SIGNETICS' IFL DEVELOPMENT SYSTEM

PROJECT CONTENT

The configuration of the product that would meet Signetics' needs is described below as being Boolean Equation Entry plus other features that highlight the design advantages of IFL. This BEE+ product consists of a programmer and the software needed to support it as well as the logic manipulation program to fill the needs of our customers.

INPUT STRUCTURE

The input formats that will be used with BEE+ allow for a wide range of data input.

1. SUM-OF-PRODUCTS EQUATIONS This form of entry is the same as PALASM. It inputs the logic equations for both combinational and sequencial circuits with the variable name that the user has assigned to that signal. The structure of the equations would be constrained to be in the sum-of-products format. This is the natural form for FPLA's. The syntax and symbols used will be determined jointly by Signetics and the vendor. The attached format is an example of the form this format might have.

2. STATE MACHINE NOTATION To take advantage of the FPLS structure of IFL, BEE+ will allow the user to define a sequencer in terms of sequencial state numbers and transition numbers. The designer will not have to go through the sometimes impossible task of converting from this format to the excitation equations.

3. TRUTH TABLE This is similar to the input format now used by most programmer manuf. to program IFL. BEE+ will allow the use of H and L's to define the input polarity and could support the hex, octal or decimal equivilants of the input word, transition or state number.

4. GRAPHICS For the TTL designer, BEE+ will have a graphics package with which he can place and route macros of the complexity of MSI circuits. This will give him the level of user-friendliness that he requires. The graphics could be a simple, low-resolution form that presents outlines and reads the position of the cursor.

5. GENERAL PURPOSE To have maximum utility, this system must support all Programmable Logic Devices. Signetics will supply all specification for programming IFL and it will be the responsibility of the vendor to obtain similar infornation from the other IC manufacturers.

FUNCTIONS PERFORMED

BEE+ will utilize the above input structure to access the various levels of programming that will allow the user to select any of the following functions.

1. ASSEMBLE IFL CODE One major purpose of BEE+ is to assemble any of the various input formats into the serial format needed for the different programmers that support IFL.

2. PROGRAMMER DOWNLOAD This function will transmit the assembled IFL code to the programmer in the format that it needs. This function could also be used to send the code via a data link to a remote programming center.

3. LOGIC SIMULATION If the ouput response to an input word is needed to generate functional patterns, this function could be used. Fault grading could be included so that the user might transmit to us or his own programming center the necessary test vectors to guarantee the operation of the programmed part.

4. LOGIC MINIMIZATION The top of the support package will take the input logic and reduce it to a minimum configuration for any IFL device. This is a very advanced program and is available now only on main frame computers.

OUTPUT FILES

To complete the user's documentation, BEE+ will output in any of the formats listed below. The user may select whichever format he wants to print or transfer to his company's central computer memory.

1. SUM-OF-PRODUCTS EQUATIONS This is the same as the input format.

2. STATE MACHINE The output may be in either the coded state machine notation or in the state machine diagram that is used in many firms as basic documentation.

3. TRUTH TABLE This output file would replicate the

program table that IFL uses now.

4. FUNCTIONAL TEST PATTERNS For the user who needs to specify the patterns by which the device is to be tested, BEE+ will output a file of test vectors.

5. SERIAL PROGRAMMER FILE BEE+ will construct a file for transmission to a programmer.

ENVIRONMENT

To offer the level of user-friendliness required by the different groups of customers, BEE+ will need to have several levels of operation. The choice of which is to used for any input session is selectable by the user. In this way, the inexperienced user may select a full prompting and menu driven format while the sophisticated user may select a faster, more direct style. This selectability may not add much to the initial task of program design but will add greatly to the utility of the program by the largest group of users.

COMPUTER

BEE+ must be able to be used on virtually any computer from the smallest micro to a main frame. Some of the more complex programs may not be able to be run on the small machines but the goal of the designer should be to have some level of the software that will run on the various levels. This may mean that the minimization program takes more computer time or runs with fewer inputs on the smaller machines. This may limit the overall utility but some level of operation is better than none for the user. The software must also be written in a language and operating system that allows for maximim transportability. In this way, a data link can quickly install the program in any machine.

The minimum support required is:

COMPUTERS	OPERATING SYSTEMS
APPLE II (with SOFTCARD)	CP/M
IBM PC	UNIX
VAX 11/780	VM/S
INTEL MDS	
IBM COMPATIBLE MAIN-FRAME	

PROGRAMMER

The hardware portion of BEE+ will satisfy the need of the designer for a dedicated programmable logic programmer. The features of the programmer are:

1. STANDARD RS-232 INTERFACE The use of this standard will allow the programmer to be used in the largest number of systems with a minimum of special adaptation.

2. SMALL AND LOW COST   The size of the unit should allow for easy transportation by the field salesman or FAE in his briefcase. This will give the unit maximim utility. The cost of the unit should place it within the price easily authorized by the working level engineer. This would usually indicate a manufacturing cost of under $1000..

3. COMPUTER PERIPHERAL   This feature will give the unit the ability to talk with the user's central computer to build files for transfer between groups and between companies. This will enhance the ability of BEE+ to satisfy the documentation needs of most of our customers. The unit will be able to function with any level computer from a micro to a main frame.

4. HANDLER INTERFACE   The programmer must be of a size that will easily mate to industry standard handlers and must have the interface electronics to allow for production operation with the handler. This would include a start-up test to check the continuity of the handler connection.

5. PRODUCTION USE   For use in the production environment, the programmer must be able to perform functions such as load from master, load from tape or file and program without the host computer.

6. UNIVERSIAL   The capability to program all Programmable Logic Devices will help the acceptance of this system.

PRODUCT SUMMARY

IDEAL BEE+

FEATURES	PRIORITY
SOFTWARE	A1
INPUTS ALLOWED	
SUM-OF-PRODUCTS EQUATIONS	1
STATE MACHINE NOTATION	1
TRUTH TABLE FILE	1
(IN BINARY, HEX OR DECIMAL)	4
SIMPLE GRAPHICS TO MSI LEVEL	2
FUNCTIONS DONE	
ASSEMBLE ANY PLD CODE	1
DOWNLOAD TO ANY PROGRAMMER	1
LOGIC SIMULATION	1
LOGIC MINIMIZATION	3
FAULT GRADING	3
OUTPUT FILES	
SUM-OF-PRODUCTS EQUATIONS	1
STATE MACHINE CHART	1
TRUTH TABLE	1
FUNCTIONAL TEST PATTERNS	2
SERIAL PROGRAMMER FILE	1
ENVIRONMENT	
SELECTABLE DEGREE OF USER-FRIENDLINESS	2
HIGH LEVEL OF DIAGNOSTICS	1
HIGH LEVEL OF DOCUMENTATION	1
COMPUTER	
SOFTWARE TRANSPORTABLE	1
WORK ON "ANY" MACHINE	1
HARDWARE	2
PROGRAMMER	
STANDARD RS-232 INTERFACE	1
SMALL	1
LOW-COST	1
COMPUTER PERIPHERAL	1
SUPPORT ALL PLD'S	1

INTEGRATED FUSE LOGIC ENTRY FORMAT

This attachment represents the preferred format for the Signetics' IFL Boolean Equation Entry program, BEE+. The symbology and structure is suggested as the most universal language for describing logic equations in the computer environment. Signetics has surveyed its own designers and those to whom Signetics sells IFL and this proposed language and format best meets these designers' needs.

STRUCTURE

The session and resulting document are divided into three basic sections for convenience; the header, pin list and logic entry.

HEADER

The HEADER BLOCK should contain the overhead information needed to differentiate the design and revision level from other designs. It should include the documentation numbers normally used by companies to identify devices in their system. This would include Drawing Number, Revision Level, Device Type, Date, Company Name, Designer's Name, and Device Function. The organization of this and similar data should be readily found on the final documentation and be a part of the data base on the mass storage media.

PIN LIST

The information relating names and function to a specific pin is contained in the PIN BLOCK. A default or standard name and function is given to the pin. The default listing for the pin signal name may be changed by the user to any name that is desired. The user may also select from a list of permitted functions for each pin. The function selected by the user will control the polarity option for all outputs and the direction control function for the bidirectional ports of the IFL-20. The names for any COMMON functions are also given in this block.

LOGIC ENTRY

Logic equations, state machine notations or flow chart information is entered into the LOGIC ENTRY BLOCK to describe each output, flip-flop or sequencial transition. This block should be fully prompting to prevent the unfamiliar user from making errors. The definition of the logical operators is critical to the acceptance of the language and will be mutually defined during the program generation.

The use of and the structure of each of these blocks will be demonstrated in the examples that follow.

EXAMPLES OF DATA ENTRY

This format could be used for the CRT entry of the data for any programmable logic device. The computer prompts are shown in all upper case letters and any comments are shown in parenthesis.

(HEADER BLOCK)

```
          DATE: 7/15/82                DESIGNER: BWS
 DESIGN NUMBER: 1234A                  FUNCTION: CPU INTERFACE
DRAWING NUMBER: 10-234-1098           REV LEVEL: B
   DEVICE TYPE: 82S100                  COMPANY: Better Computers
```

(Any updates would reference the DESIGN NUMBER.)

(The HEADING BLOCK would be the same for all device types.)

(82S100 entry example.)
(PIN BLOCK)

```
             PIN LISTING FOR 82S100 DESIGN NUMBER 1234A

PIN    FUNCTION   SIGNAL              PIN    FUNCTION   SIGNAL
NO.               NAME                NO.               NAME
1        N/C                          15        O         F0
2         I         I0                16        O         F1
3         I         I1                17        O         F2
4         I         I2                18        O         F3
5         I         I3                19      ENABLE      EN
6         I         I4                20        I         I8
7         I         I5                21        I         I9
8         I         I6                22        I         I10
9         I         I7                23        I         I11
10        O         F8                24        I         I12
11        O         F7                25        I         I13
12        O         F6                26        I         I14
13        O         F5                27        I         I15
14      GROUND      GND               28      POWER       VCC
COMMON TERMS: (User enters string of names.);
FUNCTION CHANGES ARE ALLOWED ON OUTPUTS AS:
ACTIVE HIGH = O (Default condition.)
ACTIVE LOW = NO (Programs polarity fuse.)
```

(The program prints the form with the default values and places the cursor at the input field locations where change is allowed. After changing any function and signal names, the user is given the opportunity to name any common terms. These will be defined later in the equation block.)

```
             LOGIC ENTRY FOR 82S100 DESIGN NUMBER 1234A

COMMON TERMS:
     COM1:=(Logic expression);
     COM2:=(Logic expression);
OUTPUTS:
      F0:=(Expression-may include name of common terms);
      F1:=(Expression);
      F2:=(Expression);
```

(Prompting will continue until all common terms and outputs are defined. If a line feed(null entry) is given in response to an output prompt, that output is left unprogrammed.)

INTEGRATED FUSE LOGIC ENTRY FORMAT

```
(82S105 Entry example.)

 (PIN BLOCK)

              PIN LISTING FOR 82S105 DESIGN NUMBER 1234A

PIN    FUNCTION    SIGNAL                PIN     FUNCTION    SIGNAL
NO.                 NAME                 NO.                  NAME
1        CLOCK      CLK                   15        O          FO
2          I         IO                   16        O          F1
3          I         I1                   17        O          F2
4          I         I2                   18        O          F3
5          I         I3                   19      ENABLE      PR/OE
6          I         I4                   20        I          I8
7          I         I5                   21        I          I9
8          I         I6                   22        I          I10
9          I         I7                   23        I          I11
10         O         F4                   24        I          I12
11         O         F5                   25        I          I13
12         O         F6                   26        I          I14
13         O         F7                   27        I          I15
14       GROUND     GND                   28      POWER        VCC

COMP ARRAY           CA
STATE MACHINE FLIP-FLOPS:
 UTPUT       NAME            OUTPUT      NAME
  O           QO               3          Q3
  1           Q1               4          Q4
  2           Q2               5          Q5
COMMON TERMS: (STRING);
FUNCTION CHANGES ARE ALLOWED ON OUTPUTS AS:
               ACTIVE HIGH = O
                ACTIVE LOW = NO

 (LOGIC BLOCK)
              LOGIC ENTRY FOR 82S105 DESIGN NUMBER 1234A

COMMON TERMS:
     COM1:= (Logic expression);
     COM2:= (Logic expression);
(Prompting will continue until all common terms are defined.)
COMPLEMENT ARRAY:
     CA:= (Expression);
ENTER "EQ" FOR EQUATION ENTRY OR "SM" FOR STATE MACHINE
sm (User chose state machine nomenclature entry.)
TRANSITION 1:
    IF PS:= (Expression);
     & IN:= (Expression);
  THEN NS:= (Expression);
    & OUT:= (Expression);
TRANSITION 2:
    IF PS:=
(Null entry causes prompting to terminate.)
```

```
(82S153 FPLA EXAMPLE)

           PIN LISTING FOR 82S153 DESIGN NUMBER 1234A

PIN    FUNCTION   SIGNAL            PIN     FUNCTION   SIGNAL
NO.                NAME             NO.                 NAME
1          I        I0               11         B         B1
2          I        I1               12         B         B2
3          I        I2               13         B         B3
4          I        I3               14         B         B4
5          I        I4               15         B         B5
6          I        I5               16         B         B6
7          I        I6               17         B         B7
8          I        I7               18         B         B8
9          B        B0               19         B         B9
10      GROUND      GND              20       POWER       VCC
COMMON TERMS: (LISTING);
 FUNCTION CHANGES ARE ALLOWED FOR BI-DIRECTIONAL PORTS AS:
 ACTIVE HIGH OUTPUT = O (Program all D fuses for this pin.)
 ACTIVE LOW OUTPUT = NO (Program D fuses & polarity fuse.)
 ACTIVE HIGH I/O PORT = B (Default condition.)
 ACTIVE LOW I/O PORT = NB (Program polarity fuse.)
 INPUT = I (Leave all D fuses intact & do not prompt.)
(Equations for direction control terms will receive prompting in
the equation block only if the pin's function is a "B" or "NB".)

           LOGIC ENTRY FOR 82S153 DESIGN NUMBER 1234A

COMMON TERMS:
   COM1:= (Expression);
   COM2:= (Expression);
OUTPUT O9:= (Expression);
OUTPUT O6:= (Expression);
I/O PORT B8:= (Expression);
 CONTROL D8:= (Expression);

I/O PORT B7:= (Null entry leaves OR term unprogramed.)

I/O PORT B0:= (Expression);
 CONTROL D0:= (Expression);
```