INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE CENTRATOM TRIESTE

H4.SMR. 403/26

FIFTH COLLEGE ON MICROPROCESSORS: TECHNOLOGY AND APPLICATIONS IN PHYSICS

2 - 27 October 1989

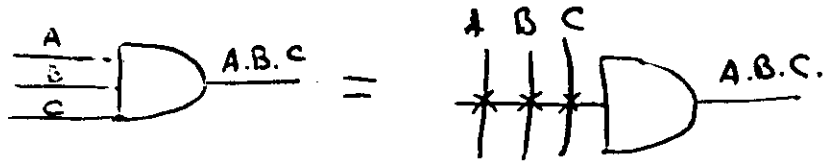
Programmable Logic: Introduction

S. VENKATARAM
Microprocessors Laboratory, ICTP, Trieste, Italy

PROGRAMMABLE LOGIC.

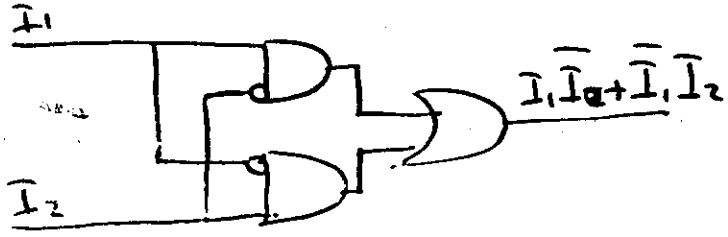
- PROVIDES A SYSTEMATIC METHOD TO GENERATE COMPLEX LOGIC FUNCTIONS.
- PROGRAMMABLE MEANS - THAT THE FUNCTIONS MAY BE SPECIFIED AFTER THE CHIP HAS BEEN MANUFACTURED.
- PROGRAMMED BY 'BLOWING FUSES' ALONG THE CHIP'S INTERNAL DATA LINES.
- THE 'FUSE' IS A NARROW STRIP OF METAL OR ANY OTHER CONDUCTOR DEPOSITED DURING MANUFACTURE.
- WHEN THE FUSE IS PRESENT IT ALLOW AN INPUT SIGNAL TO PROCEED UNIMPEDED.
- WHEN THE FUSE IS BLOWN, ITS INPUT IS DISCONNECTED FOREVER.
- EXPRESSED LOGIC AS A SUM-OF-PRODUCTS.

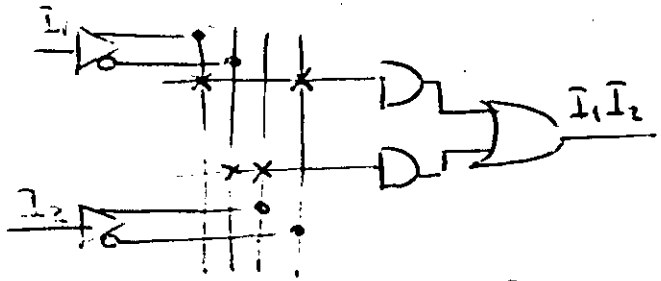
PROGRAMMABLE LOGIC NOTATION



EXAMPLE

$$\text{OUTPUT} = \bar{I}_1 . \bar{I}_2 + \bar{I}_1 I_2$$





A LITTLE FURTHER.

(PROGRAMMABLE READ ONLY MEMORY)

PROMS: CONSISTS OF A FIXED AND ARRAY WHOSE OUTPUT FEEDS A PROGRAMMABLE OR ARRAY

- THEY ARE LOW COST
- EASY TO PROGRAM
- AVAILABLE IN A VARIETY OF SIZES AND ORGANIZATIONS

USED GENERALLY TO STORE COMPUTER PROGRAMS AND DATA. THE INPUTS ARE GENERALLY AND ADDRESS, THE OUTPUTS THE CONTENT OF THE MEMORY LOCATION.

PLA: (PROGRAMMABLE LOGIC ARRAY)

CONSISTS OF A PROGRAMMABLE AND ARRAY WHOSE OUTPUT FEED A FIXED OR ARRAY.

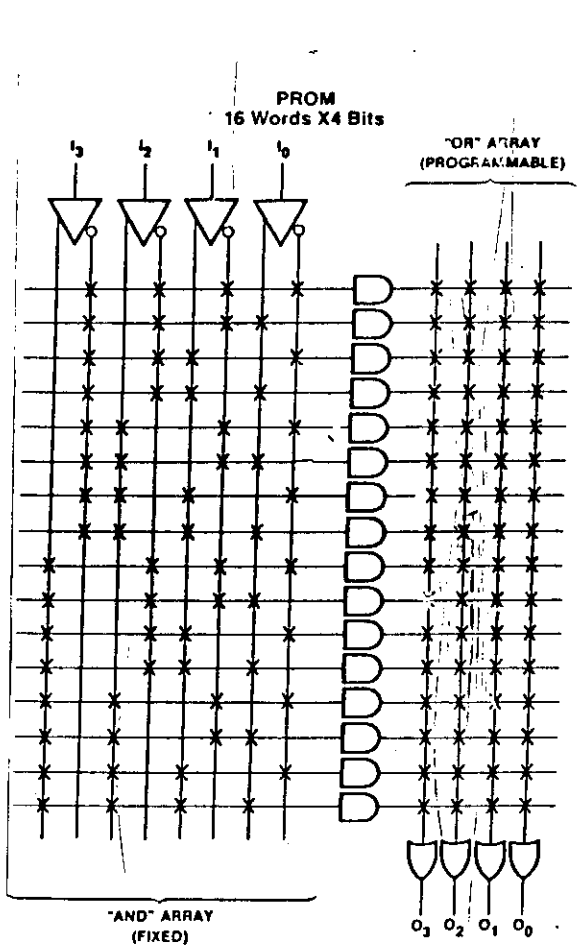
VERY FLEXIBLE BUT VERY COMPLICATED TO BUILD. (HENCE MORE EXPENSIVE THAN PALS).

PAL: (PROGRAMMABLE ARRAY LOGIC)

CONSISTS OF A PROGRAMMABLE AND ARRAY THAT FEEDS A FIXED (BUT INCOMPLETE) OR ARRAY

- LOW COST
- AVAILABLE WITH TRI-STATE OUTPUTS, REGISTERED FEED BACK.

USED AS "GLUE", ~~ETC~~ ETC.

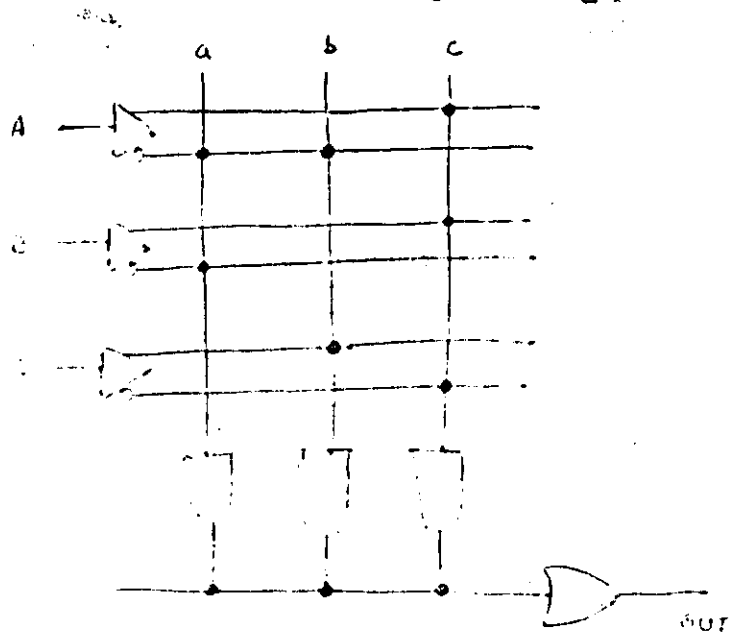


THE PROGRAMMABLE LOGIC ARRAY (PLA)

- IT IS A GENERAL STRUCTURE THAT ALLOWS PROGRAMMING OF ARBITRARY PRODUCTS & ARBITRARY SUMS.
- USED IN VLSI DESIGN TO GENERATE COMPLEX LOGIC FUNCTIONS IN A STRUCTURED MANNER
- BECAUSE OF ITS GENERALITY, REQUIRING MANY INPUTS & OUTPUTS, THE PLA IS NOT WIDELY AVAILABLE AS A DISCRETE IC.

EXAMPLE:

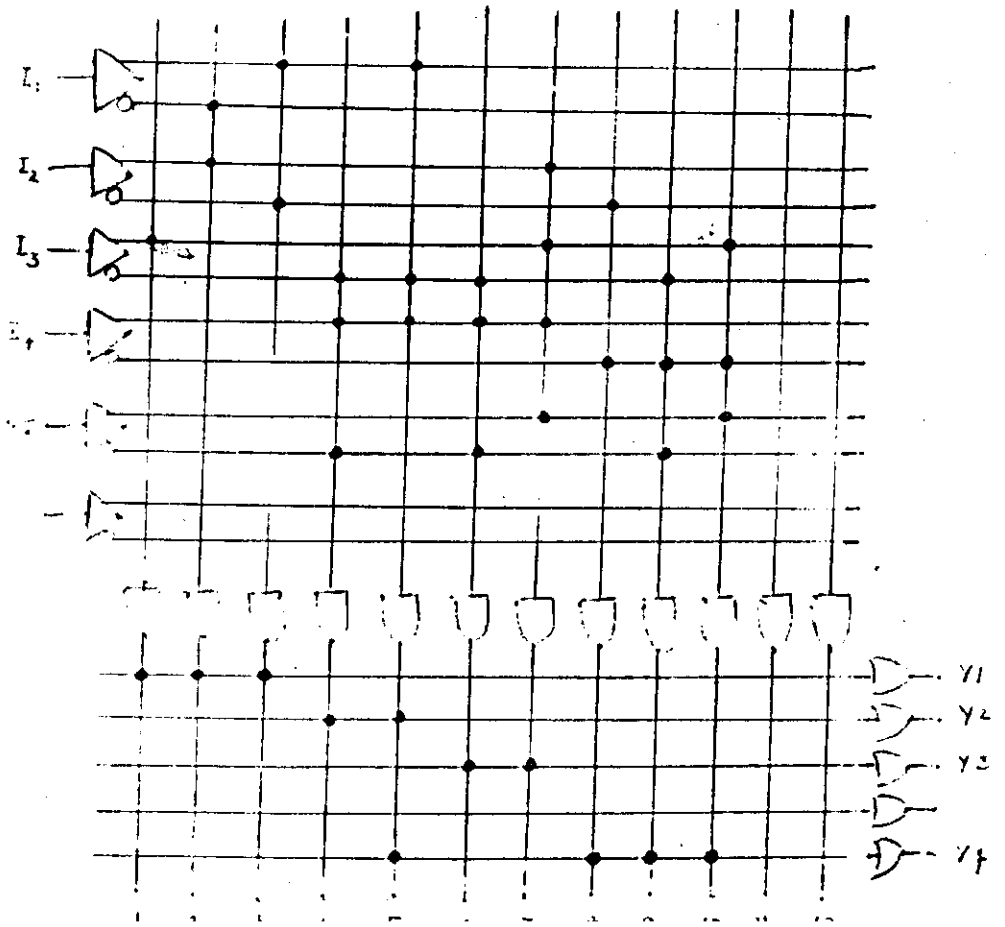
$$OUT = \underbrace{\bar{A}.\bar{B}}_{a} + \underbrace{\bar{A}.C}_{b} + \underbrace{1.B.\bar{C}}_{c}$$



- BUT WE WOULD LIKE TO CREATE SEVERAL OUTPUT FUNCTIONS USING THE SAME INPUT VARIABLES.

EXAMPLE

$$Y_1 = I_0 + \bar{I}_1 . I_2 + I_4 . \bar{I}_2$$
$$Y_2 = \bar{I}_0 . I_4 . \bar{I}_5 + I_1 . \bar{I}_0 . I_4$$
$$Y_3 = \bar{I}_0 . I_4 . \bar{I}_5 + I_2 . I_3 . I_4 . I_5 + \bar{I}_2 . \bar{I}_4$$
$$Y_4 = I_1 . \bar{I}_3 . I_4 + \bar{I}_2 . \bar{I}_4 + \bar{I}_3 . \bar{I}_4 . \bar{I}_5 + I_3 . \bar{I}_4 . I_5$$



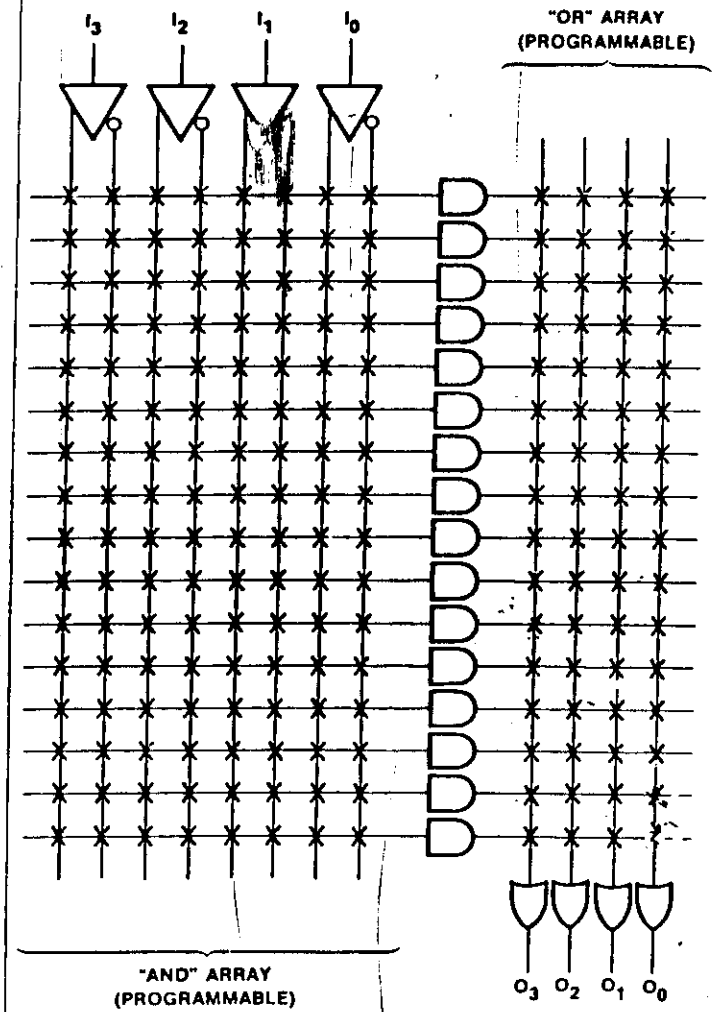
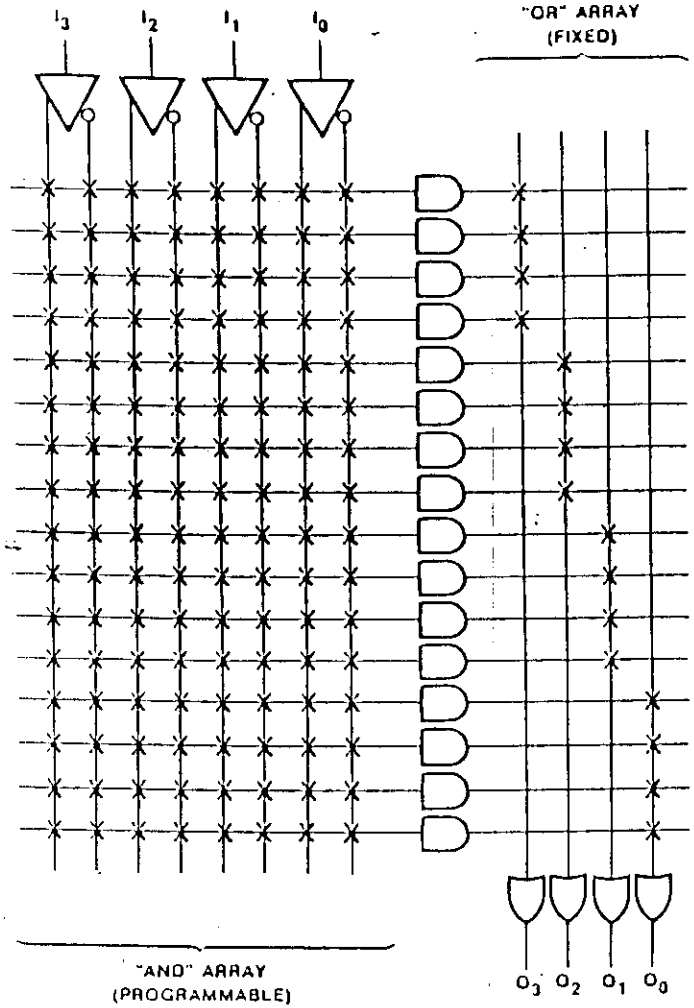

FPLA
4 In•4 Out•16 Products
I_3 I_2 I_1 I_0
"OR" ARRAY
(PROGRAMMABLE)
"AND" ARRAY
(PROGRAMMABLE)
O_3 O_2 O_1 O_0

Figure 7


PAL
4 In•4 Out•16 Products
I_3 I_2 I_1 I_0
"OR" ARRAY
(FIXED)
"AND" ARRAY
(PROGRAMMABLE)
O_3 O_2 O_1 O_0

PROGRAMMABLE ARRAY LOGIC (PAL)

THE CONCEPT:

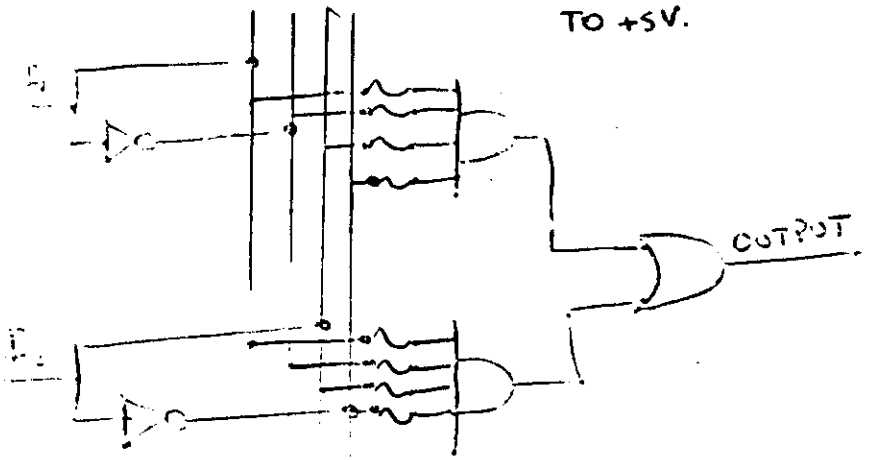
ASSUME WE HAVE THREE INPUT SIGNALS A, B, C AND ONE OUTPUT FUNCTION OUT. THE MOST GENERAL COMBINATORIAL FUNCTION CAN BE WRITTEN AS:

$$OUT = f_1 A.B.C + f_2 \bar{A}.B.C + f_3 A.\bar{B}.C + f_4 A.B.\bar{C} + f_5 \bar{A}.\bar{B}.C + f_6 A.\bar{B}.\bar{C} + f_7 \bar{A}.B.\bar{C} + f_8 \bar{A}.\bar{B}.\bar{C}$$

WHERE $f_i = \begin{cases} 0 & \text{THE MINTERM DOESN'T INTERVENE} \\ 1 & \text{" " DOES "} \end{cases}$

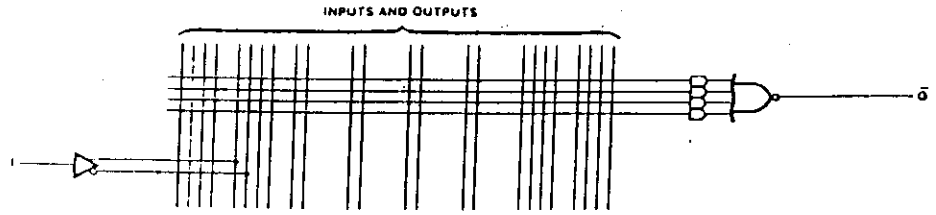
USUALLY ONLY A FEW MINTERMS ARE INVOLVED IN A FUNCTION.

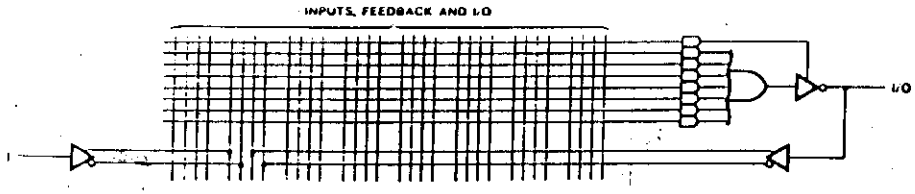
WHEN AN AND INPUT HAS ITS FUSE BLOWN IT IS "PULLED UP" TO +5V.



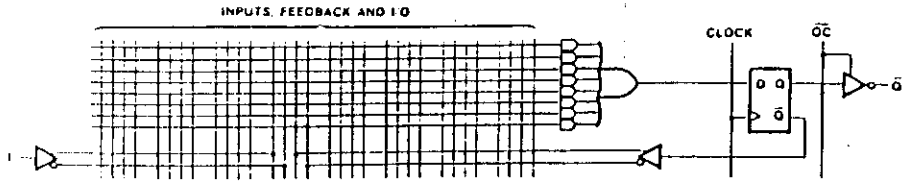
UP TO

MY FUNCTION CAN HAVE ~~only~~ TWO MINTERMS OUT

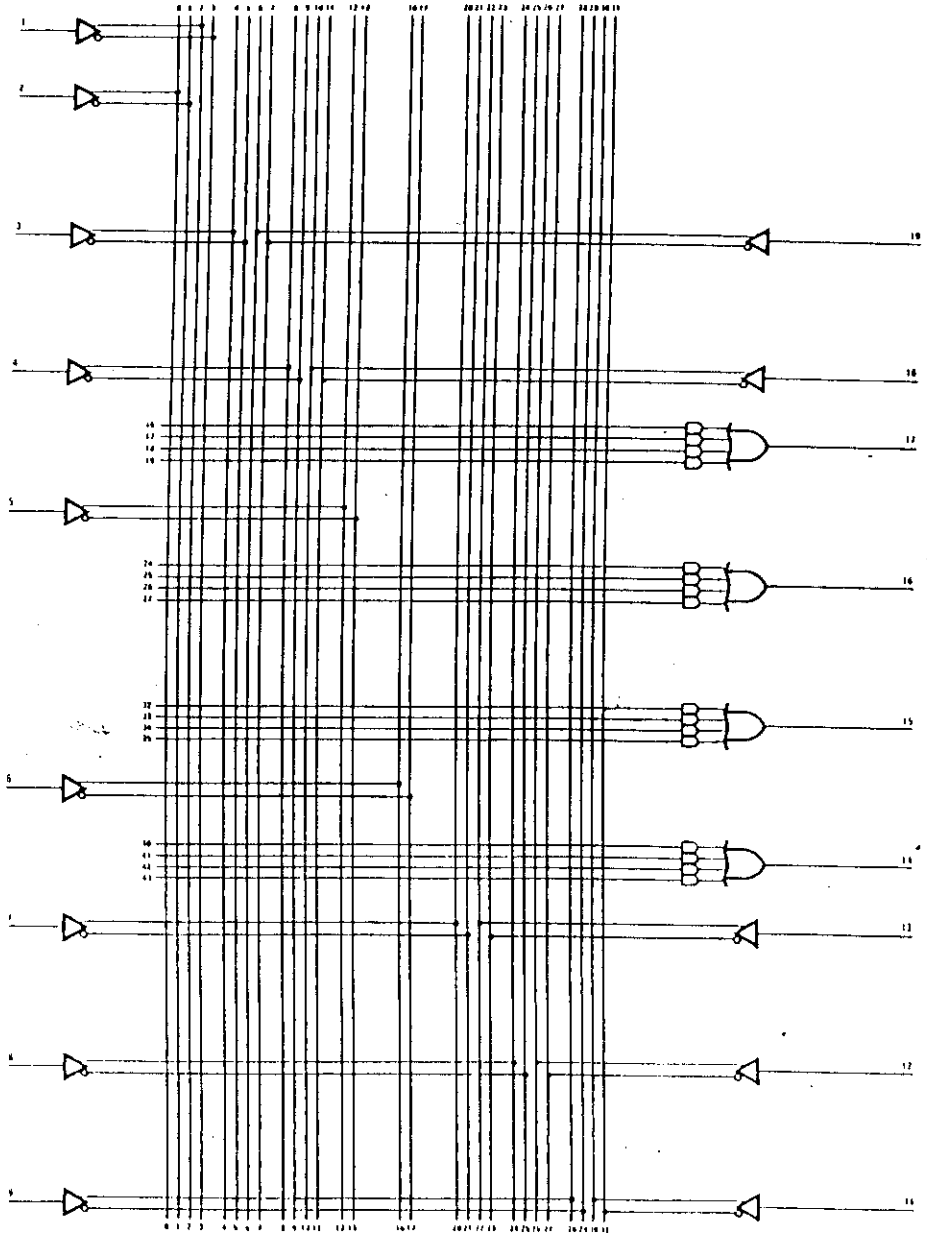




Programmable I/O

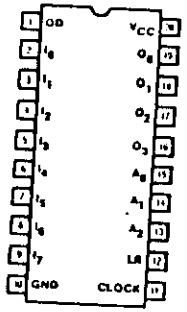


Registered Outputs with Feedback

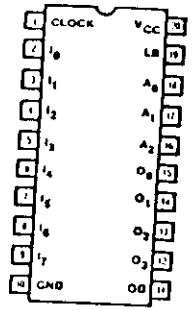


Programming Pin Configurations

PRODUCTS 0 THRU 31



PRODUCTS 32 THRU 63



Voltage Legend

L = Low-level input voltage, V_{IL}
H = High-level input voltage, V_{IH}
HH = High-level program voltage, V_{IHH}
Z = High impedance (e.g., 10kΩ to 5.0V)

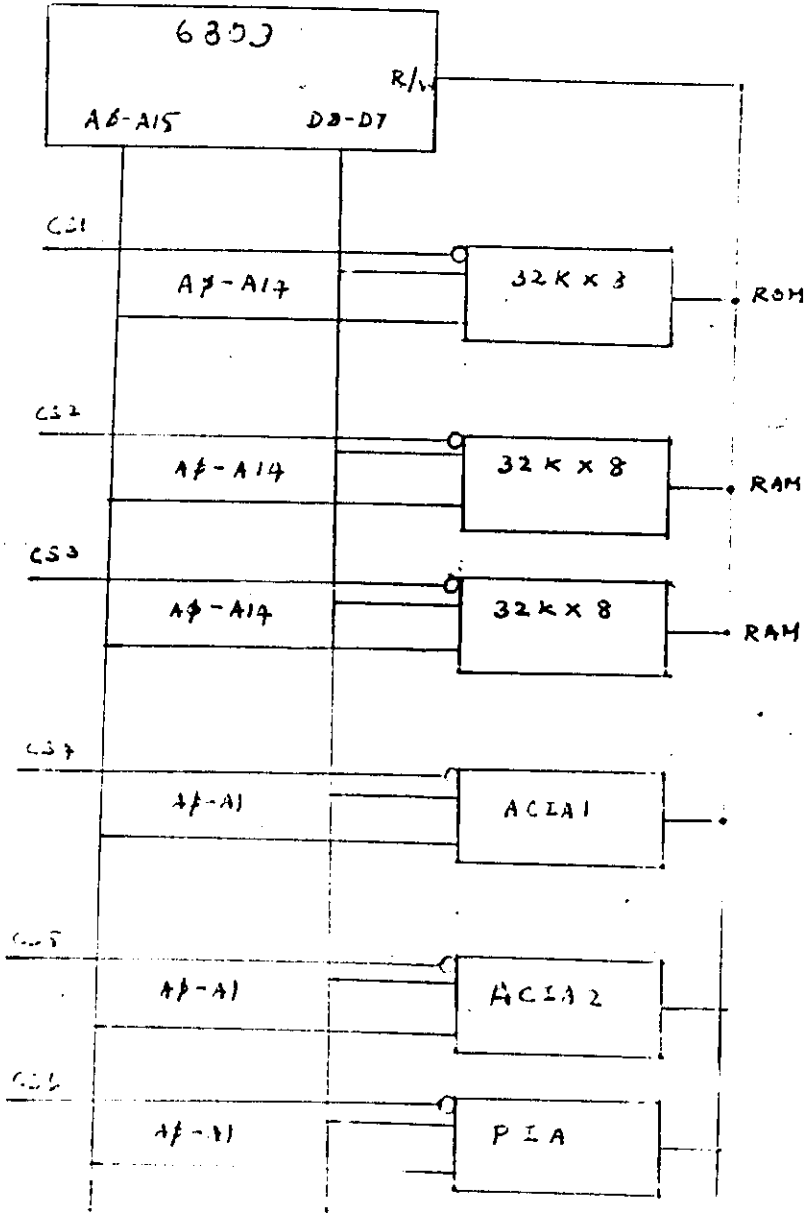
INPUT LINE NUMBER	PIN IDENTIFICATION								
	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	L/R
0	HH	HH	HH	HH	HH	HH	HH	L	Z
1	HH	HH	HH	HH	HH	HH	HH	H	Z
2	HH	HH	HH	HH	HH	HH	HH	L	HH
3	HH	HH	HH	HH	HH	HH	HH	H	HH
4	HH	HH	HH	HH	HH	HH	L	HH	Z
5	HH	HH	HH	HH	HH	HH	H	HH	Z
6	HH	HH	HH	HH	HH	HH	L	HH	HH
7	HH	HH	HH	HH	HH	HH	H	HH	HH
8	HH	HH	HH	HH	HH	L	HH	HH	Z
9	HH	HH	HH	HH	HH	H	HH	HH	Z
10	HH	HH	HH	HH	HH	L	HH	HH	HH
11	HH	HH	HH	HH	HH	H	HH	HH	HH
12	HH	HH	HH	HH	L	HH	HH	HH	Z
13	HH	HH	HH	HH	H	HH	HH	HH	Z
14	HH	HH	HH	HH	L	HH	HH	HH	HH
15	HH	HH	HH	HH	H	HH	HH	HH	HH
16	HH	HH	HH	L	HH	HH	HH	HH	Z
17	HH	HH	HH	H	HH	HH	HH	HH	Z
18	HH	HH	HH	L	HH	HH	HH	HH	HH
19	HH	HH	HH	H	HH	HH	HH	HH	HH
20	HH	HH	L	HH	HH	HH	HH	HH	Z
21	HH	HH	H	HH	HH	HH	HH	HH	Z
22	HH	HH	L	HH	HH	HH	HH	HH	HH
23	HH	HH	H	HH	HH	HH	HH	HH	HH
24	HH	L	HH	HH	HH	HH	HH	HH	Z
25	HH	H	HH	HH	HH	HH	HH	HH	Z
26	HH	L	HH	HH	HH	HH	HH	HH	HH
27	HH	H	HH	HH	HH	HH	HH	HH	HH
28	L	HH	HH	HH	HH	HH	HH	HH	Z
29	H	HH	HH	HH	HH	HH	HH	HH	Z
30	L	HH	HH	HH	HH	HH	HH	HH	HH
31	H	HH	HH	HH	HH	HH	HH	HH	HH

Table 1 Input Line Select

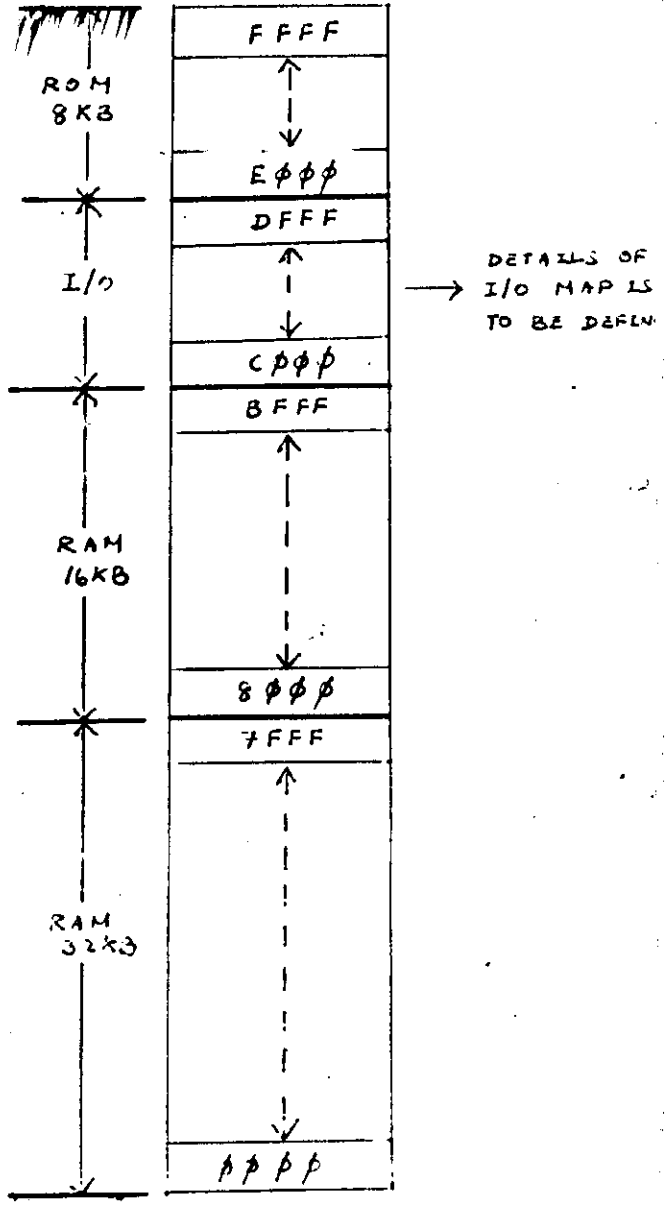
PRODUCT LINE NUMBER	PIN IDENTIFICATION						
	O_3	O_2	O_1	O_0	A_2	A_1	A_0
0, 32	Z	Z	Z	HH	Z	Z	Z
1, 33	Z	Z	Z	HH	Z	Z	HH
2, 34	Z	Z	Z	HH	Z	HH	Z
3, 35	Z	Z	Z	HH	Z	HH	HH
4, 36	Z	Z	Z	HH	HH	Z	Z
5, 37	Z	Z	Z	HH	HH	Z	HH
6, 38	Z	Z	Z	HH	HH	HH	Z
7, 39	Z	Z	Z	HH	HH	HH	HH
8, 40	Z	Z	HH	Z	Z	Z	Z
9, 41	Z	Z	HH	Z	Z	Z	HH
10, 42	Z	Z	HH	Z	Z	HH	Z
11, 43	Z	Z	HH	Z	Z	HH	HH
12, 44	Z	Z	HH	Z	HH	Z	Z
13, 45	Z	Z	HH	Z	HH	Z	HH
14, 46	Z	Z	HH	Z	HH	HH	Z
15, 47	Z	Z	HH	Z	HH	HH	HH
16, 48	Z	HH	Z	Z	Z	Z	Z
17, 49	Z	HH	Z	Z	Z	Z	HH
18, 50	Z	HH	Z	Z	Z	HH	Z
19, 51	Z	HH	Z	Z	Z	HH	HH
20, 52	Z	HH	Z	Z	HH	Z	Z
21, 53	Z	HH	Z	Z	HH	Z	HH
22, 54	Z	HH	Z	Z	HH	HH	Z
23, 55	Z	HH	Z	Z	HH	HH	HH
24, 56	HH	Z	Z	Z	Z	Z	Z
25, 57	HH	Z	Z	Z	Z	Z	HH
26, 58	HH	Z	Z	Z	Z	HH	Z
27, 59	HH	Z	Z	Z	Z	HH	HH
28, 60	HH	Z	Z	Z	HH	Z	Z
29, 61	HH	Z	Z	Z	HH	Z	HH
30, 62	HH	Z	Z	Z	HH	HH	Z
31, 63	HH	Z	Z	Z	HH	HH	HH

Table 2 Product Line Select

DESIGNING A COMPUTER



WE NEED TO DEFINE A MEMORY MAP.



— NOW COMES THE TIME TO DESIGN THE "GLUE" OR LINKING LOGIC.

STEPS TO FOLLOW WHEN USING THE "KARNAUGH METHOD FOR SIMPLIFYING BOOLEAN EQUATIONS

STEP1 : BUILD YOUR TRUTH TABLE ACCORDING TO THE REQUIREMENTS OF YOUR APPLICATION.

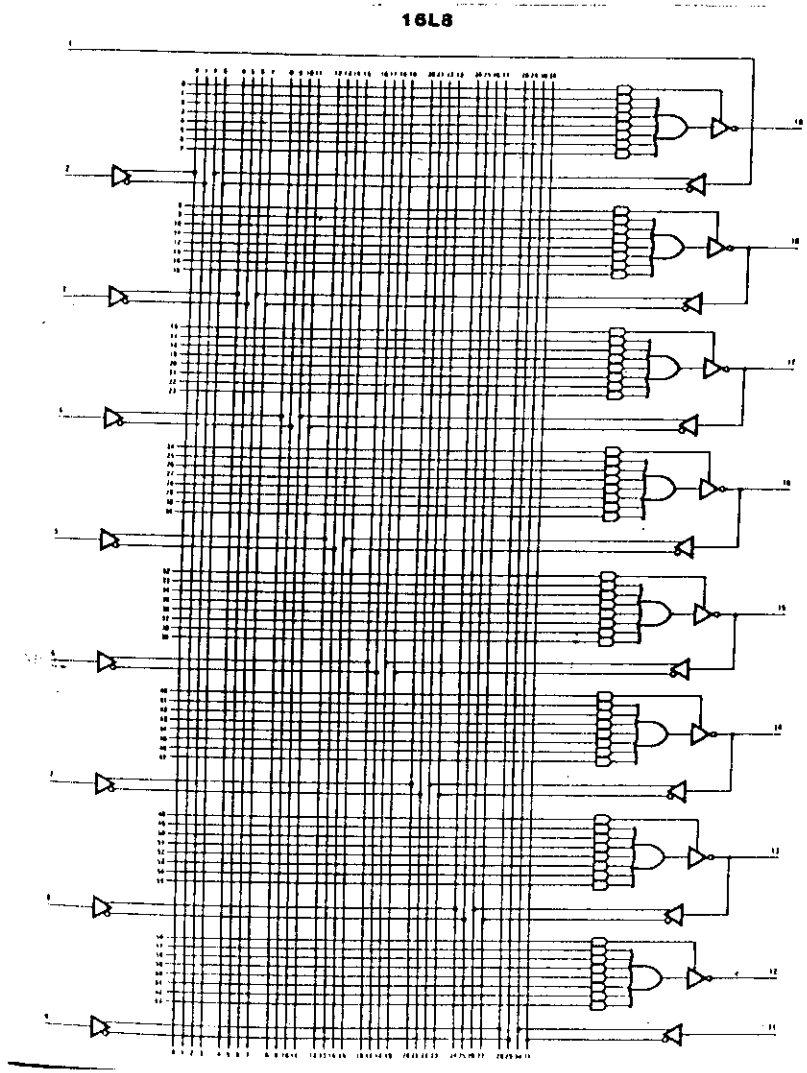
STEP2 : DRAW "1's" ON THE KARNAUGH MAP FOR EACH FUNDAMENTAL PRODUCT NECESSARY TO PRODUCE THE "1's" AT THE OUTPUTS.

STEP3 : ENCIRCLE OCTETS, QUADS AND PAIRS ENCIRCLE THE ISOLATED "1's"

STEP4 : ELIMINATE REDUNDANT GROUPS.

STEP5 : WRITE THE BOOLEAN EQUATION BY ORying THE PRODUCTS CORRESPONDING TO THE GROUPS.

STEP6 : IF NEEDED, REWRITE USING ONLY NAND's OR ORs.



ADVANTAGES OF "PROGRAMMABLE LOGIC"

- YOU NEED ONLY A FEW DIFFERENT CHIPS TO IMPLEMENT MOST FUNCTIONS.
- SAVES SPACE ON THE PCB BOARD.
- SAVES POWER
- YOU CAN CHANGE THE "LOGIC" (MEMORY MAP, ETC) BY REPLACING JUST ONE CHIP, GIVING LARGER FLEXIBILITY TO YOUR DESIGN.
- IF YOU HAVE AS INPUTS THE RIGHT SIGNALS AND YOU MADE A "LOGIC MISTAKE" YOU CAN REMEDY IT ALSO BY REPROGRAMMING AND SIMPLE REPLACEMENT OF THE CHIP.
- YOU CAN AVOID GOING THRU A KARNAUGH MAP WITH ≥ 10 INPUTS.
- IF YOU HAVE PALASM, ABEL, ETC YOU DESIGN USING MORE UNDERSTANDABLE STATEMENTS SUCH AS:

$$/CS1 := (\text{ADDRESS} \leq 17FFF)$$

↑ NOT

SOURCE FILE DECOD.PLD FOR CUPL SOFTWARE.

```
Name     DECODE;
Partno   NIL;
Date     16/10/89;
Revision 00;
Designer VENKA;
Company  MICROLAB;
Assembly NIL;
Location ICTP;

/*************************************************************************/
/*Decoding for a microprocessor to access ROM, RAM & I/O ports           */
/*                                                                       */
/*                                                                       */
/*************************************************************************/
/*  Allowable Target Device Types: 16L8                                  */
/*************************************************************************/

/**  Inputs  **/

Pin 1        = a15;
Pin 2        = a14;
Pin 3        = a13;
Pin 4        = a3;
Pin 5        = a2;

/**  Outputs  **/

Pin 19       = !ROM;
Pin 18       = !RAM1;
Pin 17       = !RAM2;
Pin 16       = !ACIA1;
Pin 15       = !ACIA2;
Pin 14       = !PIA;
Pin 13       = !SPARE;

/** Declarations and Intermediate Variable Definitions **/

FIELD BLK_ADDR = [a15..a13];

/**  Logic Equations  **/

ROM   = (BLK_ADDR:[E000..FFFF]);
RAM1  = (BLK_ADDR:[0000..7FFF]);
RAM2  = (BLK_ADDR:[8000..BFFF]);
ACIA1= (BLK_ADDR:[C000..DFFF]) & (!a3 & !a2);
ACIA2= (BLK_ADDR:[C000..DFFF]) & (!a3 &  a2);
PIA   = (BLK_ADDR:[C000..DFFF]) & ( a3 & !a2);
SPARE= (BLK_ADDR:[C000..DFFF]) & ( a3 &  a2);
```

(11) PART OF DECOD.DOC OUTPUT OF CUPL SOFTWARE

```
*******************************************************************************
                                  DECODE
*******************************************************************************

CUPL          2.15b Serial# 6-00002-198
Device        p1618  Library DLIB-h-24-8
Created       Mon Oct 16 17:45:22 1989
Name          DECODE
Partno        NIL
Revision      00
Date          16/10/89
Designer      VENKA
Company       MICROLAB
Assembly      NIL
Location      ICTP

===============================================================================
                                 Fuse Plot
===============================================================================

Pin #19
 0000 --------------------------------
 0032 x-x-x---------------------------
 0064 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0096 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0128 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0160 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0192 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0224 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #18
 0256 --------------------------------
 0288 ---x----------------------------
 0320 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0352 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0384 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0416 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0448 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0480 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #17
 0512 --------------------------------
 0544 -xx-----------------------------
 0576 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0608 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0640 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0672 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0704 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0736 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #16
 0768 --------------------------------
 0800 x-x--x---x---x------------------
 0832 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0864 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0896 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0928 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0960 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0992 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #15
 1024 --------------------------------
 1056 x-x--x---x--x-------------------
 1088 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1120 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1152 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1184 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1216 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1248 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #14
 1280 --------------------------------
 1312 x-x--x--x----x------------------
 1344 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1376 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1408 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1440 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1472 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1504 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #13
 1536 --------------------------------
 1568 x-x--x--x---x-------------------
 1600 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1632 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1664 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1696 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1728 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1760 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #12
 1792 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1824 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1856 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1888 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1920 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1952 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 1984 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 2016 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

(12)

PART OF DECOD.DOC OUTPUT OF CUPL SOFTWARE

```
===============================================================================
                                Chip Diagram
===============================================================================

                          ______________
                         |    DECODE    |
                 a15 x---|1           20|---x Vcc
                 a14 x---|2           19|---x !ROM
                 a13 x---|3           18|---x !RAM1
                  a3 x---|4           17|---x !RAM2
                  a2 x---|5           16|---x !ACIA1
                     x---|6           15|---x !ACIA2
                     x---|7           14|---x !PIA
                     x---|8           13|---x !SPARE
                     x---|9           12|---x
                 GND x---|10          11|---x
                         |______________|
```

(24)

```
Name        DECODE;
Partno      NIL;
Date        16/10/89;
Revision    00;
Designer    VENKA;
Company     MICROLAB;
Assembly    NIL;
Location    ICTP;

ORDER:
        a15, %2, a14, %2, a13, %2, a3, %2, a2, %4,
        !ROM, %2, !RAM1, %2, !RAM2, %2, !ACIA1, %2,
        !ACIA2, %2, !PIA, %2, !SPARE;

VECTORS:
$MSG "          a  a  a  a  a    !  !  !  !  !  !  !";
$MSG "          1  1  1  3  2    R  R  R  A  A  P  S";
$MSG "          5  4  3          O  A  A  C  C  I  P";
$MSG "                           M  M  M  I  I  A  A";
$MSG "                              1  2  A  A     R";
$MSG "                                 1  2        E";
$MSG "          ------------   ------------------";
$MSG " Tests for RAM1 chip select signal";
          0  0  0  X  X    H  L  H  H  H  H  H
          0  1  1  X  X    H  L  H  H  H  H  H
$MSG " Tests for RAM2 chip select signal";
          1  0  0  X  X    H  H  L  H  H  H  H
          1  0  1  X  X    H  H  L  H  H  H  H
$MSG " Tests for ROM chip select signal";
          1  1  1  X  X    L  H  H  H  H  H  H
$MSG " Tests for ACIA1 chip select signal";
          1  1  0  0  0    H  H  H  L  H  H  H
$MSG " Tests for ACIA2 chip select";
          1  1  0  0  1    H  H  H  H  L  H  H
$MSG " Tests for PIA chip select";
          1  1  0  1  0    H  H  H  H  H  L  H
$MSG " Tests for SPARE chip select";
          1  1  0  1  1    H  H  H  H  H  H  L
$MSG " End of tests";
```

C.U.P.L. FILE DECOD.PL FOR THE CSIM SOFTWARE

(24)

SCREEN OUTPUT OF THE CSIM SOFTWARE FOR DECOD.PL.

```
-v dddd decod
CSIM: CUPL Simulation Program
Version 2.15b Serial# 6-00002-198
Copyright (C)  1983,1987 Personal CAD Systems, Inc.

csima

        a  a  a  a  a    !  !  !  !  !  !  !
        1  1  1  3  2    R  R  R  A  A  P  S
        5  4  3          O  A  A  C  C  I  P
                         M  M  M  I  I  A  A
                            1  2  A  A     R
                                  1  2        E

        ------------   ------------------
 Tests for RAM1 chip select signal
0001: 0  0  0  X  X    H  L  H  H  H  H  H
0002: 0  1  1  X  X    H  L  H  H  H  H  H
 Tests for RAM2 chip select signal
0003: 1  0  0  X  X    H  H  L  H  H  H  H
0004: 1  0  1  X  X    H  H  L  H  H  H  H
 Tests for ROM chip select signal
0005: 1  1  1  X  X    L  H  H  H  H  H  H
 Tests for ACIA1 chip select signal
0006: 1  1  0  0  0    H  H  H  L  H  H  H
 Tests for ACIA2 chip select
0007: 1  1  0  0  1    H  H  H  H  L  H  H
 Tests for PIA chip select
0008: 1  1  0  1  0    H  H  H  H  H  L  H
 Tests for SPARE chip select
0009: 1  1  0  1  1    H  H  H  H  H  H  L
 End of tests
time: 5 secs
total time: 5 secs

C:\CUPL>
```

(22)

```
CUPL            2.15b  Serial# 6-00002-198
Device          p1618  Library DLIB-h-24-8
Created         Mon Oct 16 17:45:21 1989
Name            DECODE
Partno          NIL
Revision        00
Date            16/10/89
Designer        VENKA
Company         MICROLAB
Assembly        NIL
Location        ICTP
*QF20
*QF2048
*QV9
*Q3
*F0
*L0000 11111111111111111111111111111111
*L0032 01010111111111111111111111111111
*L0256 11111111111111111111111111111111
*L0288 11101111111111111111111111111111
*L0512 11111111111111111111111111111111
*L0544 10011111111111111111111111111111
*L0768 11111111111111111111111111111111
*L0300 01011011101110111111111111111111
*L1024 11111111111111111111111111111111
*L1056 01011011101110111111111111111111
*L1280 11111111111111111111111111111111
*L1312 01011011101110111111111111111111
*L1536 11111111111111111111111111111111
*L1568 01011011101110111111111111111111
*C36AB
*P 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*V0001 000XXXXXXNXXHHHHHLHN
*V0002 011XXXXXXNXXHHHHHLHN
*V0003 100XXXXXXNXXHHHHLHHN
*V0004 101XXXXXXNXXHHHHLHHN
*V0005 111XXXXXXNXXHHHHHHLN
*V0006 11000XXXXNXXHHHLHHHN
*V0007 11001XXXXNXXHHLHHHHN
*V0008 11010XXXXNXXHLHHHHHN
*V0009 11011XXXXNXXLHHHHHHN
*03F5
```

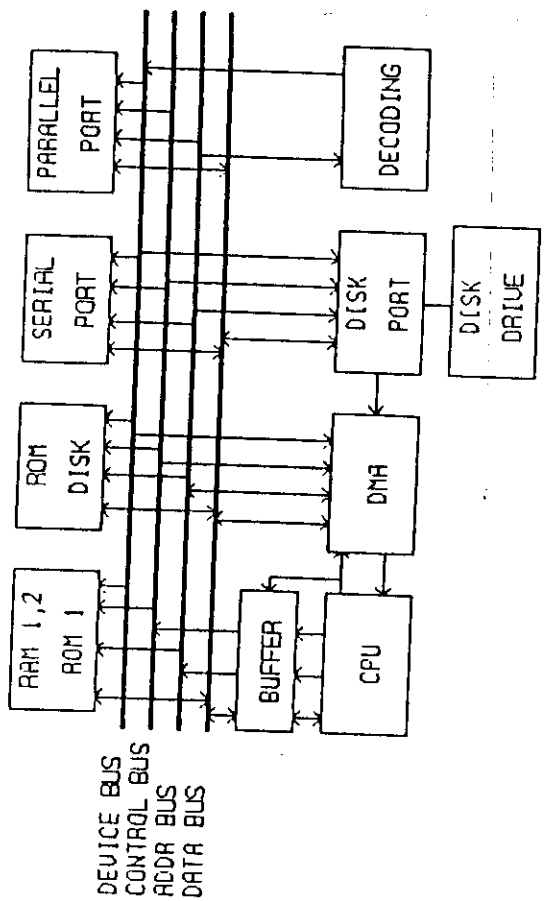



Table 1 Main memory mapping

Address	Rosy/Flex	OS-9
F000-FFFF	Rosy monitor	OS-9 kernel
EC00-EFFF	IO device	IO device
E800-EBFF	Floppy driver	Floppy driver
C000-E7FF	RAM2	RAM2
A000-BFFF	Rosy monitor	RAM2
8000-9FFF	RAM2	RAM2
0000-9FFF	RAM1	RAM1

Table 2 IO device memory mapping

Address	Device	Use
EF40-EF7F	I/O BUS	Logidule
EF30-EF38	Bus connector	User port C
EF28-EF2F	Bus connector	User port B
EF20-EF28	Bus connector	User port A
EF18-EF18	74LS273 (U12)	RMSK (Logical sector number)
EF00-EF03	SY6551A (U20)	ACIA2 (serial printer)
EE00-EEFF	ROM2	Romdisk Window
ECC0-ECDF	MC68B44(U24)	DMA channel
EC84-EC87	WD2793	WD (floppy controller)
EC82-EC82	74LS74(U29)	DIC (disk interrupt controller)
EC81-EC81	74LS245(U30)	PIR(Polling interrupt signal)
EC80-EC80	74LS273(U32)	DRS(drive select)
EC54-EC55	MC68B50(U18)	ACIA1(host communication
EC18-EC1F	MC68B40(U13)	PTM(trace timer)
EC14-EC15	MC68B50(U19)	ACIA0(terminal)
EC10-EC13	MC68B21(U23)	PIA0 (parallel port)