one clock cycle for every chip boundary crossing.
–Pop– (data read in Stack mode)	14	If on same chip. Add one clock cycle for every chip boundary crossing.

2

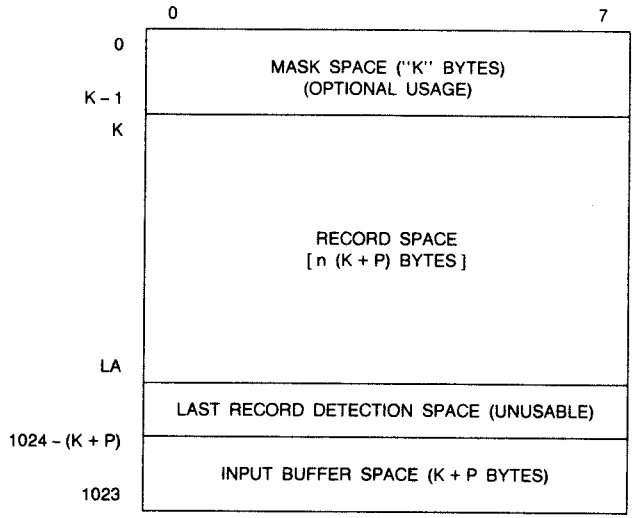


Figure 1. Am95C85 CADM Physical Model

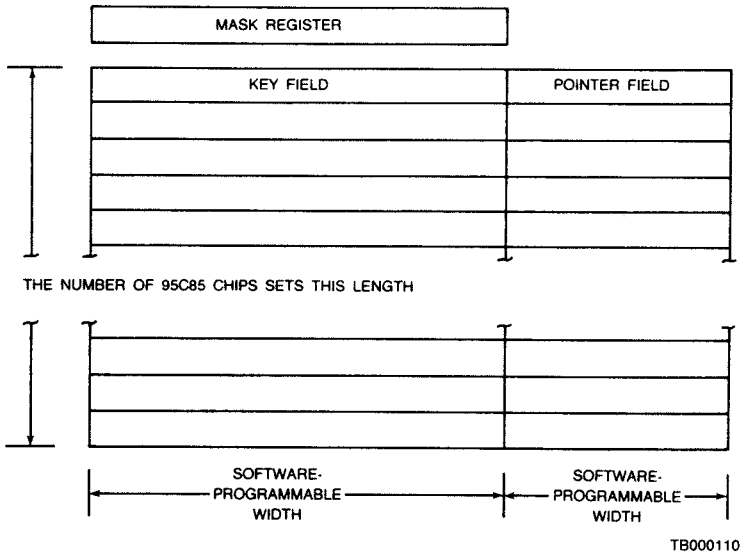


Figure 2. Am95C85 CADM Programmer's Model

Performance

"FND (Find)" Performance Equation

The "Find" performance equation assumes that the non-matching keys are different from the search key (in the most significant byte), and that the first match is found at the end of the longest possible binary search. The Find command's binary search is executed in parallel by all the CADM devices in an array. The device that finds the first occurrence terminates the operation by pulling $\overline{DONE}$ LOW. This explains why, for multi-chip arrays, the "Find" performance is independent of the total number of records. This equation includes the time required to load the search key.

"SOF (Sort-Off-Line)" Performance Equations

The Am95C85 CADM sorting performance is data-dependent. Best-case performance, quickest sort, is achieved from previously sorted data with no matching most significant bytes. The data which takes the longest time to sort is already sorted in reverse or descending order, and contains matching most significant bytes, where only the least significant bytes differ. The following two equations establish performance bounds for these two extremes.

"FIND" PERFORMANCE EQUATION

$$T_F = \frac{39 + 5K + (5.5 + 3K)\ (\lfloor \log_2 (n) \rfloor + 1)}{F}$$

"SORT OFF-LINE" PERFORMANCE EQUATIONS

Best-Case Performance:

$$T_{SB} = \frac{9 + N\left[\ 20 + 6\ (K + P) + 8.5\ (\lfloor \log_2\ (n + 1) \rfloor)\right]}{F}$$

Worst-Case Performance:

$$T_{SW} = \frac{9 + N\left[21 + \left(9 + \left\lceil \frac{N}{n} \right\rceil\right)\ (K + P) + (\lfloor \log_2 (n) \rfloor + 1)\ (5.5 + 3K)\right]}{F}$$

Where:

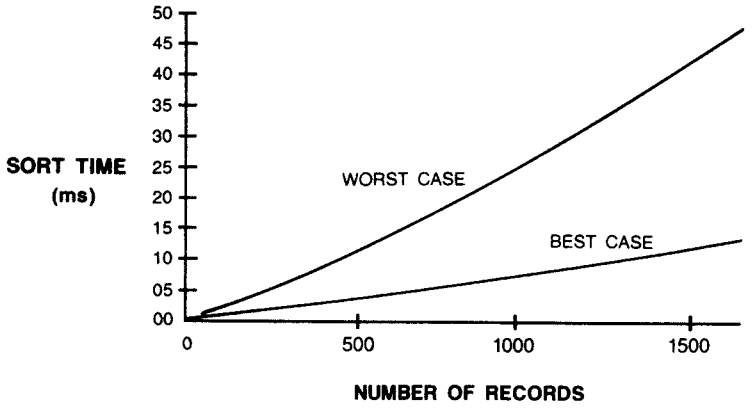
- N = Total no. of records
- n = No. of records in each chip
- K = No. of bytes/key
- P = No. of bytes/pointer
- F = Frequency of Am95C85 clock
- $\lfloor\ \rfloor$ = Truncate notation
- $\lceil\ \rceil$ = Round up notation
- T_{SB} = Time for sort (best case)
- T_{SW} = Time for sort (worst case)
- T_F = Time for find

$$n = \left\lfloor \frac{1024 - K - P - M}{K + P} \right\rfloor$$

M = K (if masking is used)
0 (if masking not used)

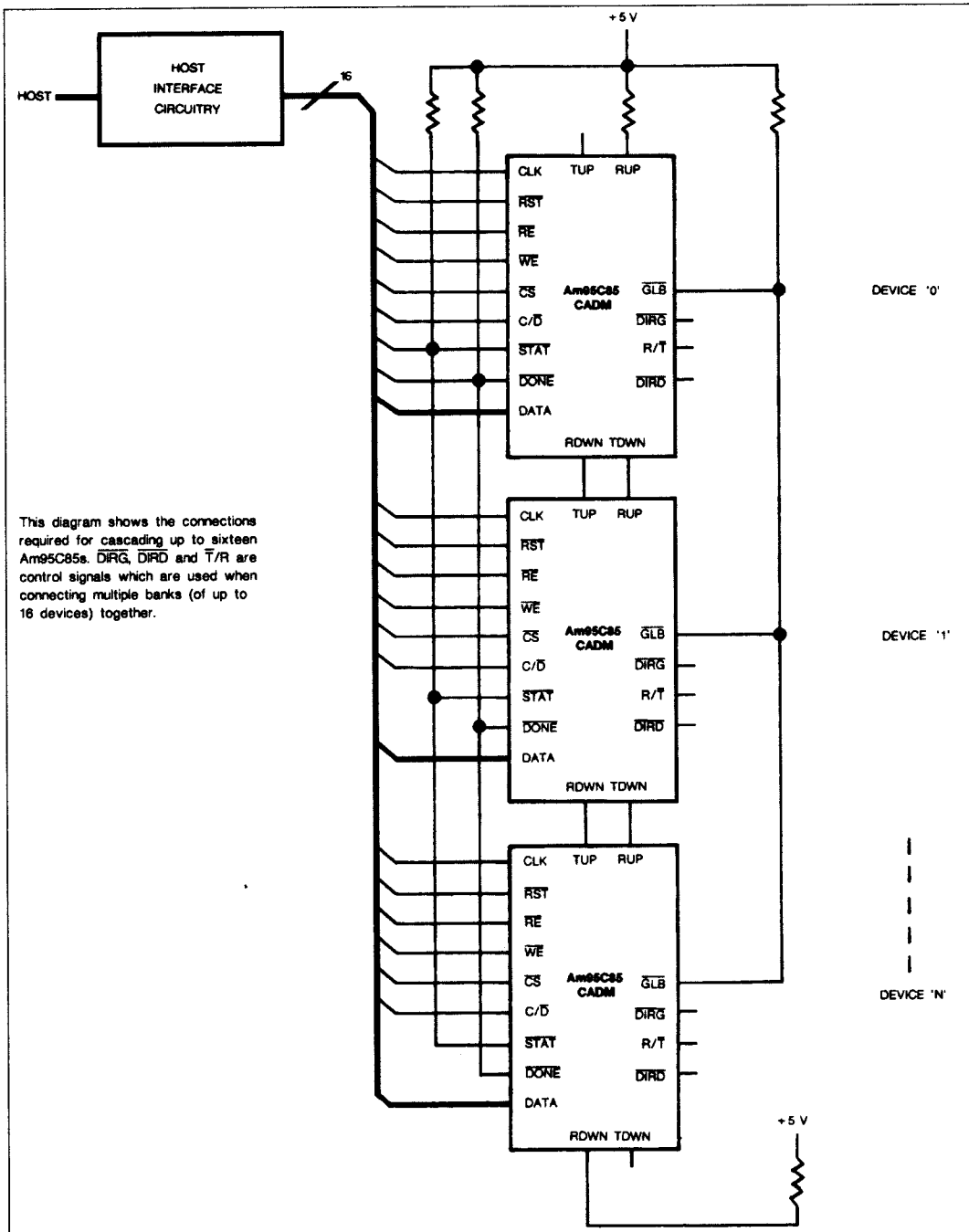
Am95C85 CADM SORT PERFORMANCE

(K = 8, P = 2)



OP001821

For This Case:
T_F = 6.4 μs
CLK = 16 MHz



This diagram shows the connections required for cascading up to sixteen Am95C85s. $\overline{DIRG}$, $\overline{DIRD}$ and $\overline{T}/R$ are control signals which are used when connecting multiple banks (of up to 16 devices) together.

Figure 3. Am95C85 Cascade Circuit

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to GND ..... −0.5 to +7

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	16 MHz, 12 MHz		Units
			Min.	Max.	
V_{OH}	Output HIGH Voltage	I_{OH} = −12 mA (I_{OH} = −1 mA for TUP, TDWN)	2.4	V_{CC}	V
V_{OL}	Output LOW Voltage	I_{OL} = 12 mA (I_{OH} = 1 mA for TUP, TDWN)	0	0.45	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + 0.5	V
V_{IL}	Input LOW Voltage		−0.5	0.8	V
I_{IL}	Input Leakage Current	0 V < V_{IN} < V_{CC}		±10	μA
I_{LO}	Output Leakage Current	0 V < V_{OUT} < V_{CC}		±10	μA
I_{CC}	Maximum Average Power Supply Current	V_{CC} = 5.5 V, 16 MHz, Outputs loaded, worst-case data shifts during Push		200	mA
I_{CCS}	Maximum Average Power Supply Standby Current	V_{CC} = 5.5 V, 16 MHz, Outputs unloaded, No-ops		125	mA
V_{CC}	Power Supply Voltage		4.5	5.5	V

CAPACITANCE*

Parameter Symbol	Parameter Description	Test Conditions	16 MHz, 12 MHz		Units
			Min.	Max.	
C_i	Input Capacitance ($\overline{RE}$, $\overline{WE}$, $\overline{CS}$, C/$\overline{D}$, $\overline{RST}$, RUP, RDWN, CLK)	F_C = 1 MHz, V_{CC} = 0 V, GND = 0 V, unmeasured pins floating		8	pF
C_o	Output Capacitance (TUP, TDWN, R/$\overline{T}$)	F_C = 1 MHz, V_{CC} = 0 V, GND = 0 V, unmeasured pins floating		10	pF
C_{io}	I/O Capacitance ($\overline{DONE}$, $\overline{DIRD}$, $\overline{GLB}$, $\overline{DIRG}$, $\overline{STAT}$, DATABUS)	F_C = 1 MHz, V_{CC} = 5 V, GND = 0 V, unmeasured pins floating		12	pF

*The capacitance values are guaranteed by design and are not tested.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

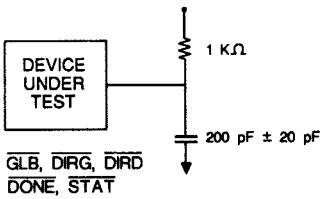
No.	Parameter Symbol	Parameter Description	16 MHz		12 MHz		Units
			Min.	Max.	Min.	Max.	
SYSTEM-TO-CADM TIMINGS							
1	t_{CC}	CLK Period	62	1000	82	1000	ns
2	t_{CH}	CLK HIGH Time	26		36		ns
3	t_{CL}	CLK LOW Time	26		36		ns
4	t_{DS}	Data Setup Before CLK LOW for Write	5		5		ns
5	t_{DH}	Data Hold After CLK LOW for Write	24		24		ns
6	t_{WS}	$\overline{WE}$ Setup Before CLK LOW	10		10		ns
7	t_{WW}	$\overline{WE}$ Pulse Width	86		106		ns
8	t_{WH}	$\overline{WE}$ Hold After CLK LOW	14		14		ns
9	t_{RS}	$\overline{RE}$ Setup Before CLK LOW	10		10		ns
10	t_{RR}	$\overline{RE}$ Pulse Width	86		106		ns
11	t_{RH}	$\overline{RE}$ Hold After CLK LOW	14		14		ns
12	t_{CSS}	$\overline{CS}$ Setup Before CLK LOW	10		10		ns
13	t_{CSW}	$\overline{CS}$ Pulse Width	86		106		ns
14	t_{CSH}	$\overline{CS}$ Hold After CLK LOW	14		14		ns
15	t_{CDS}	C/$\overline{D}$ Setup Before CLK LOW	10		10		ns
16	t_{CDW}	C/$\overline{D}$ Pulse Width	86		106		ns
17	t_{CDH}	C/$\overline{D}$ Hold After CLK LOW	14		14		ns
18	t_{SS}	$\overline{RST}$ Setup Before CLK LOW	10		10		ns
19	t_{SW}	$\overline{RST}$ Pulse Width	210		270		ns
20	t_{SH}	$\overline{RST}$ Hold After CLK LOW	14		14		ns
CADM-TO-SYSTEM TIMINGS							
21	t_{LDVR}	CLK LOW to Data Valid for Read		26		37	ns
22	t_{HDTR}	CLK HIGH to Data Three-State for Read		20		30	ns
23	t_{CHDH}	CLK HIGH to $\overline{DONE}$, $\overline{DIRD}$ HIGH		20		30	ns
24	t_{CLDT}	CLK LOW to $\overline{DONE}$, $\overline{DIRD}$ Three State		20		30	ns
25	t_{CHDL}	CLK HIGH to $\overline{DONE}$, $\overline{DIRD}$ LOW		20		30	ns
26	t_{CHSH}	CLK HIGH to $\overline{STAT}$ HIGH		20		30	ns
27	t_{CLST}	CLK LOW to $\overline{STAT}$ Three State		20		30	ns
28	t_{CHSL}	CLK HIGH to $\overline{STAT}$ LOW		20		30	ns
29	t_{CLRL}	CLK LOW to R/$\overline{T}$ LOW		26		37	ns
30	t_{CHRH}	CLK HIGH to R/$\overline{T}$ HIGH		20		30	ns

SWITCHING CHARACTERISTIC (continued)

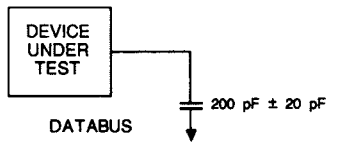
No.	Parameter Symbol	Parameter Description	16 MHz		12 MHz		Units
			Min.	Max.	Min.	Max.	
CADM-TO-CADM TIMINGS							
31	t_{CHRL}	CLK HIGH to R/$\overline{T}$ LOW for Interchip Data Move		20		30	ns
32	t_{HDVI}	CLK HIGH to Data Valid for Interchip Data Move	5	20	5	30	ns
33	t_{HDTI}	CLK HIGH to Data Three State for Interchip Data Move		20		30	ns
34	t_{CHGH}	CLK HIGH to $\overline{GLB}$ HIGH		20		30	ns
35	t_{CLGT}	CLK LOW to $\overline{GLB}$ Three State		20		30	ns
36	t_{CHGL}	CLK HIGH to $\overline{GLB}$, $\overline{DIRG}$ LOW		20		30	ns
37	t_{LDGH}	CLK LOW to $\overline{DIRG}$ HIGH		26		37	ns
38	t_{HDGT}	CLK HIGH to $\overline{DIRG}$ Three State		20		30	ns
39	t_{HTUH}	CLK HIGH to TUP HIGH		20		30	ns
40	t_{HTUL}	CLK HIGH to TUP LOW		20		30	ns
41	t_{LTUL}	CLK LOW to TUP LOW		15		20	ns
42	t_{RDSL}	RDWN Setup Before CLK LOW	6		6		ns
43	t_{RDHL}	RDWN Hold After CLK LOW	26		36		ns
44	t_{RUSL}	RUP Setup Before CLK LOW	6		6		ns
45	t_{RUHL}	RUP Hold After CLK LOW	26		36		ns
46	t_{HTDH}	CLK HIGH to TDWN HIGH		20		30	ns
47	t_{HTDL}	CLK HIGH to TDWN LOW		20		30	ns
48	t_{LTDH}	CLK LOW to TDWN HIGH		15		20	ns
49	t_{RUDT}	RUP HIGH to Data Three State	0	15	0	20	ns
50	t_{RURH}	RUP HIGH to R/$\overline{T}$ HIGH (for Pop)	0	15	0	25	ns
51	t_{RDDT}	RDWN LOW to Data Three State	0	15	0	20	ns
52	t_{RDRH}	RDWN LOW to R/$\overline{T}$ HIGH (for Push)	0	15	0	25	ns
53	t_{DHTD}	Data Hold After TDWN HIGH for Interchip Data Move (for Pop)	0		0		ns
54	t_{DHTU}	Data Hold After TUP LOW for Interchip Data Move (for Push)	0		0		ns
55	t_{DSL}	$\overline{DONE}$ Setup Before CLK LOW	6		6		ns
56	t_{DHL}	$\overline{DONE}$ Hold After CLK LOW	26		36		ns
57	t_{GSL}	$\overline{GLB}$ Setup Before CLK LOW	6		6		ns
58	t_{GHL}	$\overline{GLB}$ Hold After CLK LOW	26		36		ns

SWITCHING TEST CIRCUITS

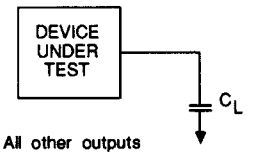
AC Loading



TC003982



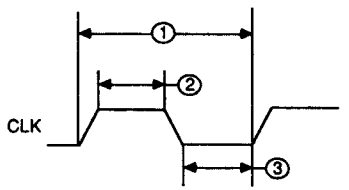
TC004002



All other outputs
where C_L = 100 pF ± 20 pF for R/$\overline{T}$
C_L = 20 pF for TUP, TDWN

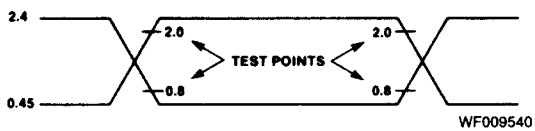
TC003991

SWITCHING TEST WAVEFORMS



WF023670

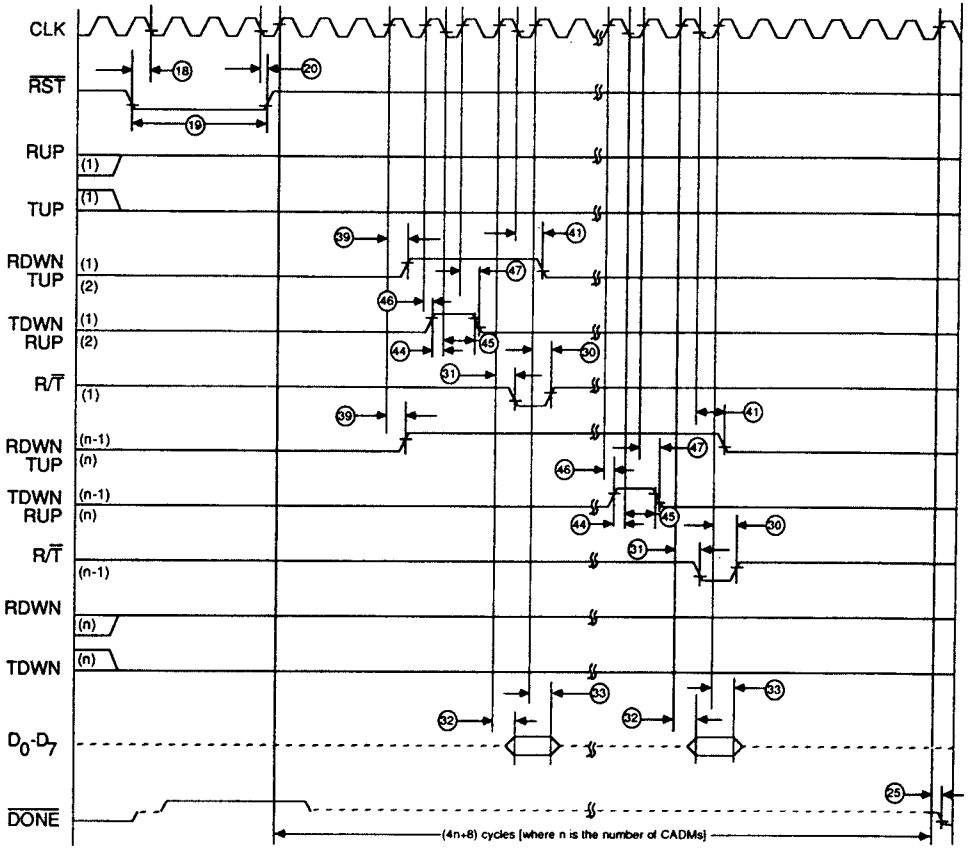
Clock



WF009540

Input/Output

SWITCHING WAVEFORMS

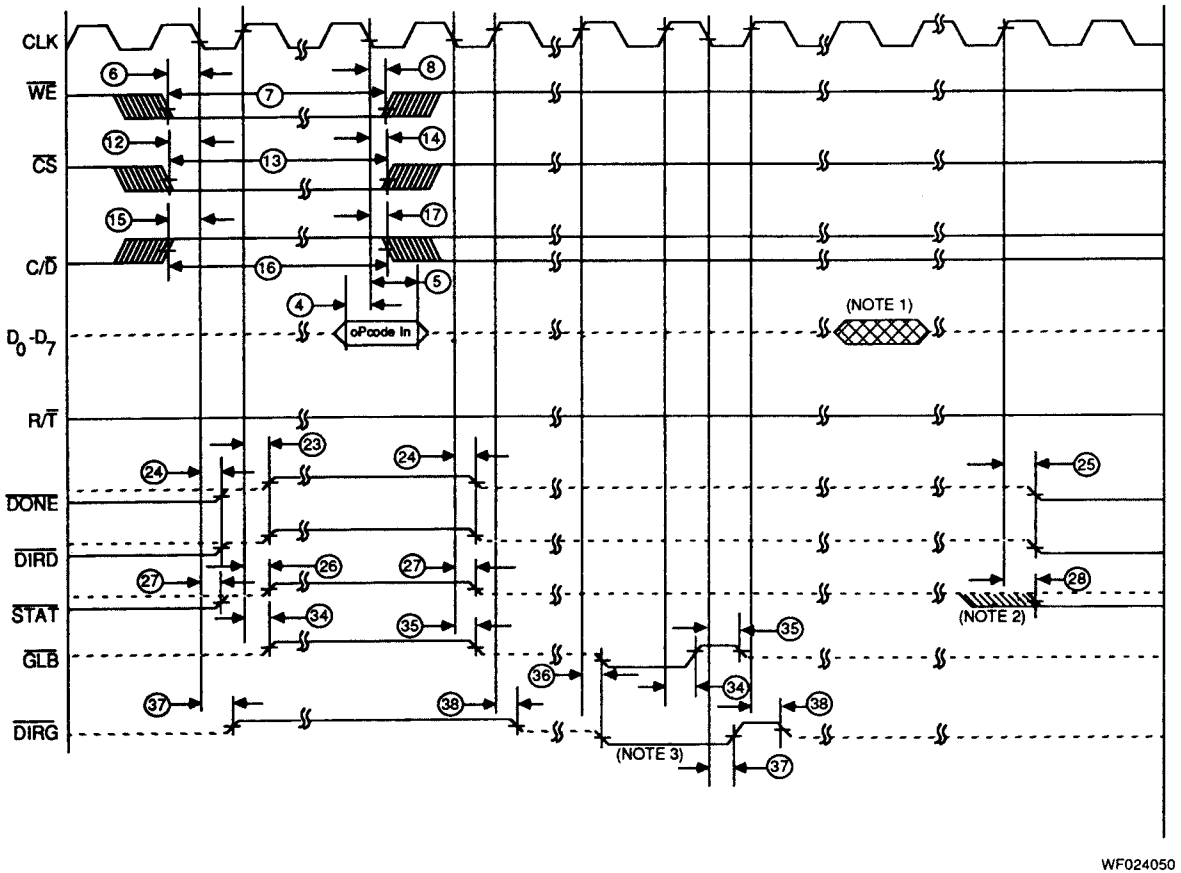


WF024040

Hardware Reset Timing

Note: TUP and TDWN are guaranteed to be LOW only when the CLK is LOW.

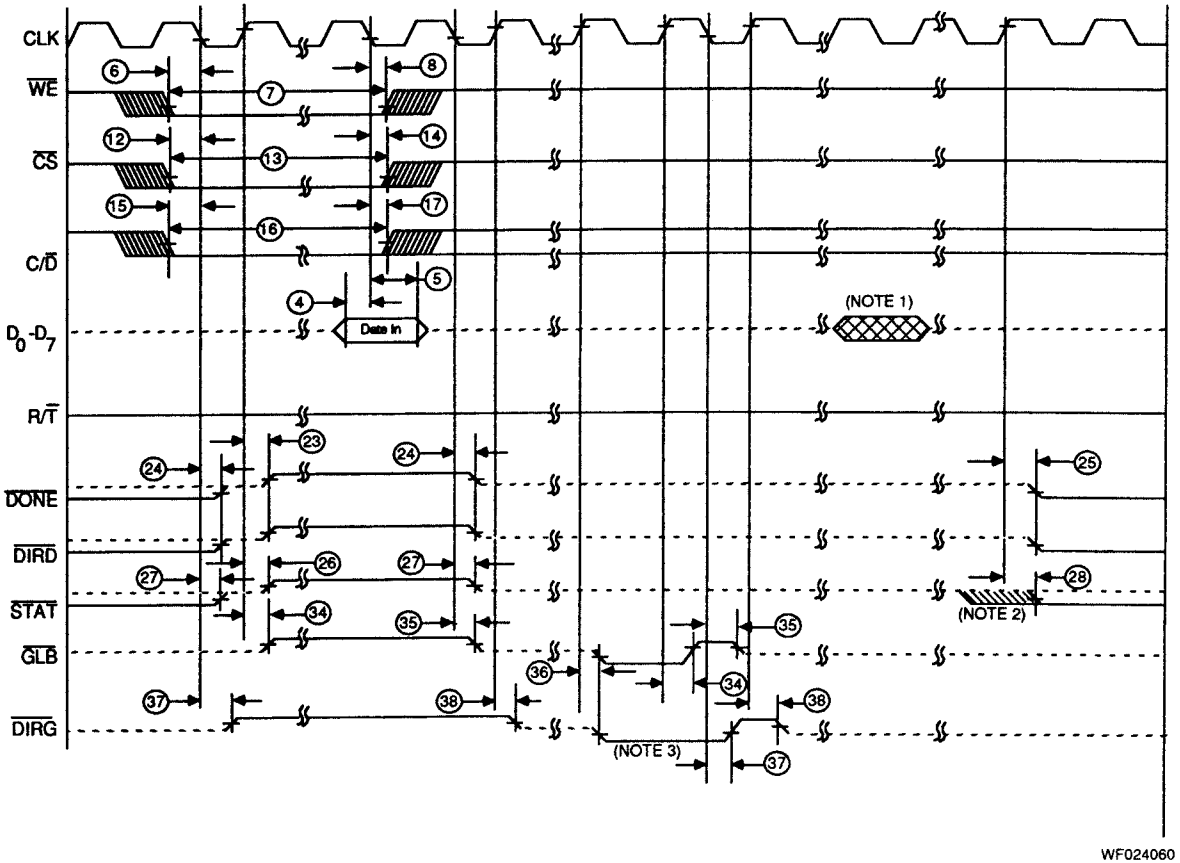
SWITCHING WAVEFORMS (continued)



Command Write Timing

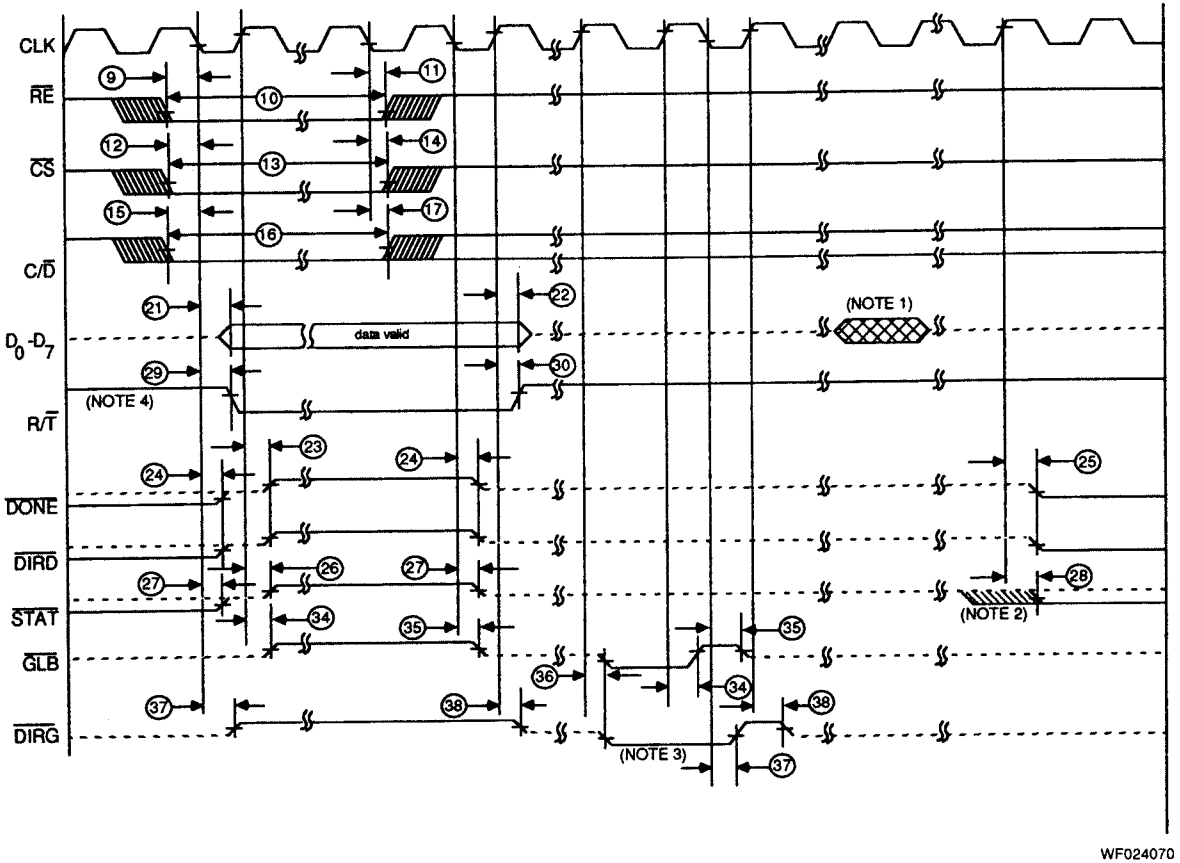
Notes: 1. Any CADM may drive the data bus at any time during the command.
2. $\overline{STAT}$ is indeterminate one cycle before $\overline{DONE}$. It must be qualified with $\overline{DONE}$ being asserted.
3. $\overline{GLB}$ and $\overline{DIRG}$ may occur multiple times or not at all during the command.

SWITCHING WAVEFORMS (continued)



Data Write Timing

Notes: 1. Any CADM may drive the data bus at any time during the write.
2. $\overline{STAT}$ is indeterminate one cycle before $\overline{DONE}$. It must be qualified with $\overline{DONE}$ being asserted.
3. $\overline{GLB}$ and $\overline{DIRG}$ may occur multiple times or not at all during the write.

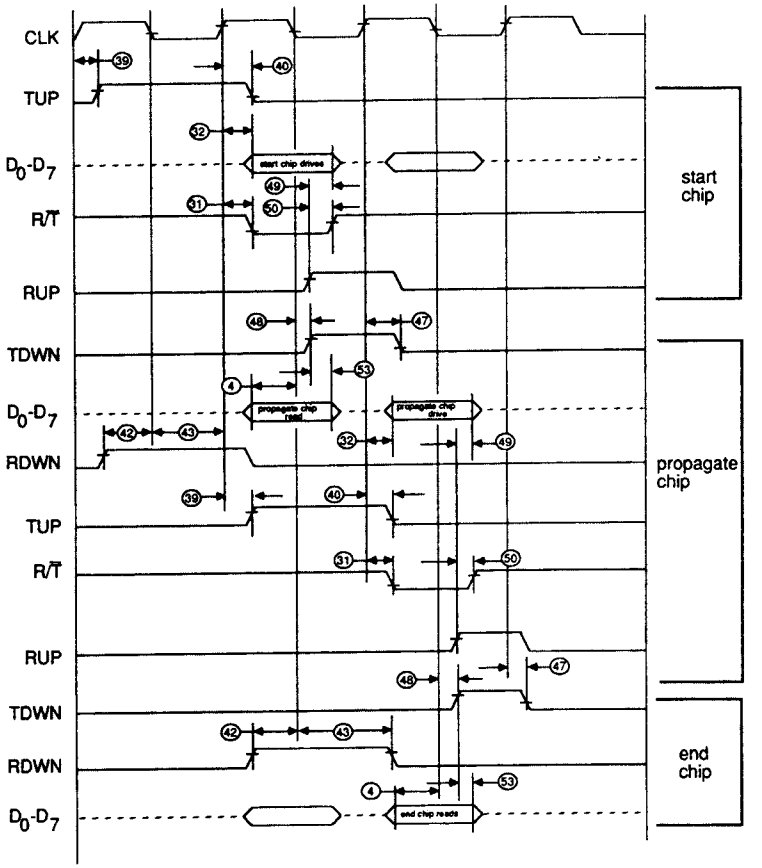


WF024070

Data Read Timing

Notes: 1. Any CADM may drive the data bus at any time during the read.
2. $\overline{STAT}$ is indeterminate one cycle before $\overline{DONE}$. It must be qualified with $\overline{DONE}$ being asserted.
3. $\overline{GLB}$ and $\overline{DIRG}$ may or may not occur during the read.
4. TUP and TDWN are not guaranteed to be LOW during the cycle when $\overline{T}/R$ switches LOW during a read.

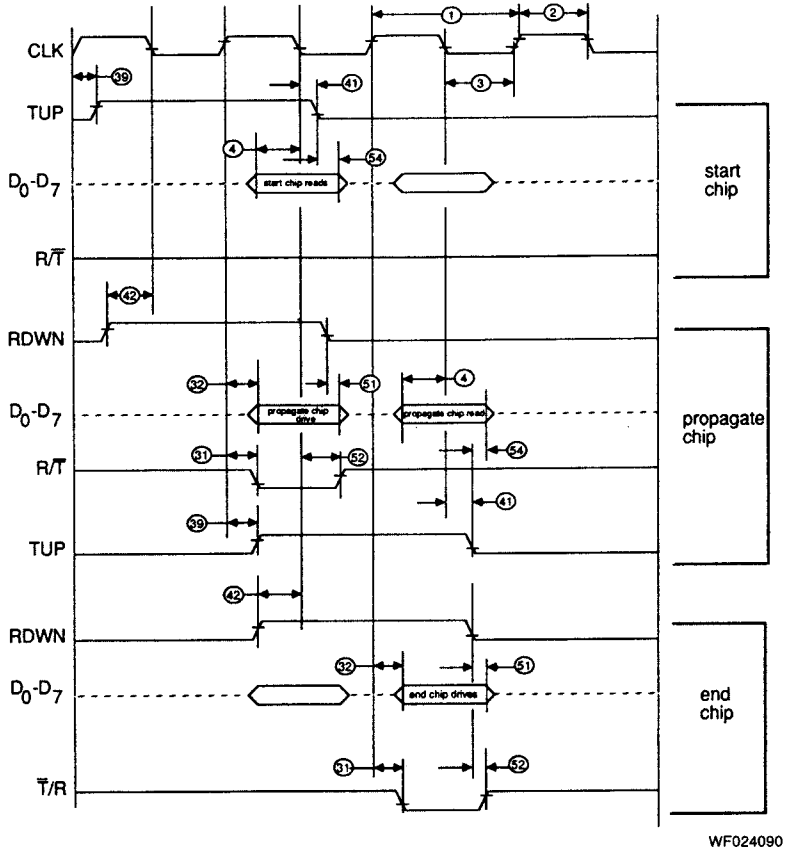
SWITCHING WAVEFORMS (continued)



Pop Timing

Note: TUP and TDWN are guaranteed to be LOW only when the CLK is LOW.

SWITCHING WAVEFORMS (continued)

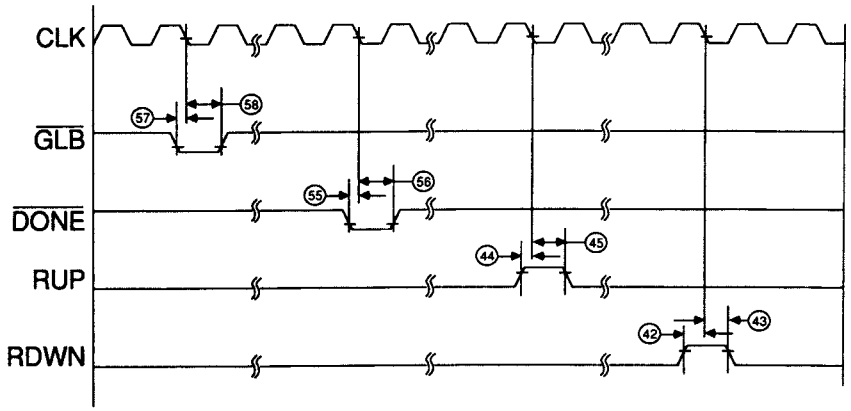


WF024090

Push Timing

Note: TUP and TDWN are guaranteed to be LOW only when the CLK is LOW.

SWITCHING WAVEFORMS (continued)

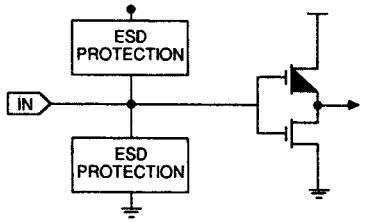


WF024100

Drive $\overline{\text{GLB}}$, $\overline{\text{DONE}}$, RUP, RDWN Timing

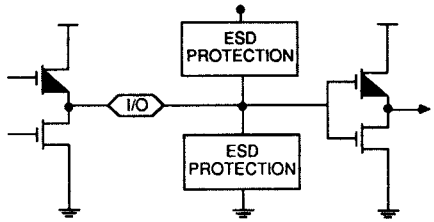
INPUT/OUTPUT CIRCUIT DIAGRAMS

Inputs: $\overline{RE}$, $\overline{WE}$, $\overline{CS}$, C/$\overline{D}$, $\overline{RST}$, RDWN, RUP



TC004110

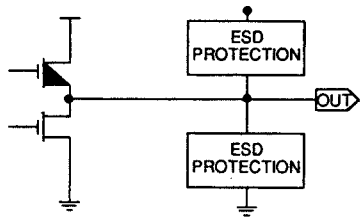
Bi-Directional: $\overline{DONE}$, D0-7, $\overline{GLB}$, TUP, TDWN*



TC004120

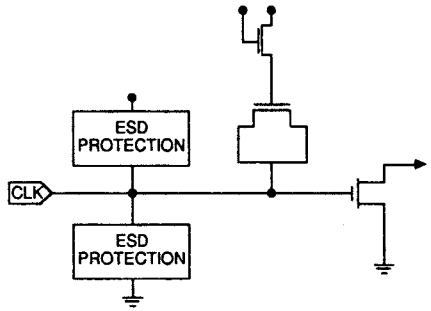
*TUP and TDWN are inputs for test mode only

Outputs: $\overline{STAT}$, $\overline{DIRG}$, $\overline{DIRD}$, R/$\overline{T}$



TC004130

Clocks: CLK



TC004140