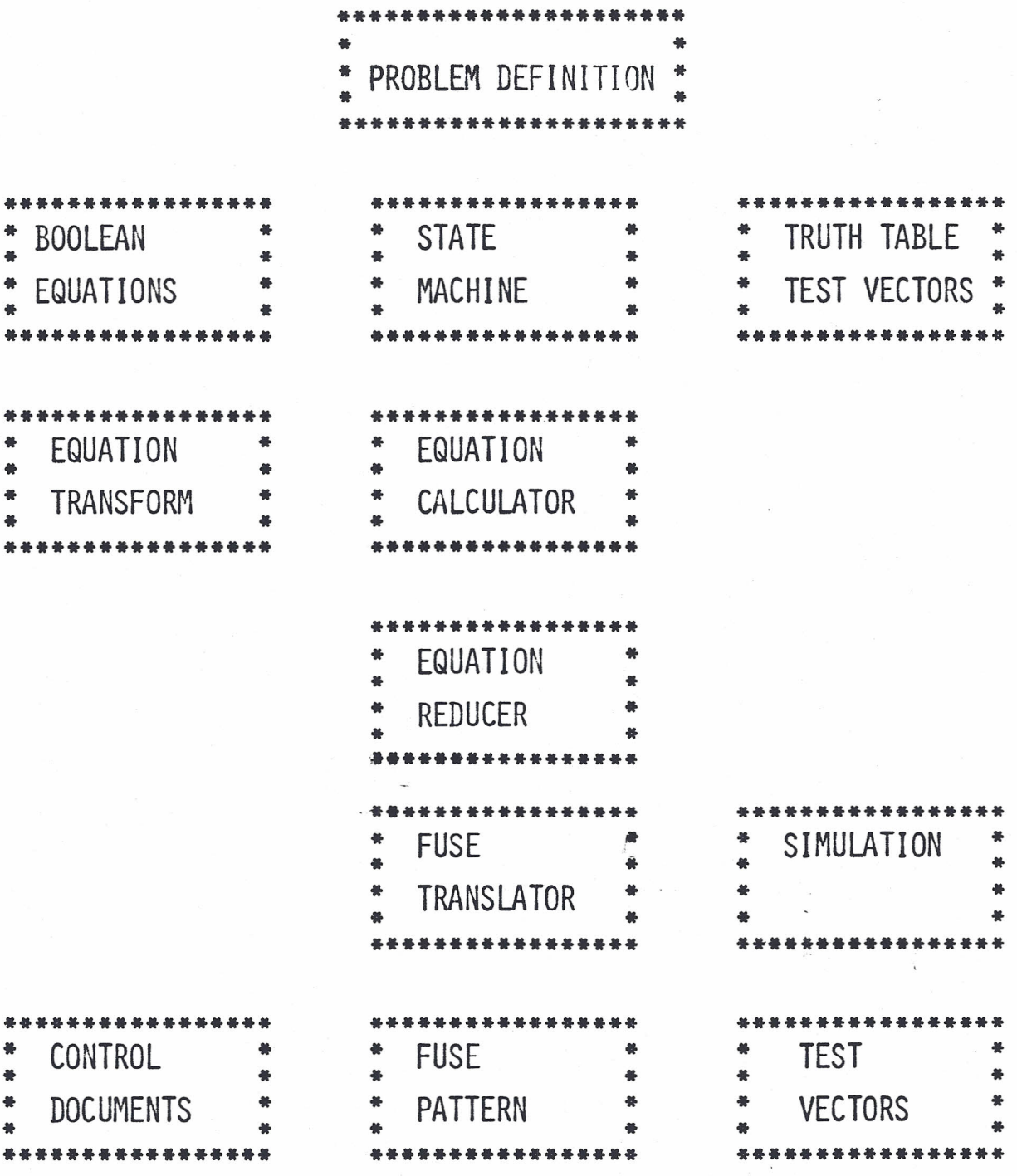
ABEL : AN INTEGRATED APPROACH

MAJOR OBJECTIVES

CONVERT A LOGIC DESCRIPTION INTO A FUSE PATTERN AND TEST VECTORS.

SUPPORT ALL EXISTING LOGIC DEVICES (PROMS, PAL, IFL).

OPERATE ON A COMPUTER SYSTEM AVAILABLE TO DATA I/O'S CUSTOMERS.


PROBLEM DEFINITION
BOOLEAN EQUATIONS
STATE MACHINE
TRUTH TABLE TEST VECTORS
EQUATION TRANSFORM
EQUATION CALCULATOR
EQUATION REDUCER
FUSE TRANSLATOR
SIMULATION
CONTROL DOCUMENTS
FUSE PATTERN
TEST VECTORS

DATA I/O

PRIORITIES

1 ABEL PROTOTYPE, FOR IN-HOUSE USE ONLY.

- TRUTH TABLE ENTRY
- BOOLEAN SUM-OF-PRODUCT & PRODUCT-OF-SUM EQUATIONS
- RANDOM LOGIC BOOLEAN EQUATIONS (NO FEEDBACK)
- SYNCHRONOUS STATE MACHINE INPUT (SUBSET)
- MULTIPLE PROJECTS IN ONE DEVICES
- MULTIPLE DEVICES (MANUAL PARTITIONING)
- SUPPORT PALs FPLAs FPLSs ROMs
- ERROR REPORTING (SYNTACTICAL AND LOGICAL)
- DATA TRANSFER WITH JEDEC STANDARD

2 ABEL FIRST RELEASE.

- PALASM TO ABEL TRANSLATOR
- MACROS
- "INCLUDE" FILES
- PALASM STYLE OUTPUT DOCUMENTATION
- SELECTABLE LEVELS OF ERROR REPORTING

3 FUTURE RELEASES.

- AUTOMATIC GENERATION OF TEST VECTORS
- FAULT GRADING
- ASYNCHRONOUS STATE MACHINE INPUT (SUBSET)
- AUTO PARTITIONING OF MULTIPLE DEVICES
- ON-LINE HELP
- SOFTWARE TRANSPORTABLE TO MOST POPULAR MACHINES

4 NOT IN PRESENT PLANS.

- ABEL PROMPTING EDITOR (INTERACTIVE ABEL)
- GRAPHIC INPUT AND OUTPUT

ABEL SYNTAX

FREE-FORMAT (NO LINE OR COLUMN NUMBERS)

SELF-DOCUMENTING (NO CRYPTIC KEYWORDS)

MODULE BEGIN AND END CONSTRUCT

ONE-PASS COMPILATION

CONSTANT DECLARATIONS

LIBRARIES OF MACROS

PINS AND NODES CAN BE DEFINED IN ANY ORDER

GROUPS OF PINS CAN BE DEFINED IN SET-LIKE NOTATION

CLOCKED AND SEQUENTIAL ASSIGNMENTS

UNARY OPERATORS : NOT, MINUS

BINARY OPERATORS : AND, OR, XOR, SHIFT

ARITHMETIC OPERATORS : ADD, SUB, MUL, DIV, MOD

EQUALITY OPERATORS : LT, LE, GT, GE, EQ, NE

PARENTHESIZED SUB-EXPRESSIONS

BIT PATTERNS : BINARY, OCTAL, DECIMAL, HEX, ASCII

DATA I/O

```
/*****************************************/
/* A CONCEPTUAL ABEL EXAMPLE             */
/* 06 DEC 1982                           */
/*****************************************/

HEADER
 DEVICE XYZ123;

 PINS
     MA0, MA1, MA2, A0, A1, A2, A8, A9, A10,
     ROW, HOLD, TOGGLE
 END_PINS;
END_HEADER.

FUNCTION MUX( A, B, SELECT);
    MUX = A * SELECT + B * /SELECT;
END_FUNCTION MUX.

EQUATIONS
    MA0 = MUX( A0, A8,  ROW);
    MA1 = MUX( A1, A9,  ROW);
    MA2 = MUX( A2, A11, ROW);
END_EQUATIONS.
```

```
"      CONCEPTUAL STATE MACHINE EXAMPLE          "
"       ABEL              12 DEC 1982             "

TAPECON [P3, P2, P1, P0] : STATE_MACHINE;

STOP: STATE 1;
FAST_FWD, FWD, REWIND : STATE 2, 3, 4;
STOP_EOT, STOP_BOT : STATE 5, 6;
EMPTY, EJECT : STATE 7, 8;

STATE_DIAGRAM TAPECON;
STOP : " STOPPED, WAITING FOR COMMAND"
     CASE
     /IN_PLACE                  : EMPTY;
      IN_PLACE & /SELECT        : STOP;
      INPLACE & SELECT & FWD    : FWD;
      ..............

EMPTY :  "NO TAPE IN PLACE"
     CASE
     /INPLACE                   : EMPTY;
      INPLACE                   : REWIND;

      ..................................
EQUATIONS TAPECON;
SEL_EJECT = TAPECON [EMPTY];
TAPE_RUN  = TAPECON [FAST_FWD, FWD, REWIND];
```

DATA I/O

```
CONTROL  /* 3 BLOCKS IN TWO DEVICES */
   DEVICE1 = PDQ123;
   INCLUDE BLOCK1,BLOCK2;
   PINS   /RAS = 1,   A12 = 5,
          /CAS = 2,   A13 = 6,   /ROMSEL = 11,
           E   = 3,   A14 = 7,   /IOSEL  = 12,
           Q   = 4,   A15 = 8,
   END PINS;

   DEVICE2 = XYZ123;
   INCLUDE BLOCK3;
   PINS   RW = 1,    /IOSEL   = 3,
          E  = 2,    /WRITEIO = 4,
                     /READIO  = 5,
   END PINS;
END CONTROL;
```

```
CONTROL  /* 3 BLOCKS IN A SINGLE DEVICE  */
   DEVICE = BIG123;
   INCLUDE BLOCK1,BLOCK2,BLOCK3;
   PINS   /RAS = 1,   A12 = 5,
          /CAS = 2,   A13 = 6,   /ROMSEL = 11,
           E   = 3,   A14 = 7,   /WRITEIO =12,
           Q   = 4,   A15 = 8,   /READIO = 13,
    END PINS;
END CONTROL;
```

```
/*   A CONCEPTUAL EXAMPLE OF MULTIPLE DEVICES  */
/*   6 DEC 1982                                */

BLOCK1 /* CAS AND RAS */
  INPUTS E, Q, /RAMSEL;
  OUTPUTS /RAS, /CAS;
  EQUATIONS
   /RAS = E + Q;
   /CAS = E * Q * /RAMSEL;
  END EQUATIONS
END BLOCK1

BLOCK2 /* DECODE */
   INPUTS A12, A13, A14, A15;
   OUTPUTS /RAMSEL, /IOSEL, /ROMSEL;
   EQUATIONS
    /RAMSEL = /A15 + A15*/A14 ;
    /ROMSEL = A15*A14*A13*A12 ;
    /IOSEL  = A15*A14*A13*/A12 ;
  END EQUATIONS
END BLOCK1

BLOCK3 /* I/O READ WRITE  */
    INPUTS /IOSEL, RW, E;
    OUTPUTS /WRITEIO, /READIO;
    EQUATIONS
     /WRITEIO = /RW * E * /IOSEL;
     /READIO  =  RW * E * /IOSEL;
    END EQUATIONS;
END BLOCK3;
```

DATA I/O