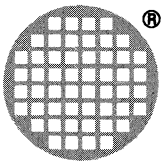
inmos



IMS B430 Prototyping TRAM User Guide

42 1478 00

October 1990

INMOS document number: 42-1478-00

Contents

1 Introduction

1.1 Layout of this User Guide

The first four sections of this document provide essential information on the IMS B430, introducing the features of the product and providing sufficient information to set up and operate the transputer system. All users are advised to read this material before beginning any prototyping task.

Section five is intended to provide experienced designers with a fuller technical discussion of the product. General familiarity with microprocessor system design is assumed, although specific knowledge of the transputer is not.

1.2 Purpose of the Prototyping TRAM

The IMS B430 combines a minimal transputer system with a general purpose prototyping area. The product supports feasibility investigations, software development, demonstrations, and other activities requiring construction of a small number of units to a specific design. Inclusion of a functional transputer system speeds up the prototyping process significantly. The user is relieved of the tasks of defining, constructing, and debugging such a system, and need only be concerned with the peripheral hardware specific to the intended application. To make the module useful in the widest possible range of applications, the transputer system may be configured in many different ways, and a variety of memory devices accommodated. The prototyping area is simply a matrix of through-holes, on which additional devices may be installed and connected to the transputer system in any desired arrangement. These devices can perform any desired function, such as extending the I/O interface capabilities of the system, implementing special-purpose interfaces and protocols, or even enhancing the transputer's computing power.

The IMS B430 conforms to the INMOS TRAM (TRAnsputer Module) standard. This published specification defines a system architecture for modules and motherboards in terms of electrical and mechanical details. INMOS IQ Systems offers a broad range of products based on this standard, featuring modules for compute and I/O functions, supported by motherboards to suit a variety of host systems. Hardware and software products from this range can be integrated very simply to create a development platform tailored to suit the task at hand.

Full details of the TRAM/motherboard architecture may be found in 'The Transputer Applications Notebook - Systems and Performance' [INMOS document number 72-TRN-205-00] (see especially sections 7 and 8). Users of the IMS B430 are strongly recommended to gain familiarity with the ideas described.

2 Pin Descriptions

Pin	In/Out	Function	Pin No.
System Services			
VCC, GND		Power supply and return	3,14
ClockIn	in	5MHz clock signal	8
Reset	in	Transputer reset	10
Analyse	in	Transputer error analysis	9
notError	out	Transputer error indicator (inverted)	11
Links			
LinkIn0-3	in	INMOS serial link inputs to transputer	13,5,2,16
LinkOut0-3	out	INMOS serial link outputs from transputer	12,4,1,15
LinkspeedA,B	in	Transputer link speed selection	6,7

Table 1 IMS B427 Pin designations

Notes:

1 Signal names are prefixed by **not** if they are active low; otherwise they are active high.

3 Standard TRAM signals

A TRAM can be regarded as a transputer with extra RAM attached but with only 16 signals brought out to the TRAM pins. The majority of the TRAM pins function in exactly the same way as the corresponding transputer signals, which are detailed in *The Transputer Databook*. However, a few of these signals are slightly different from the transputer specification as follows:

3.1 notError (pin 11)

This is an open collector signal. It is driven low when there is an error; otherwise it is pulled high by a resistor on the motherboard. This enables the **notError** outputs on several TRAMs to be wire-ORed together. (The TRAM specification recommends that no more than 10 **notError** outputs are connected together).

3.2 LinkSpeedA and LinkSpeedB (pins 6 and 7)

LinkspeedA and **LinkspeedB** set the speed of transputer link 0 and links 1-3 respectively. When the appropriate input is low the link(s) operate at 10 Mbits/s and when high the link(s) operate at 20 Mbits/s.

3.3 Link signals

Whilst the links obey a protocol identical to that described in the *Transputer Databook*, there are some differences in the electrical characteristics.

LinkIn0-3 The link inputs have pull-down resistors to ensure that they are disabled when they are not connected. Diodes are also included for protection against electrostatic discharge.

LinkOut0-3 The link outputs have resistors connected in series for matching to a 100 ohm transmission line.

4 Outline of IMS B430 Hardware

This section should be read by all users before beginning any prototyping task, to gain an understanding of the possibilities offered by the TRAM.

4.1 Printed Circuit Board

The board itself is of high quality four layer construction. All holes are through plated.

The two inner layers carry Power and Ground planes to all devices, including the prototyping area. This form of supply distribution supports reliable operation of high speed digital devices such as transputers, minimising problems of ground-bounce and noise. The two outer layers carry all signal tracks, and there are NO blind or buried vias. Therefore, the user can gain access to all signal connections.

4.2 Onboard transputer system

The IMS B430 TRAM has a T222-20 transputer with 4K bytes of fast internal RAM. There is also a socket for a 20-pin Programmable Logic Device (PLD), and two sockets for 28-pin memory devices. (Both the PLD and memory functions are discussed in more detail below). The transputer itself is socketed (IC1), so that it may be upgraded at some future date (e.g. to a T225), or replaced in case of damage. **Warning:** device removal and refitting should always be carried out with great care to avoid damaging the device and/or the board, and should not be performed unless absolutely necessary.

All the transputer signals - address bus, data bus, etc. - are brought to pads near the prototyping area. These provide convenient points for running signal wires to the prototyping devices, and for the attachment of test probe points for signal monitoring. Where possible, the pads are labelled with signal names (although in some cases the component density prevents this). A diagram identifying ALL pads and jumpers by signal name appears in Appendix A of this guide.

4.3 Prototyping area

This is made up of 1.0 mm (0.040 in.) through-plated holes in a 2.54 mm (0.1 in.) matrix. These holes will accept the leads of commonly available components, sockets, wire-wrapping terminals, etc. There are 16 columns of holes across the 1.6 in. width of the area, and 43 rows of holes along the 4.3 in. length of the area.

In twelve of the sixteen columns, the holes are surrounded by *circular* pads, both on the front and back of the board. These holes are completely uncommitted, i.e. they are NOT connected to any signals on the board, nor to the internal supply planes. However, in every fourth column, the holes are surrounded by *square* pads (again, both on the front and back of the board). These holes are connected to the internal supply distribution planes, alternate holes being connected to the power and ground layers respectively. Viewing the board from the component side, with the small yellow triangle at the top left corner, the overall arrangement is like this:

u	G	u u u	G	u u u	G	u u u	G	u u
u	V	u u u	V	u u u	V	u u u	V	u u
u	G	u u u	G	u u u	G	u u u	G	u u
		etc		etc		etc		
u	V	u u u	V	u u u	V	u u u	V	u u
u	G	u u u	G	u u u	G	u u u	G	u u
u	V	u u u	V	u u u	V	u u u	V	u u
u	G	u u u	G	u u u	G	u u u	G	u u

u = uncommitted pin

V = pin connected to internal VCC layer

G = pin connected to internal Ground layer

This can accommodate many different component geometries in whatever arrangement the user considers appropriate to the application. For example, common 0.3 in. and/or 0.6 in. pitch DIP package devices may be mixed in any or all of the arrangements in figure 1.

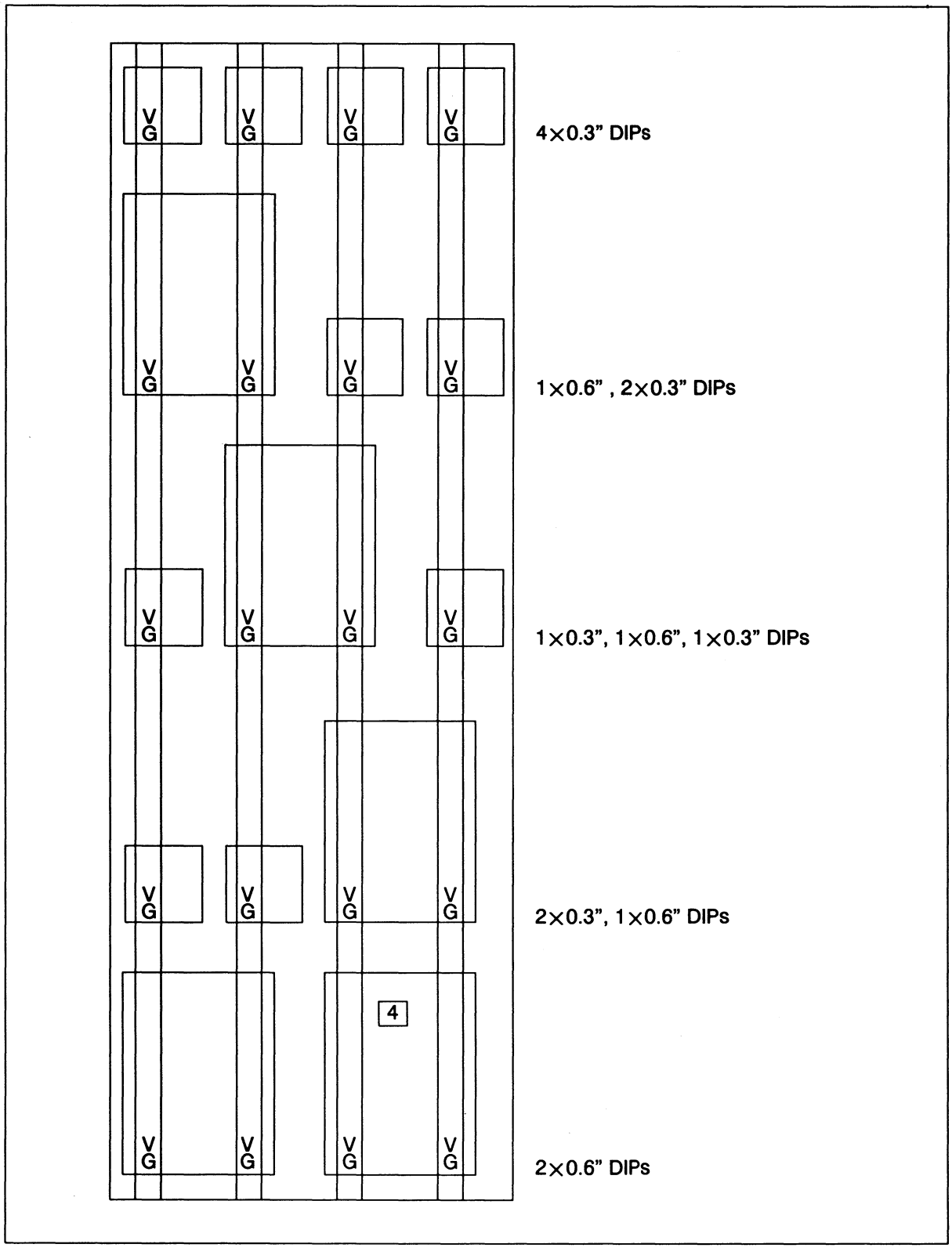

VG
4×0.3" DIPs
1×0.6" , 2×0.3" DIPs
1×0.3", 1×0.6", 1×0.3" DIPs
2×0.3", 1×0.6" DIPs
4
2×0.6" DIPs

Figure 1

IMS B430 layout possibilities

Note: To assist in planning and checking device layouts, the silk screen printing is used to highlight the arrangement of supply holes in the matrix. Lines *along the length* of the area highlight the relevant columns, and lines *across the width* the area indicate the relative placement of power and ground holes within these rows. Text at the right-hand edge of the board indicates the relative location of Ground layer holes (marked as G) from that of Power layer holes (marked as V).

These visual distinctions are present both on the front and back of the board, and allow the different hole types to be distinguished at a glance.

4.4 Programmable Logic Device PLD (IC4)

The pin connections at this socket are arranged to accept a 20-pin programmable logic device, such as a PAL 16R4, or the more flexible GAL16V8, programmed as discussed later in this user guide (or with modifications suitable to the user's application). Readers unfamiliar with PLDs are referred to the relevant data books (e.g. ST Microelectronics' Programmable Logic Manual - GAL Products) for an introduction to this technology.

The resources of the device are allocated to two distinct support functions. The first half of the device is used to decode the transputer's address and control signals. It generates chip enable signals for the two memory devices IC2B/W and IC3B/W. It also derives two chip enable signals which are uncommitted, and can be used to control devices in the prototyping area. The programming of this portion of the device recognises that different prototype designs may have different memory mapping requirements. The device programming allows for four different memory maps, which were chosen to meet the broadest possible range of applications. The user selects the most appropriate map for the intended application via the settings of jumpers JP10-11. Full details of the options are given in the later section on configuring the IMS B430.

The remainder of the device is configured as a programmable wait state generator. Since the transputer can transfer data faster than most low-cost devices, this allows the cycle time of data transfers to be extended as necessary. Again, full details of the options available are given in the later section on IMS B430 configuration.

4.5 Memory device sockets IC2B/W and IC3B/W

These allow external memory devices to be added to the transputer system, to supplement the internal RAM and/or contain non-volatile code or data. A glance at the IMS B430 will show that these socket sites actually have doubled rows of pins, so that devices may be fitted *nearer* to the edge of the board - positions 'B' - or *further* from the edge of the board - positions 'W'. The **B** positions carry the appropriate transputer signals for **B**yte-wide operation, whereas the **W** positions are arranged for **W**ord-wide operation. This allows considerable flexibility. These are just some of the possibilities:

- **Code in onboard ROM:** In an application requiring the transputer to run some infrequently altered code after Reset, a single ROM could be fitted to run byte-wide as IC3B, and the IC2B position left empty.
- **Download code to RAM:** In an application where the software may be changed frequently, IC2B could be fitted with a single RAM to be run byte-wide, allowing extra capacity (beyond the transputer's 4K internal RAM) for programs to be loaded via a transputer link. The IC3B position could be left empty.
- **High-performance:** In a more demanding application, IC2W and IC3W could both be fitted with high-speed SRAMs or ROMs to create a word-wide fast-access extension to transputer memory.

Many possibilities besides these concerning device size, speed, function, etc., are available as jumper selections, allowing trade-offs of cost against performance, etc. All the options are discussed in detail in the following section on configuring the IMS B430.

5 Configuring the IMS B430

5.1 Address decoding

Two of IC4's outputs are programmed to give active-low chip enable signals for the two memory devices (IC2 B/W and IC3 B/W). The other two outputs are active-low strobes which may be connected by the user to devices installed in the prototyping area. The address ranges in which these four signals are active constitute the memory map of the transputer system. Many different maps are possible; four of the most useful have been chosen, and incorporated into the device programming equations. The map to be used is selected via jumpers JP10 and 11.

JP10 link	JP11 link	Map	Width	RAM	ROM
Present	Present	0	Word	24K + 24K	0K
Present	Absent	1	Word	0K	24K + 24K
Absent	Present	2	Byte	36K	8K
Absent	Absent	3	Byte	16K	32K

The figures for RAM and ROM sizes in the table above indicate the amount of *usable* memory in each configuration, which in some cases is not equal to the total physical capacity of the devices fitted. Note that JP10 effectively selects between the *word-wide* and *byte-wide* sets of maps. This is because JP10 not only selects the memory map at IC4 – the jumper setting is (normally) also connected to the transputer's MemBAcc pin via jumper JP14, to select the memory access mode. A byte-wide map **MUST** be used if only a single memory device is to be fitted (whether ROM or RAM), or when one ROM and one RAM are to be fitted.

Word access can **ONLY** be used if two devices of the same size and function are fitted, e.g. two 8K RAMs, or two 32K ROMs, etc. The access mode can be altered dynamically if desired, so that byte wide and word wide devices may be mixed within a single system. This is covered in the later section on technical information.

5.2 IMS B430 Memory Map options

The following sections explain the details of the four different memory mapping schemes selectable by jumpers JP10 and 11.

Caution to software writers – incomplete address decoding:

Software writers should note that because of hardware constraints, several of the map options involve 'incomplete' decoding, i.e. a single range of, for example, 8 Kbyte locations in physical storage may be accessible in two or more 8K byte non-overlapping ranges of machine addresses. If overlooked, this can lead to errors. For example, naive routines to test the amount of RAM present in a system typically check addresses in isolation, counting a location as valid if a set of test data values can be written to it and retrieved from it correctly. With incomplete address decoding, such a strategy would report valid memory in all of the multiple address ranges enabled by the decoding, causing physical storage locations to be checked and counted as valid more than once. The consequent overestimate of the amount of RAM actually present in the system could lead to storage allocation faults and confusing run-time errors. In the following discussions, warnings are given in each case involving incomplete decoding.

Map 0 word wide, 24K+24K RAM, OK ROM

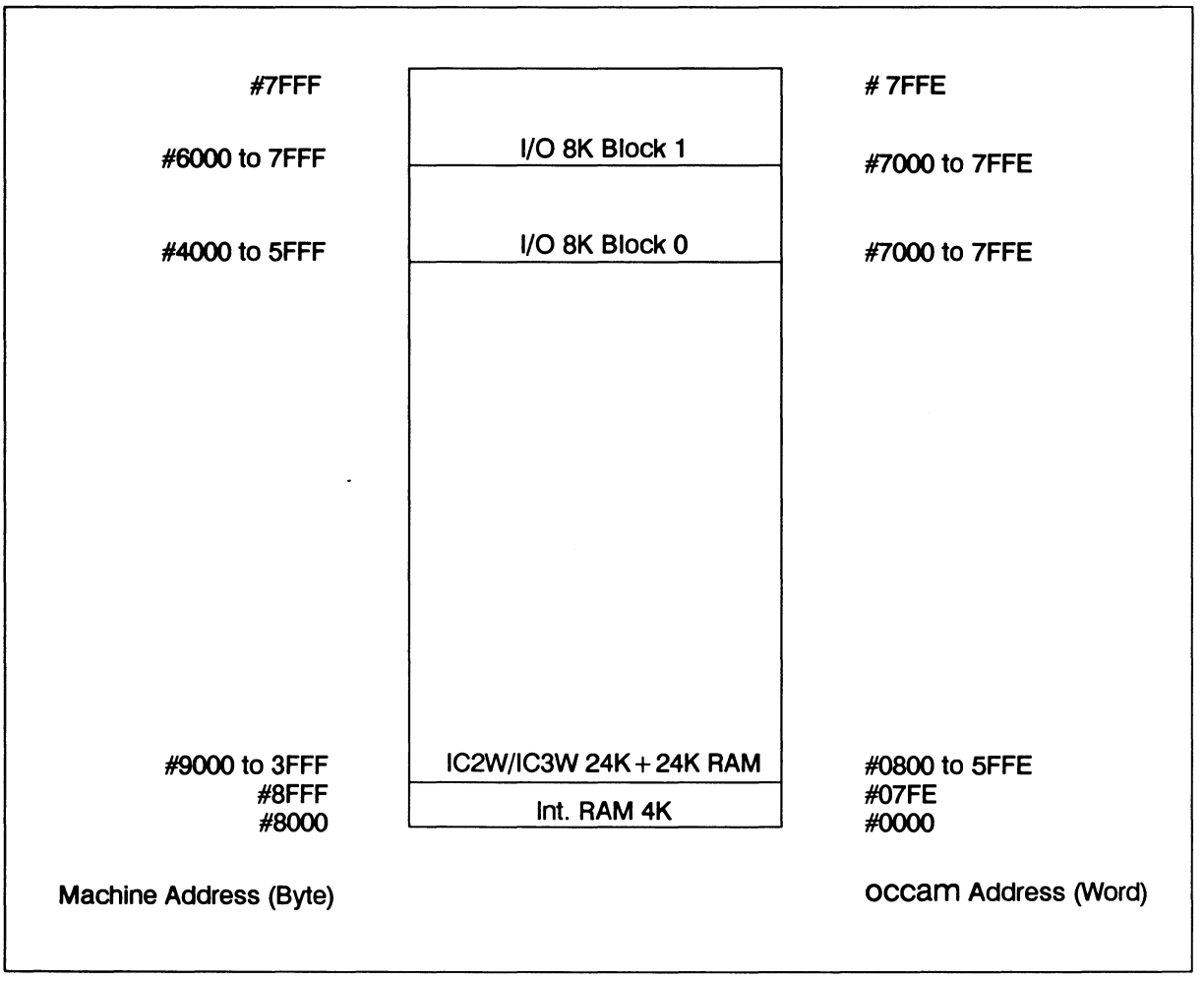


Figure 2 IMS B430 memory map 0

This map maximises the area available for word-wide RAM.

IC2W and IC3W are enabled throughout the 48Kbyte machine address range #8000 to #3FFF. Note that even numbered addresses in this area (i.e. the low byte of each word) enable IC2W, whereas odd numbered addresses enable IC3W.

The transputer's internal memory may overlay the bottom 4Kbytes of this range, and is enabled or disabled via the DisableIntRAM pin, easily set via jumper JP3.

When internal RAM is active, any value driven onto the data bus by IC2 and/or IC3 is simply ignored by the transputer. Internal RAM contents never appear at the transputer's data pins, so there is no possibility of contention with external memory.

It is anticipated that the devices fitted as IC2W and IC3W will be 32Kbyte RAMs although 8Kbyte devices may be used.

With the transputer's internal RAM disabled and with 32Kbyte devices fitted, 48K bytes of the total 64K bytes capacity are available in the machine address range from #8000 to #3FFF (except for a few locations at the bottom of memory reserved for transputer internal hardware). If 8Kbyte devices are used, the memory accessible in the 16Kbyte machine address range #8000 to #BFFF will also be accessible in the ranges #C000 to #FFFF, and #0000 to #3FFF. See the caution regarding incomplete address decoding at the beginning of this section.

With internal RAM enabled and with 32Kbyte devices fitted, the 4Kbyte machine address range from #9000 to #9FFF will access internal RAM, leaving 44K bytes of external memory accessible. With 8Kbyte devices fitted, the memory accessible in the 12K machine address range #9000 to #BFFF will also be accessible in the ranges #D000 to #FFFF and #1000 to #3FFF; also, a single 4K region of external RAM may be accessed in machine address range #C000 to #CFFF or in the range #0000 to #0FFF. See the caution regarding incomplete address decoding at the beginning of this section. The address range for which the I/O Block 1 decode signal is active includes the reset vector (at machine address #7FFE), and the transputer may be set to boot via this vector by removing the shorting link from jumper JP2. The hardware to supply the reset vector would have to be constructed in the prototyping area of the board. Note both IC2W and IC3W MUST be fitted with devices, or it will be impossible to execute code from external RAM.

To make use of this memory map, the following points should be checked:

- Set up jumpers JP10–11 to select Map 0.
- Set up jumpers JP15–27 appropriately for the device sizes to be used. Refer to the section below on memory device options for details.
- Fit memory devices in positions IC2W **and** IC3W, for word-wide operation. Take care NOT to fit the devices in IC2B or IC3B positions (for byte-wide operation) by mistake.

Map 1 word wide, OK RAM, 24K+24K ROM

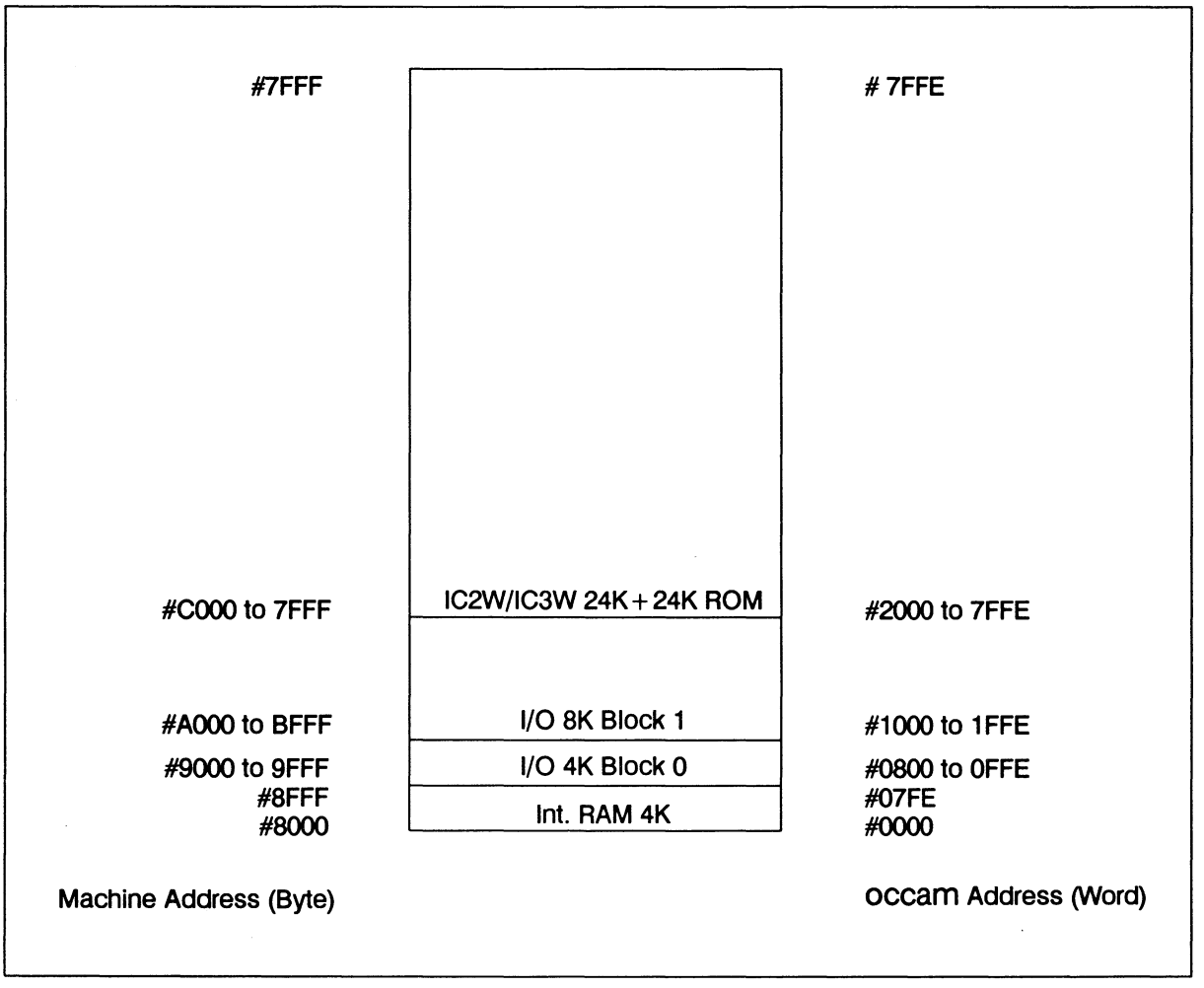


Figure 3 IMS B430 Memory map 1

This map maximises the area available for word-wide ROM.

It is anticipated that the devices fitted as IC2W and IC3W will be 32Kbyte ROMs, although 8Kbyte devices could be used. IC2W and IC3W are enabled throughout the 48Kbyte machine address range #C000 to #7FFF. Note that even numbered addresses in this area (i.e. the low byte of each word) enable IC2W, whereas odd numbered addresses enable IC3W. If 8Kbyte devices are fitted, the external memory accessible in the 16K machine address range #4000 to #7FFF will also be accessible in the ranges #0000 to #3FFF and #C000 to #FFFF. See the caution at the beginning of this section regarding incomplete address decoding.

The address range in which these devices are mapped includes the reset vector (at machine address #7FFE). The transputer may be set to boot from ROM by removing the shorting link from jumper JP2.

The I/O Block 0 enable signal is activated by machine addresses in the 8K byte range #8000 to 9FFF. The transputer's internal memory may overlay the bottom 4Kbytes of this range, and is enabled or disabled via the DisableIntRAM pin, easily set via jumper JP3. With the transputer's internal RAM disabled, the entire 8K byte machine address range from #8000 to #9FFF is available for access to external I/O devices (except for a few locations at the bottom of memory reserved for transputer internal hardware). With internal RAM enabled, external I/O hardware can only be accessed in the 4Kbyte machine address range from #9000 to #9FFF. Devices mapped in the 4K byte range below this i.e. machine addresses #8000 to #8FFF will be overlaid by the transputer internal RAM. When internal RAM is active, any value driven onto the data bus by external devices is simply ignored by the transputer. Internal RAM contents are not driven onto the transputer's data pins, so there is no possibility of contention with external devices. Note that both IC2W and IC3W MUST be fitted with devices, or it will be impossible to execute code from external ROM.

To make use of this memory map, the following points should be checked:

- Set up jumpers JP10–11 to select Map 1.
- Set up jumpers JP15–27 appropriately for the device sizes to be used. Refer to the section below on memory device options for details.
- Fit memory devices in positions IC2W **and** IC3W, for word-wide operation. Take care NOT to fit the devices in IC2B or IC3B positions (for byte-wide operation) by mistake.

Map 2 byte wide, 36K RAM + 8K ROM

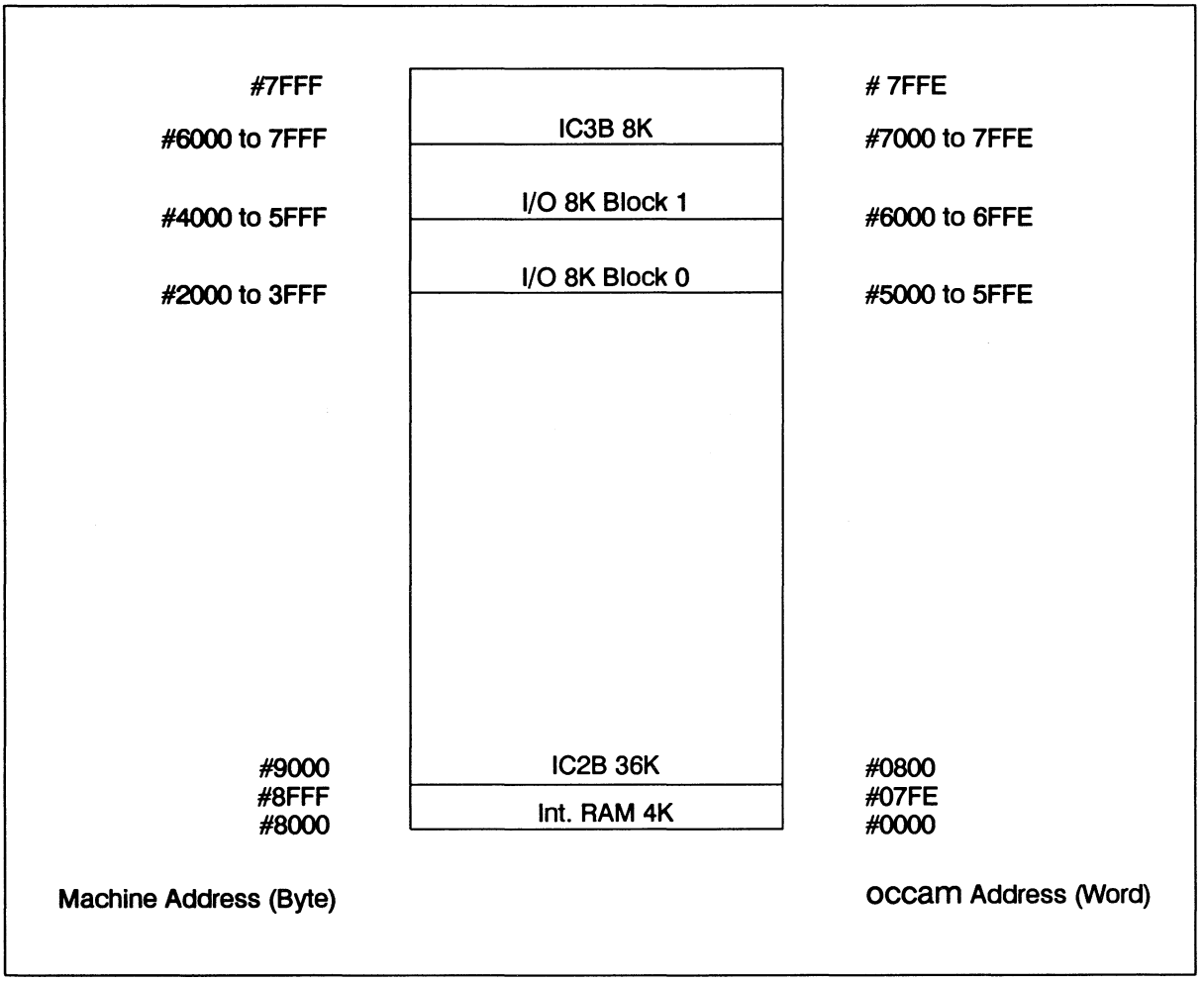


Figure 4 IMS B430 Memory map 2

This map is intended to maximise the space available for extra RAM, allowing full use of a 32K device.

IC2B is enabled throughout the 40Kbyte machine address range #8000 to #1FFF. The transputer's internal memory may overlay the bottom 4K of this range, and is enabled or disabled via the DisableIntRAM pin, easily set via jumper JP3. When internal RAM is active, any value driven onto the data bus by IC2B is simply ignored by the transputer. Internal RAM contents are not driven onto the transputer's data pins, so there is no possibility of contention with external memory.

It is anticipated that the device fitted as IC2B will be a 32Kbyte RAM, although an 8Kbyte device could be used.

The bottom 4K area of IC2B may be overlaid by the transputer's internal RAM, if this is enabled. However, the same physical area of the device is also accessed by machine addresses in the range #0000 to 0FFF, so use of the full 32K is retained.

The 4K area of external RAM mapped immediately above the internal RAM, i.e. at machine addresses #9000 to #9FFF, will also be accessed by machine addresses in the range #1000 to 1FFF. Similarly, if internal transputer RAM is disabled, the 4K area of RAM at machine addresses #8000 to 8FFF will also be accessed at addresses #0000 to 0FFF.

In either case, the entire capacity of a 32K device will be accessible between machine addresses #9000 and #0FFF. If an 8K device is used, the 8K range of the device will always be accessible in machine address ranges #C000 to #DFFF, #E000 to #FFFF, and #0000 to #1FFF. If transputer internal RAM is disabled, the

device will also be accessible in the 8K machine address range #8000 to #9FFF. See the caution to programmers at the beginning of this section with regard to partial decoding.

It is anticipated that the device fitted as IC3B will be an 8K ROM. (A 32Kbyte device could be fitted, but the useable space would still only be 8Kbyte.) The address range in which this device is mapped includes the reset vector (at machine address #7FFE). The transputer may be set to boot from ROM by removing the shorting link from jumper JP2. Some applications may require extra RAM but need no ROM. In such cases, there is no need to fit a device in IC3B. Similarly, for applications which need ROM but no extra RAM, IC2B can be left empty. To make use of this memory map, the following points should be checked:

- Set up jumpers JP10–11 to select Map 2.
- Set up jumpers JP15–27 appropriately for the device sizes to be used. Refer to the section below on memory device options for details.
- Fit memory devices in positions IC2B and/or IC3B, for byte-wide operation. Take care NOT to fit the devices in IC2W or IC3W positions (for word-wide operation) by mistake.

Map 3 byte wide, 16K RAM + 32K ROM

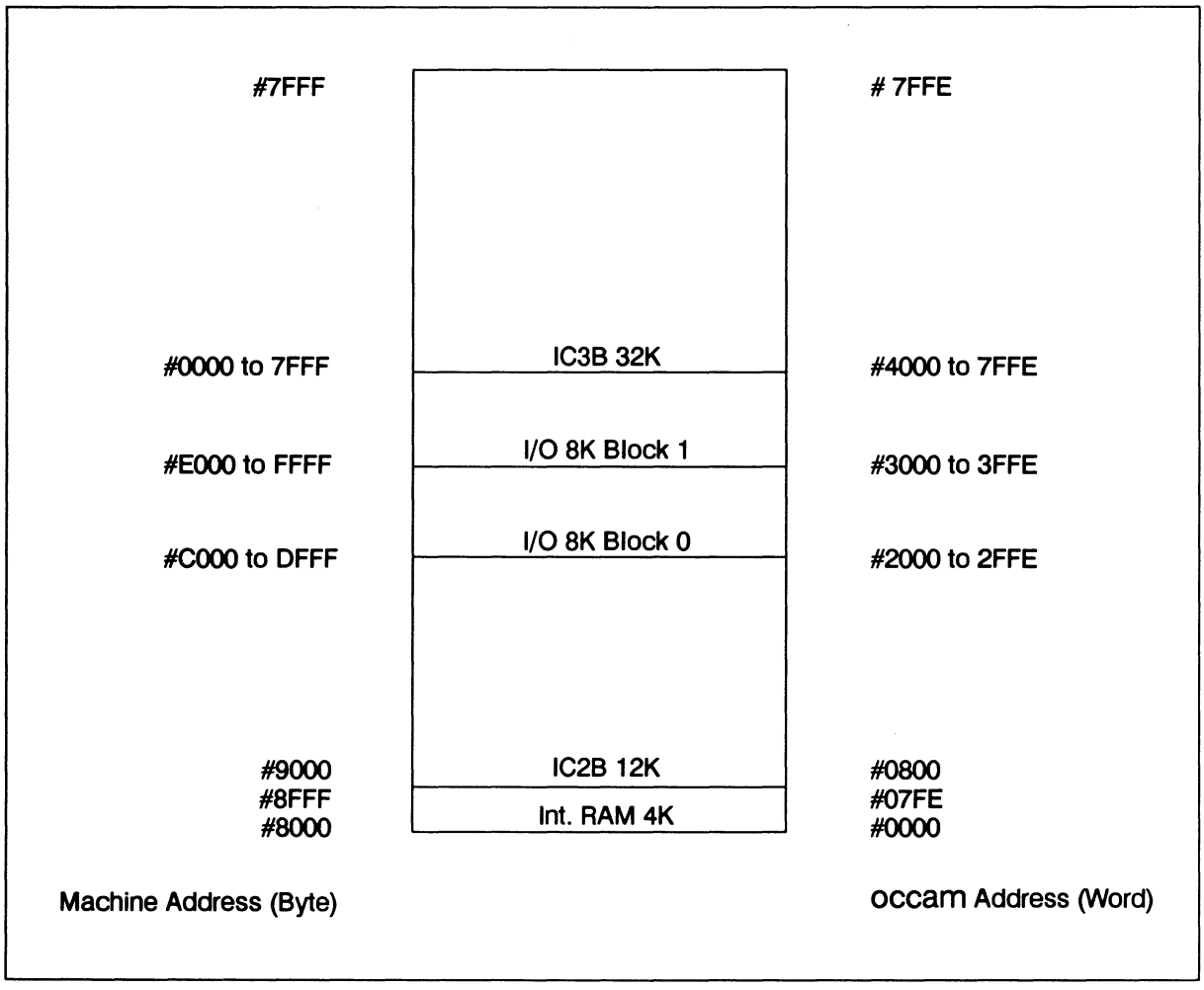


Figure 5 IMS B430 Memory map 3

This map maximises available ROM space, allowing full use of a 32Kbyte device.

IC2B is enabled throughout the 16Kbyte machine address range #8000 to #BFFF. The transputer's internal memory may overlay the bottom 4K of this range, and is enabled or disabled via the DisableIntRAM, easily set via jumper JP3. When internal RAM is active, any value driven onto the data bus by IC2B is simply

ignored by the transputer. Internal RAM contents are not driven onto the transputer's data pins, so there is no possibility of contention with external memory.

It is anticipated that the device fitted as IC2B will be an 8K or 32Kbyte RAM (since 16K types are not commonly available). With a 32K device fitted and the transputer's internal RAM disabled, half of the device capacity can be used (except for a few locations at the bottom of memory reserved for transputer internal hardware). With internal RAM enabled, only 12K of IC2B will be usable. With an 8K device fitted and the transputer's internal RAM disabled, the entire device will appear twice in the memory map, i.e. machine addresses in the 8K range #8000 to #9FFF will access the same physical storage locations as addresses #A000 to #BFFF. With internal RAM enabled, the 4K machine address range from #9000 to #9FFF will access the same physical storage locations as addresses B000 to #BFFF. See the caution to programmers at the beginning of this section with regard to partial decoding.

It is anticipated that the device fitted as IC3B will be an 8K or 16K ROM. The address range in which this device is mapped includes the reset vector (at machine address #7FFE). The transputer may be set to boot from ROM by removing the shorting link from jumper JP2. Some applications may require extra RAM but need no ROM. In such cases, there is no need to fit a device in IC3B. Similarly, for applications which need ROM but no extra RAM, IC2B can be left empty. To make use of this memory map, the following points should be checked:

- Set up jumpers JP10–11 to select Map 3.
- Set up jumpers JP15–27 appropriately for the device sizes to be used. Refer to the section below on memory device options for details.
- Fit memory devices in positions IC2B and/or IC3B, for byte-wide operation. Take care NOT to fit the devices in IC2W or IC3W positions (for word-wide operation) by mistake.

5.3 Wait state generator

Half of IC4 is used to implement a programmable wait state generator. As mentioned previously, the transputer is capable of transferring data values much faster than most low-cost memory and support devices, requiring only two cycles of the ProcClockOut signal to complete each transfer operation. The transputer may therefore be caused to defer termination of external data transfers for one or more 'wait states', i.e. additional cycles of the ProcClockOut signal. With the 20MHz transputer supplied on the IMS B430, each wait state inserted into a data transfer extends it by 50ns. To extend all the transputer's external data transfer cycles by a fixed amount, the user need only fit and/or remove the appropriate links at jumpers JP8 and 9. The checklist at the end of this section includes a table, relating device access times to the appropriate jumper settings. The number of wait states can be selected dynamically, so that data transfers are extended only by the minimum amount required for each device. However, this is not possible with the existing wait state generator, for reasons explained in the later section giving technical information on the IMS B430.

5.4 Memory size and type options

The IMS B430 can accept almost any memory devices conforming to the JEDEC standard 28-pin format in the IC2B/W and IC3B/W sockets. The possibilities include 8K and 32K static RAMs, 8K,16K and 32K ROMs, and 8K and 32K electrically alterable ROM devices of various technologies. The signals required at certain pins on these sockets must be chosen to suit the device type and size to be fitted. For example, a write enable signal must be fed to RAM devices, whereas EPROMs can make no use of it. Similarly, a 32Kbyte device can accept up to 15 address lines, whereas an 8K can only use up to 13 lines. The situation is further complicated by differences in pin assignment between ROMs and RAMs, even for devices of the same capacity.

To accommodate all the necessary options, the signals and pins involved can be connected in the appropriate configurations by jumpers JP15–18 (for IC2B/W) and jumpers JP19–23 (for IC3B/W). The following configurations are available.

Data width selection

Jumpers JP24 and 25 must be set to reflect whether devices have been fitted in IC2B and/or IC3B (Byte wide operation), or in IC2W and IC3W (Word wide operation). These settings route the correct high-order address lines to the other memory option jumpers - refer to the IMS B430 circuit diagram and the jumper/ signal pad layout drawing for clarification.

Data Width	JP24 link (A13/14)	JP25 link (A14/15)
Word	Up (A14)	Up (A14)
Byte	Down(A13)	Down(A13)

Note: 'down' and 'up' in the above table assume the board to be viewed from the component side, with the small yellow triangle at the top left corner.

IC2B/W options

Jumpers JP15-18 must be set to reflect whether the device fitted as IC2B is of 8Kbyte or 32Kbyte capacity, and whether it is a RAM or a ROM. The settings shown are for correct for standard JEDEC pinout devices. If in doubt, check the device pinout against the signals and pin numbers given in the second and third rows of the table respectively. Note that all pins are tied to VCC via resistors, so that if all jumpers relating to a pin are removed it will be pulled up.

Device	JP15 link	JP16 link	JP17 link	JP18llink
(From) (To)	(nMemWrB0) (Pin 27)	(A14/15) (Pin 27)	(A13/14) (Pin 26)	(A14/15) (Pin 1)
8K RAM	Fit	Remove	Remove	Remove
8K ROM	Remove	Remove	Remove	Remove
32K RAM	Fit	Remove	Fit	Fit
32K ROM	Remove	Fit	Fit	Remove

Jumper JP26 will normally have a shorting link fitted, so that the chip enable signal at pin 20 of the device is also connected to the output enable (pin 22). If desired, the user may remove this link and apply an alternative output enable signal at THP21.

IC3B/W options

Jumpers JP19 and 20 must be set to reflect whether the device fitted as IC3B/W is a ROM or a RAM, and in the latter case whether it is operated byte- or word-wide.

Device	JP19 link	JP20 link
(From) (To)	(nMemWrB1) (Pin 27)	(nMemWrB0) (Pin 27)
RAM (byte)	Remove	Fit
RAM (word)	Fit	Remove
ROM (b/w)	Remove	Remove

Jumpers JP21 to 23 must be set to reflect whether the device fitted as IC3B/W is of 8Kbyte or 32Kbyte capacity, and whether it is a RAM or a ROM. The settings shown are for correct for standard JEDEC pinout devices. If in doubt, check the device pinout against the signals and pin numbers given in the second and third rows of the table respectively. Note that all pins are tied to VCC via resistors, so that if all jumpers relating to a pin are removed it will be pulled up.

Device	JP21 link	JP22 link	JP23 link
(From) (To)	(A14/15) (Pin 27)	(A13/14) (Pin 26)	(A14/15) (Pin 1)
8K RAM	Remove	Remove	Remove
8K ROM	Remove	Remove	Remove
32K RAM	Remove	Fit	Fit
32K ROM	Fit	Fit	Remove

Jumper JP27 will normally have a shorting link fitted, so that the chip enable signal at pin 20 of the device is also connected to the output enable (pin 22). If desired, the user may remove this link and apply an alternative output enable signal at THP22.

5.5 IMS B430 jumper setup checklist

This section is presented in the form of a questionnaire. The intention is that the designer can work through the sheet in order, marking off the answers to the questions about the intended application. This will result in a complete list of the option settings needed, useful for future reference.

The options marked 'default' indicate the settings of the IMS B430 as originally shipped.

JP1: Event. Will the transputer's Event pin be driven by an externally generated signal, to synchronise the program to some real-time occurrence?

NO: (default) fit the shorting link at JP1. Leave THP2 unconnected.

YES: remove the shorting link at JP1, and wire the external signal to THP2. Note that an appropriate software Event handler process must be included in the transputer program.

JP2: Boot. Will the transputer be required to download bootable code from one of its serial links after Reset, or to boot into code stored in a suitably programmed onboard ROM?

BOOT FROM LINK: (default) Fit the shorting link at JP2. Leave THP3 unconnected.

BOOT FROM ROM: Remove the shorting link from JP2. (It is possible to drive this signal at THP3 if desired, although such dynamic alteration of boot behaviour would only be useful in very specialised applications).

JP3: Int RAM. Will the transputer's internal RAM be enabled or disabled?

ENABLED: (default) Fit the shorting link at JP3. Leave THP4 unconnected.

DISABLED: Remove the shorting link at JP3. (It is possible to drive this signal at THP4 if desired, although such dynamic alteration of internal RAM usage would only be useful in very specialised applications).

JP4: MemReq. Will the transputer's MemReq pin be driven by an externally generated signal, to allow external hardware to gain control of the system address, data and control lines?

NO: (default) Fit the shorting link at JP4. Leave THP5 unconnected.

YES: Remove the shorting link at JP4. Connect an appropriate signal at THP5. Note that the signal is active HIGH at the transputer - if the shorting link is removed and no appropriate signal connected, the transputer will be unable to perform any external data accesses.

JP5,6,7: ProcSpeedSel. Is the transputer fitted to the board a T222 (as shipped), or a T225 (or other upgrade)?

T222: (default) Fit the three shorting links at all three jumpers, JP5 through JP7.

T225(etc): Refer to the appropriate device engineering data for clock speed options available, then fit/remove shorting links from JP5 through JP7 as appropriate. Note that fitting of a shorting link connects the respective signal to Ground, i.e. logic low.

JP8,9: WaitSel. Are wait states required for reliable operation of devices fitted to the board?

The following table relates the external device access time to the number of wait states required, and the corresponding settings of JP8 and 9.

Access time (ns)	Waits	Total cycles	JP8 link	JP9 link
less than 31	0	2	Present	Present
31 to 81	1	3	Absent	Present
81 to 131	2	4	Present	Absent
131 to 181	3	5	Absent	Absent

Note: the default setting is JP8 link Absent, JP9 link Absent i.e. 3 wait states selected. The number of wait states should be set to accommodate the slowest device in the system. If devices are installed in the prototyping area, be sure to check the access time they require in the manufacturer's data.

JP10,11: MapSel Which of the available memory maps will be used? Refer to the previous section for details of the map options.

The following table is reproduced here for convenience:

JP10 link	JP11 link	Map	Width	RAM	ROM
Present	Present	0	Word	24K + 24K	0K
Present	Absent	1	Word	0K	24K + 24K
Absent	Present	2	Byte	36K	8K
Absent	Absent	3	Byte	16K	32K

Note: that the default setting is JP10 link Absent, JP11 link Present i.e. Map 2 selected (to suit the single 32Kbyte RAM device fitted as IC2B).

JP12 (None)

JP13: MemBAcc Will the transputer's access mode (Word or Byte) be governed by the Map selection at jumpers JP10,11 or driven by an externally derived signal?

SELECTED WITH MAP: (default) Fit the shorting link at jumper JP13. Leave THP7 unconnected.

DRIVEN: Remove the shorting link from JP13 and connect the signal to THP7. Note that a HIGH level at the transputer selects BYTE access, and that the signal must meet the timing specified in the transputer engineering data.

JP14: MemWait Will insertion of wait states be governed by the selection at jumper JP8,9 or will it be controlled by an externally generated signal?

SELECTED BY JUMPERS: (default) Fit the shorting link at jumper JP14. Leave THP8 unconnected.

DRIVEN: Remove the jumper at JP14, and wire the external signal to THP8. Note that the signal must meet the timing specified in the transputer engineering data.

JP15-18: IC2B/W What type and size of device will be fitted as IC2?

Refer to the previous subsection for details of the options available. Check that any device to be fitted conforms to the Jedec standard pinout for its size and function.

JP19-23: IC3B/W What type and size of device will be fitted as IC3?

Refer to the previous subsection for details of the options available. Check that any device to be fitted conforms to the Jedec standard pinout for its size and function.

JP24,25: Data width Will the memory sockets IC2B/W and IC3B/W be operated Byte-wide (device(s) fitted in either or both B positions) or Word-wide (devices fitted in W positions - must be TWO devices)?

Viewing the board from the component side, with the yellow triangle at the top left corner:

BYTE: (default) Fit both the jumper links in the 'down' position.

WORD: Fit both the jumper links in the 'up' positions.

JP26,27: Output Enables Will the device(s) fitted as IC2 and/or IC3 be output-enabled by the chip-enable signals, or be driven by externally generated signals, to control the output drive of the device(s)?

CHIP ENABLE: (default) Fit the shorting links at jumpers JP26 and/or JP27 as appropriate. Leave THP21 and/or THP22 unconnected.

DRIVEN: Remove the shorting links from jumpers JP26 and/or JP27, and wire appropriate signals to THP21 and/or THP22.

6 IMS B430 Technical Information

This section gives full explanations of the IMS B430 hardware functions. It assumes thorough familiarity with hardware design terminology in general, and reasonable knowledge of programmable logic devices. The attention of the designer is also drawn to certain aspects of transputer operation which are easily overlooked by those unfamiliar with the device.

6.1 Wait state generator

Description

Three macrocells in IC4 are arranged as a shift register/state machine, clocked by the transputer's ProcClockOut signal and synchronised to its notMemCE signal. A fourth macrocell acts as a multiplexer. This selects the output of any one of the three shifter stages, or is forced to logic low, under control of the inputs Waitsel0 and Waitsel1 (at pins 6 and 7). The transputer sets notMemCE active i.e. low near the beginning of each external data transfer. This leading edge on notMemCE is clocked through the shift register stages by the rising edge of ProcClockOut, appearing at the outputs after delays of one, two, and three clock cycles.

By feeding this multiplexer output to the transputer's MemWait pin via jumper JP14, memory cycles can be extended by zero, one, two, or three 'wait states' - i.e. cycles of the transputer's ProcClockOut signal. The outputs of the three shift register stages are available at pads THP15-17. When interfacing peripheral devices which are not directly compatible with the transputer in terms of their timing characteristics, availability of these shift register outputs can simplify the additional logic needed to generate appropriate control signals.

Note: Beware of confusion: the transputer's MemWait pin is active high, i.e. inserts wait states when set to a *high* logic level. Most microprocessors have a similar pin function, but operate in the opposite logic sense, i.e. the pin is pulled low to insert wait states, then set high again for execution to proceed. If the transputer's MemWait pin is set high, the device will wait indefinitely – it will not execute any code, nor will it respond to any serial Link activity. With the 20MHz transputer supplied, each added wait state extends the memory cycle by 50ns.

Timing Analysis

The following paragraphs give a full analysis of external memory cycle timing on the IMS B430, to give a full picture of the constraints on memory timing. Most of the discussion is equally applicable to support devices.

The T222-20 supplied with the IMS B430 is capable of accessing external devices in only two cycles of ProcClockOut, i.e. in 100ns. The data sheet guarantees that the notMemCE signal will be active (i.e. low) for a minimum of 68ns. During a read cycle, the transputer requires data to be set up at least 22ns before the end of this 68ns period. Therefore, the read data must be presented to the transputer (68 - 22) = 46ns after notMemCE becomes active, at the latest. The external device will not begin its access until enabled by notMemCE propagating through IC4 to its chip enable. Since the device supplied as IC4 has a maximum propagation delay of 15ns, the external device access time must be 46 - 15 = 31ns or faster. Similarly, in write cycles the transputer guarantees data to be available during a 50ns period before notMemCE ceases to be active, but may remove it only 5ns after this. Most SRAM devices will terminate a write cycle as soon as either the write signal or the chip enable signal go false (but note that this is not always true of I/O devices). On the IMS B430, the notMemWrB transputer signals are wired directly to the device write enables of IC2b/w and IC3b/w, whereas the device chip enables are derived via IC4 (and are therefore terminated later). Therefore, the beginning of a write cycle is governed by the the falling edge of notMemCE delayed via some address decoding, and the end of the write cycle by the notMemWrB strobes. The guaranteed write data setup time without wait states is only 68ns less the address decode delay, and the hold time only 5ns.

At the time of writing, memory devices which can operate under these timing conditions are relatively expensive, and many support functions simply are not available in versions which can run this fast. Therefore, in the context of a prototyping TRAM, the wait state generator is likely to be of use in many designs. With a device programmed as supplied, each memory cycle can be extended by up to three wait states, i.e. by up to an additional 150ns with the 20 MHz transputer supplied. Thus, devices with read access times of up to 31 + 150 = 181ns can be accommodated. Write data setup time can be extended to 50 + 150 = 200ns. Write data hold time can be extended up to 5 + 150 = 155ns, by using some extra logic (possibly based on the shift register outputs) to terminate the write cycle at the external device *before* it terminates at the transputer. Note that this involves trading off some setup time, since the combined extensions to the setup and hold time cannot exceed the total 150ns extension available.

Extra logic could be added to insert even more wait states if necessary, e.g. by adding more stages to the existing shift register. In this case, the shorting link should be removed from jumper JP14 and the user-derived signal connected to the transputer's MemWait pin via THP8.

Dynamic control of wait states

In many systems, there will be a mix of fast and slow external devices. In such cases, it is clearly desirable to switch the number of wait states to match each device as it is addressed, inserting only the minimum delay into each data transfer for reliable operation. Because of cost and space constraints, the wait state generator implemented on the IMS B430 as shipped is a simple synchronous design. It is NOT suitable for this form of operation.

To understand this, note that if the transputer is required to extend a data transfer, its MemWait signal must be asserted - at the latest - 25ns after the rising edge of the ProcClockOut signal which is the reference for the beginning of the transfer. The wait state multiplexer output is configured as a synchronous output, to allow implementation within the 16R4 PAL architecture. Therefore (assuming Waitsel1 and 0 to have stabilised with adequate setup time) it will drive the MemWait pin high no more than 12ns (the device clock to output propagation delay) after the rising edge of ProcClockOut, leaving a margin of 13ns.

The requirement that Waitsel1 and Waitsel0 are set up relative to the rising edge of ProcClockOut means that they cannot be a driven by logic functions of the address lines, since the address value to be accessed in the forthcoming cycle is not available this early in the transfer. This makes it impossible to set the number of wait states according to the device to be accessed - by the earliest point in the transfer at which the required number of wait states can be deduced, the WaitSel lines should already have been set up. This is why the IMS B430 wait state generator must effectively be used with preset levels on the WaitSel lines, inserting the same extension into ALL the transputer's external data transfers. A somewhat altered pro-

gramming configuration could overcome this limitation. The multiplexer output would be configured as a combinatorial (rather than registered) function. It could then be arranged to activate the MemWait pin by decoding appropriate combinations of address value and shift register output. However, the signal at the transputer's MemWait pin could not be guaranteed valid until (a) the shift register outputs had stabilised after the rising edge of ProcClockOut (clock to output delay,) and (b) these new shift stage output values had propagated through the combinatorial multiplexer (combinatorial device delay). With the 15ns device supplied, the shifter outputs may not be stable until 12ns after the rising edge of ProcClockOut, and the multiplexer output may not be valid until after a further 15ns, i.e. 27ns in total after the edge on ProcClockOut - 2ns *outside* the specified limit. Therefore, the programmable logic device would have to be of a higher speed grade than the 15ns part supplied. Also, it would be necessary to use a device with a more flexible architecture such as a GAL16V8, which (in contrast to the PAL16R4) would allow the multiplexer output to be reassigned as a combinatorial function without any pinout change, or alternatively to reassign this output to one of the other pins (e.g. by sacrificing one of the I/O block decodes).

As a final note, the designer may prefer to control insertion of wait states via a counter, rather than a shift register. This allows N output macrocells to count 2 to the power N wait states, as compared with N wait states for the shift register approach. However, it becomes necessary to decode the counter output to generate timing signals in extended data transfers, rather than merely connect to the appropriate shift register output. To assist in this, it is recommended that a Gray coded (rather than binary-coded) counter be used - this has the property that only one output changes at each clock transition, making it easier to avoid glitches at the decoded MemWait output by inserting logically redundant 'cover' terms.

6.2 Word and Byte access modes

The following paragraphs explain the distinctions between these modes, drawing the designer's attention to some of the finer points involved.

Word access mode

When operating in word access mode,i.e. with the MemBAcc pin set at logic low, the transputer expects the external device to be capable of moving a full sixteen-bit wide value in each data transfer cycle. However, to take advantage of this fast operation, the memory system or I/O device must also be capable of transferring sixteen-bit data values. Most low-cost memory devices are only able to transfer eight-bit wide values, so the usual price of obtaining this high performance is that TWO devices must be operated side by side, each connected to one half of the data bus.. Note also that in word-access mode, no use is made of the transputer's address line A0, since only even-numbered addresses are presented on the address bus. The transputer expects to access pairs of bytes simultaneously, even if the operation in progress only requires one of the bytes. In the case of byte reads, The unwanted byte is discarded internally by the transputer. In the case of byte writes, only one of the notMemWrB strobes is activated, so the unwritten device will perform a redundant read access which the transputer ignores. A consequence of this is that the address lines of memory devices to be operated word-wide are shifted relative to the transputer's address lines - i.e. rather than wiring memory A0 to transputer A0 (which does nothing), memory A0 should connect to transputer A1. Similarly, memory A1 connects to transputer A2, etc.

Byte access mode

When operating in byte access mode, i.e. with the MemBAcc pin set at logic high, the transputer expects the external device to be capable of transferring only eight-bit wide values in each data transfer cycle, i.e. to have an 8-bit data bus connected to the transputer's Data 7 through Data 0 lines; logic internal to the transputer automatically splits operations involving sixteen-bit values (including instruction fetches) into pairs of byte-wide operations. The first byte addressed is always even-numbered, i.e. with address line A0 at logic low, and the second byte the immediately consecutive odd-numbered byte, i.e. with all address bits unchanged except that A0 is now high. In this mode, the memory address lines are connected to the transputer address lines with the same significance, i.e. memory A0 connects to transputer A0, memory A1 to transputer A1, etc.

This makes it possible to execute code from a single byte-wide memory, with the penalty of doubling the time required for each operation involving the external device. In view of the explanation above, it should be clear that byte access **must** be used when only one memory device is fitted, or when both a ROM and a RAM are to be added to the system (since they will always be enabled one at a time).

Dynamic control of Word/Byte access

Systems may include a mix of word-wide and byte-wide devices. When this is done, the shorting links should be removed from jumper JP13, and an appropriate logic signal connected to the transputer's MemBAcc pin via THP7. The timing characteristics of this signal should meet the requirements of the T222 engineering data. Note in particular that to guarantee entry into byte access mode, the signal should be asserted (i.e. set high) throughout BOTH byte-wide operations of a Byte access mode transfer, and through any wait states which may be inserted to extend the byte-wide operations.

6.3 Byte operations in transputer systems

When running in byte access mode, the transputer *always* accesses two adjacent bytes, one after the other – even for operations which actually manipulate single bytes, such as the load byte instruction, lb. The first of the adjacent byte addresses will be even-numbered, i.e. with address line A0 low, and the second will be odd-numbered, i.e. with address line A0 high. A similar situation exists when the transputer is operated in word access mode. Since the notMemCE signal governs the timing for both halves of the data bus, it is impossible to distinguish externally between byte and word read operations. In the case of byte writes, only one of the notMemWrB strobes will be active – but for the other half of the data bus, the operation is indistinguishable from a read in terms of transputer signal activity. When reading or writing a single byte, there will therefore always be an accompanying redundant read of the byte immediately above or below that which is actually supposed to be accessed (depending whether the latter has an even or odd numbered address).

This unintentional read cannot be suppressed, though ordinarily it has no adverse effect on system operation. For example, a memory device which experiences such a redundant read will still contain the same data after the operation as it did before, so its function is not impaired. However, a problem can arise with certain peripheral devices, including some types of interface controllers. These contain status flags which are automatically cleared internally after being read.

If two successive byte addresses (the first even, the second odd-numbered) were mapped to contain flags of this type, an attempt to access ONE of the addresses – with a byte operation – would actually read the other address also. The status flag(s) mapped at this adjacent address would therefore be cleared internally by the peripheral device, and the information obtained in the redundant read discarded within the transputer. The unintended reading of the address would in fact destroy the information contained in the status flags before the transputer had made any use of it. As a result, critical interface conditions might go undetected, characters be lost, etc.

To avoid this, there are two possible strategies. Firstly, the memory map can be arranged to separate the addresses of such flags by at least *two* bytes, so that it is impossible for a single operation to affect both locations. Alternatively, the software may be made subject to the constraint that the sensitive address(es) are always accessed with word (rather than byte) instructions, so that an entire sixteen-bit value is always read or written.

7 Mechanical details

Figure 6 shows the outline and dimensions of a single IMS B430 TRAM.

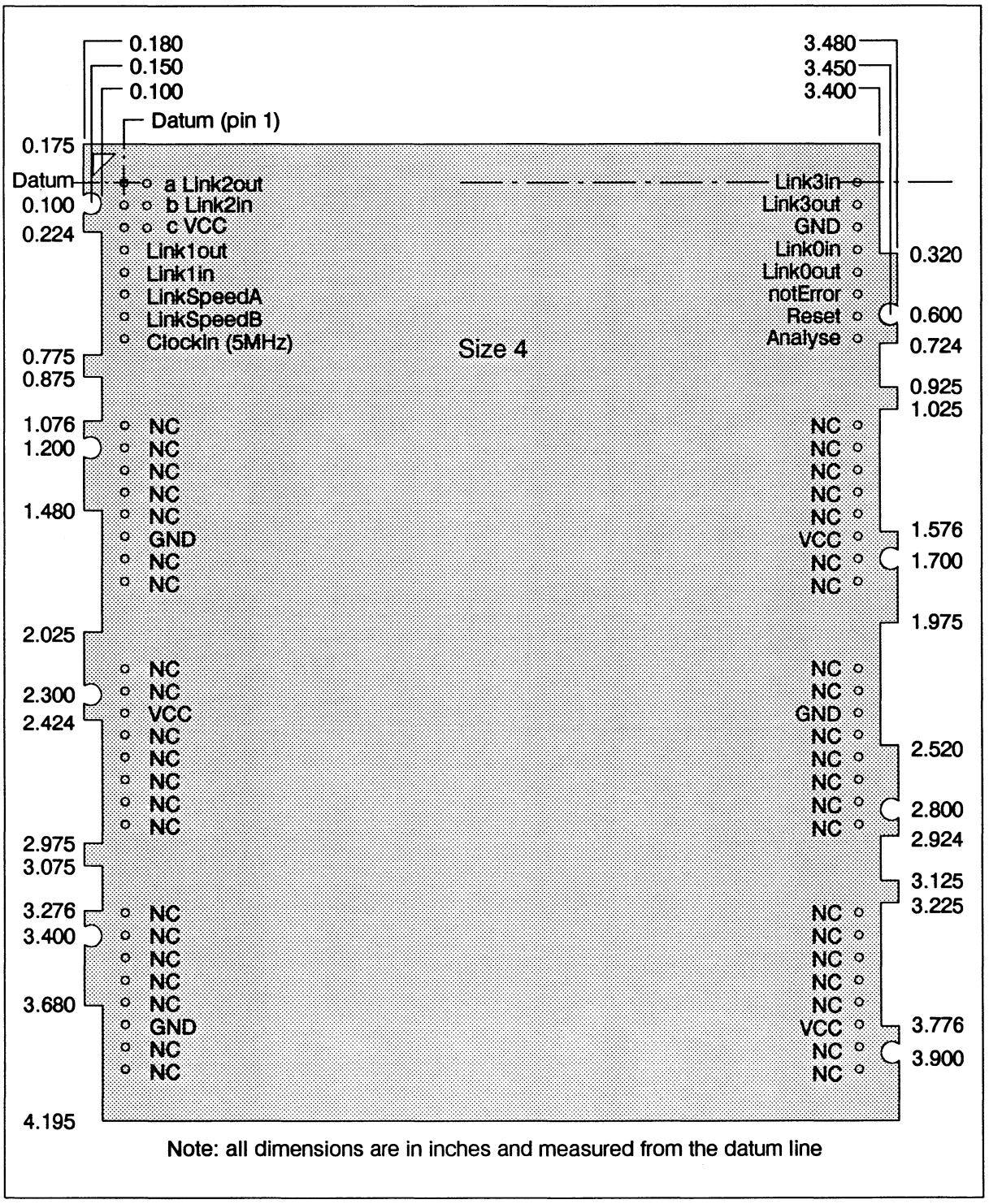


Figure 6 IMS B430 outline drawing (All dimensions in inches)

8 Installation

Since the IMS B430 contains CMOS components, all normal precautions to prevent static damage should be taken.

The IMS B430 is supplied with spacer pin strips attached to the TRAM pins on the underside of the board. These spacers perform two functions. Firstly, they help to protect the TRAM pins during transit. Secondly, they can be used to space the TRAMs off the motherboard. If there are no components mounted on the motherboard TRAM slot, then the spacer strips should be removed before the TRAM is inserted.

Plug the IMS B430 carefully into the motherboard. Where the IMS B430 is being used with an INMOS motherboard, the silk screened triangle marking pin 1 on the IMS B430 (see figure 6) should be aligned with the silk screened triangle that appears in the corner of the appropriate TRAM slot. If it is envisaged that the assembly is likely to be subjected to any vibrations, it is recommended that the TRAM is secured to the motherboard using nylon M3 nuts and bolts. The bolts should be inserted through the fixing holes on the motherboard, and through the castlations on two edges of the TRAM. A number of these nuts and bolts are supplied with each of the INMOS motherboards.

Should it be necessary to unplug the IMS B430, it is advised that, having removed any retaining nuts and bolts, it is gently levered out while keeping it as flat as possible. As soon as the IMS B430 is removed, the spacer pin strips should be refitted to the TRAM to protect the pins.

9 Specification

TRAM feature	IMS B430	Units	Notes
Transputer type	IMS T222-20		
Number of transputers	1		
Number of INMOS serial links	4		
Amount of RAM	4–64	KBytes	
Memory Wait States	Prog'able 0–3		
Memory cycle time	100	ns	
Subsystem controller	No		
Peripheral circuitry	Prototype area		
Memory Parity	No		
Size (TRAM Size)	4		
Length	3.66	Inch	
Pitch between pins	3.30	Inch	
Width	4.35	Inch	
Component height above PCB	9.2	mm	
Component height below PCB	3.7	mm	1
Weight	–	g	
Storage temperature	0–70	°C	
Operating temperature	10–40	°C	2
Power supply voltage (VCC)	4.75–5.25	Volt	
Power consumption	1	Watt	3

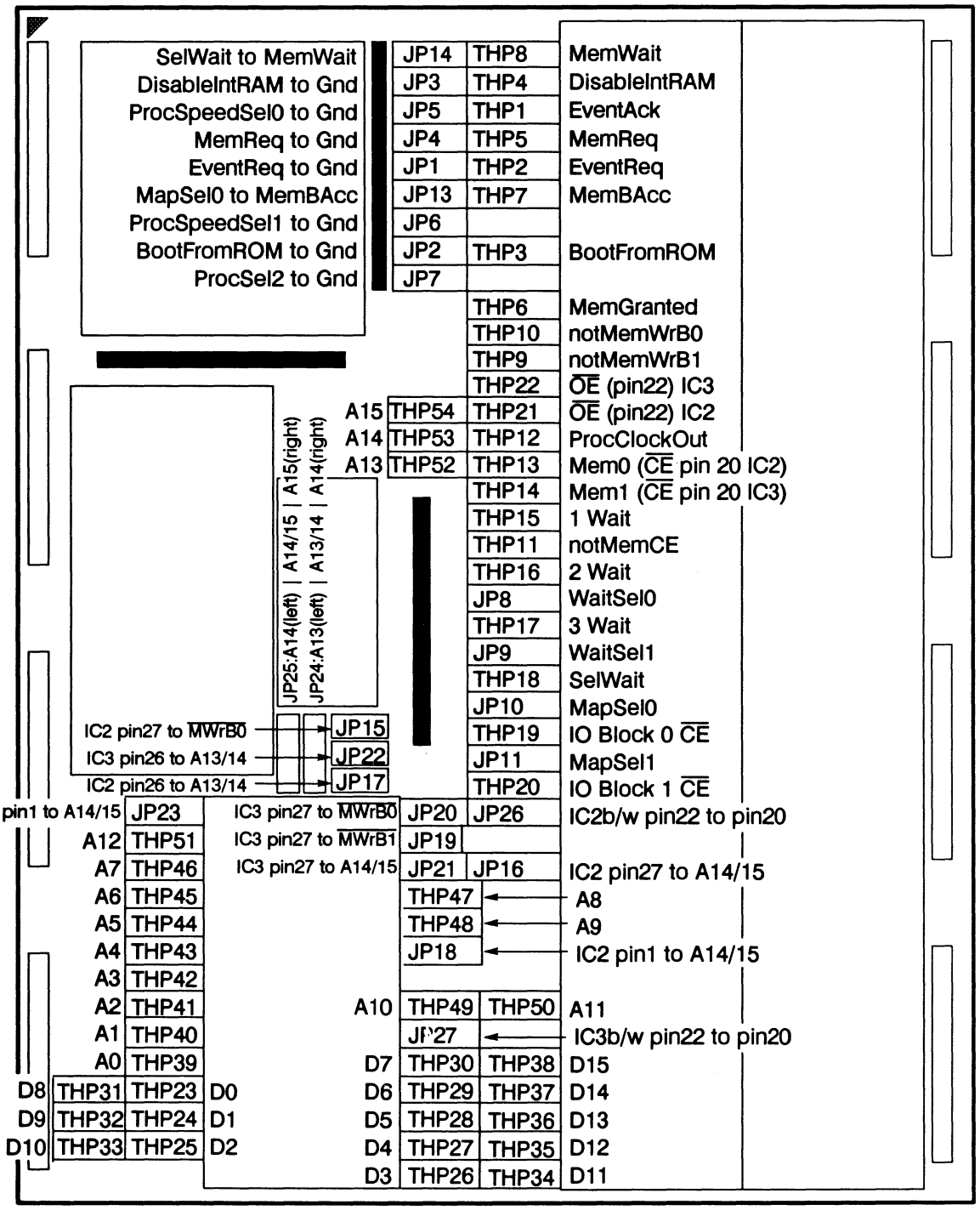
Notes

1 This dimension includes the thickness of the PCB.

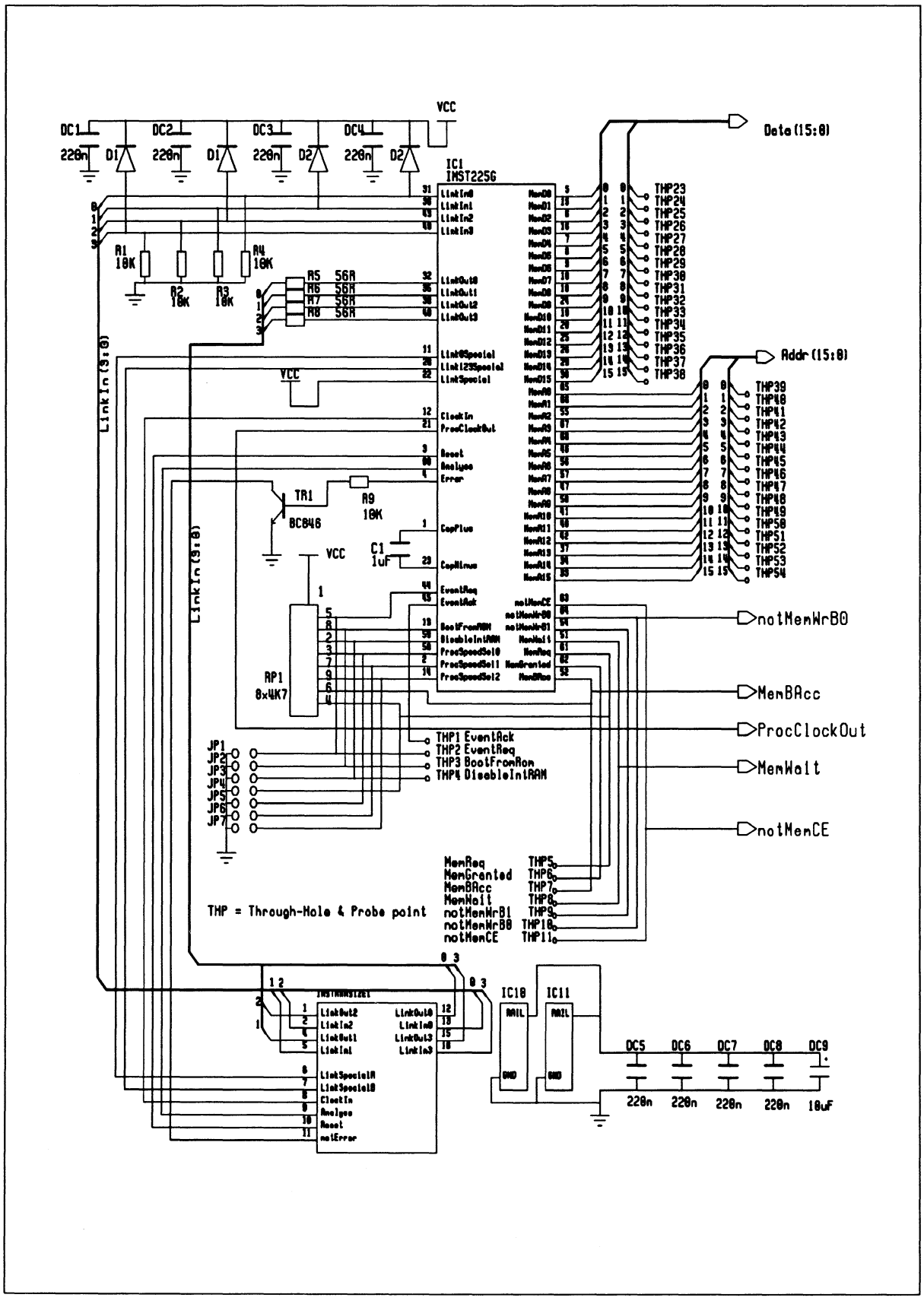
2 The figure quoted refers to the ambient air temperature.

3 The power consumption is the worst case value obtained when a sample of IMS B430 TRAMs was tested (running a program that utilised all four links and accessed memory simultaneously) at a supply voltage (VCC) of 5.25V.

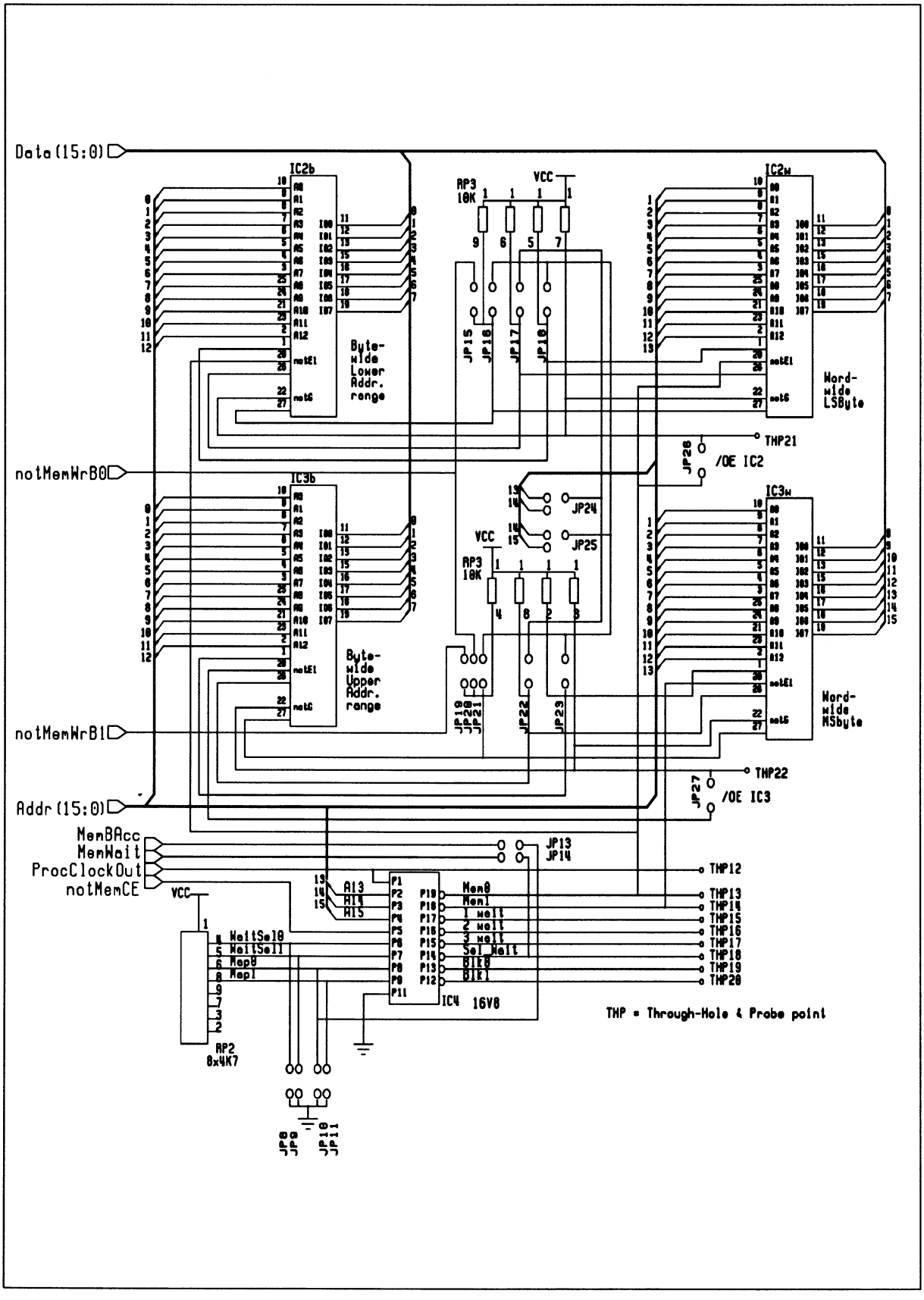
A Jumper and signal pad layout



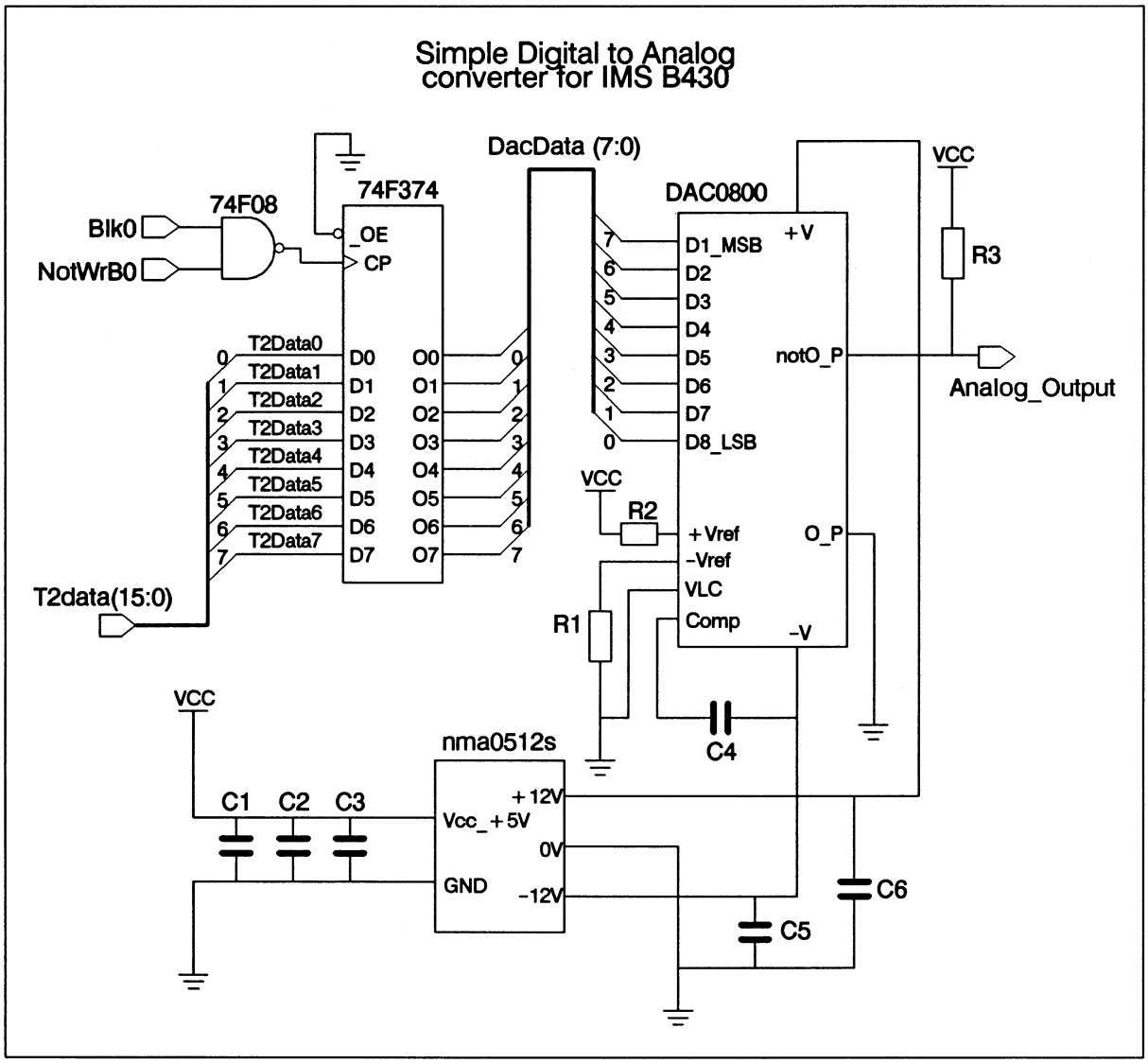
B IMS B430 circuit diagram



Circuit diagram continued

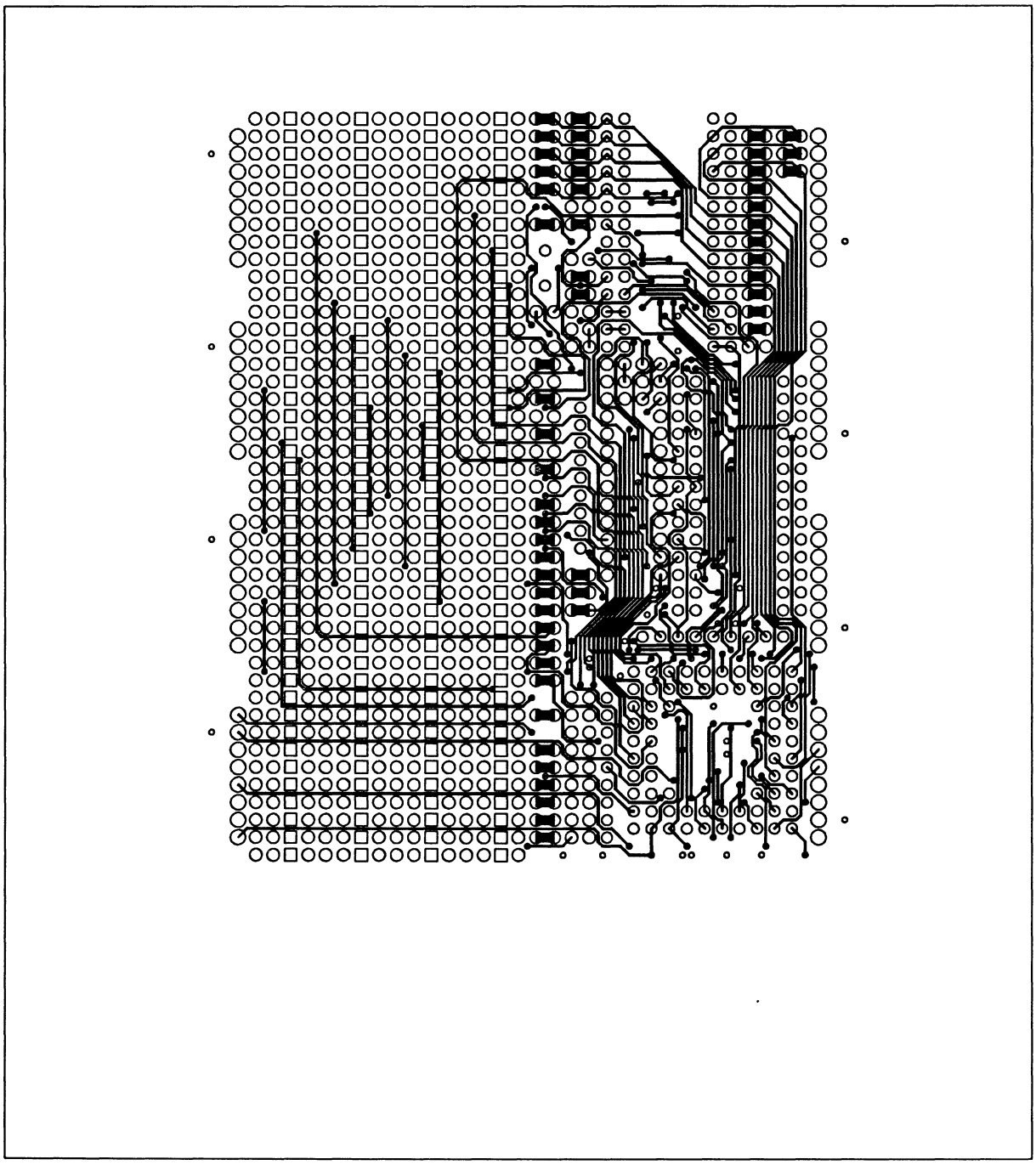


C Example application of the IMS B430 - simple digital to analogue converter

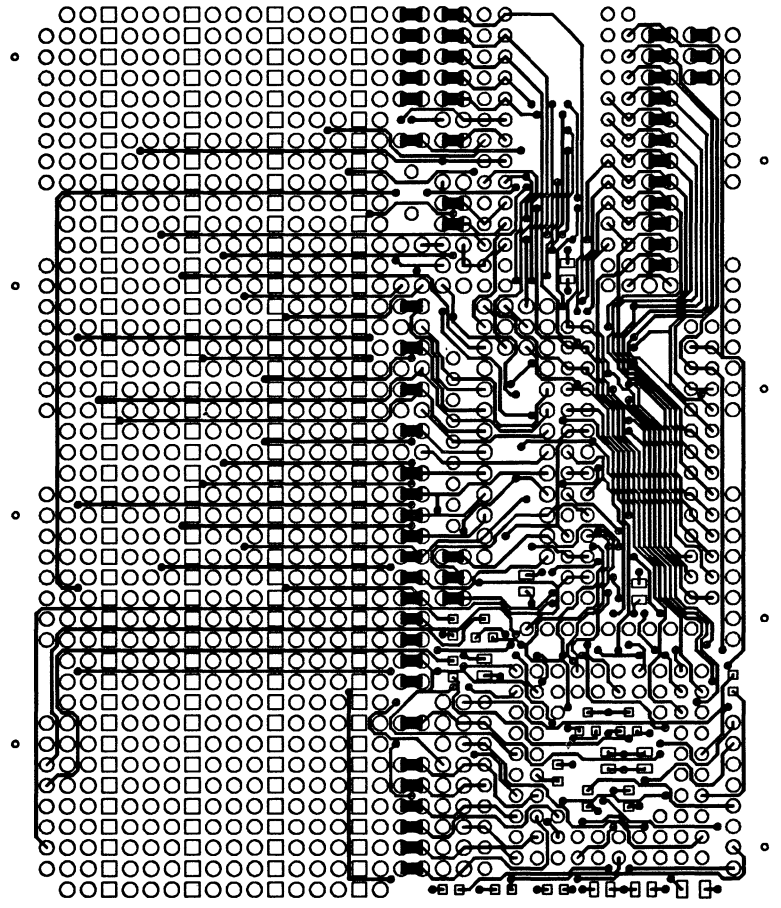


D IMS B430 PCB Layout

D.1 Top layer



D.2 Bottom layer



E The PALASM programming source

The PALASM programming source file for the IMS B430 programmable logic device is reproduced below. The PALASM device programming software is available from distributors of AMD products for a modest charge. Conversion to other logic device programming languages should be straightforward.

```
TITLE        B430 Decode & Waitstate Gen.
PATTERN      DRW-0398
REVISION     03
AUTHOR       P.COBB
COMPANY      INMOS Ltd.
DATE         28/9/90

CHIP DECWAIT PAL16R4

; NOTE: this programming data can be also be used
; implemented using a GAL16V8 type device, since
; its architecture can emulate that of the PAL16R4.

PIN  1 CLK              ; clock input
PIN  2 A13      COMB    ; input
PIN  3 A14      COMB    ; input
PIN  4 A15      COMB    ; input
PIN  5 notMemCE COMB    ; input
PIN  6 Waitsel0 COMB    ; input
PIN  7 Waitsel1 COMB    ; input
PIN  8 Map0     COMB    ; input
PIN  9 Map1     COMB    ; input
PIN 10 GND              ; Ground
PIN 12 BLK1     COMB    ; output
PIN 13 BLK0     COMB    ; output
PIN 14 Selwait  REG     ; output
PIN 15 3Wait    REG     ; output
PIN 16 2Wait    REG     ; output
PIN 17 1Wait    REG     ; output
PIN 18 Mem1     COMB    ; output
PIN 19 Mem0     COMB    ; output
PIN 20 VCC              ; Vcc

; STRING DECLARATION SECTION
; The following strings make the equations below
; much easier to read and write

; These four strings pick out the condition for
; the four different memory map options
STRING  MAP_0 `(/Map1*/Map0*/notMemCE)`
STRING  MAP_1 `( Map1*/Map0*/notMemCE)`
STRING  MAP_2 `(/Map1* Map0*/notMemCE)`
STRING  MAP_3 `( Map1* Map0*/notMemCE)`

; These eight strings divide the address space
; into eight blocks of 8Kbytes

STRING  BLOCK_0 ' (A15*/A14*/A13)`
STRING  BLOCK_1 ' (A15*/A14* A13)`
STRING  BLOCK_2 ' (A15* A14*/A13)`
STRING  BLOCK_3 ' (A15* A14* A13)`
STRING  BLOCK_4 `(/A15*/A14*/A13)`
STRING  BLOCK_5 `(/A15*/A14* A13)`
STRING  BLOCK_6 `(/A15* A14*/A13)`
STRING  BLOCK_7 `(/A15* A14* A13)`
```

```
EQUATIONS

; The first three equations define the shift
; register equations.

1Wait := notMemCE                          ; Follow notMemCE unless -
      +(/Waitsel1*Waitsel0*/2Wait)         ;   1 wait selected and done
      +(Waitsel1*/Waitsel0*/3Wait)         ;   2 waits selected and done
      +(Waitsel1*Waitsel0*/3Wait*2Wait)    ;   3 waits selected and done


2Wait := 1Wait                             ; Follow 1Wait unless -
      +notMemCE                            ;   notMemCE goes high
      +(/Waitsel1*Waitsel0*/2Wait)         ;   1 wait selected and done
      +/3Wait                              ;   3 waits selected and done

3Wait := 2Wait                             ; Follow 2Wait unless -
      +notMemCE                            ;   notMemCE goes high
      +(/Waitsel1*Waitsel0)                ;   1 wait selected (never low)
      +(Waitsel1*/Waitsel0*/3Wait)         ;   2 waits selected and done

; The next equation defines the output of the
; wait state multiplexer output

Selwait := /Waitsel1*Waitsel0*(notMemCE+/2Wait)
        + Waitsel1*/Waitsel0*(1Wait+notMemCE+/3Wait)
        + Waitsel1*Waitsel0*(2Wait+notMemCE+(/3Wait*2Wait))

; The next two equations define the memory chip
; enable outputs

/Mem0 = MAP_0*(/BLOCK_6*/BLOCK_7)                ; 32K RAM wordwide in 0-5
      + MAP_1*(/BLOCK_0*/BLOCK_1)                ; 32K ROM wordwide in 2-7
      + MAP_2*(/BLOCK_5*/BLOCK_6*/BLOCK_7)       ; 32K RAM bytewide in 0-4
      + MAP_3*(BLOCK_0+BLOCK_1)                  ; 16K RAM bytewide in 0-1

/Mem1 = MAP_0*(/BLOCK_6*/BLOCK_7)                ; 32K RAM wordwide in 0-5
      + MAP_1*(/BLOCK_0*/BLOCK_1)                ; 32K ROM wordwide in 2-7
      + MAP_2*(BLOCK_7)                          ; 8K ROM bytewide in 7
      + MAP_3*(BLOCK_4+BLOCK_5+BLOCK_6+BLOCK_7)  ; 32K ROM bytewide in 4-7

; The last two signals define the I/O block
; output enable signals.

/Blk0 = MAP_0*BLOCK_6
      + MAP_1*BLOCK_0
      + MAP_2*BLOCK_5
      + MAP_3*BLOCK_2

/Blk1 = MAP_0*BLOCK_7
      + MAP_1*BLOCK_1
      + MAP_2*BLOCK_6
      + MAP_3*BLOCK_3
```

Addresses

Worldwide Headquarters

INMOS Limited
1000 Aztec West
Almondsbury
Bristol BS12 4SQ
UNITED KINGDOM
Telephone (0454) 616616
Fax (0454) 617910

Worldwide Business Centres

USA

INMOS Business Centre
Headquarters (USA)
SGS–THOMSON Microelectronics Inc.
2225 Executive Circle
PO Box 16000
Colorado Springs
Colorado 80935–6000
Telephone (719) 630 4000
Fax (719) 630 4325

SGS–THOMSON Microelectronics Inc.
Sales and Marketing Headquarters (USA)
1000 East Bell Road
Phoenix
Arizona 85022
Telephone (602) 867 6100
Fax (602) 867 6102

INMOS Business Centre
SGS–THOMSON Microelectronics Inc.
Lincoln North
55 Old Bedford Road
Lincoln
Massachusetts 01773
Telephone (617) 259 0300
Fax (617) 259 4420

INMOS Business Centre
SGS–THOMSON Microelectronics Inc.
1310 Electronics Drive
Carrollton
Texas 75006
Telephone (214) 466 8844
Fax (214) 466 7352

INMOS Business Centre
SGS–THOMSON Microelectronics Inc.
9861 Broken Land Parkway
Suite 320
Columbia
Maryland 21045
Telephone (301) 995 6952
Fax (301) 290 7047

INMOS Business Centre
SGS–THOMSON Microelectronics Inc.
200 East Sandpointe
Suite 650
Santa Ana
California 92707
Telephone (714) 957 6018
Fax (714) 957 3281

INMOS Business Centre
SGS–THOMSON Microelectronics Inc.
2055 Gateway Place
Suite 300
San Jose
California 95110
Telephone (408) 452 9122
Fax (408) 452 0218

EUROPE

United Kingdom

INMOS Business Centre
SGS–THOMSON Microelectronics Ltd.
Planar House
Parkway Globe Park
Marlow
Bucks SL7 1YL
Telephone (0628) 890 800
Fax (0628) 890 391

France

INMOS Business Centre
SGS–THOMSON Microelectronics SA
7 Avenue Gallieni
BP 93
94253 Gentilly Cedex
Telephone (1) 47 40 75 75
FAX (1) 47 40 79 27

West Germany

INMOS Business Centre
SGS–THOMSON Microelectronics GmbH
Bretonischer Ring 4
8011 Grasbrunn
Telephone (089) 46 00 60
Fax (089) 46 00 61 40

Italy

INMOS Business Centre
SGS–THOMSON Microelectronics SpA
V.le Milanofiori
Strada 4
Palazzo A/4/A
20090 Assago (MI)
Telephone (2) 89213 1
Fax (2) 8250449

ASIA PACIFIC

Japan

INMOS Business Centre
SGS–THOMSON Microelectronics K.K.
Nisseki Takanawa Building, 4th Floor
18–10 Takanawa 2-chome
Minato-ku
Tokyo 108
Telephone (03) 280 4125
Fax (03) 280 4131

Singapore

INMOS Business Centre
SGS–THOMSON Microelectronics Pte Ltd.
28 Ang Mo Kio Industrial Park 2
Singapore 2056
Telephone (65) 482 14 11
Fax (65) 482 02 40