

This document is © Copyright Timatic Systems Ltd. All rights reserved.

Scanned, OCR'd and re-typeset by Kevin Thacker.

XROM

SIDEWAYS ROM AND EPROM PROGRAMMER

NO MACHINE CODE KNOWLEDGE
REQUIRED TO PROGRAM
"BASIC" & "BINARY" ON ROM

ACCEPTS 4 = SIDEWAYS ROMS
(2764 or 27128) SWITCH SELECTABLE

PROGRAMS 2764 & 27128 EPROMS
(SOFTWARE SELECT)

FAREHAM MARKET, FAREHAM, HANTS PO16 0LB

Telephone: Fareham (0329) 239953/236727

*** TIMATIC XROM ***
*** EPROM PROGRAMMER ***
***SIDEWAYS ROM CARD ***

Thank you for purchasing the **TIMATIC XROM CARD**. We hope that it will suit your needs and that you will have trouble free use of the device.

This device is sold on an "as is" basis and no guarantee can be given that it will suit your specific purpose. We can however guarantee that it will function as both a **SIDEWAYS ROM CARD** and **EPROM PROGRAMMER** on the **AMSTRAD (464 - 664 - 6128)**.

The card is guaranteed for a period of **SIX** months against any defects of manufacture. It is essential that you therefore fill out the guarantee card enclosed with the device. Should you have cause to suspect malfunction then please return the **XROM** to us. We will repair or replace free of charge if a fault in manufacture is found. To help us please enclose as detailed a description as possible of the problem.

This guarantee does not affect your statutory rights.

If you would like details of further **TIMATIC** products please tick the box provided when filling out the guarantee. No guarantee is valid until we receive your completed form.

Please note that the **ZIF SOCKET** is not covered by this guarantee.

**THE XROM DESIGN AND THIS DOCUMENT ARE THE
COPYRIGHT OF TIMATIC SYSTEMS LTD.
UNAUTHORISED REPRODUCTION IS ILLEGAL AND WILL
LEAD TO PROSECUTION.**

*** TIMATIC XROM FITTING ***

*** IN ALL CASES PLEASE FIRST ENSURE THAT YOUR *
* COMPUTER IS SWITCHED OFF ***

PLACE THE CONNECTOR ON THE **XROM** RIBBON CABLE ONTO THE SLOT MARKED EXPANSION ON THE REAR OF YOUR COMPUTER. SPECIAL CARE SHOULD BE TAKEN TO ENSURE THAT THE CONNECTOR HAS MADE A GOOD CONNECTION AND THAT THE CONNECTOR IS THE CORRECT WAY UP. THIS IS SHOWN IN DIAGRAM (2).

FOR OWNERS OF THE **464** PLUG THE DISC DRIVE ONTO THE CONNECTOR PROVIDED ENSURING THAT THE DISC CABLE GOES TO THE RIGHT OF THE MACHINE.

IT IS ONLY NECESSARY TO PLUG IN THE MAINS ADAPTOR IF YOU ARE GOING TO BE PROGRAMMING **EPROMS**. IF THE BOARD IS ONLY BEING USED AS A ROM CARD THEN LEAVE THE MAINS ADAPTOR OFF.

*** SPECIAL NOTE ***

IN ALL CASES IF THE POWER SUPPLY IS USED THEN THE MAINS TO THE PROGRAMMER SHOULD BE SWITCHED ON BEFORE POWER TO THE COMPUTER.

*** ROM TYPES ***

The **XROM** will accept two types of **ROMS**. These are **8K** and **16K ROMS** also known as **2764's** and **27128's**. Amstrad recommend that these have an access time of **200 nsec**. We have found that the more normal **250 nsec ROMS** will work although these should only be used for development work. The difference in price is now so little that using **250's** is probably a false economy.

*** SIDEWAYS ROM CARD ***

The provision of the Amstrad to accept sideways **ROMS** has meant that there are now several **ROMS** available which can operate on the **AMSTRAD 464 - 664 - 6128**.

*** ROM ADDRESSING ***

The **XROM** can accept up to four **ROMS** each with a unique number. There are some rules to do with the type of **ROMS** and the numbers that these may occupy. There are three types of **ROMS** available to the Amstrad, these are. Background, Foreground and Extension.

Basically the rules are:-

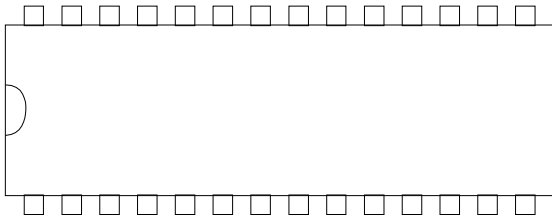
1. Background **ROMS** have numbers from 1 to 7
2. Foreground **ROMS** have numbers from 0 to 251
3. No two **ROMS** may have the same number
4. Foreground **ROMS** must not have empty spaces before them

The **XROM** allows each socket to be numbered 1 to 6 by use of the related switch. Each switch is numbered to make selection as easy as possible. If a **ROM** socket is unoccupied the related bank should have all switches off.

The position of each socket is shown by dia.(4). The number is selected when the switch is in the **ON** position as shown in dia.(5)

All **ROMS** must be fitted with the notch facing towards the **ZIF** socket on the far end of the board. Your attention is drawn to the following hints:-

Always insert the **ROM** the correct way around,



DIA.3

PIN 1

Always check that none of the **ROM** pins has become bent during insertion.

Use great care when removing **ROMS**, use a flat blunt object like a screwdriver and try to remove as evenly as possible. Broken pins are very hard to re-solder!

*** ROM SELECTION ***

There are four commands available after the machine has powered up and the **XROM** has signed on. As examples these are:-

|E or **le** selects **XROM PROGRAMMER MENU**

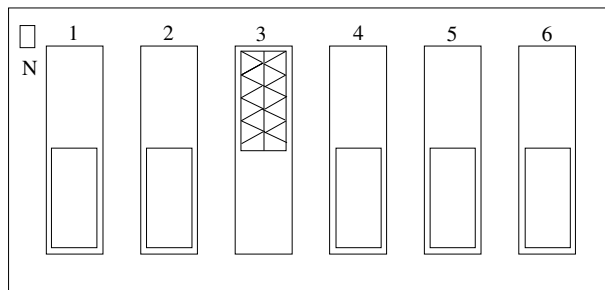
|EPROM or **leprom** selects **XROM PROGRAMMER MENU**

|E.ID or **le.id** displays **XROM** version number

|XROM.OFF or **lxrom.off** switches off **XROM EPROM** and allows computer to be used as if the **XROM EPROM** was not fitted, without having to remove hardware.

Other sideways **ROMS** have their own commands, please read their instructions.

DIA.5



ROM IS SELECTED WHEN SWITCH IS IN
POSITION SHOWN BY SWITCH NO.3.
EACH SWITCH RELATES IT'S OWN ROM
NUMBER AS SHOWN.

*** EPROM PROGRAMMER ***

The **XROM PROGRAMMER** will program two types of **EPROMS**, these are:-

2764 8K EPROM
27128 16K EPROM

Amstrad recommend that these should have an access time of 200 nsecs in order to ensure consistent results.

The **PROGRAMMER** should be connected to the computer with all power supplies **OFF**. The **PROGRAMMER** should then be switched on **BEFORE** the computer. This should be done **WITHOUT** a **ROM** inserted in the **ZIF** socket. Failure to observe this rule could damage the **PROGRAMMER** and the inserted **ROM**.

Access to the **PROGRAMMER** software is obtained by the use of the commands **!EPROM** or **!E**.

(THE BAR IS ACCESSED BY THE SHIFT AND @ KEY)

Control then passes to the **PROGRAMMER ROM** and all the options are displayed in the form of a menu.

Please note that once the **PROGRAMMER** menu is called it will use areas of **RAM** for programming. It will therefore corrupt any programs that are in memory or that you have been working on. These should first be saved before entering the **PROGRAMMER** menu.

*** XROM PROGRAMMER ***

*** COMMANDS ***

THERE ARE FOUR MENUS AVAILABLE FOR PROGRAMMING EPROMS AND GENERAL UTILITIES. THESE CAN BE USED FOR PROGRAMMING AND CAN BE USED FOR GENERAL WORK AS A UTILITY ROM.

THE MENUS ARE:

- 1 COMMAND MENU
- 2 PROGRAMMING MENU
- 3 UTILITY MENU
- 4 HOUSEKEEPING MENU

THE MAIN MENU IS ACCESSED WITH THE COMMANDS **IE** OR **IEPROM**. ALL THE OPTIONS FROM WITHIN THE MENUS ARE SELECTED WITH THE FUNCTION KEYS ON THE RIGHT HAND SIDE OF YOUR MACHINE ON THE **464** THESE ARE MARKED 1 THRU TO 9 AND ON THE **664-6128** THESE ARE MARKED AS F0 THRU TO F9.

ALWAYS INSERT WITH NOTCH FACING LEVER DIA (6)

COMMAND MENU

f1	PROGRAMMER COMMANDS
f2	UTILITY COMMANDS
f3	HOUSEKEEPING COMMAND
f4	EPROM SIZE
f5	PROGRAMMING SPEED
f6	COMMAND SUMMARY
ESC	QUIT TO BASIC

*** PROGRAMMER COMMANDS ***

This command allows entry to the **PROGRAMMER** menu.

*** UTILITY COMMANDS ***

This command allows entry to the **UTILITY** menu.

*** HOUSEKEEPING COMMANDS ***

This command allows entry to the **HOUSEKEEPING** menu.

*** EPROM SIZE ***

This option selects the amount or size of **EPROM** to program in single bytes. The default on powering up the **XROM** is **&2000** which is equal to **8K** or a **2764 EPROM**. The most usual options are:-

&4000 - 16K - 27128

&2000 - 8K - 2764

The third option is intended to speed up the programming process if required. The Data size in single bytes should be input when requested. This will then program from **&6000** to the length you gave, i.e. if you input 01 the **PROGRAMMER** will program from **&6000** to **&6001**. It makes no difference to the **ROM** how much you program but do err on the side of caution, remember one byte short can stop the **ROM** from working where one too many won't affect it. If you have selected the third option you will also be asked for a new **EPROM** start address;- this can be any address. Selecting for example **EPROM** size of **&01** would program one single byte. If then **EPROM** start address were changed to **&1000** this would program one single byte from **EPROM** address **&1000**.

This command is very useful for development work and as an example we developed the **XROM** software in the following stages.

First we wrote a **MENU** screen and then placed this on an **EPROM**. Each of the **MENU** options jumped to a preset location on the **EPROM**.

i.e. EPROM ADD	COMMAND JUMP
&0200	PROGRAMMER COMMANDS
&0400	UTILITY COMMANDS
&0600	HOUSEKEEPING COMMANDS
&0800	EPROM SIZE

etc.

We then developed each option in **RAM** and tested. Once the **RAM** version was de- bugged we placed the routine at it's correct address on the **EPROM** and further tested. This saved having to test several routines at the same time and also saved on the number of **EPROMS** required.

*** PROGRAMMING SPEED ***

This command has two options;- **FAST** and **SLOW**.

The **FAST** will program a **2764 8K EPROM** in approximately 2 minutes and a **27128 16K EPROM** in approximately 5 minutes. **SLOW** speed will program the **EPROM** with the maximum pulse for each location and for a **2764 8K EPROM** will take approximately 6 minutes and for a **27128 16K EPROM** will take approximately 14 minutes. This **SLOW** speed is recommended for **MASTER EPROMS**.

*** COMMAND SUMMARY ***

This option will display a brief description of the commands available from within the ROM.

UTILITY MENU

f1	HARD COPY
f2	PROBE ROMS
f3	DUMP MEMORY
f4	EDIT MEMORY
f5	CLEAR BUFFER
f6	MOVE MEMORY
f7	FILL MEMORY
ESC	RETURN TO COMMAND MENU

*** ROMPROBE ***

This command will check all the selected **ROMS** and print their **ROM** names and their **ROM** number with relation to the machine and their version number. This is most useful for checking if a suspect **ROM** is actually being seen by the machine etc. This command will also output to the printer if the **HARD COPY** option is on.

*** DUMP MEMORY ***

This command will dump memory to the screen from the first address given. To pause this listing press the **ESCAPE** key once. Pressing **ESCAPE** again will exit you back to the menu, any other key will continue the display. This command displays both **ASCII** and **HEX**. All **HEX** codes will be displayed but only **ASCII** character codes recognised are displayed. All other codes are displayed as a “.”.

*** EDIT MEMORY ***

This command is used to edit anywhere in memory. There are two modes of editing. These are **ASCII** and **HEX**. To switch from one to the other the **TAB** key should be used. This key is used as a toggle and the cursor will stay in the chosen field until the **TAB** key is again pressed. To exit back to the menu you should press the **ESCAPE** key when in the **HEX** field. Please note that data to be programmed onto an **EPROM** should start at **&6000**.

*** CLEAR BUFFER ***

This command will clear memory from **EPROM** address to the size set with the **EPROM SIZE** command and then fill memory between these addresses with **&00**.

*** MOVE MEMORY ***

This command will move memory from any given address, and with any given length. The command will ask for the first address, then the end address, and finally the destination address.

*** FILL MEMORY ***

This command will fill memory from any given address, and with any given length with the **HEX** bytes you specify. This value must be in the range **&00** thru to **&FF**.

*** HARD COPY ***

This command will redirect output of the **DUMP MEMORY**, **FILE INFO**, **CAT**, and **ROMPROBE** to the printer if one is fitted. This command is a toggle and is active until switched off by reselecting the command. The status of the **HARD COPY** command is displayed as either:

PRINTER COPY ON / PRINTER COPY OFF

PROGRAMMING MENU

f1	PROGRAM BASIC
f2	PROGRAM BINARY
f3	PROGRAM EPROM
f4	COPY EPROM
f5	VERIFY EPROM
f6	READ EPROM
f7	BLANK CHECK EPROM
ESC	RETURN TO COMMAND MENU

*** PROGRAM BASIC ***

This option will take a basic program and place a header on the first 100 bytes which will enable the program to be called from the **ROM**.

The option will ask for a filename which must be a valid **DISC** file name:

i.e. **TOOLS** (will work)
 TOOLS ^* (will not work)

The program will take the given name and load your basic program ready for programming.

Place blank **EPROM** in **ZIF** and follow prompts.

This program will also ask if you wish to have a new name for the **ROM**. Any name given will be used as the powerup, sign-on. Pressing **RETURN** at this prompt will use the default name with which the program was loaded.

**PLEASE NOTE THAT ALL FILE NAMES MUST BE IN ASCII
BETWEEN (A - Z)**

Once the ROM has been programmed it can then be fitted into the Romcard. It will sign on with the same name as that which you gave when loading unless you have specified a new name.

**i.e Program Name =TOOLS
Rom Name =TOOLS
Rsx Name =TOOLS**

As can be seen from above there are three names. The first is your original name. The second is the name the **ROM** will sign on with on power up or reset. The third is the name you should use to call the **ROM**. This must be used with the | command. To call the above example you have to type either of the following:

|tools or |TOOLS

This will then call the header program which will load in the **BASIC** program as if it came from disc or cassette but with the advantage of no waiting to load. Once the **ROM** has been called it can be used exactly as a normal basic program.

i.e. It can be **RUN, EDITED, LISTED, SAVED**, etc.

Note: This option will use the last size set with the **EPROM SIZE** option. It will give error messages if the program is too long and you should then try to compact, It is a good idea to check the length and compact the program before putting on **ROM**. This can be done very easily by removing Rems, Spaces, Having smaller variable names, and concentrating the lines. Remember one full line is smaller than. two short lines which have the same basic.

example (1), 27 bytes

```
10 PRINT "HELLO"  
20 PRINT "GOODBYE"
```

The first program is longer than the following example even though they both are the same program. In this example it is mainly because only one line no. needs to be saved by the interpreter.

example (2), 23 bytes

```
10 PRINT "HELLO":PRINT "GOODBYE"
```

The intention of **PROGRAM BASIC** option is to allow the non machine code programmer to create their own **ROMS** which can contain anything from utilities, games or frequently used tools etc.

Note: - This option will use the last size set with the **EPROM SIZE** option.

*** PROGRAM BIN ***

This option will take a binary program and place a header on the first 100 bytes which will enable the program to be called from the **ROM**.

The option will ask for a filename which must be a valid DISC name:

i.e. **BINTOOL** (will work)
 BINTO.^* (will not work)

The program will take the given name and load your binary program ready for programming.

Place blank **EPROM** in **ZIF** and follow prompts.

This command will also ask if you Wish to have a new name for the **ROM**. Any name given will be used as the power-up sign-on. Pressing **RETURN** at this prompt will use the default name with which the program was loaded.

**PLEASE NOTE THAT ALL FILE NAMES MUST BE IN ASCII
BETWEEN (A - Z)**

Once the ROM has been programmed it can then be fitted into the Romcard. It will sign on with the same name as that which you gave when loading unless you have specified a new name.

**i.e Program Name =BINTOOL
Rom Name =BINTOOL
Rsx Name =BINTOOL**

As can be seen from above there are three names. The first is your original name. The second is the name the **ROM** will sign on with on power-up or reset. The third is the name you should use to call the **ROM**. This must be used with the | command. To call the above example you have to type either of the following:

|BINTOOL or |bintool

This will then call the header program which will load in the **BINARY** program as if it came from disc with the exception that this command will reset **HIMEM** to the correct value. It will then jump to the **EXECUTE ADDRESS** if one was found on loading. If none was found it will jump to the program **START ADDRESS**.

Note: - This option will use the last size set with the **EPROM SIZE** option.

*** PROGRAM EPROM ***

This will take data from memory address **&6000** onwards, and program an **EPROM**. The size of the **EPROM** should first be set with the **EPROM SIZE** command. This command will program from **ROM** address **0000** thru to **ROM** address nnnn. This process is not interruptable and the escape key is disabled, If you know the length of data you wish to program then use of the **EPROM SIZE** command can speed up the programming process. A single byte counter will display each byte as it is programmed. If the third option was chosen in the **EPROM SIZE** command then the **EPROM** start address will be that which was specified.

*** COPY EPROM ***

This will read an **EPROM** into memory at address **&6000**. This data will then be used to program a copy of the **EPROM** read. Please follow the 'on screen' prompts. This option should not be used to copy commercial **ROMS**.

Please be aware of the copyright laws.

*** VERIFY EPROM ***

This command checks the data in the now programmed **EPROM**. With the data starting at address **&6000**. This command will display either the first address that does not compare with memory or a message to inform you that the **EPROM** is correctly programmed.

*** READ EPROM ***

This will read an **EPROM** into memory at address **&6000**. This data can then be edited or saved as required. This command will read an **EPROM** of the size set with the **EPROM SIZE** command. The default on power-up for this is **&2000** and this is equal to a **2764 (8K) EPROM**.

*** BLANK CHECK ***

Before programming this command can be used to check if an **EPROM** is empty. i.e. - (all locations contain **&FF**). This command will return to the menu if the **EPROM** was found to be empty. If not, it will return with **DATA FOUND**. If a check of the data found is required then the **READ EPROM** and **DUMP MEMORY** option should be used. If the **EPROM** was found to have data then it should be erased before trying to program. Please note that an **EPROM** is considered empty when it contains **&FF**. An **EPROM** can only be erased by the use of an ultraviolet light of the correct wavelength.

HOUSEKEEPING MENU

f1	FILE INFORMATION
f2	LOAD FILE
f3	SAVE FILE
f4	ERASE FILE
f5	RENAME FILE
f6	CATALOGUE DISC
f7	RSX COMMANDS
ESC	RETURN TO COMMAND MENU

*** FILE INFORMATION ***

This command will give header information on all files. The information given will be:

START ADDRESS	(IN HEX)
FILE LENGTH	(IN HEX)
EXECUTION ADDRESS	(IN HEX)
FILE TYPE	

*** LOAD FILE ***

This command will ask you for the name of the file to load. It does not require the file extension unless when using disc there are two files of the same name.

i.e. **MYROM.BAS** and **MYROM.BIN**.

With these two names the file **MYROM.BAS** would be loaded first. All data loaded is read into **&6000** ready for programming. An error will be reported if the file couldn't be found or for some reason couldn't be loaded. This command will also report if the file was bigger than the size set with the **EPROM SIZE** command. You should either compact the program or choose a new **EPROM** size.

*** SAVE FILE ***

This will save a file whose length will depend on the last **ROM** size selected. The default for this at powerup, is the **2764 8K**. To save disc or cassette space you can leave the **PROGRAMMER** with the **QUIT** command and perform a normal binary save.

ie:- `save"file.bin",b,&6000,&length,&exec`

*** ERASE FILE ***

This command will ask for the name of the file to erase. (**DISC ONLY**).

*** RENAME FILE ***

This command will ask for the name of the file to rename and for the files' new name. (**DISC ONLY**).

*** CATALOGUE DISC ***

This will produce a standard **AMSTRAD** catalogue of a disc or cassette. If **HARD COPY** is on this will output to the printer.

*** RSX COMMANDS ***

This command will allow you to access any **RSX** commands present in the machine. It is very useful for switching from one **ROM** to another so for example you could use the **READ EPROM** command and then the **RSX** command to select **IM** for **MAXAM** (if fitted) and disassemble the code that was last loaded with the programmer. This saves having to reload the code, All other **RSX**'s previously loaded in memory are also available.

AMMENDMENTS

The following changes have been made to the XROM EPROM PROGRAMMER software. These superceed any other information found in this manual.

As some commands have been changed or added they may not all appear as in the manual. Those that are in the manual may be in a slightly different order.

ENTRY TO MENU'S AND COMMAND ENTRY

All keyboard inputs are now single key entries unless asked for specific input. This means that is is not necessary to press RETURN or any other keys unless you are prompted for them.

EXIT FROM XROM

It is now necessary to type the word "BYE" from the COMMAND MENU before it is possible to leave XROM. Press ESCAPE first and then type BYE. This is the only legal way to leave XROM without memory corruption etc. This will also leave the new defaults you have set in memory and these will remain when you re-enter XROM. This will work as long as these values are not over written. They are stored at around &AC00 depending on your machine.

For the EDIT memory and the DUMP memory commands pressing RETURN at the START address prompt will default to the XROM programming buffer, ie &5000.

Also for any command that uses the disc a RETURN will be accepted as default to DRIVE A:.

XROM DEFAULTS

The XROM programming buffer has been moved to &5000. This was necessary in order to enable the use of RSX commands which were being over written when XROM loaded. All references in the manual to &6000 should be changed to &5000.

Xrom when initialised now sets the following defaults.

EPROM SIZE	=	3fff 27128 16k
PRINTER COPY	=	OFF
AUTO VERIFY	=	ON
PROGRAMMING SPEED	=	FAST
EPROM START ADDRESS	=	&0000
BUFFER EMPTY	=	EMPTY

PROGRAMMING

XROM can now use the standard INTEL programming algorithm. This means than if you wish to over program an eprom you will have to use either the FAST or SLOW programming speeds. This is because the algorithm has to verify each byte programmed. This time taken to verify is then used as a time variable for programming the rest of the eprom. If the eprom has already been programmed correctly then the algorithm will reprogram with the shortest time allowed and this is not always desired.

AUTO VERIFY

The FAST and SLOW programming routines are the only routines that use the AUTO VERIFY routine. This routine will carry out a BLANK check before programming and then a VERIFY eprom after it has finished programming. This routine is a toggle and will be either turned ON or OFF.

COPY EPROM/PROGRAM BASIC/PROGRAM BINARY

When you choose one of the above commands and the file or eprom is loaded you will be asked if you wish to alter anything. Pressing ESCAPE will return you to the menu. Any other key will continue with the programming process. When YOU have finished editing you should use the PROGRAM EPROM command.

When you choose the PROGRAM EPROM option this will ask if you require to load a file. Answer "Y" to load file or any other key to program with the buffer.

We hope that you find these ammendments usefull, they have ALL been added after users have talked their with their ideas with us.

THANKS

We would like to say a special thanks to all those who have helped with the development of this device and the XROM software. It is always easy to criticize others work. it is much harder to offer solutions.

For those interested in such things this manual was written with the ARNOR PROTEXT word processor the software with ARNOR MAXAM assembler and the parts to do with basic with BEEBUGS TOOLKIT.

All of these are available on EPROM and will fit in to the XROM. These will make the device one of the most powerfull programming tools you could require.

GUARANTEE CARD

NAME : _____
ADDRESS 1 : _____
ADDRESS 2 : _____
ADDRESS 3 : _____
POST CODE : _____

DATE PURCHASED : _____
SERIAL NUMBER : _____
SUPPLIERS STAMP : _____

In order to help us with further products please fill in the lower part of this form with any ideas or comments you wish to make. This is not part of the guarantee and need not be filled in.

DO YOU REQUIRE DETAILS OF FURTHER AMSTRAD PERIPHERALS?
YES / NO

WHICH MACHINE DO YOU OWN? 464 / 664 / 6128 / 8256

WHERE DID YOU SEE THIS PRODUCT ADVERTISED?

COMMENTS

