UDK 621.3:(53+54+621+66), ISSN 0352-9045 Informacije MIDEM 20(1990)4, Ljubljana

GALs - GENERIC ARRAY LOGIC (Part III) (A member of PLDs family)

Zlatko Bele

KEY WORDS: Programmable logic circuits (PLDs), integrated circuits, Aplication specific integrated circuits (ASIC), EECMOS, Output Logic Macro Cell (OLMC), GAL

ABSTRACT: In previous parts of the article an overview and detailed architecture of GALs has been presented.Part III deals with programming software and hardware for GAL devices as well as programming example. A way of emulating conventional PAL devices with GAL devices is also presented.

GALi - Generične logične mreže (III.del)

KLJUČNE BESEDE: programabilna logična vezja (PLDs), integrirana vezja, vezja po naročilu (ASIC), EECMOS, izhodne logične makrocelice (OLMC), GAL

POVZETEK: V prvih dveh delih članka je bil podan pregled in podrobna zgradba GAL programabilnih logičnih vezij, tretji del pa opisuje programska in strojna orodja za programiranje GAL vezij skupaj s konkretnim primerom programiranja.Podan je tudi način emulacije konvencionalnih PAL vezij z GAL vezji.

1. INTRODUCTION

Apperance of first PLD devices (bipolar PAL's) falls into early 70's. At that time this devices had a quite difficult time being accepted by system's and board designers due to the lack of good programming software. It was necessary to load each individual fuse location into the devices after extensive analysis of the design requirements.This was a slow and cumbersome process requiring the designer to learn the architecture of many different devices in addition to the fact that logic errors could not be automatically identified.

Development and availability of user-friendly and functional software tools had the main contribution to the tremendous upsurge in the usage of the PLD devices in late 80's.

2. SOFTWARE TOOLS

The most popular early assembler based programming software for PAL devices was PALASM from Monolithic Memories. This assembler allows inputs only as Boolean equations, has a difficult command structure, allows equations only in SOP (sum-of-products) format, works on PAL devices only and has no intelligence i.e. unable to do logic minimization or identify specific device types which will or will not work with a given set of Boolean equations. The most severe restrictions of this type of approach was the inflexbility of the software to work on other vendors devices thus forcing designers to learn many different assemblers in order to have more than one device supplier to choose from.

The development of compiler based software in the early 80's was the response to the need for more flexibility and utility in development tools. The original packages were developed by third-party manufacturers, not device vendors, with the goal of supporting all device types and all manufacturers. This original packages as "CUPL" from Personal CAD systems and "ABEL" from DATA I/O Corp. had the capability of logic equation minimization,macros,truth table and state machine syntax and self-documentation.

The latest advance in the PLD development software has occured in the mid-80's. These programs allow schematic capture using pre-programmed macros in the software which allow a designer to simply create a logic shematic as the input to a translator. The translator converts the graphic representation to a network list that is then compiled to the fuse maps by the software tool. All the other functions of the software such as logic minimization are then available to streamline the design before it is downloaded into a device. The most widely used of these tools are "CAE-1" from Personal CAD systems and "DASH" from DATA I/O.

3. HARDWARE TOOLS

The hardware used to program GAL devices can be divided roughly into two types:

a) Universal programmers
b) GAL-only programmers

a) Universal programmers

In this case "universal" means with respect to PLD devices only, this terminology should not be confused with the broader sense of "universal" programmers, meaning those that program EPROM memories or EPROM arrays in microprocessors as well as PLD's.

In these category of universal type programmers are those from DATA I/O and STAG MICROSYSTEMS as well as many others. These programmers support many different PLD devices, including ECL, CMOS EPROM, standard bipolar PALs and GAL EECMOS device types. These universal programmers also support many advanced functions such as test vectors, register preload, and even automatic chip handler control in a production environment.

b) GAL-only programmers

Main advantage of GAL-only programmers is, of course, their low price. Well known vendors of such programmers are QWERTY and PROGRAMMABLE LOGIC TECHNOLOGIES. The QWERTY also supports test vectors and registers preload for full functional testing of GAL devices.

The type of tool choosen should reflect the environment it will be used in. This means that GAL-only programmer could be considered in an operation where GAL only development and small volume production is occuring in a situation where a low-cost evaluation and programming of GALs is necessary. However in a large development lab where many types of PLDs are being evaluated or a high volume production environment where automated handling of devices is necessary, a universal type programmer with chip handler may be more appropriate. A prime consideration should also be the necessary functions of the programer as well. For example altough a 100% programming yield of GAL devices is guaranteed by vendor, test vector and register preload capability is recommended for the designer to verify that the device is doing exactly what was planned.

4. PROGRAMMING PROCESS

Programming GAL device is the process of providing it with so called "JEDEC" file. This file has got its name by the standards organization JEDEC with the representatives from major semiconductor companies on its committees, which has approved a standard for the interchange of PLD data. So JEDEC file is used as the medium of transfer from the development computer environment to that of the hardware device programmer. Included in the file are control bits that determine the status of programming cells, status of security cell, test vectors and data-transsmision checksum. Test vectors, if included, indicate the stimulus and response for a PLD and serve primarily to validate the functionallity of a design source file.

JEDEC file is written into the device by applying a series of specific voltage pulses."Responsibility" of the programmer manufacturer refers to his ability to provide the correct voltages and timing pulses and make the correct measurements on the outputs,if applicable,for the device.

JEDEC file is produced from a design source file which is written in a specific syntax and compiled with a compiler based programming software. As mentioned before one of the most known such a software is that of CUPL from Personal CAD systems.

First what we have to do developing a design source file using CUPL or in general some other development software is to tell the software which type of GAL device will be used. Then entry of some optional informations such as company name, design description, designer name is provided.

The device pinout and pin labels need to be specified next. Convenient names should be used since the software doesn't care what the pin is called as long as we are consistent. Pin definition example is as folows:

```
/ * inputs * /
pin (1,2) = ( A, !B);

/ * outputs * /
pin (18,19) = (Y,!Z);
```

It is also a good idea to specify pin names in a format that is consistent with the actual pin state. In the above definitions, signals A and Y are active high, while B and Z are active low. An eclamation point prefixing label is used to indicate active-low data signals. The use of active-high variation of these signals in subsequent design statements will automatically be resolved by the software compiler.

Entry of logic functions is next. Traditionally, this entry is in the form of Boolean equations. Current revisions of CUPL software allow truthtable, state-macine and shematic-entry formats, as well. In example below a traditional equation-entry format is used to create an AND function on Y (pin 18) and an XOR function on Z (pin 19). Since Z has been defined as an active-low signal, however, function on pin 19 is actually XNOR:

```
/ * logic equations * )
Y = A & B;
Z = !A & B # A & !B;
```

The opeartors used in the CUPL language are "!" for invert, "&" for the AND function and "#" for the OR function. The equations are written exactly as needed. All of the inversions for active-low inputs and outputs will be automatically resolved, a routine procedure for the compiler software. Altough these are simple equations, if they had been complex ones that needed automatic reduction to a specific number of product terms for a given PLD, the software would have performed that reduction, as well.

Next, the CUPL compiler needs to be invoked to process the " source" file. Through compiling process a JEDEC file is produced together with so called "documentation file". The purpose of this file is to provide a

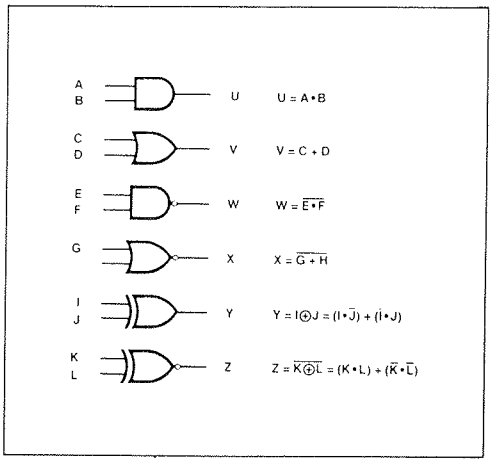

Fig. 1 Basic Logic Gates

hard-copy documentation of the final (reduced) equations, the cell map and chip diagram, if desired.

As an actual example an implementation of the basic logic gates AND, OR, NAND, NOR, and XNOR (see Figure 1) in a GAL16V8 device is presented next.

Since 12 inputs and 6 outputs are needed (Figure 2), 2 Output Logic Macrocells (OLMCs) must be configured as dedicated inputs and 6 as dedicated combinational outputs. Programming software automatically handles this task. Active-high or low outputs are no problem either, becouse of the programmable polarity feature of the GAL16V8.

Figure 3 shows a CUPL design input source file as it has to be prepared to program GAL16V8 with basic logic gates from Figure 1.

Once this file is compiled with CUPL compiler a JEDEC file as shown on Figure 4 is produced. Zeros in the cell's field represent active cells.

Detailed and complete cell map or so called 'fuse plot' for mentioned example is presented on Figure 4. As can be seen all inactive cells have to be programmed.

As mentioned, for testing the functionality of the device and design verification a "test vectors" has to provided by means of CUPL simulation file. In this case JEDEC file includes also test vectors. Both files are shown on Figure 5 and Figure 6 respectively.

At the end of programming process a 'documentation' file is generated which consists of:

- Expanded product terms (Figure 7)
- Symbol table (Figure 8)
- Chip diagram (Figure 9)
- Fuse plot (allready shown on Figure 4)

The patterning of the GAL device array is done using a parallel programming scheme. This allows the device to be programmed very fast and in fact is less than a second on most programming hardware. This is up to an order of magnitude faster than device using the UV-CMOS approach. During this programnming cycle, the logic array, the architecture matrix programming and

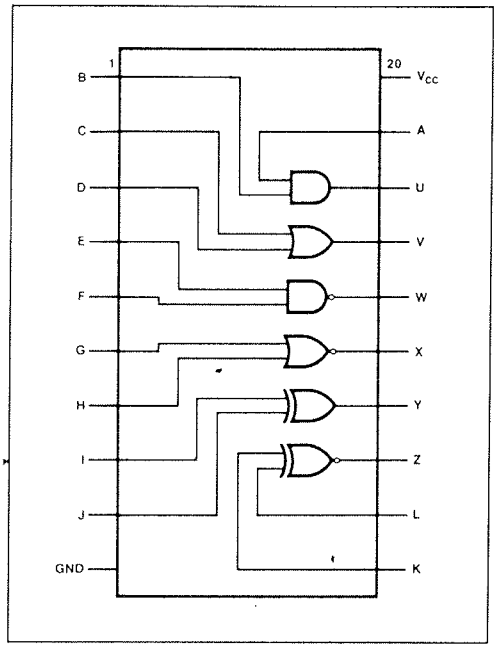

Fig. 2 GAL16V8 Basic Gates Pinout

```
Name     GATES;
Partno   00001;
Date     29/06/87;
Revision 01;
Designer ALBERTO;
Company  SGS-THOMSON Microelectronics
Assembly PC AT;
Location PLZN1;
FORMAT   JEDEC;
/*************************************************************************/
/*                                                                       */
/*  BASIC GATES  :  INPUT FILE                                           */
/*                                                                       */
/*************************************************************************/
/*  Allowable Target Device Types: G16V8                                 */
/*************************************************************************/

/**  Inputs  **/

Pin   [19,1] = [A,B]     ;                /* INPUTS AND          */
Pin    [2,3] = [C,D]     ;                /* INPUTS OR           */
Pin    [4,5] = [E,F]     ;                /* INPUTS NAND         */
Pin    [6,7] = [G,H]     ;                /* INPUTS NOR          */
Pin    [8,9] = [I,J]     ;                /* INPUTS XOR          */
Pin  [11,12] = [K,L]     ;                /* INPUTS XNOR         */

/**  Outputs  **/

Pin       18 = U         ;                /* OUTPUT AND          */
Pin       17 = V         ;                /* OUTPUT OR           */
Pin       16 = W         ;                /* OUTPUT NAND         */
Pin       15 = X         ;                /* OUTPUT NOR          */
Pin       14 = Y         ;                /* OUTPUT XOR          */
Pin       13 = Z         ;                /* OUTPUT XNOR         */

/** Declarations and Intermediate Variable Definitions **/



/**  Logic Equations  **/

     U = A & B           ;                /*   A N D             */
     V = C # D           ;                /*   O R               */
     W = !(E & F)        ;                /*   N A N D           */
     X = !(G # H)        ;                /*   N O R             */
     Y = I $ J           ;                /*   X O R             */
     Z = !(K $ L)        ;                /*   X N O R           */
```

Fig. 3 CUPL Design Input File

the verify cycle are executed. The verify cycle check programming and margins conservatively such that a minimun data reteintion of 20 years is ensured.

```
Syn   2192 - Ac0   2193 x

Pin #19  2048  Pol x  2120  Ac1 -     Pin #15  2052  Pol x  2124  Ac1 x
 0000 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1024 ----------------x---------------
 0032 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1056 --------------------x-----------
 0064 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1088 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0096 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1120 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0128 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1152 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0160 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1184 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0192 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1216 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0224 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1248 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #18  2049  Pol -  2121  Ac1 x     Pin #14  2053  Pol -  2125  Ac1 x
 0256 --x---x-------------------------   1280 ------------------------x----x--
 0288 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1312 -------------------------x--x---
 0320 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1344 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0352 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1376 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0384 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1408 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0416 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1440 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0448 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1472 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0480 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1504 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #17  2050  Pol -  2122  Ac1 x     Pin #13  2054  Pol x  2126  Ac1 x
 0512 x-------------------------------   1536 ---------------------------x--x-
 0544 ----x---------------------------   1568 --------------------------x----x
 0576 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1600 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0608 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1632 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0640 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1664 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0672 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1696 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0704 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1728 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0736 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1760 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Pin #16  2051  Pol x  2123  Ac1 x     Pin #12  2055  Pol x  2127  Ac1 -
 0768 --------x---x-------------------   1792 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0800 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1824 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0832 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1856 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0864 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1888 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0896 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1920 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0928 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1952 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0960 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   1984 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0992 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx   2016 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

                                  LEGEND    X : fuse not blown
                                            - : fuse blown
```

Fig. 4 *CUPL Fuse Plot*

```
CSIM  Version 2.11b Serial# 5-00001-154
Copyright (C) 1983,1986 Personal CAD Systems, Inc.
CREATED Fri Sep 18 17:15:04 1987

LISTING FOR SIMULATION FILE: GATES.si

   1: Name            GATES;
   2: Partno          00001;
   3: Revision        01;
   4: Date            29/06/87;
   5: Designer        ALBERTO;
   6: Company         SGS-THOMSON Microelectronics
   7: Location        PLZN1;
   8: Assembly        PC AT;
   9: Format          JEDEC;
  10: /*****************************************************************/
  11: /*                                                               */
  12: /*  BASIC GATES   :   SIMULATION FILE                            */
  13: /*                                                               */
  14: /*****************************************************************/
  15: /*      Target Devices: G16V8                                    */
  16: /*****************************************************************/
  17:
  18:
  19: Order: A,B,%2,U,%3,C,D,%2,V,%3,E,F,%2,W,%3,G,H,%2,X,
                 %3,I,J,%2,Y,%3,K,L,%2,Z;
  20:
  21:

==========================================================================
                          Simulation Results
==========================================================================

                               N                             X
              A                A         N         X         N
              N      O         N         O         O         O
          AB  D  CD  R    EF   D    GH   R    IJ   R    KL   R

0001:     00  L  00  L    00   H    00   H    00   L    00   H
0002:     10  L  10  H    10   H    10   L    10   H    10   L
0003:     01  L  01  H    01   H    01   L    01   H    01   L
0004:     11  H  11  H    11   L    11   L    11   L    11   H
```

Fig. 5 *CUPL Simulation File*

```
CUPL         2.11b  Serial# 5-00001-154
Device       g16v8s  Library DLIB-f-23-8
Created      Fri Sep 18 17:07:25 1987
Name         GATES
Partno       00001
Revision     01
Date         29/06/87
Designer     ALBERTO
Company      SGS-THOMSON Microelectronics
Assembly     PC AT
Location     PLZN1
*QP20
*QF2194
*QV4
*G0
*F0
*L0256 11011101111111111111111111111111
*L0512 01111111111111111111111111111111
*L0544 11110111111111111111111111111111
*L0768 11111111011101111111111111111111
*L1024 11111111111111111011111111111111
*L1056 11111111111111111111101111111111
*L1280 11111111111111111111111011110111
*L1312 11111111111111111111111110110111
*L1536 11111111111111111111111111101101
*L1568 11111111111111111111111111011110
*L2048 01100100000000000000000000000000
*L2112 00000000100000001111111111111111
*L2144 11111111111111111111111111111111
*L2176 1111111111111110
*C2F02
*P 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*V0001 000000000N00HLHHLL0N
*V0002 01010101 0N10LHLHHL1N
*V0003 10101010 1N01LHLHHL0N
*V0004 11111111N11HLLLHH1N
*DF00
```

Fig. 6 CUPL JEDEC File with Test Vectors

```
*******************************************************
                         GATES
*******************************************************
CUPL         2.11b Serial# 5-00001-154
Device       g16v8s  Library DLIB-f-23-8
Created      Fri Sep 18 17:07:24 1987
Name         GATES
Partno       00001
Revision     01
Date         29/06/87
Designer     ALBERTO
Company      SGS-THOMSON Microelectronics
Assembly     PC AT
Location     PLZN1

=======================================================
                          Expanded Product Terms
=======================================================

U =>
    A & B

V =>
    C
  # D

W =>
    E & F

X =>
    G
  # H

Y =>
    I & !J
  # !I & J

Z =>
    K & !L
  # !K & L
```

Fig. 7 CUPL Expanded Product Terms

```
==============================================================================
                              Symbol Table
==============================================================================

Pin Variable                                    Pterms   Max     Min
Pol   Name              Ext     Pin     Type     Used    Pterms  Level
--- --------            ---     ---     ----    ------   ------  -----

     A                          19       V        -        -       -
     B                          1        V        -        -       -
     C                          2        V        -        -       -
     D                          3        V        -        -       -
     E                          4        V        -        -       -
     F                          5        V        -        -       -
     G                          6        V        -        -       -
     H                          7        V        -        -       -
     I                          8        V        -        -       -
     J                          9        V        -        -       -
     K                          11       V        -        -       -
     L                          12       V        -        -       -
     U                          18       V        1        8       1
     V                          17       V        2        8       1
     W                          16       V        1        8       1
     X                          15       V        2        8       1
     Y                          14       V        2        8       1
     Z                          13       V        2        8       1

LEGEND    F : field      D : default variable       M : extended node
          N : node       I : intermediate variable  T : function
          V : variable   X : extended variable      U : undefined
```

Fig. 8 CUPL Symbol Table

```
            ***********     ***********
            *             * *           *
         ****                          ****
       B * 1                           20 * VCC
         ****                          ****
            *          GATES            *
         ****                          ****
       C * 2                           19 * A
         ****                          ****
            *                           *
         ****                          ****
       D * 3                           18 * U
         ****                          ****
            *                           *
         ****                          ****
       E * 4                           17 * V
         ****                          ****
            *                           *
         ****                          ****
       F * 5                           16 * W
         ****                          ****
            *                           *
         ****                          ****
       G * 6                           15 * X
         ****                          ****
            *                           *
         ****                          ****
       H * 7                           14 * Y
         ****                          ****
            *                           *
         ****                          ****
       I * 8                           13 * Z
         ****                          ****
            *                           *
         ****                          ****
       J * 9                           12 * L
         ****                          ****
            *                           *
         ****                          ****
     GND * 10                          11 * K
         ****                          ****
            *                           *
            ****************************
```

Fig. 9 CUPL Chip Diagram

References:

SGS-THOMSON: Programmable logic manula-GAL products

DATA I/O: Programmable logic

Zlatko Bele,dipl.ing.
MIKROIKS d.o.o.
Titova 36A
61000 Ljubljana

Prispelo: 05.05.1990 Sprejeto: 20.11.1990