COMPUTER SYSTEMS LABORATORY

STANFORD UNIVERSITY . STANFORD, CA 943054055

DESIGN AUTOMATION AT STANFORD

Edited by

W.M. vanCleemput

TECHNICAL REPORT NO. 178

July 1979

This work was supported by gift funds from
Hewlett-Packard Company and Tektronix, Inc.

SEL-79-031

DESIGN AUTOMATION AT STANFORD

Edited by

W. M. vanCleemput

TECHNICAL REPORT NO. 178

July 1979

COMPUTER SYSTEMS LABORATORY
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305

This work was supported by gift funds from Hewlett-Packard Company and Tektronix, Inc.

DESIGN AUTOMATION AT STANFORD

An Overview of Design Automation at Stanford University

W. M. vanCleemput

TENCHICAL REPORT NO. 178

July 1979

COMPUTER SYSTEMS LABORATCRY
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305

ABSTRACT

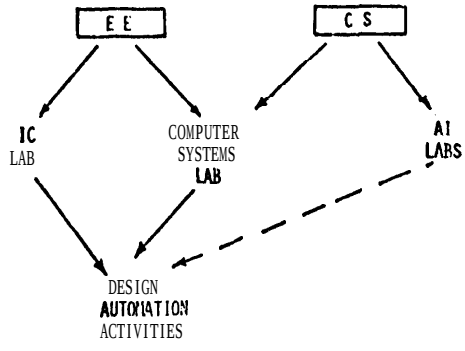
This report contains a copy of the visual aids used by the authors during the presentation oftheirwork at the First Workshop on Design Automation at Stanford, held on July 3 - 4, 1979.

The topics covered range from circuit level simulation and integrated circuit process modelling to high level languages and design techniques. The presentations are a survey of the activities in design automation at Stanford University.

TABLE OF CONTENTS

DESIGN AUTOMATION AT STANFORD

Overview of Design Automation at Stanford
(W.M. vanCleemput)



COMPUTER SYSTEMS LAB

ARCHITECTURE / NETWORKS

SOFTWARE SYSTEMS

PROGRAM VERIFICATION

RELIABLE COMPUTING

DESIGN AUTOMATION

10 FACULTY
50 STUDENTS

COMPUTING ENVIRONMENT

SLAC
2 IBM 168
1 IBM 91

STANFORD
IBM 3033
DEC 2055 (LOTS)

CS/EE
DEC 2060 (SCORE)
XEROX ETHERNET/ALTOS
IBM 4331 / S-1

DA
HP 3000
3 HP 1000
DEC VAX

COMPUTER DESIGN AT STANFORD

SUPER FOONLEY (A1) ; EARLY 70'S
- GREATLY IMPROVED PDP-10
- ECL INSTEAD OF TTL
- BASIS FOR DEC KL10 DESIGN

EMMY (FLYNN) : 1974-5
- UNIVERSAL EMULATION:
  - IBM360, CDC 6000, PDP-13, Nova, 8080, ETC.

STANFORD-1 : (1976 - ON)
- 16 CPU SYSTEM
- 36 BIT CPU, 10 MIPS
- LARGE SHARED MEMORY
- CROSS-BAR SWITCH

VGT (VIDEO-GRAPHICS TERMINAL):
- 1975 - ON (BASKETT)

SUNS

INTERACTIVE DRAWING SYSTEM

BLOCK/ LOGIC DIAGRAMS

WIRELISTER

SIMPLE → PC LAYOUT
SIMPLE → LOGIC SIMULATION

ASSEMBLER ON DEC10

HARD TO MAINTAIN

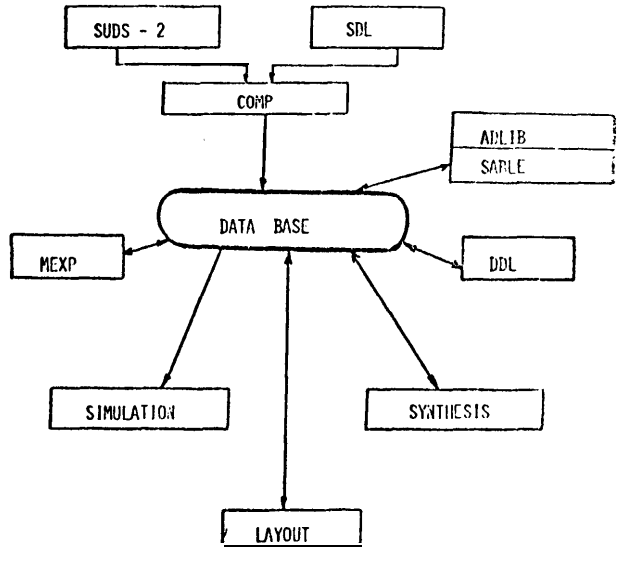
VECTOR GRAPHICS TERMINAL

SCALD

TOM MCWILLIAMS
CURT WIDDOES

STRUCTURED HARDWARE DESIGN

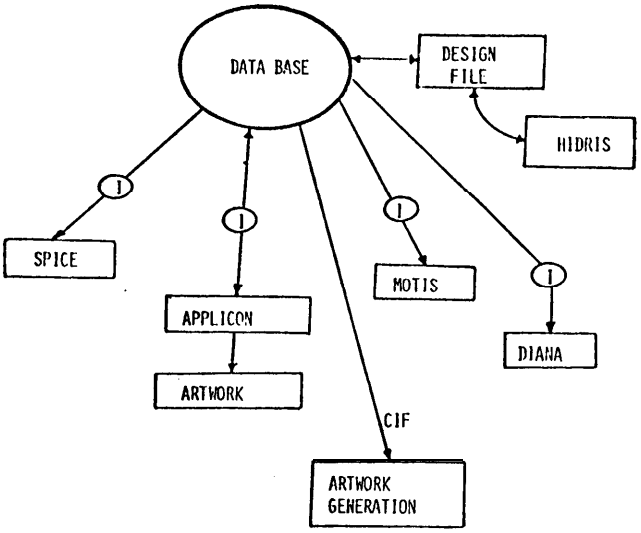
- SUDS FOR DESIGN ENTRY
- MACRO EXPANDER
- WIREWRAP DESIGN SYSTEM
- ECO PROCESSOR



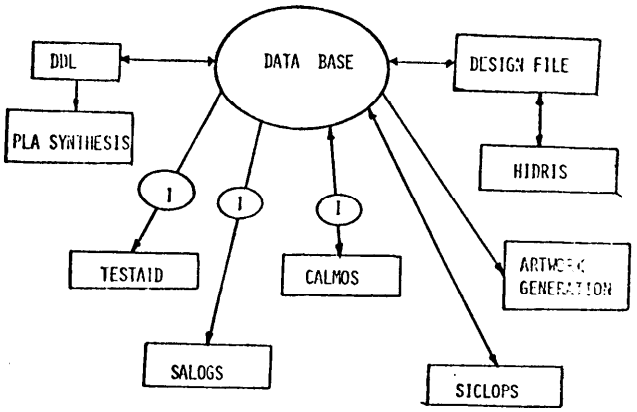
WHY BUILD DA SYSTEMS AT STANFORD ?

1. TO SUPPORT HARDWARE DESIGN (SUPPORT OTHER RESEARCH)
2. TO GIVE STUDENT REAL-LIFE ENVIRONMENT (EDUCATIONAL)
3. TO PROVIDE A TESTBED FOR NEW ALGORITHMS AND APPROACHES (RESEARCH IN DA)

IC CELL DESIGN

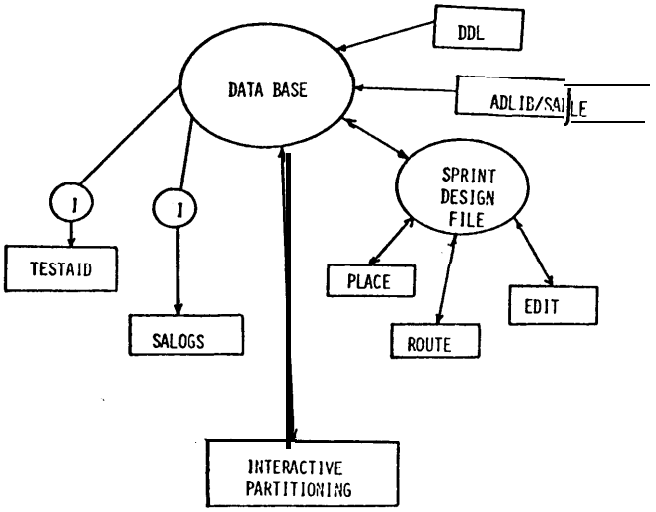


IC MACROCELL DESIGN

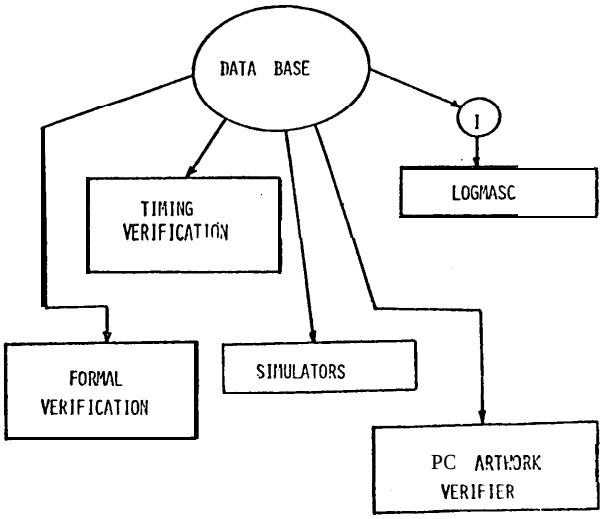


IC MASTERSLICE DESIGN

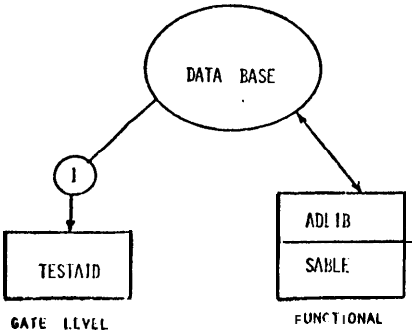
K DESIGN



DESIGN VERIFICATION



TESTING



The SCALD Design System
(T. McWilliams)

THE S-I (SCALD) DESIGN SYSTEM

THE GOAL

- To substantially reduce the large and growing design-cycle costs and time-lags . for high-performance computers.

CONVENTIONAL LOGIC DESIGN

- Designers use one or a few fixed levels of abstraction.
  - Gates, flip-flops, and other available devices.
- Computer-aided layout and wire-listing is often available.
- Computer-assisted drawing is sometimes available.
- Large-computer developments typically cost >100 man-years in the design stage.
  - Amdahl
  - Burroughs
  - CDC
  - IBM
- Design costs hrve usually been small fractions of total product cost (high volume systems).
- Economic penalty is in technological obsolescence of marketed systems.
  - Has become stiff only recently (LSI revolution).
  - industry is beginning to automate logic design.

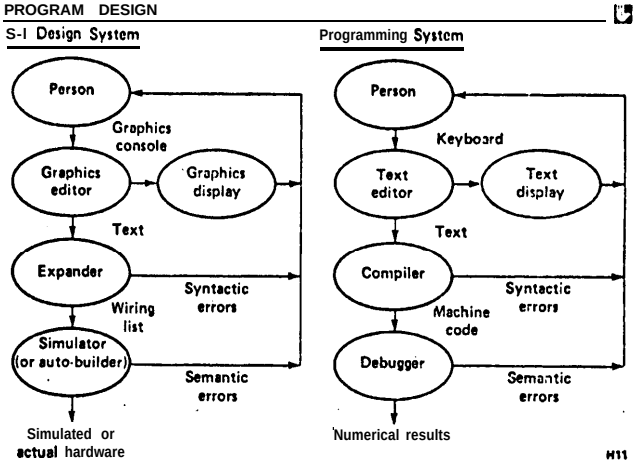
SCALD: THE FUNDAMENTAL DIFFERENCE

- SCALD is a high-level hardware-language compiler.
  - Closely analogous to a high-level software-language compiler.
  - Inputs a high-level description.
  - Outputs hardware.
- Arbitrary modules are designed,
  - each in terms of a few other modules,
  - relatively independently,
  - to communicate through well-defined Interfaces.
- SCALD edvantages are:
  - Increased understandability of resulting design.
    - Reducing design time.
    - Enhancmg design correctness.
  - Facilitation of final documentation.
  - Increased changeability of design.
  - Increased computer-verifiability of design.

SCALD OVERVIEW

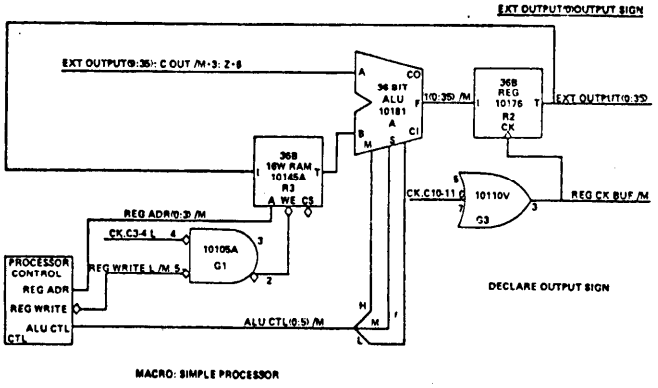
- Consists of 30,000 lines of Pascal source code.
- Accepts high-level graphics input.
- Automatically produces
  - design-aid documentation
  - automatic implementation tapes
  - implementation debug files
  - maintenance documentation
  - diagnostics
- Is largely technology independent.
- Is transportable.
- Is extendable.

COMPUTER AIDED LOGIC DESIGN VERSUS COMPUTER AIDED PROGRAM DESIGN

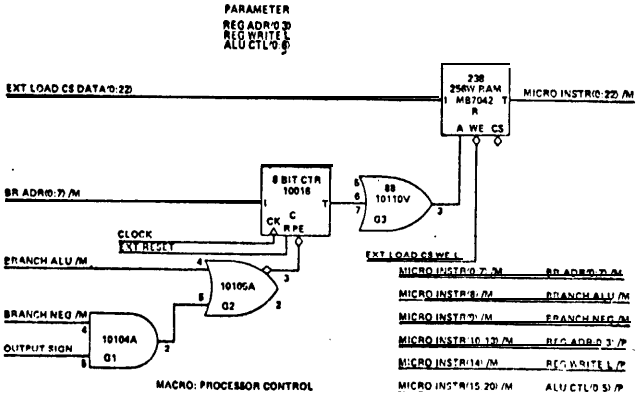


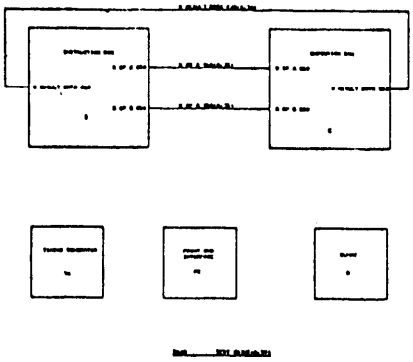
H11

EXAMPLE SCALD MACRO DEFINITION-SIMPLE PROCESSOR

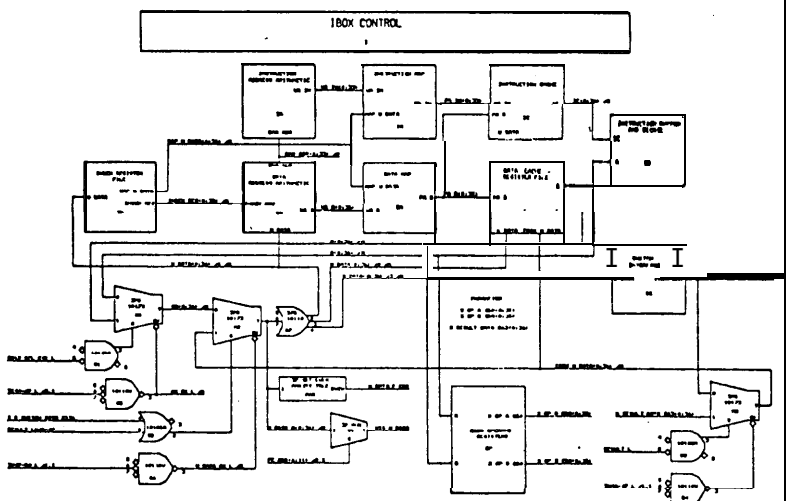


EXAMPLE SCALD MACRO DEFINITION-PROCESSOR CONTROL

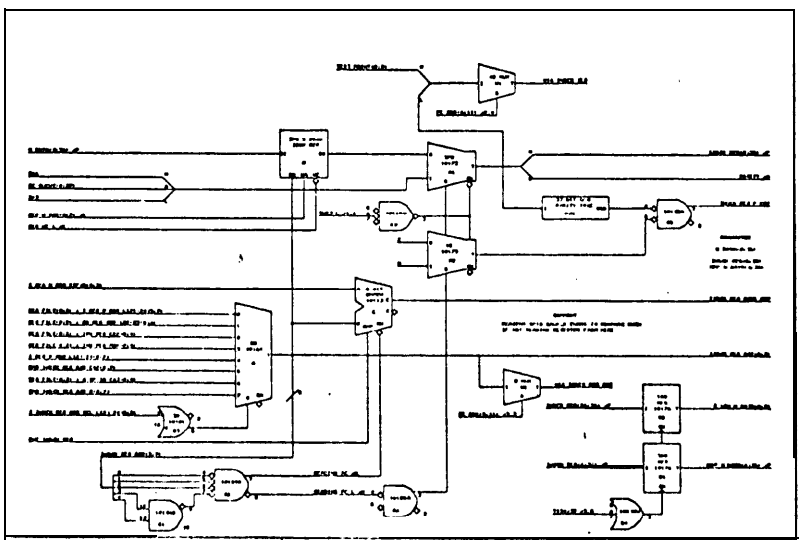




S-1 PROJECT	PROCESSOR 1/2	01-JAN-79 14:18 PROC1[MK1,S1]	
ORAYN BY:	PAGE OF	NUMBER	REV.
APPROVED BY:	PROJECT: PROC		3

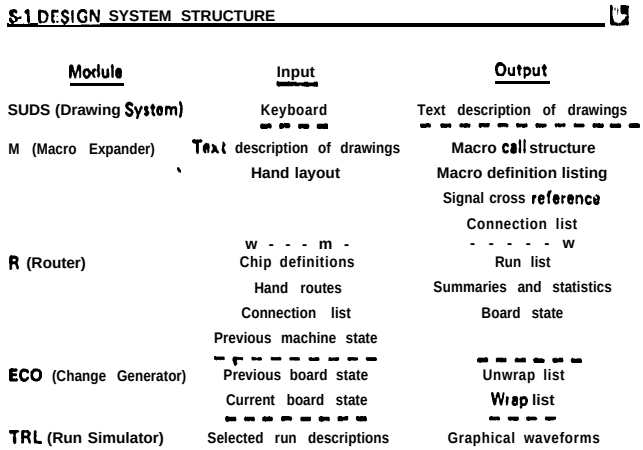


S-1 PROJECT	INSTRUCTION SOY	11-JAN-79 15:28 IBOX[MK1,S1]	
ORAYN BY:	PAGE OF	NUMBER	REV.
APPROVED BY:	PROJECT: IBOX		6



S-1 PROJECT	INDEX REGISTER FILE	01-JAN-79 15:29 IXREGF[MK1,S1]	
DRAWN BY:	PAGE OF	NUMBER	REV.
APPROVED BY:	PROJECT: IXREGF		10

S-1 DESIGN SYSTEM STRUCTURE



S-1 DESIGN SYSTEM STATISTICS

	SUDS	M	TRL:O	
Programming language used	FAIL	PASCAL	PASCAL	PASCAL
Program sizes (lines)	30K	10K	15K	2K
Programming time (man-months)	Unknown	5	6	2
Compute time per signal run, msec. (IBM 370/168)	30 hrs total (DEC KL-10)	28	64	1000

S-I MARK I DESIGN STATISTICS

Object machine size	5500 13-pin DIPS; 20K signal runs
Low level drawings (architecture-independent)	130
High level drawings (technology-independent)	150
Design time (man-months)	24
Layout time (men-months)	3

HIDRIS, An Interactive IC Design System
(E. Slutz)

HIERARCHICAL DESIGN STRATEGY

CURRENT IC DESIGN

OFTEN NO INTEGRATED SYSTEM
HIGH-VOLUME: HAND LAYOUT/DIGITIZING
LOW-VOLUEIE : MASTERSLICE OR ROW BASED
MINIMAL USE OF CIRCUIT SIMULATION
LITTLE OR NO HIGH-LEVEL VERIFICATION

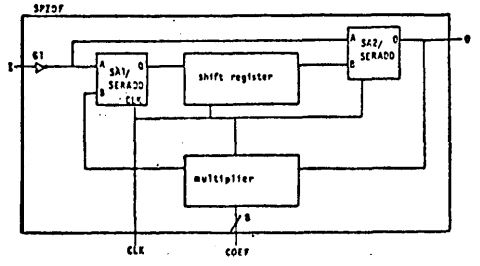
TOP-DOWN DESIGN
BOTTOM-UP IMPLEMENTION

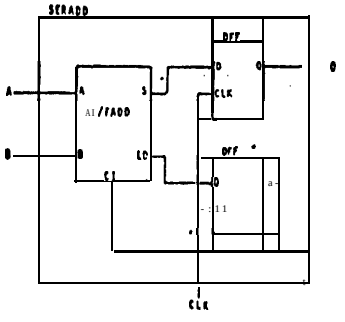
POTENTIAL ADVANTAGES .

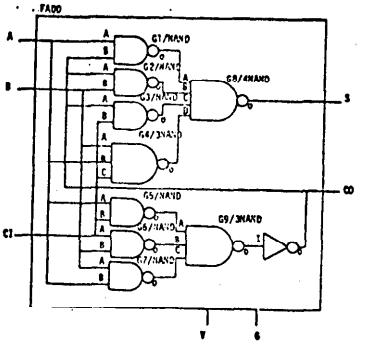
CORRECTNESS
SPEED
COMPLEX IC's (10-50K GATES)

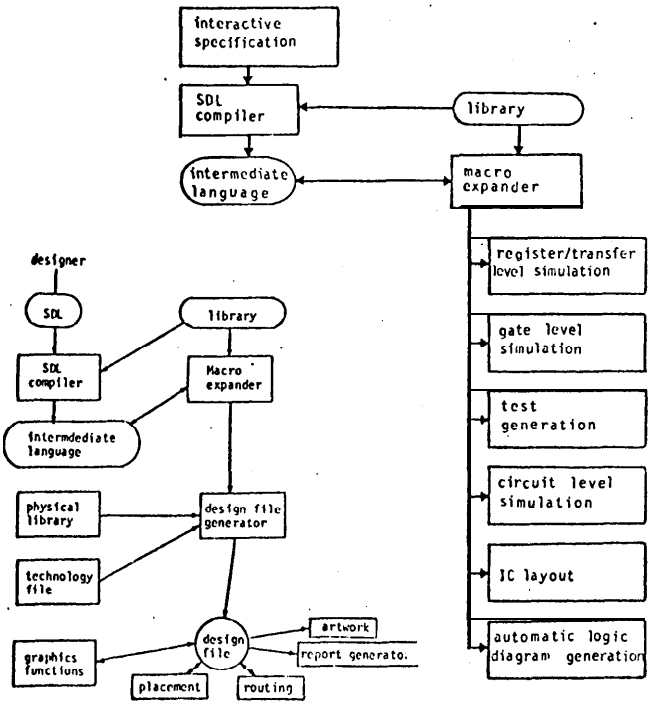
SYSTEM CHARACTERISTICS

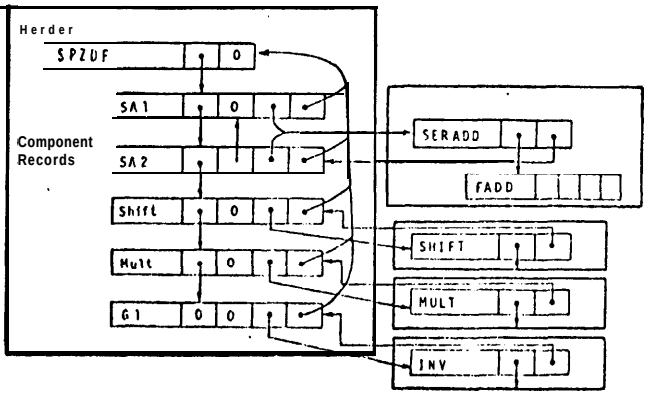
TOTAL DESIGN ENVIRONMENT
LOW & HIGH VOLUME
ALLOW COMBINATION OF MANUAL AND AUTOMATIC LAYOUT
-DYNAMIC DESIGN RULE VERIFICATION
DYNAMIC CHECK OF CONNECTIVITY
FUNCTIONAL VERIFICATION BY LOGIC & CIRCUIT SIMULATION
TECHNOLOGY INDEPENDENT
INTERACTIVE GRAPHICS ESSENTIAL
INTERFACE TO EXISTING PROGRAMS
-(DDL, TESTAID, SPICE, MOTIS, ETC.)
DESIGNER AT 1 LEVEL OF ABSTRACTION AT A TIME











GRAPHICS COMMANDS

CONSTRUCTIVE:

PLACE
BUILD
MODIFY
DELETE
CONNECT

VIEWING:

ZOOM
LAYERS
RULER
LABEL

TRANSLATION:

MOVE
ROTATE
MIRROR

ALGORITHMIC:

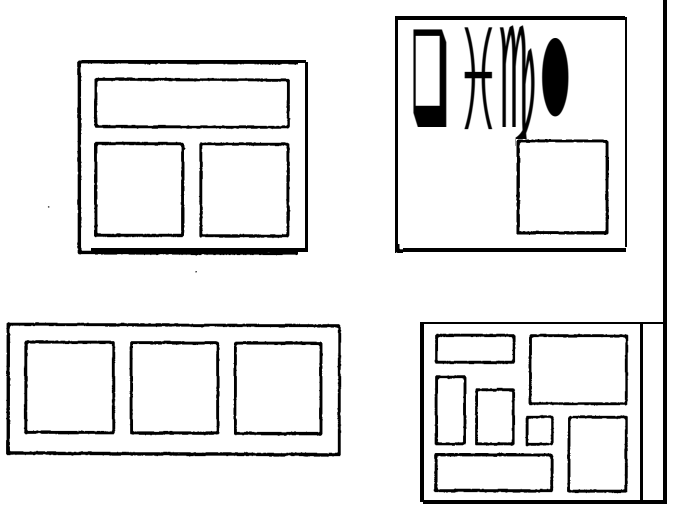
ADJUST

MEASURMENT:

MEASURE

AUTOMATED DESIGN ALGORITHMS

PLACEMENT AND ROUTING OF ARBITRARY RECTANGLES
AREA ESTIMATION FOR OPTIMAL SHAPE DETERMINATION
SUBCIRCUIT OUTPUT IN SDL TO DDL, TESTAID, MOTIS, SPICE



MANUAL LAYOUT PROCEDURE

PHASE 1:

1 ESTIMATE SIZE AT EVERY LEVEL
2 INITIAL PLACEMENT
3 DETERMINE SHAPES FOR DENSE PACKING
4 2 & 3 FOR EACH COMPONENT AT THIS LEVEL
5 DO FOR ALL LEVELS DOWN
6 REITERATE WITH NEW SIZE ESTIMATES'

PHASE 2:

BOTTOM UP IMPLEMENTION
AND OPTIONALLY
CELL MODIFICATION

CURRENT STATUS

SUBSYSTEMS IMPLEMENTED:

SDL COMPILER / MACRO EXPANDER
INTERFACE TO TESTAID
IC LAYOUT NUCLEUS
AUTOMATED LOGIC DIAGRAMS
DDL COMPILER / SIMULATOR

UNDER DEVELOPMENT :

INTERACTIVE DESIGN SPECIFICATION
AUTOMATED IC LAYOUT ALGORITHMS
INTERACTIVE COLOR GRAPHICS

Automatic **Partitioning** Algorithms
(T. Pdyne)

PARTITIONING ALGORITHMS

- Used to assign components to modules'
- Can apply at several levels
- Will look at components on boards

Types of Algorithms

Constructive
- -Sequential
- -Parallel

Improvement .
- -Iterative
- -Interactive

Sequential Constructive

Assumes that: .
- A complete logic diagram exists
- A connection limit exists
- A space limit exists

It ● tteapts to minimize the number of boards.

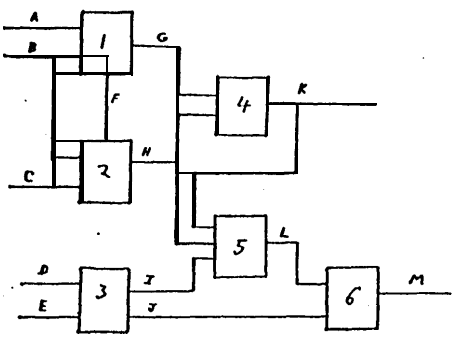
It assigns components to a board until the board fills then starts the next board.

It assigns components when they
- are heavily connected
- and
- add minimally to constraints.

An implementation

Start with board 1.

1. Select a seed component.
   - Random
   - User input
2. Assign the component to the board,
3. Update the external connection list.
4. Update possible component list.
5. Select the next component.
   - Connections added
   - Space ● dd,ad
   - Board full?
   - List? ■□◆▽⇙



Seed = 2
External connections = B , C , F . H
Possible next components = 1 , 4 , 5

Results

- System with about 1000 ECL gates
- 100 connections per board limit

space (ics)	manual (ics/board)	sequential (ics/board)
18	17	15
55	33	17

Advantages
- Fast
- Easy to implement
- Always finds a feasible partition

Parallel Constructivr

Assumes that:
- A complete logic diagram exists
- A connection limit exists
- A space limit exists

It attempts to find a valid partition onto a given numtcr of boards.

It constructs all the boards in a parallel manner so assignment decisions can be made based on information in all boards instead of just one board.

An implementation

Phase 1

1. Select a target number *of. boards.*
2. Seed each board.
   - Random
   - Urcr Input
3. Pick a new component for each board.
   - Much like sequential selection
   - Do not place any components that connect to more than one board
4. Repeat until no more components can be placed.

Phase i?

the remaining *components* were *not* placed because

- They connect to components on *more* than one board
- They do not fit on the board they connect to
- they do not connect to any placed component

Place the *remaining* components by finding the board it fits on the best.

- Place heavily connected components first
- Look at
  - Connections added
  - Space added
  - Board fullness
- If a component will not fit anywhere then the partitioning has failed

Results

- System ulth about 1000 ECL gates
- 100 connections per board limit

space (ics)	manual (ics/board)	parallel (ics/board)
18	17	15
55	33	17

Advantages

- Easy to implement
- Each iteration is fart

Disadvantages

- Ray not succeed

Improvement Algorithms

Assume that:

- Some initial partitioning ● xlstr
  - Random
  - Constructive
  - Manual
- A connoctlon limit ● xlrtr
- A ● paco limit exists

Attempt to improve the partitioning by rerrranglng components in some manner.

Interactlvc Improvement

- (In initial partition is *formed* using a constructive algorithm
- The user removes components interactively
  - Limits are checked
- The user can replace these components by hand
  - again limits *are* checked
- The components can be *rc*placed automatically using tho last phase of the parallel algorithm

Palrwlse Interchange

1. Pick two components
2. Test to see if interchanging them will improve the partitioning
3. If yes then interchange them
4. Repeat 1-3 until
   - Time limit
   - Interchange limit

User lots of time

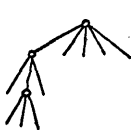
Problems

- Reliability
- Testability
- Use of standard or repeated boards

future Work

- Use of hierarchical design information
- Interactive program
- Bit slicing
- Logic structure,recognition

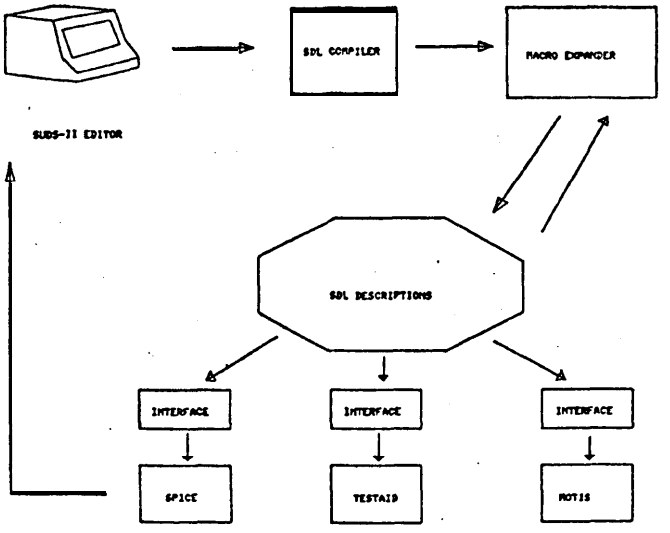
Hierarchical Logical Description Tree



- Functional grouping
- Structural Information

Physical *Tree*

System
Racks
Boards
Chips



DESIGN INFORMATION

STRUCTURE
(CONNECTIVITY)
(BLOCK DIAGRAMS)

BEHAVIOR
(FUNCTIONAL SPECIFICATION)
(SIMULATION MODEL)

PROBLEM :

MULTIPLE REPRESENTATION FOR STRUCTURE AND BEHAVIOR

RESULT :

DESIGNER HAS TO RECODE DESCRIPTION

TO SAVE TIME, DO NOT USE SOME TOOLS

THE GOAL:

A SINGLE LANGUAGE FOR STRUCTURE': SDL

A SINGLE LANGUAGE FOR BEHAVIOR: ADLIB

ADVANTAGES

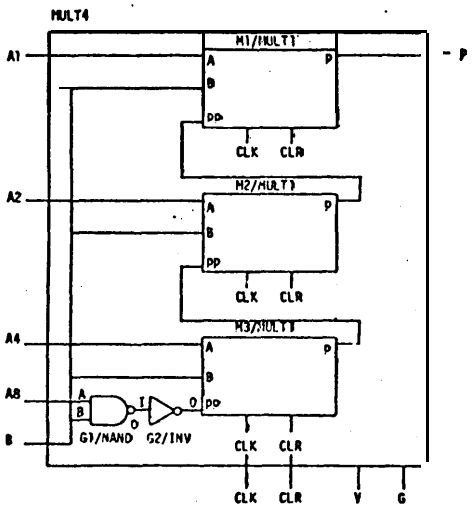
- A SINGLE NOTATION
- SAVES TIME
- ALLOWS USE OF LARGE NUMBER OF TOOLS
- PROMOTES EASY DESIGN INTERCHANGE
- ALLOWS CONSISTENCY CHECKS
- HAVE AUTOMATED MAPPING THRU MACRO-EXPANSION

STRUCTURAL DESCRIPTION

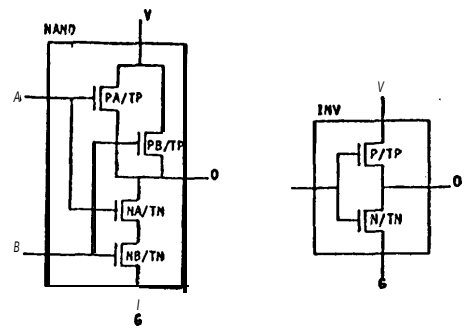
- SUBSYSTEMS (BLACK 'BOXES)
  ->COMPONENTS
- EXTERNAL CONNECTIONS
  ->EXTERNAL CONNECTORS
- CONNECTIONS BETWEEN SUBSYSTEMS
  ->NETS AND BUSSES

ABSTRACTION LEVELS

ARCHITECTURE
REGISTER
LOGIC DESIGN
CIRCUIT




```
USER:'WMUC';

(* GENERAL CIRCUIT INFORMATION FOLLOWS *)
NAME:MULT4;
PURPOSE:ICDESIGN;
LEVEL:CHIP;

(* DESCRIPTION OF EXTERNAL CONNECTORS FOLLOW *)
EXT:EC:A1,A2,A4,A8,B,CLK,CLR,PRODUCT,V,GND;
INPUTS:EC.<A1,A2,A4,A8,B,CLK,CLR>;
OUTPUTS:EC.PRODUCT;

(* DECLARATION OF THE PRIMITIVE COMPONENT TYPES FOLLOWS *)
TYPES:MULT1,NAND,INV;

(* DECLARATION OF COMPONENTS AND THEIR TYPES FOLLOWS *)
MULT1:M<1:3>;
INV:G2;
NAND:G1;
END;


(* CONNECTIVITY BETWEEN COMPONENTS FOLLOWS IN TERM OF NETS *)
NETSEGMENT;
N1=EC.A1,M<1>.A;
N2=EC.A2,M<2>.A;
N3=EC.A4,M<3>.A;
N4=EC.A8,G1.A;
N5=EC.B,G1.B,M<1:3>.B;
N6=M<1>.P,EC.PRODUCT;
N7=EC.C.K,M<1;3>.CLK;
N8=EC.CLR,M<1:3>.CLR;
N9=M<1>.PP,M<3>.P;
N10=M<2>.PP,M<3>.P;
N11=G1.O,G2.I;
N12=G2.O,M3.PP;
ENDNETS;
ENDC;
```




```
(* SDL NAND CIRCUIT DESCRIPTION *)
NAME:NAND;
PURPOSE:ICDESIGN;
LEVEL:TRANS;

(* EXTERNAL CONNECTOR INFORMATION *)
EXT:EC:A,B,O,V,GND;
INPUTS:EC.<A.B>;
OUTPUTS:EC.O;

(* COMPONENT TYPES AND COMPONENT DECLARATIONS FOLLOW *)
TYPES:TP,TN;
TN:NA,NB;
TP:PA,PB;

(* CONNECTIVITY OF CIRCUIT FOLLOWS *)
NETSEGMENT;
N1=EC.A,PA.G,NA.G;
N2=PA.D,PB.D,NA.D,EC.O;
N3=EC.GND,NB.S;
N4=EC.V,PA.S,PB.S;
N5=EC.B,PB.G,NB.G;
N6=NA.S,NB.D;
ENDNETS;
ENDC;
@
```

CURRENT STATUS

COMPILER/SYNTAX CHECKER OPERATIONAL ON
- DEC 20 SYSTEM
- IBM 370/168
- HP 3000
- HP 1000

INTERFACES BUILT TO
- SPICE (CIRCUIT ANALYSIS)
- TESTAID (LOGIC SIMULATOR, FAULT TEST GENERATION)

INTERFACES PLANNED:
- NOTIS (MOS TIMING SIMULATOR)
- CALMOS (IC LAYOUT)
- SICLOPS (IC LAYOUT)

DDL-P, an RTL Behavioral Description Language
(W. Cory)

DDL-P: DICITRL DESIGN LANGUAGE

A behavioral descr ptfon *language*
for digital *systems*

- Finite state machine notation
- Parallelism
- Language design considerations

References:

D.L. Dietmeyer and J.R. Duley, "Register Transfer Languages and Their Trnnslation" in DIGITAL SYSTEM DESIGN AUTOMATION: LANGUAGES, SIMULATION AND DATA EASE. Computer Science Press, Inc., 1975, pp. 117-218.

W.E. Cory, J.R. Duley, and W.M. vanCleemput, "AN INTRODUCTION TO THE DDL-P LANGUAGE. Stanford University, CSL, March 1979, 97 pp.

W.E. Cory, J.R. Duly, and W.M. vanCleemput, DDL-P COMMAND LANGUAGE MANUAL. Stanford University, CSL, March 1979, 39 pp.

Brief history of DDL

- Formulated by Jim Duley, U. Wisconsin 1967
- Some work by Jim Duley, Don Dietmeyer, R.L. Arndt, L.E.R. Soares at U. Wisconsin
- Large subset implemented at H.P. Labs in 1971-73 by Jim Duley, Becky Clark, and John Welsch
- Rewritten *in* Pascal at Stanford by Warren Cory in 1978

DDL SYNTAX

COMMENTS — enclosed in quotes

```
' THIS IS A COMMENT. '

' COMMENT TO END OF LINE...
'   (NO TRAILING QUOTE REQUIRED)
```

IDENTIFIERS

- start with a letter
- up to 132 alphanumeric characters

```
A
B
REG
NamesCanBeQuiteLong
```

CONSTANTS

	Length-Base-Ualue
1B1	1
6D22	010110
8B101	00000101
2B101	01
16Q2321	0000000010111001
11Q1367	01011110111
6H3C	111100
8B.101	10100000
2B.101	10
10H.74	0111019800
100	0000000001100100

OVERVIEW - DDL Description has following sections:

REGISTER	Synchronous flip-flops
MEMORY	Asynchronous latches
TERMINAL	Combinational networks
OPERATION	Define transfers which may occur, optional timing information
CONTROL	Finite state machine controlling use of previously defined facilities

MEMORY/REGISTER DECLARATIONS

```
( REGISTER )
(  MEMORY  ) A, B, C,
             DC5: 103, EC16:0],
             FC10],    'equivalent to FC1:10]'
             GC0:1023,15:0]
             END
```

TERMINAL DECLARATIONS

```
TERMINAL

  'Inputs, function not specified '
  H, IC5:20], JC0:10,5:0],

  'Inputs, function specified'
  ONEC1:8] = 8B1,
  XL17 = X(17,
  SUMXYC1:16] = X(+)Y TAIL 16,

  'Some inputs unspecified'
  SUM(X,Y)C1:12] = (10D0 CON X(+)Y) TAIL 12

  END
```

BOOLEAN EXPRESSIONS - FACILITY REFERENCES

```
'assume declaration
'  REGISTER A, B, C, DC5:10], EC16:0], FC107,
            GC0:1023,15:0] END
' TERMINAL SUM(X,Y)C1:12] = X(+)Y TAIL 12 END

  A
  DC6]
  EC10:7]
  F
  GC1000,14]      GC1000]C14]
  GC1,15:3]       GC1]C15:3]
  GC25]

  SUM(E,GCD])

  G C D C C CON D : B (+) 2 ] , F : EC4:0] ]
```

BOOLEAN EXPRESSIONS - OPERATORS

'almost' in order of precedence:

Addition and subtraction

```
1B1 (+) 4B1011      . 5B01100
4B1111 (+) 4 B1111  = 5B11110
1B1 (+) 1B0         . 2B01
1B1 (-) 4B1011      . 5B10110
4B1111 (-) 4B1111   = 5B00000
1B1 (-) 1B0         = 2B01
1B0 (-) 3B110       = 4B1010
(-) 3B110           = 3B010
(-) 3B001           . 3B111
```

Arithmetic relational operators

Unsigned arguments assumed
Result is 1B1 (true) or 1B0 (false)

```
2B10 > 16D1     . 1B1    true
10D1 8  1B1     . 1B0    '#' is 'not equa 1'
8D2 (=) 8D3     : 1B0    '(=)' is 'equal'
1B1 >= 233      = 1B0    false
```

Substring operators

```
3B101       EXT  3  = 9B101101101
8B11010110 HERD 4  = 4B1101
8B11010110 TAIL 2  = 2B10
```

Concatenation

```
4B1101 CON  6B1   . 10B1101000001
4B1    CON 6B.1   = 10B0001100000
```

One's complement .

```
- 1B1          = 1B0
- 6B110101     = 6B001010
- 1081473      = 10B0011000100
```

Binary logical operators

```
5B10110  *  5B00101  = 5B00100     AND
5B10110 [+] 5B00101 = 51318011     XOR
5B10110  t  5B00101  = 5B10111     OR
5B10110 * 7B1111111  = 7B0010110   *WARNING*
```

Reduction operators

```
+   R E D  5B00010 = 1B1
*   R E D  5B11111 = 1B1
[+] RED 5800101 = 1B0
(+) RED 5B11101  = 16B0000000000000100
```

Parentheses may be used freely in expressions to alter the order of operators:

```
(A CON B) [+] (C + D)
```

Conditional expression

```
CASE selector DO value if selector=1
              DO value if selector=2
                 . . .
              DO value otherwise
              ENDCASE

A selector ^ value if selector=1 ;
             value if selector=2 ;
                . . .
             value otherwise .

IF selector THEN value if selector=1
            ELSE value otherwise
            ENDIF

CASE J CON K   DO 1B0   DO 1B1
               DO -0    DO Q
               ENDCASE

IF  A THEN YC1]
    ELSE IF B THEN YC2] ELSE YC3] ENDIF
    ENDIF
```

OPERATION ACTIONS

Assignment `<-` Delayed store
. Immediate store

```
A=B,  B=A   will not swp values
M-B,  B<-A will swap values
```

Left hand side concatenation:

```
A CON B = expression
AC1:3] CON BC1:5] . 8B10101100
  --> AC1:3] . 3B101,
      BC1:5] . 5B01100
```

A @ equivalent to 'A=1B1'

TINE expression gives length of current operation

INPUT, OUTPUT actions available
INPUT(1,PORTC1]),
OUTPUT(6, PORiC2.3)

Conditional syntax used in Boolean expresstons also available for operations, along with labels, goto's

OPERATION DECLARATION

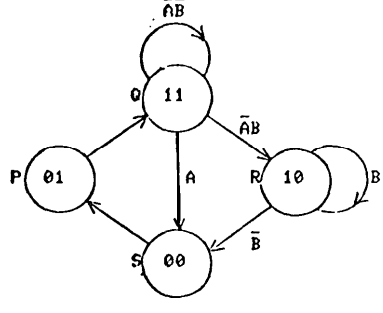
"Operation is named sequence of actions"

OPERATION

```
INIT = [      ADDR = 8B0,
       LABEL: MEADDR] = 8B0,
              ADDR = ADDR(+)1 TAIL 8,
              I F  ADDR#0 THEN -> LABEL
                              ENDIF
       ] ,

GETSTROBE = [INPUT(1,STROBE)],

GETSUM(X,Y) = [REGC1:12] <- SUM(X,Y)]

END
```

CONTROL DECLARATION

Example



. DDL-P specification for previous example .

```
REGISTER $SSRC0:1] END
TERMINAL A,B END
OPEHRTION SETSSR(N) = CSSR(-N] END
CONTROL   P(2B01): ->Q/
          Q(2B11): IF A THEN ->S
                     ELSE IF B THEN ->R
                                ELSE ->Q
                   ENDIF     ENDIF/
          R(2B10): I F  B THEN SETSSR(2B10)
                   ELSE SETSSR(2B00 ) ENDIF/
          S(2B00): SETSSR(2B01)/
          END
```

States invoke operations to carry out desired data transfers

All operations in stato executed in parallel

Delayed stores finished just before next state is entered

```
OPERATION
    SWAP = [R(-S, S(-R],. . ,
CONTROL
    P: ... SWAP ... ->Q/
    Q: ...
```

ADLIB/SABLE, a Multilevel Functional Simulator
(D. Hill)

SABLE

STRUCTURE AND BEHAVIOR LINKING ENVIRONMENT

DWIGHT D. HILL
WILLEM VANCLEEMPUT

STANFORD UNIVERSITY

INTRODUCTION

USEFULNESS OF SIMULATION
PROBLEMS WITH EXISTING SIMULATORS
- NARROW PHASE OF DESIGN
- FLAT STRUCTURE - HIGH COMPUTATION LOADS
- FRAGMENTATION OF DESIGN
- COMPATIBILITY CHECKING DIFFICULT

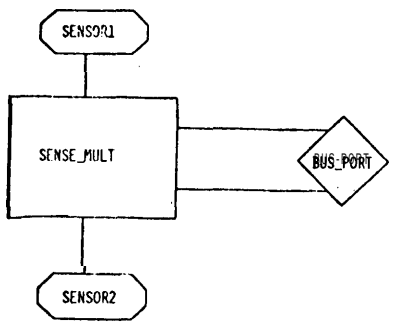
GOALS OF SABLE PROJECT
- USEFUL FROM CONCEPTION THROUGH MAINTENANCE
- CAPTURE DESIGN EAQLY
- MERGE STRUCTURE WITH BEHAVIOR
- ACCURATE MODEL
  - OMIT NOTHING IMPORTANT
  - INTRODUCE NO ARTIFACTS
- SUPPORT ALL LEVELS OF ABSTRACTION
- ALLOW MIXING OF LEVELS
- REASONABLY EFFICIENT
- EASY TO LEARN, USE, READ

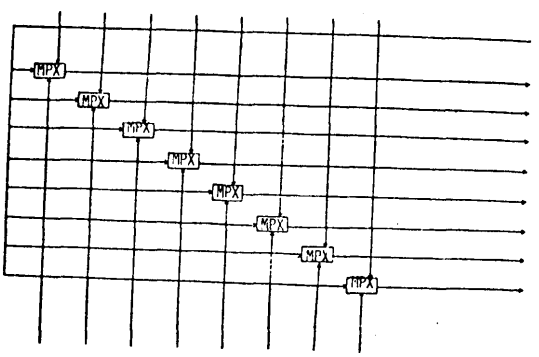
APPROACH
- VERY GENERAL MODEL
- NO STRUCTURE CONSTRAINTS
- GENERAL PURPOSE HIGH LEVEL LANGUAGE
- NO LEVEL SPECIFIC FEATURES
- EXTENDABLE

SABLE MODEL OF COMPUTER SYSTEMS
- MANY INDEPENDENT PPOCESSES
- INTERCONNECTED WITH "NETS"
- SYNCHRONOUS OR ASYNCHRONOUS
  - CLOCK DRIVEN
  - STIMULATION DRIVEN
- NETS CARRY DATA AND CONTROL
- ARBITRARY TIME SCALE

USER INTERFACE
- STRUCTURE
  - SDL - "STRUCTURAL DESIGN LANGUAGE"
  - GRAPHICAL EDITOR
- BEHAVIOR
  - PASCAL BASED LANGUAGE

STRUCTURE EDITOR - SUDS II
- RUNS ON TEK 4014
- MANIPULATE POLYGONS, LINES AND TEXT
- "PUSH" AND 'POP" OPERATIONS
  - HIERARCHICAL DESIGN
  - MACRO EXPANSION
- GENERATES SDL
- WORKS WITH OTHER B.A. TOOLS



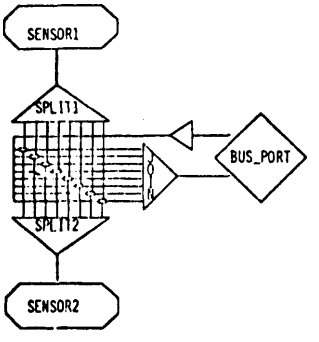


'ADLJB' - A DESIGN LANGUAGE FOR INDICATING BEHAVIOR
- CONTAINS PASCAL (ALMOST)
  - USER DEFINEDTYPES
  - TYPE CHECKING ENFORCED
  - FLEXIBLE CONTROL STRUCTURE
  - SMALL LANGUAGE
- ADDED FEATURES
  - COMMUNICATING VIA NETS
  - COROUTINING

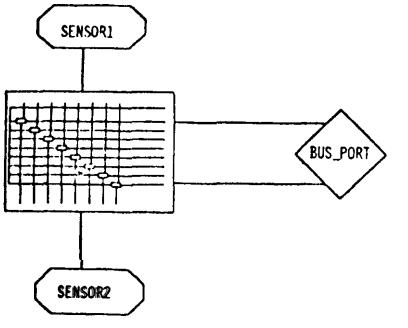
FORMAT OF AN ADLIB PROGRAM
- GLOBAL INFORMATION SHARED BETWEEN COMPONENTS
  - NET TYPE DEFINITIONS
  - CLOCK DEFINITIONS
  - TYPES, CONSTANTS, VARIABLES, ROUTINES
- COMPONENT TYPE DEFINITIONS
- NO "MAIN BODY"

NET TYPE DEFINITIONS (NETTYPES)
- DATA STRUCTURE + INTERPRETATION
- INTERFACE BETWEEN COMPONENTS
- LEVEL OF COMPONENT - LEVEL OF NETTYPES
- COMPACT APPROXIMATION - HIGH LEVEL
  - BOOLEAN / MULTI-VALUED
  - CHAR / SET OF BITS

```
ADLIB CONTROL PRIMITIVES
        ASSIGN      ASSIGN REGA • REGB TO BUS SYNC CLK PHASE 2
                    ASSIGN RISING TO OUT DELAY 3.0E-10
        WAITFOR     WAITFOR INTERRUPT SYNC
                    WAITFOR ENABLE CHECK ENABLE


BEGIN
WHILE TRUE DO CASE CNTRL OF
        PATH1: BEGIN
                    PERMIT(CHANNEL_A);
                    WAITFOR CNTRL PATH1 CHECK CNTRL;
                    INHIBIT(CHANNEL_A);
                    END;
        PATH2 : BEGIN
                    PERMIT(CHANNEL_B);
                    WAITFOR CNTRL PATH2 CHECK CNTRL;
                    INHIBIT(CHANNEL_B);
                    END;
        END; (* OF CASE *)
END;


NETTYPE
BYTE-BUS = SET OF [BIT1,BIT2,BIT3,BIT4,BIT5,BIT6,BIT7,BIT8];
SELECTOR_NET = (PATH1, PATH2);
BOOLNET = BOOLEAN;

                COMPTYPE MULTIPLEXOR;
                INPUT
                DATA1, DATA2 : BYTE_BUS:
                CNTRL : SELECTOR-NET;
                OUTPUT
                DATA-OUT : BYTE_BUS;

    SUBPROCESS
    CHANNEL_A : TRANSMIT DATA1 TO DATA_OUT DELAY 10.0;
    CHANNEL_B : TRANSMIT DATA2 TO DATA-OUT DELAY 10.0;


SUBPROCESSES (CONT)
        EACH HAS NAME
        CONTROLLED WITH 'INHIBIT' AND "PERMIT"
        RUN INDEPENDENTLY OF RAIN BODY
        NON - PROCEDURAL
```



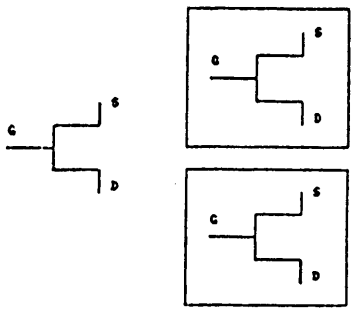

```
TRANSLATOR SPLITTER:
INPUT
BUS : BYTE-BUS;
OUTPUT
OUT1, OUT2, OUT3, OUT4, OUT5, OUT6, OUT7, OUT8: BOOLNET;
BEGIN
WHILE TRUE DO BEGIN
        ASSIGN (BIT1 IN BUS) TO OUT1;
        ASSIGN (BIT2 IN BUS) TO OUT2;
        ASSIGN (BIT3 IN BUS) 10 OUT3;
        ASSIGN (BIT4 IN BUS) TO OUT4;
        ASSIGN (BIT5 IN BUS) TO OUT51
        ASSIGN (BIT6 IN BUS) TO OUT6;
        ASSIGN (BIT7 IN BUS) TO OUT7;
        ASSIGN (BIT8 IN BUS) 10 OUT8;
        WAITFOR CHECK BUS:
        END;
END;


TRANSLATORS
        SAME FORM AS COMPTYPES
        TRANSLATE BETWEEN NETTYPES
        INSERTED TO PRODUCE MULTI-LEVEL SIMULATION
```




```
EXPERIENCE
        USERS KNFW PASCAL
        BASIC IDEAS GRASPED QUICKLY
        INTERACTING COMPONENTS WITH PRACTICE
        FAST INTERACTION APPRECIATED
        PREPROCESSOR PROBLEMS
        LACK OF BIT MANIPULATION FACILITIES


CONCLUSIONS
        BASIC PRINCIPLES SOUND
        IMPLEMENTATION ADAQUATE, BUT
                BETTER DIAGNOSTICS
                PREDEFINED BIT MANIPULATION
                INTEGRATED TESTING PACKAGE


FUTURE MY INCLUDE NON-SIMULATION SUPPORT
        SYMBOLIC EXECUTION
        ANALYSIS AND REWRITING
        DESIGN RULE CHECKING
        VERIFICATION
```

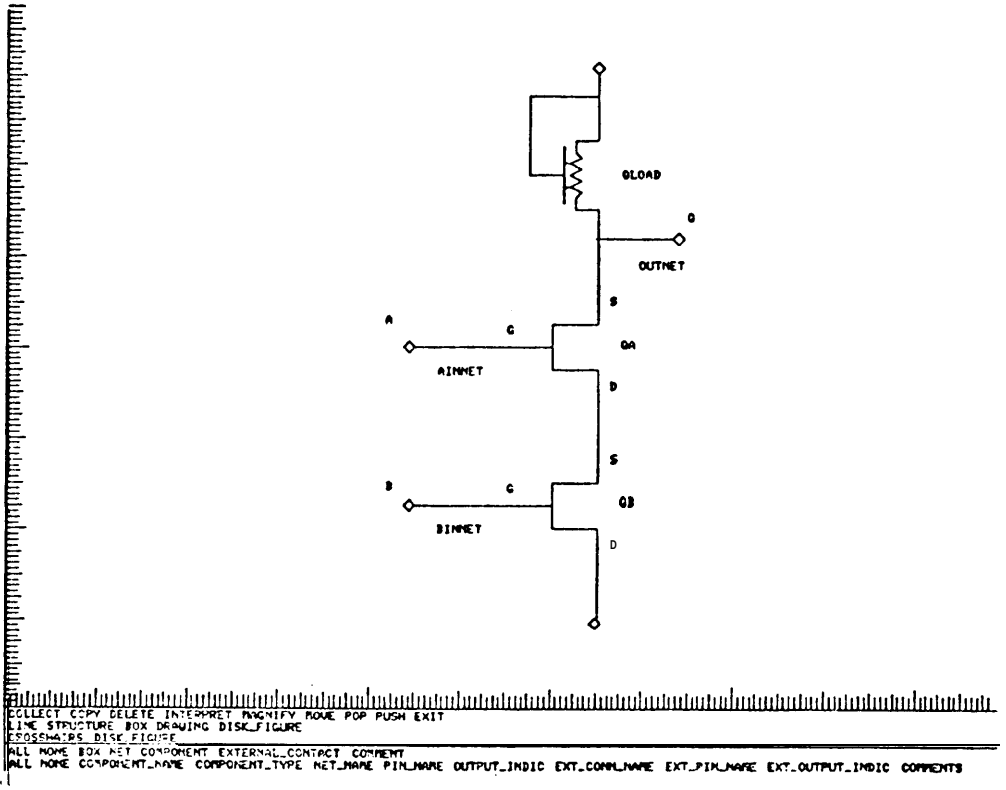
IDSPEC IC DESIGN SYSTEM

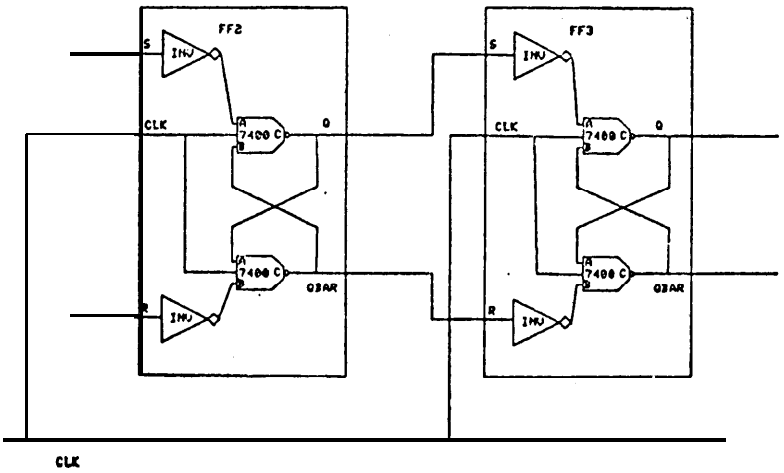
Interactive design specification,
simulation and layout of
integrated circuits.

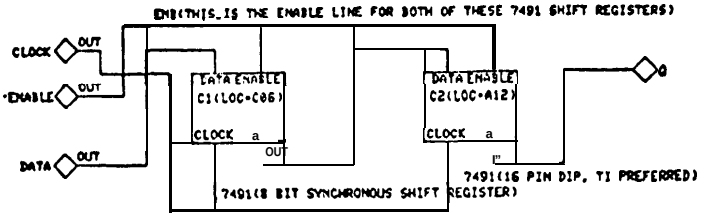


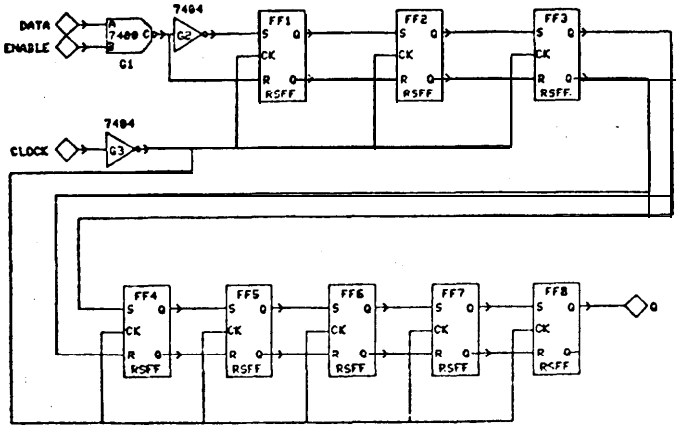
SUDS-II Edi tor

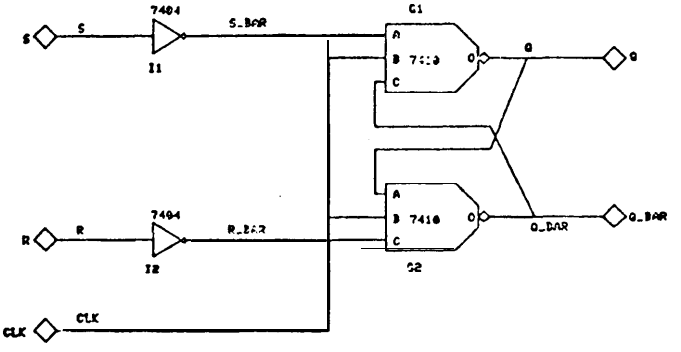
*GENERAL
*PORTABLE (PASCAL)
*CAPTURES CONNECTIVITY AND
DESCRIPTION INFORMATION









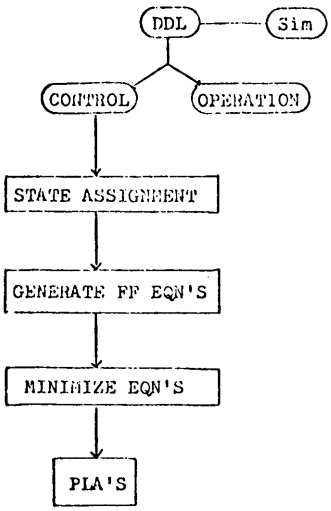



```
USER:"WHW,ESQ.";
NAME:RSFF;
PURPOSE: TEST;
LEVEL: GATE;
EXT:CONN:S,R,Q,Q_BAR,CLK;
%INIT
%LOAD RSFF
%TYPES
%CONNS
%NETSEGMENT
ENDC;
CEND;
```

```
USER:"WHW,ESQ.";
NAME:RSFF;
PURPOSE: TEST;
LEVEL: GATE;
EXT:CONN:S,R,Q,Q_BAR,CLK;
TYPES: 7104, 7410;
7404: I2, I1;
7410: G2, G1;
END;
NETSEGMENT;
CLK= G1.B, G2.B, .CLK;
Q=  .C,  G1.O,  G2.A;
Q_BAR=  .Q_BAR, G2.O, G1.C;
R   =  I2.$PN0003, .R;
R_BAR= G2.C, I2.$PN0002;
S= I1.$PN0001,  .S;
S_BAR= G1. A, I1.$PN0000;
ENDC;
CEND;
```

AUTOMATED HARDWARE SYSTHESIS
FROM
A DDL-P DESCRIPTION

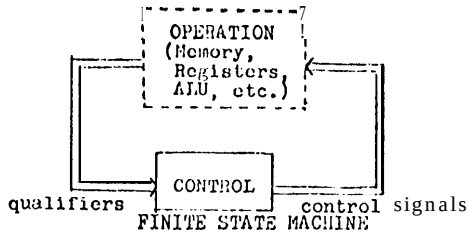
SUNGHO KANO



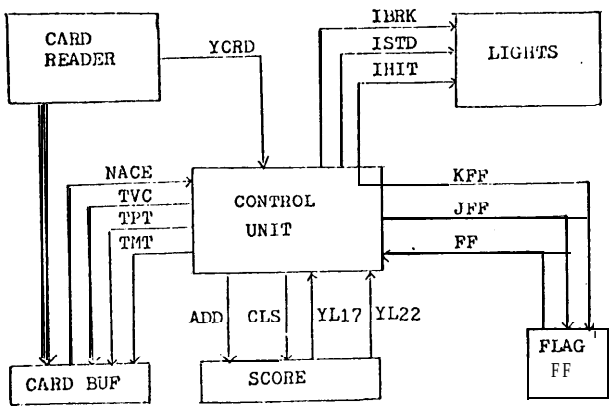
* 3 main problems

1) state assignment
2) minimization
3) partitioning in mapping

DDL MACHINE MODEL



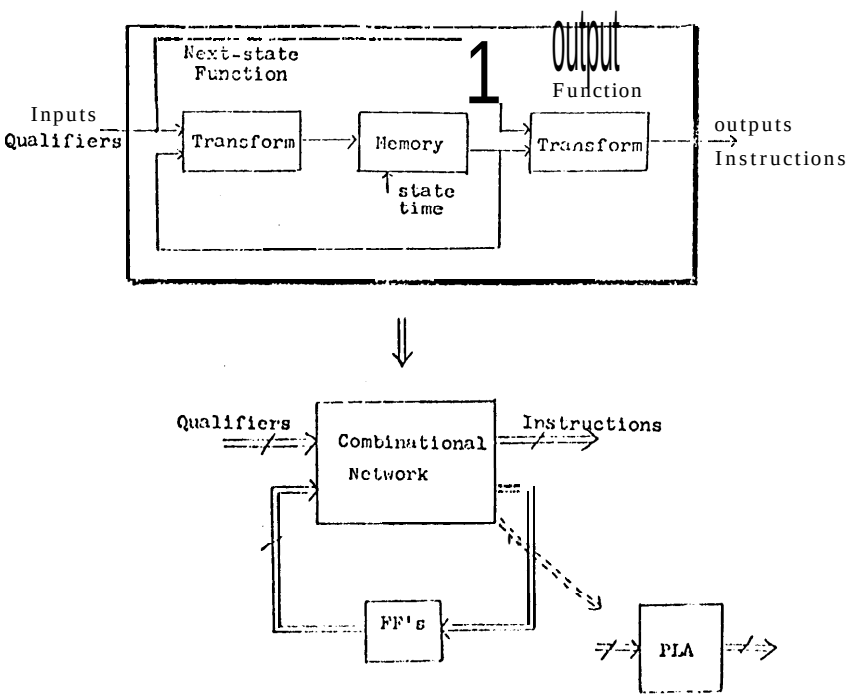
Ex.



BLACKJACKMACHINE

5 QUALIFIERS
10 CONTROL SIGNALS

STATE MACHINE



```
" B L A C K J A C K    M A C H I N E . "
REGISTER  SCORE[5], CARDBUF[5], FF.
TERMINAL  HIT, BROKE, STAND,
   VALUE[1:5] = INPUT(1,VALUE),
   YCRD = INPUT(1,YCRD),
   YL17 = SCORE<17, YL22 = SCORE<22,
   NACE = CARDBUF#1.
OPERATION
   TPT = [CARDBUF _ 5D10], TMT = [CARDBUF _ 5D22],
   TVC = [CARDBUF_ VALUE], IHIT = [HIT=1B1],
   ISTD = [STAND=1B1 3 ,  IBRK = [BROKE=1B1 ],
   CLS = [SCORE _ 5D0], AND=[SCORE _(SCORE( F)CARDBUF)TAIL 5],
   KFF = [FF_1D0], J F F  = [FF _ 1D1 ]
CONTROL
   A:  C L S ,  KFF, ->B/
   B:  IHIT, TVC, ^YCRD^ ->C; ->B. /
   c:  ^YCRD^ ->C; ->D. /
   D:  ADD,  ^NACE+FF^ ->F ,  ->E. /
   E:  JFF,  TPT,  ->D/
   F:  ^YL17^  ->B; ->G. /
   G:  ^YL22^  ->K; ->H. /
   H:  KFF, T M T ,  ^FF^ ->D; ->J . /
   J:  IBRK,  ^YCRD^  ->A; ->J. /
   K:  ISTD,  ^YCRD^  ->A; ->K. /. $
4444444 ***
```

```
CONTROL
   A:  C L S ,  KFF, ->B/
   B:  IHIT, TVC, ^YCRD^ ->C; ->B. /
   c:  ^YCRD^ ->C; ->D /
   D:  ADD.  ^NACE+FF^ ->F ;  ->E./
   E:  J F F .  TPT,  ->D/
   F:  ^YL17^  ->B;  "YL.22"  ->JK; ^FF^ ->D. , KFF. TMT. . /
   JK: ^YL22^ ISTD; IBRK. ,  ^YCRD^  ->A; ->JK. / $
```

PROBLEM :

State minimization

STATEASSIGNMENT

1. determine # of state variables

   * for r states,

     $(\log_2 r) \leq n \leq r$

   * r states

     n ntnto varinblca

     # of state assignments $= \frac{(2^n)!}{(2^n-r)!}$

     # of distinct state assignments $= \frac{(2^n-1)!}{(2^n-r)!\,n!}$

     (S-R,J-K,T flip-flops)

   (For D flip-flops, the complcmentation of the state variables results in a new state assignment $\Rightarrow \frac{(2^n)!}{(2^n-r)!n!}$)

r	n	
1	0	1
2	1	1
3	2	3
4	2	3
5	3	140
6	3	420
7	3	840
8	3	840
9	4	10,810,800

---

2. assign the codes

- requirement
  1) correct : each state must be unique
  2) race-free : asynchronous qualifiers
  3) minimum hardware
- method
  1) manual
  2) semi-automatic : interactive
  3) full-automatic
- algorithm
  1) consider the entirc machine
     a. Dolotta'3 method
     b. SHR-procedure
  2) state-by-ctatc
     a. Minimum sta tc *locus*

        state locus : tho sum of all the bit distances of all the exit paths in a state machine
     b. Rcducad dependency
        . consider tho effect of' qualifiors
        . good for f inding a race-free state assignment

SELECTION OF FF'S

D, JK, T

Q_i	Q_{in}	D	J	K	T	S	R
0	0	0	0	d	0	0	d
0	1	1	1	d	1	1	0
1	0	0	d	1	1	0	1
1	1	1	d	0	0	d	0

*Excitation Tablc

1. JK FF's tend to give a smaller hardware if used with conventional combinational network, but
2. need twice a number of' noxt state functions.

MINIMIZATION

1. QUINE-McCLUSKY Method

* generate all prime implicants and then obtain a minimal covering
* gunrsntec an optimum solution
* costly = storngo and time

2. Heuristic Algorithm

* # of products is important
  ⇒ the cost function is simplified by assigning equal weight to every implicant

EX.

Z \ XY	00	01	11	10
0	0	1	1	0
1	1	1	0	0

$F = \bar{X}Y + \bar{X}\bar{Y}Z + XY\bar{Z}$

$= \bar{X}Y + \bar{X}Z + Y\bar{Z}$

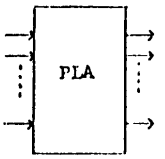
* requirement

i) simple
ii) fast
iii) use less storage
iv) *hazard free

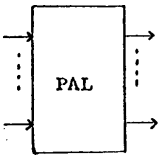
MAPPING

1. Partitioning

2. PLA or PAL

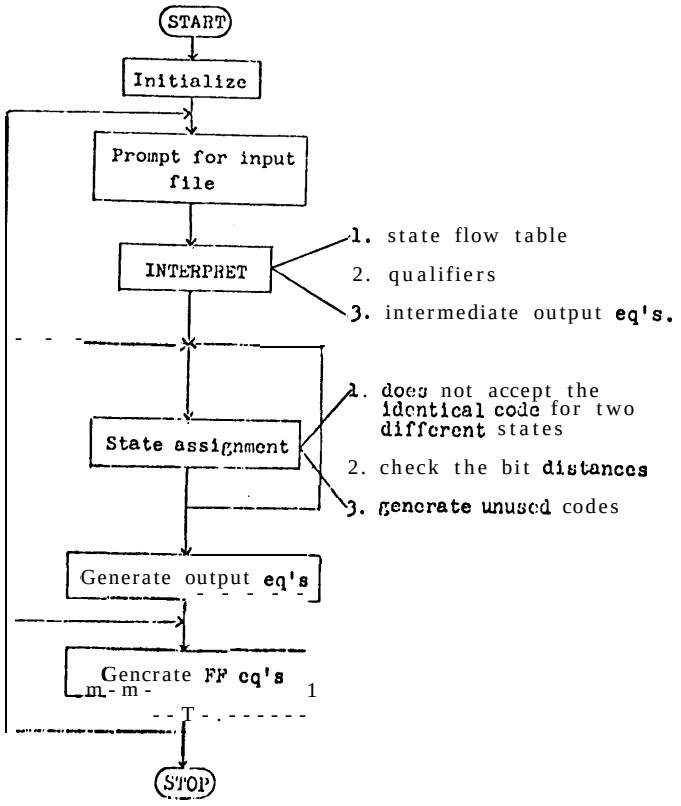


multiple input
⇒ multiple output
(programmable AND array
OR)

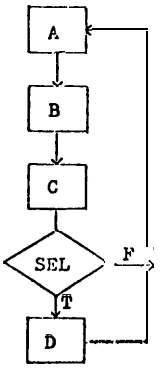


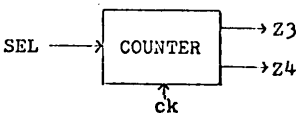
multiple input
⇒ several sin&e outputs
(Output functions do not share product terms)
(programmable AND array
fixed OR)

PROGRAM



Modulo 3/4 counter





TERMINAL

SEL, Z3, Z4

CONTROL

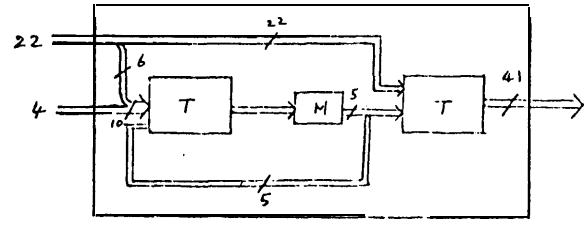
A : /
B : /
C: ^SEL^--> D ; Z3 €,-->A./
D : Z4 @,-,A./. $

Ex. Modulo 3/4 COUNTER

```
        *** STATE TRANSITION ***

1.   A(00)/B
2.   B(11)/C
3.   C(01)/D, A
4.   D(10)/A

**********************************

1.   C/D = SEL
2:   C/A = -SEL


*** STATE ASSIGNMENT SUMMARY ***

  1) WARNING : MORE THAN 1 VARIABLE CHANGES IN A TRANSITION

     1.  2 VARIABLES CHANGE FROM A(00) TO B(11).
     2.  2 VARIABLES CHANGE FROM C(01) TO D(10).

  2) UNUSED CODES FOR STATE ASSIGNMENT

       NONE

     < OUTPUT EQUATIONS >

1.   Z3 = -Q2*Q1*-SEL
2.   Z4 = Q2*-Q1

     < JK FF EQUATIONS >

1.   J1 = -Q2*-Q1 + d(Q2*Q1 + -Q2*Q1*(-SEL) + -Q2*Q1*SEL)      => -Q2
2.   K1 = -Q2*Q1*(-SEL) + -Q2*Q1*SEL + d(-Q2*-Q1 + Q2*-Q1)     => -Q2
3.   J2 = -Q2*-Q1 + -Q2*Q1*SEL + d(Q2*Q1 + Q2*-Q1)            => -Q1+SEL
4.   K2 = Q2*Q1 + Q2*-Q1 + d(-Q2*-Q1 + -Q2*Q1*(-SEL) + -Q2*Q1*SEL)
                                                              => 1

     < DFF EQUATIONS >

1.   D1 = -Q2*-Q1 + Q2*Q1
2.   D2 = -Q2*-Q1 + -Q2*Q1*SEL

     < TFF EQUATIONS >

1.   T1 = -Q2*-Q1 + -Q2*Q1*(-SEL) + -Q2*Q1*SEL
2.   T2 = -Q2*-Q1 + Q2*Q1 + -Q2*Q1*SEL + Q2*-Q1

NORMAL TERMINATION

IN STATEMENT 553 OF . MAIN. AT LEVEL 0


FASBOL STATISTICS SUMMARY-
          1358 MS. EXECUTION TIME
          1342 STATEMENTS EXECUTED
           290 STATEMENTS FAILED
             5 REGENERATIONS OF DYNAMIC STORAGE
          1011 MICROSECONDS AVG. PER STATEMENT
```

" INTEL 8008 "

```
CONTROL
   M1T1 : HS1, PCLOUT, ^INT^ HSO, CLCY, SIFF; INCPCL, CIFF. /
   M1T2 : HSO, SPCI, PCHOUT, INCPCH, INTC/
   M1TW : RDYE, RDY^ FETCH, STOIR, STORB, ->M1T3; ->M1TW. /
   M1T3 : HS2, RDYC, INTE, ' APR+ROT ARA. ,
          ^RST^ PUSH, CLPRA ,
          ^HLT^ HS1, ^INT^ ->M1T1; ->M1T3. ;
                ^M2^ ->M2T1 , ^RCF^ ->M1T1. /
   M1T4 : HS2, HS1, HSO, ^SSG^ SSSRB. , ^INR+DCR^ DDDRA. ,
          ^RTC^ POP. , ^RST^ RAPCH , ^LMR^ ->M2T1. /
   M1T5 : HS2, HSO, ^LRR^ RBDDD. , ^INR^ INRADD, ^DCR^ DCRSUR. ,
          ^APR^ ALUOP. ^ROT^ ROTRA. , ^RST^ RRBPCL. , -> M1T1/
   M2T1 : HS1, ^MRI^ LOUT, ^INP+OUT^ AOUT;
                PCLOUT, ^IFF^ HSO, CLCY; INCPCL. . . /
   M2T2 : HSO, ^MRI^ ^LMR^ SPCW; SPCR , HOUT;
                ^INP+OUT^ RBOUT; SPCR, PCHOUT, INCPCH. /
   M2TW : RDYE, ^RDY^ ^LMR^ RBOUT; "--OUT" I-ETCH, STORB. . , ->M2T3;
                ->M2TW /
   M2T3 : HS2, RDYC, ^APM+API ARA ,
          ^M3^ ->M3T1, ^LMR+OUT^ ->M1T1./
   M2T4 : HS2,HS1,HSO, ^INP FFOUT /
   M2T5 : HS2, HSO, ^LRM+LRI RBDDD , ^APM+API^ ALUOP. ,
          ^INP^ RBA , ->M1T1/
   M3T1 : HS1, ^LMI LOUT, PCLOUT, ^IFF^ HSO, CLCY; INCPCL. /
   M3T2 : HSO, ^LMI^ SPCW. HOUT; SPCR, PCHOUT, INCPCH /
   M3TW : RDYE, ^RDY^ ^LMI^ RB OUT, FETCH. STORA. , ->M3T3;
                ->M3TW /
   M3T3 : HS2, RDYC, ^CALSUB^ PUSH , ^LMI+JCF^ ->M1T1. /
   M3T4 : HS2, HS1, HSO, RAPCH/
   M3T5 : HS2, HSO, RSPCL , ->M1T1/ $
```

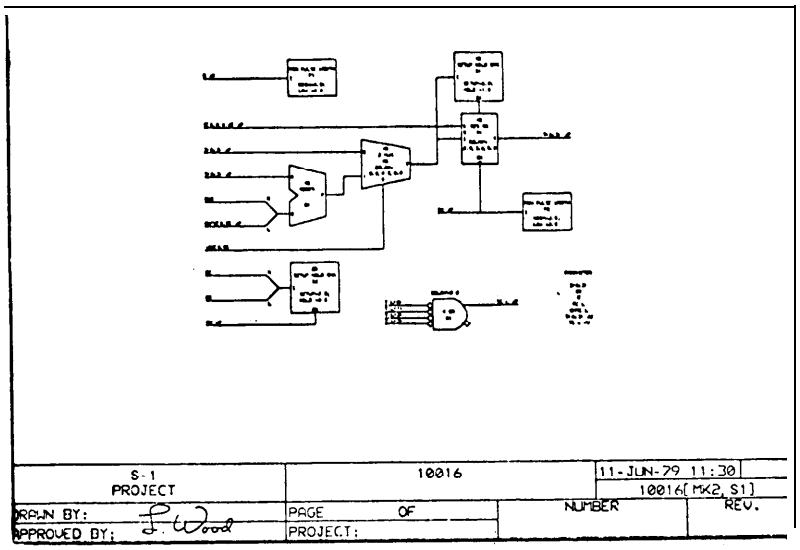


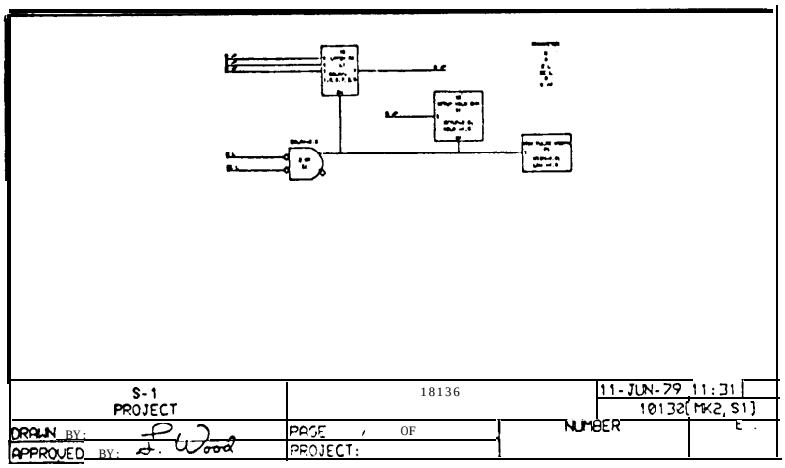
TIMING VERIFICATION IN THE SCALD SYSTEM

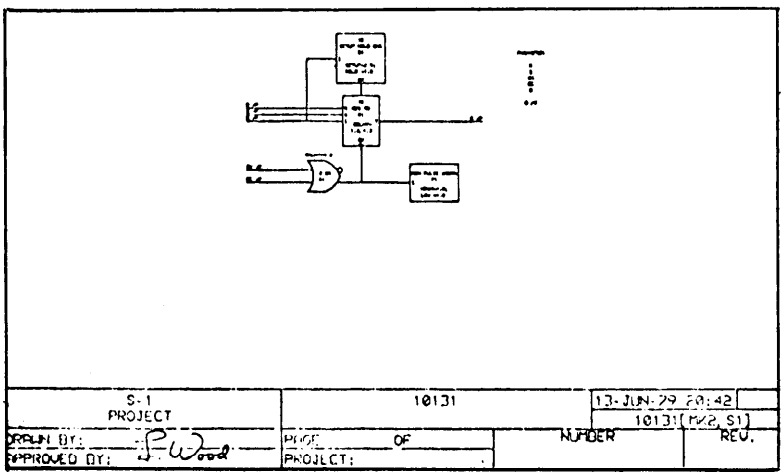
- CHECKS ALL TIMING CONSTRAINTS IN LARGE DIGITAL SYSTEMS, TAKING INTO ACCOUNT:
  - COMPONENT TIMING PROPERTIES
    - PROPAGATION DELAYS
    - SETUP AND HOLD CONSTRAINTS
    - MINIMUM PULSE WIDTH CONSTRAINTS
  - WIRE DELAYS
    - USER-SPECIFIED LIMITS
    - CALCULATED VALUES BASED ON ROUTING, CAPACITANCE, AND TRANSMISSION LINE CHARACTERISTICS
  - ADDITIONAL DESIGNER-SPECIFIED CONSTRAINTS

TIMING VERIFIER - SIGNAL VALUES

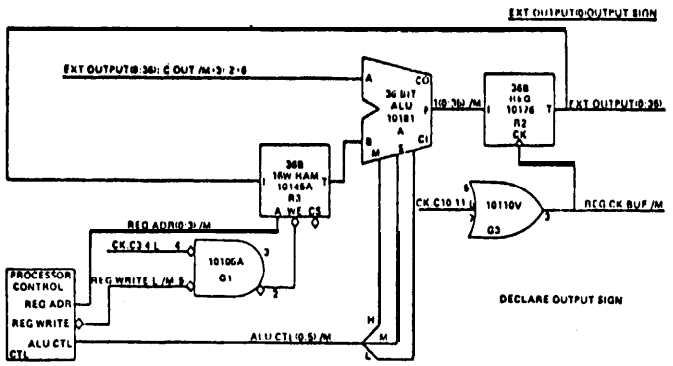
VALUE	MEANING
0	FALSE
1	TRUE
S	STABLE
C	CHANGING
R	RISING ECCC
F	FALLING EDGE
U	UNDEFINED (INITIAL VALUE)







EXAMPLE SCALD MACRO DEFINITION–SIMPLE PROCESSOR



MACRO: SIMPLE PROCESSOR

EXAMPLE SIGNALS

NAME .C1-2L &A I
XYZ .C4-7 &AN
CONTROL .V1-1.5,4-4.5

EVALUATION DIRECTIVES

EVALUATION DIRECTIVE	MEANING
A	CHECK THAT OTHER INPUTS ARE STABLE WHEN CLOCK IS ASSERTED
H	CHECK THAT OTHER INPUTS ARE STABLE WHEN CLOCK IS NOT ASSERTED
I	IGNORE OTHER INPUTS, JUST USE CLOCK
E	EVALAUTE OTHER INPUTS WITH CLOCK

Data Rase Considerations for VLSI Design
(L. Scheffer)

Data Base Considerations for VLSI Design

1) Need for additional data

2) A possible structure for the data base

3) A new rtthbdology for design

4) Plans for research

The magnitude of the problem

1) Example - Typical chip in production today

a) Smallest legible plot is 3 meters on a side

b) 6 micron geometries, 5000 microns (200 mils) on a side

c) 16000 transistors, about 1/2 random logic

d) Took 5 man-years to develop

Suppose we had a design system that encorporated all features we have seen at this meeting, so that given a chip of this size:

1) Monday, do algorithmic, functional and logical design

2) Tuesday, we do circuit design

3) Yednetday, we do the ● rtuork

4) Thursday, write test sequences

5) Friday, debug it

This is roughly a 250 times improvement over current methods

How long does it take to design a ULSI chip?

200 years !

Why Consider a Unified Data Bate?

IC's have been designed for years with separate data bases:

1) Schematics (In Notebook)

2) Circuit Simulator Input

3) Logic Diagrams

4) functional Simulation Programs

5) Test Vectors

6) Artwork

What are thr problems with this a rrangement?

1) N o consistancy

2) No error checking

3) No "MACRO" capability

4) Not in one place

5) No cohesive comments

Furthermore, problems will get worse as VLSI continues

1) No plots of chips

2) N o complete schematics

3) Hierarchical form will be the only useful (possible) form

What demands will we make of the data base?

1) Edit all Information

2) Allow access by verification programs

3) Expand any subpiece, on any type diagram

4) Ask historical questions

Yho last modified this and uhy?

5) Security

6) Simplify -- this is the only way design can be accomplished

What do we need for each module

1) One or more Jrtuork representations

a) multiple technologies

b) design rule checking

c) multiple form factors

2) Schematics, on several levels

a) Logic, several forms

b) Circuit level

c) Register transfer

3) Text

a) Functional design

b) Comments

c) lest vectors for individual pieces

4) History

5) A Procedure, perhaps null (procedural attachment)

Why a procedure uith each module?

1) Allows generalization, which cuts down work

eg. N-bit adders, shifters, etc

2) Related concepts

a) N-bit devices (S-1 design system)

b) Stretchable Cells

c) Auto-layout

d) *'Silicon compilers"

How does using a procedure instead of Jny of these other concepts help?

All of these concepts can be improved

1) N-bIt adder could do lookahead correctly

2) Stretchable cells could fold as sizes got bigger, makes them more applicable

3) Auto-routIng. NMOS is not used uith auto-routing schemes since DC power is too great. Using this technique, we can

a) Route using standard cells

b) Calculate driven capacitance

c) Reduce power and cell size (and change topologies)

d) Iterate this process until no improvement is obtained

4) SIlicon compilers. The attached procedure can also produce logic diagrams, test vectors, functional specs, etc.

Applications not done nou.

Problems

A) Data Base itself

1) Huge amounts of data

2) Fast access for commonly asked questions

3) Must offer procedural attachment

B) Language

1) Must handle text

2) Must handle numerical calculations

3) Must handle (small scale) circuit simulation

4) Rust handle artwork

5) And it must do all of these things well!!

Implementation Problems

1) Commercial or Custom data base

2) Language

a) Standard language • library

Simula, Pascal

3) Language designed for layout

a) Ron Ayers language, Caltech

b) This makes Jrtuork easy, but procedural attachment difficult

c) does not handle tent, simulation well

Placement and Routing of Arbitrary-Size Rectangular Blocks (B. Preas)

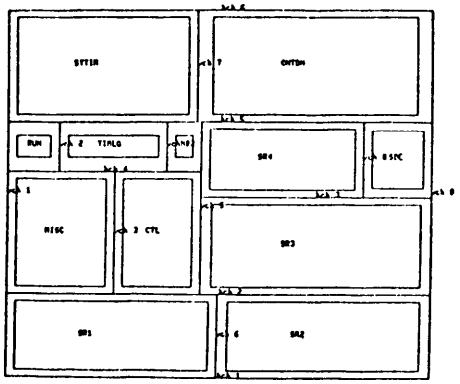
HIERARCHICAL INTEGRATED CIRCUIT LAYOUT

- MANY LARGE SCALE INTEGRATION (100,000 TRANSISTORS)
- HIERARCHICAL DECOMPOSITION
- TOP DOWN DESIGN/BOTTOM UP IMPLEMENTATION

LAYOUT OF ONE LEVEL

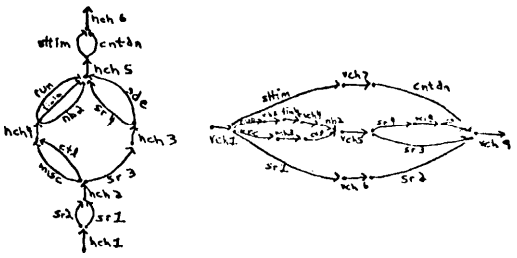
- ARBITRARILY SHAPED RECTANGULAR BLOCKS
- FIXED CONNECTION POSITIONS
- ROUTING IN CHANNELS BETWEEN BLOCKS
- AUTOMATIC PLACEMENT AND ROUTING

CONTINUED



CHIP SURFACE RADAR CONTROLLER

CHANNEL DEFINITIONS

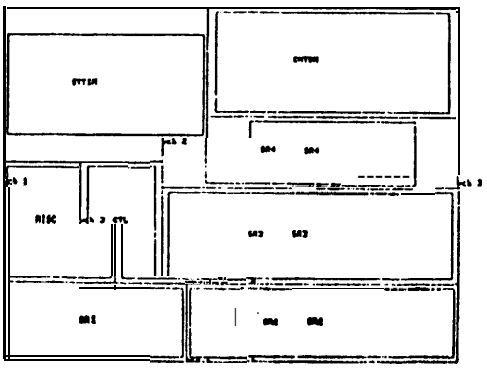


CHANNEL POSITION GRAPHS

PLACEMENT

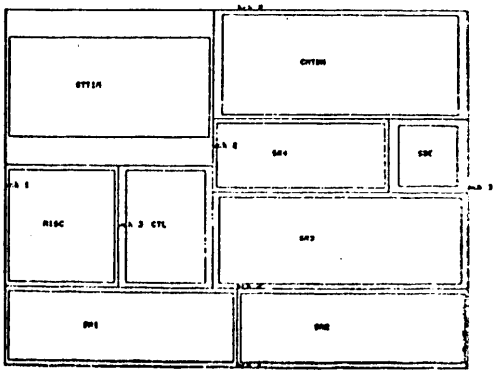
- FIND CHANNELS OF SEED
- INITIAL PLACEMENT
- PLACEMENT IMPROVEMENT

CONTINUED



INTERMEDIATE PLOT RADAR CONTROLLER SEED

PARTIAL PLACEMENT

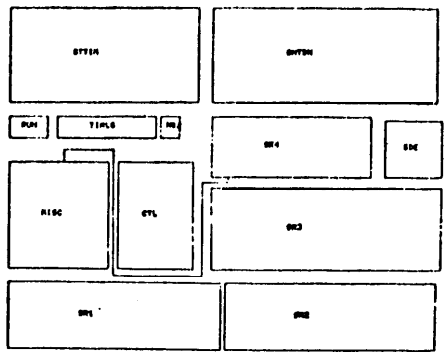


INTERMEDIATE PLOT RADAR CONTROLLER

NEXT PARTIAL PLACEMENT

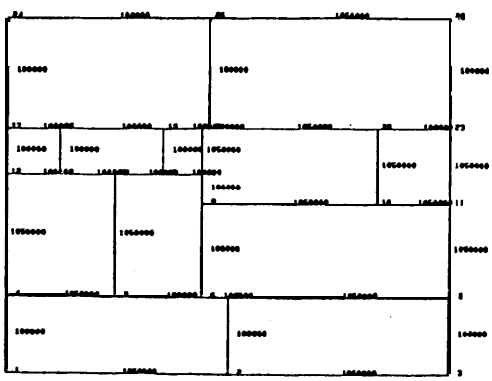
ROUTING

- FIND CHANNEL ROUTING ORDER
- INITIAL LOOSE ROUTE
- LOOSE ROUTE IMPROVEMENT
- TRACK ASSIGNMENT



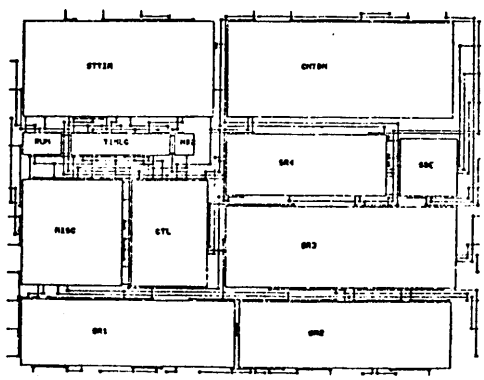
ELECTRICAL WET RADAR CONTROLLER

LOOSE ROUTE OF ONE NET



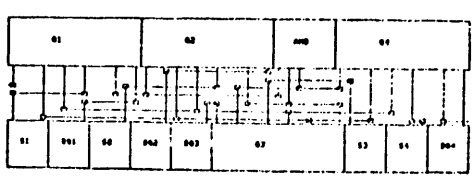
CUB CHANNEL WEIGHTS RADARCONTROLLER

CHANNEL INTERSECTION GRAPH



COMPLETE CHIP RADARCONTROLLER

M M



CONCLUSIONS

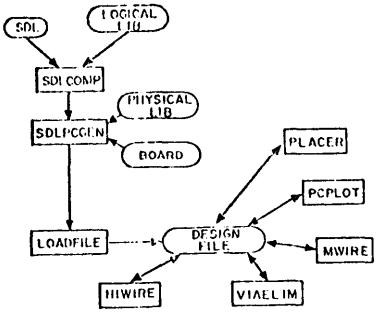
- HIERARCHICAL LAYOUT IS PRACTICAL

LAYOUT METHOD	COMPUTER TIME (MINUTES)	AREA (SQ. MILS)	PERCENT AREA
STANDARD CELL	a2	15,375	100%
HIERARCHICAL METHOD	16	12,658	82%

- PROBLEMS

The SPRINT System for PC Design
(K. Stevens)

SPRINT System Structure



Data Entry

Interactive schematics drawing system

Text input via SDL (Structural Design Language)

Possibility to interface to existing logic simulators (TEGAS, DLASAR, TESTAIDD, etc.)

Input compiler checks design for consistency

Logical Library

map:, pin names into pin numbers

loading *data*

input/ output definitions

logical equivalence data (e.g. inputs of NAND gate)

physical equivalence data (e.g. Vcc on more than one pin)

synonyms (alternate component named: 7400, SN7400)

possibility to store othor data: power dissipation
reliability
cost
vendor

Physical Library

exact pin coordinates

placement obetruction size

pad type definition (symbolic)

via/ routing obstructions

silk outline

Board Definition

physical board outline

logical board space

connector location and characteristics

via/ placement/ *routing* obstructions

pad sire definition

dcsign rules (line widths and spacing)

Design Fi ls

central data base for a single PC design

accessed by all programs

optimized for PC design appliiation

hierarchy of subfiles and dirctorles

can be extended easily

PLACER

major phases:

place critical components
initial placement (manual/ auto)
placement improvement (auto)

manual change capability at all times

free movement between phases

Automatic Placement Capabilities

user-dcfincd placement cells
user-defined placement restrictions

initial placement: manual
constructive algorithm
random

optimization: pairwisc interchange
wirelength minimization
crossing count minimization

hierarchical automatic placement o f different-size components (including discretes)

Manual Placement Capabilities

rove
exchange
delete
rotate
zooming
rulers

editing of old designs possible

Routing Subsystem (SPROUT)

Efficient heuristic line-search algorithm

Based on user-defined grid

Storage required 12 bits/ grid cell
(60 kB for 10 x 10 inch board, 50 mil grid)

Single line width in a run; several widths can be run consecutively

Dynanic dcsign rule checking

Handles 1, 2 or *2n* layers

Automatic via elimination for short wrong-aide segments

Best-fit placement o f vias

User controls search depth, path refinement

Can use off-grid vias

Ula Elimination

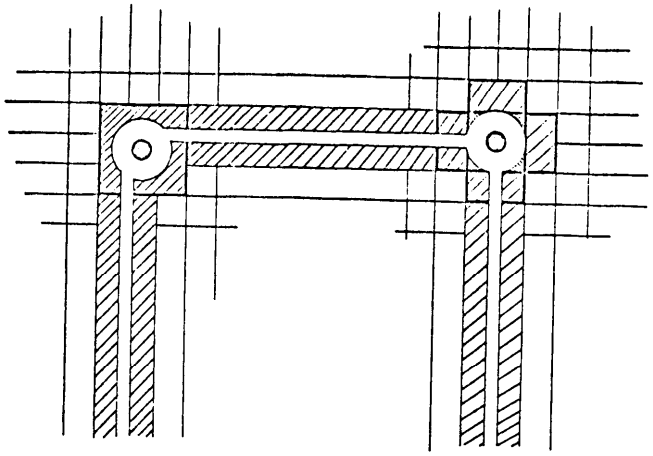
Second-order optimization

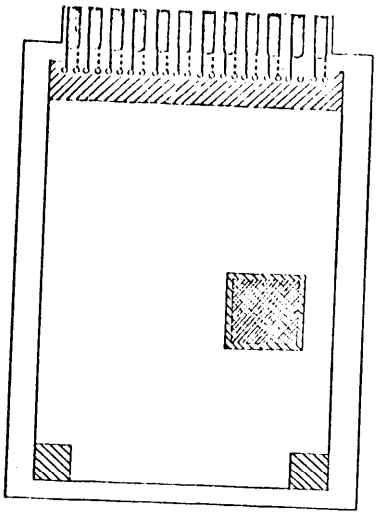
Graph model allows global minimiration

Critical *segments* not moved

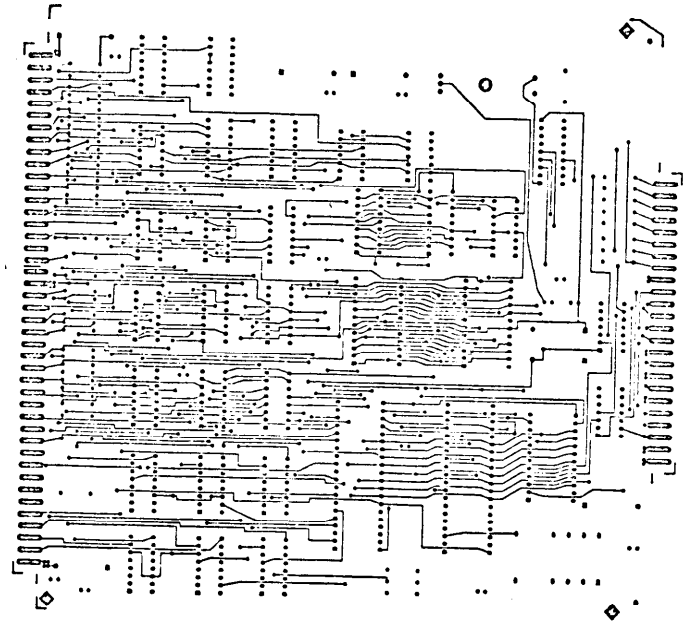
Design rules maintained

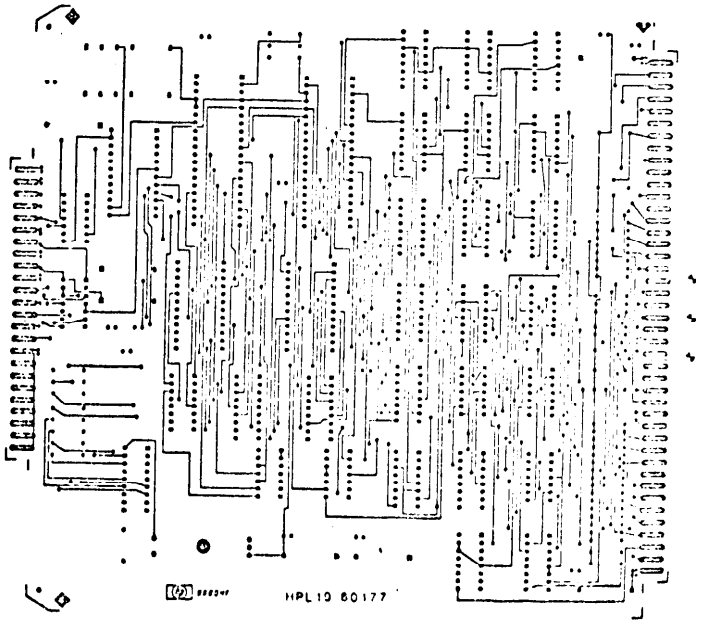
Example	T1	T2	H1
Total # pins	610	1220	2506
# LAYERS	2	2	2
Gridsize (mil)	25	25	50
Board area (sq. in.)	24	63	143
# ICs	20	48	150
# discretes and others	37	59	71
sq. in. per IC	.62	.73	.51
CPU time (sec IBM 168)	17	33	75
% conpiction	98	96	85





Board Definition





PROCESS AND DEVICE SIMULATION

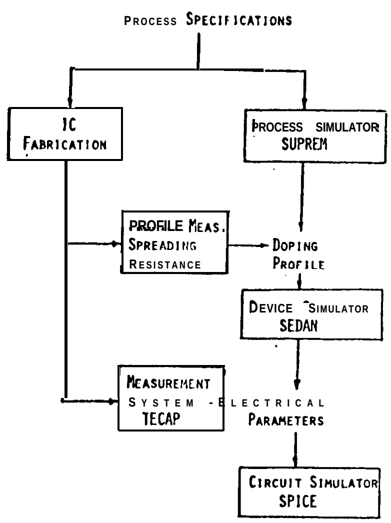
Massimo Vanzi

OVERVIEW

PROCESS SIMULATION (SUPREM)

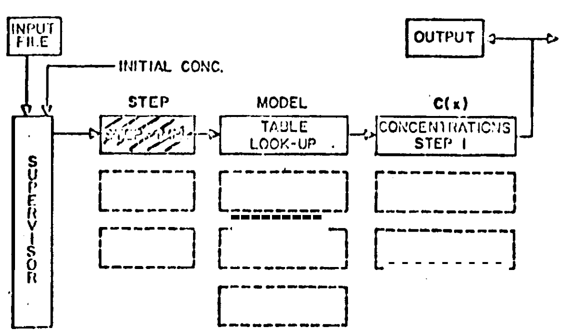
DEVICE ANALYSIS (SEDAN)

RESULTS AND APPLICATIONS



EXAMPLE :

- Boron implantation
- Oxidation
- Arsenic diffusion

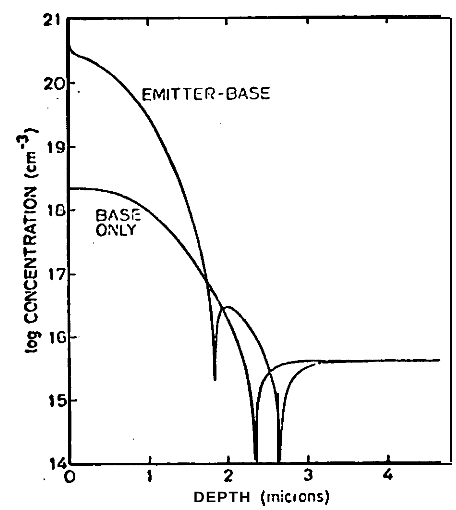


INPUT SAMPLE

```
TITLE SUPREMII EXAMPLE INPUT FILE        }
GRID  DYSI=.02, DPTH=2, TRAY=6           }INITIALIZATION
SUBS  ELEM=B, CONC=2E15, ORNT=111        }

LOAD  LUNM=22, TYPE=8                    }INPUT FROM DISK

PLOT  IDIV=Y, TOTL=Y, CHIN=15, NOCC=8, WIND=6   }
PRINT HEAD=Y, IDIV=N, TOTL=N                    }OUTPUT CONTROL

MODEL NAME=MPH1, DSXO=4E10, DSXV=4E10, ESXD=3.56   }EXT MODEL SPEC

COMM  ION IMPLANTATION                               }
STEP  TYPE=IMPL, ELEM=AS, TIME=70, CONC=1.20         }STEP 1

COMM  OXIDATION                                                   }
STEP  TYPE=OXID,TEMP=1000,TIME=10,MODL=DRYO,MODL=MPH1             }STEP 2
STEP  TYPE=OXID,TEMP=1000,TIME=20,MODL=WETO,MODL=MPH1             }STEP 3
STEP  TYPE=OXID,TEMP=1000,TIME=5,MODL=DRYO,MODL=MPH1              }STEP 4

COMM  REMOVE OXIDE                       }
STEP  TYPE=ETCH, TEMP=25                 }STEP 5

SAVE  LUNM=21, TYPE=8                    }SAVE DATA ON DISK

END
```

PROCESS MODELS :

- Ion Implantation
- Predeposition
- Oxidation / Drive In
- Epitaxial Growth
- Etching
- Oxide Deposition

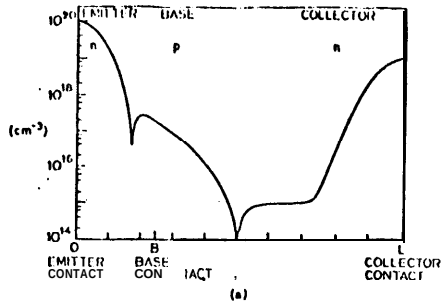


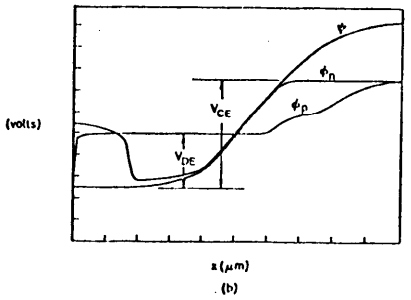
INPUT:

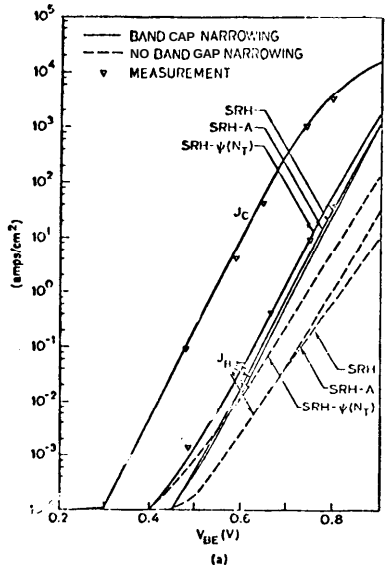
- Impurity Profile (SUPREM or analytical)
- Terminal Voltages
- Grid Specification
- Physical Model Specification
- Timestep

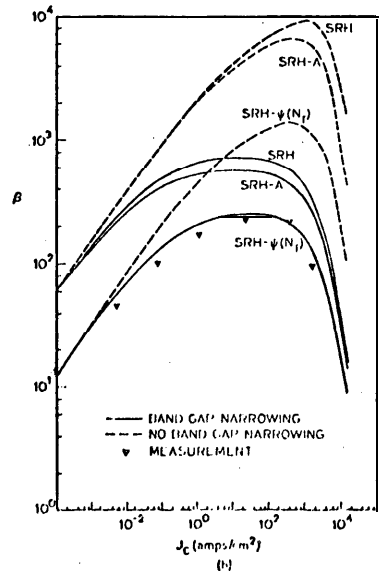
TYPE OF DEVICE :

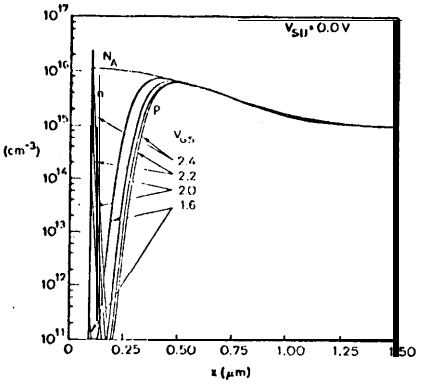
- NPN Bipolar Transistor
- PN Diode

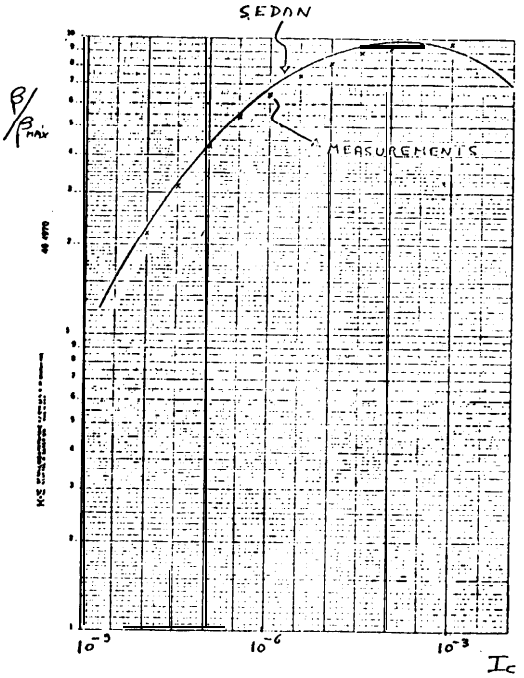

EMITTER
BASE
COLLECTOR
n
p
n
(cm⁻³)
EMITTER CONTACT
BASE CONTACT
COLLECTOR CONTACT
(a)

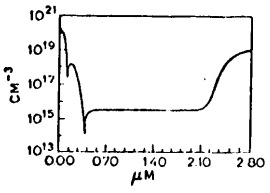

(volts)
V_{CE}
V_{DE}
x (μm)
(b)

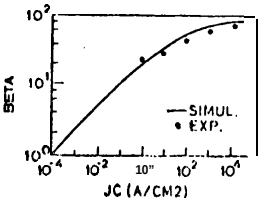

BAND GAP NARROWING
NO BAND GAP NARROWING
MEASUREMENT
SRH
SRH-A
SRH-ψ(N_T)
(amps/cm²)
V_BE (V)
(a)


SRH
SRH-A
SRH-ψ(N_T)
β
BAND GAP NARROWING
NO BAND GAP NARROWING
MEASUREMENT
J_C (amps/m²)
(b)


V_{SUB} = 0.0V
N_A
n
p
V_{BE}
2.4
2.2
2.0
1.6
(cm⁻³)
x (μm)


SEDAN
MEASUREMENTS
β/β_max
Ic


CM⁻³
μM


BETA
SIMUL.
EXP.
JC (A/CM2)

OUTPUT :

- p, n, ψ (Dependent Variables)
- Jn, Jp, ϕ_n, ϕ_p, E
- Terminal Current Densities (Jc, Jb, Je)
- Beta
- Junction Capacitances
- Sheet Resistances at each bias point
- Gummel Number t-5

PHYSICAL MODELS :

- Shockley-Read-Hall Recombination with constant or concentration dependent lifetimes
- AUGER Recombination
- Band-Gap-Narrowing
- Mobility (concentration and field dependent)

BASIC EQUATIONS

Continuity :

$$\frac{\partial p}{\partial t} = -\frac{1}{q}\frac{\partial J_p}{\partial x} - U$$

$$\frac{\partial n}{\partial t} = \frac{1}{q}\frac{\partial J_n}{\partial x} - U$$

Transport :

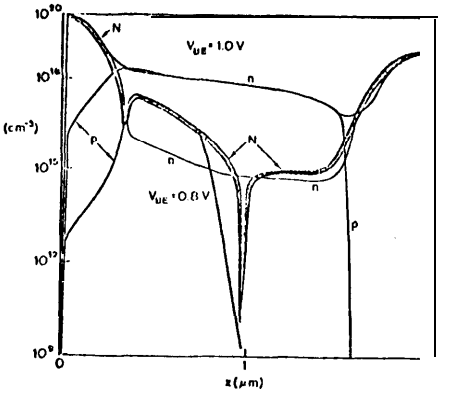
$$J_n = q\mu_n n E + q D_n \frac{\partial n}{\partial x}$$

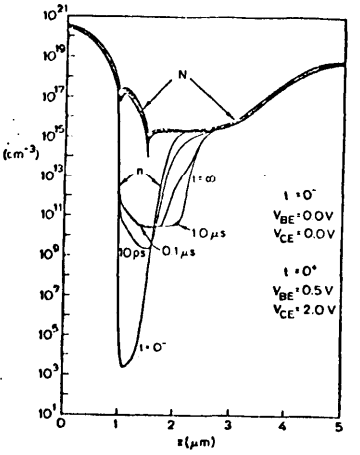
$$J_p = q\mu_p p E - q D_p \frac{\partial p}{\partial x}$$

Poisson :

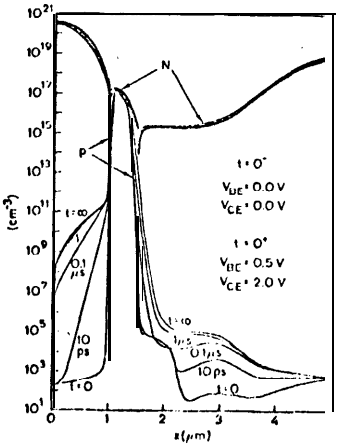
$$\frac{\partial^2 \psi}{\partial x^2} = -\frac{q}{\epsilon}\left\{p - n + N_D - N_A\right\}$$

```
TITLE     SEDAN I NPUT TEST
GRID      NREG=1, STSZ= .02, NSTP=50
GRID      NREG=2, STSZ=.01 , NSTP=100
GRID      NREG=3, STSZ=.04 , NSTP=50
LOAD      LUNN=22
DEVICE    TYPE=NPNT, BACO=1.2

PROFILE   TYPE=SUPR
BIAS      VBEF=0, VBEL=.7, VBES=.1, VCEF=0,
+         VCEL=2, VCES=1
MODEL     SRHR=Y, AUGE=N, BGNW=Y

PRINT     HEAD=Y, PRT1=Y

END
```





(a)



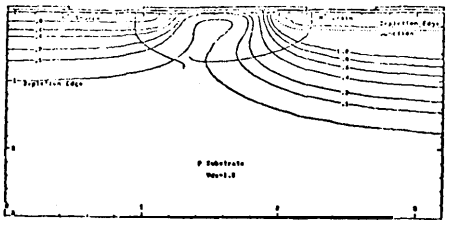
(b)

Two-Dimensional Device Analysis
(C. Price)

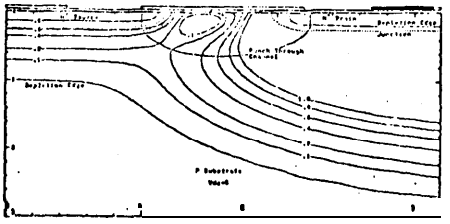
OVERVIEW

DESCRIPTION & MOTIVATION
MODELING EQUATIONS
DISCRETIZATION GRIDS
SOLUTION METHODS
TANDEM

PUNCH THROUGH

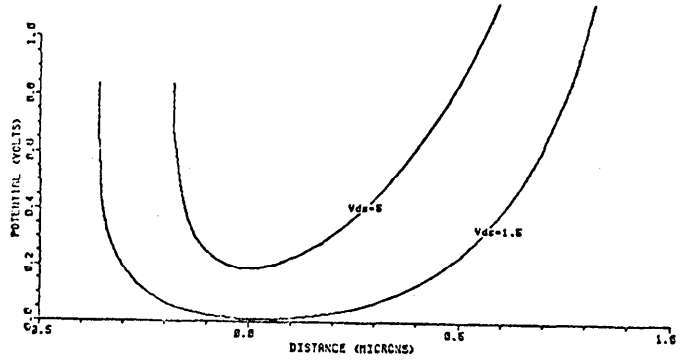


Vds = 1.5 VOLTS

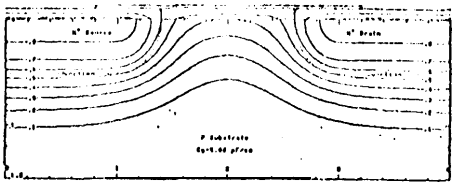


Vds = 5.8 VOLTS

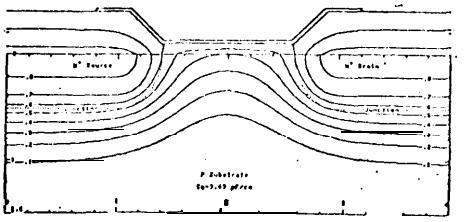
POTENTIAL ALONG PUNCH THROUGH CHANNEL



GATE CAPACITANCE



Cg = 5.88 pF/cm



Cg = 3.69 pF/cm

MODEL EQUATIONS

POISSON – $\vec{\nabla}\cdot(\epsilon\vec{\nabla}\psi) = n - p - N$

CONTINUITY – $\frac{\partial n}{\partial t} = \frac{1}{q}\vec{\nabla}\cdot\vec{J}_n + G_n - R_n$

– $\frac{\partial p}{\partial t} = -\frac{1}{q}\vec{\nabla}\cdot\vec{J}_p + G_p - R_p$

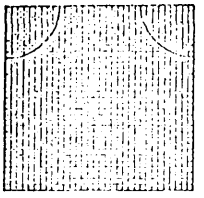
TRANSPORT – $\vec{J}_n = -q\mu_n n\vec{\nabla}\psi + qD_n\vec{\nabla}n$

– $\vec{J}_p = -q\mu_p p\vec{\nabla}\psi - qD_p\vec{\nabla}p$

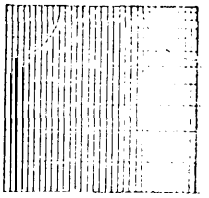
BOLTZMANN – $n = n_i \exp[q(\psi-\phi_n)/kT]$

– $p = n_i \exp[q(\phi_p-\psi)/kT]$

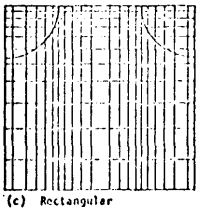
DISCRETIZATION GRIDS



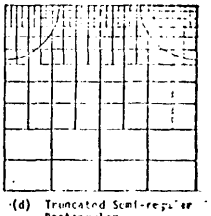
(a) Regular Rectangular



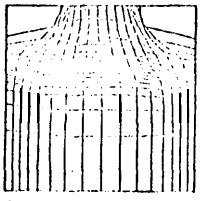
(a) Semi-regular Rectangular



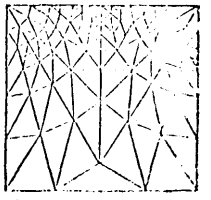
(c) Rectangular



(d) Truncated Semi-regular Rectangular



(e) Relaxed Rectangular



(f) Triangular

SOLUTION METHODS

SYSTEM OF EQUATIONS :

SIMULTANEOUS - SIZE = 3mn x 3mn
QUADRATIC CONVERGENCE

ALTERNATING - SIZE = 3 * (mn x mn)
LINEAR CONVERGENCE

MATRICES :

RELAXATION - SOR, SLOR, SBOR

DIRECT - L U DECOMPOSITION

ITERATIVE - STONE'S METHOD

TANDEM

STATIC SOLUTION (POISSON)

FINITE DIFFERENCE

NON-PLANAR SURFRCE

SLOR

STORAGE - 8mn VARIABLES

EXECUTION - 180 SEC ON HP2117F
25 S E C ON DEC-20

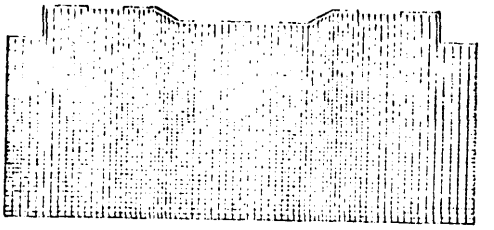
DOPING PROFILES - DIFFUSION, IMPLANT, SUPREM

OUTPUT - TERMINAL GRAPHICS

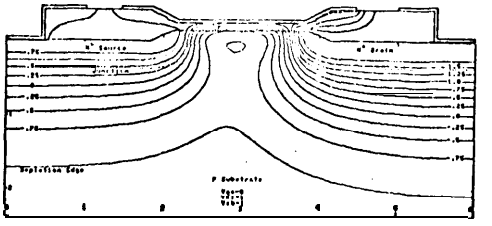
SAMPLE I NPUT

```
TITLE       SHORT CHANNEL MOSFET - IMPLANTED CHANNEL

STRUCTURE   FILE.OUT=MOS1

  SUBSTRATE  CONCENTRATION=1E15  TYPE=P  WIDTH=6  DEPTH=2.5
  CHANNEL    PEAKCONC=3E16  TYPE=P  Y.PEAK=0  Y.STDEV=.2
  SOURCE     DIFFUSION  PEAKCONC=1E19  TYPE=N  WINDOW=4  Y.PEAK=0
  +            Y.JUNCTION=.5  X.ERFCN
  DRAIN      DIFFUSION  PEAKCONC=1E19  TYPE=N  WINDOW=4  Y.PEAK=0
  +            Y.JUNCTION=.5  X.ERFCN

  INSULATOR  SOURCE  THICKNESS=.4  X.WINDOW=4  C.WINDOW=1
  INSULATOR  GATE  THICKNESS=.1
  INSULATOR  DRAIN  THICKNESS=.4  X.WINDOW=4  C.WINDOW=1

  ELECTRODE  SOURCE  WIDTH=1  WORK.FUNCTION=.7
  ELECTRODE  GATE  WIDTH=3  WORK.FUNCTION=.7
  ELECTRODE  DRAIN  WIDTH=1  WORK.FUNCTION=.7

  OSS        CONCENTRATION=5E10

  GRID       XGRD.MIN=.05  XGRD.MAX=.1  X.POINTS=78
               YGRD.MIN=.02  Y.POINTS=70

  END

SOLUTION    FILE.OUT=MOS2

  BIAS       SUBSTRATE  IMPURITY  POTENTIAL=-1

  BIAS       SOURCE  ELECTRODE  POTENTIAL=0
  BIAS       SOURCE  IMPURITY  POTENTIAL=0

  BIAS       GATE  ELECTRODE  POTENTIAL=0

  BIAS       DRAIN  ELECTRODE  POTENTIAL=1
  BIAS       DRAIN  IMPURITY  POTENTIAL=1

  END

PLOT.2D     X.MIN=0  X.MAX=6  Y.MIN=-.4  Y.MAX=2.5  BOUNDARY  GRID
  END

PLOT.2D     X.MIN=0  X.MAX=6  Y.MIN=-.4  Y.MAX=2.5
              BOUNDARY  JUNCTIONS  DEPL.EDGE
  CONTOUR   POTENTIAL  MIN.VALUE=-.75  MAX.VALUE=1.5  DEL.VALUE=.25
  END

PLOT.1D     POTENTIAL  HORZ.LEFT=0  HORZ.RIGHT=2
              VERT.BOTTOM=-1  VERT.TOP=2
  EXTREMA   X.MIN=2  X.MAX=4  Y.MIN=0  Y.MAX=1  MAXIMUM
  END
```

RESULTS Of- SAMPLE INPUT

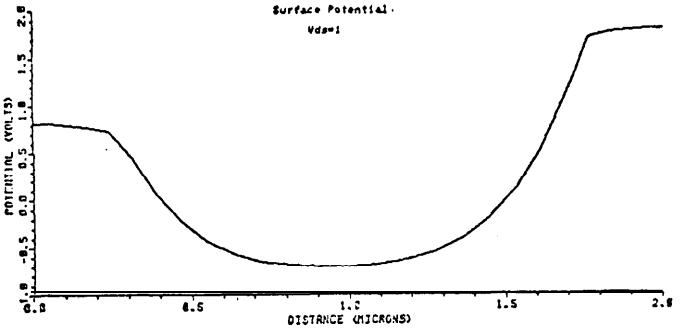


STRUCTURE AND GRID

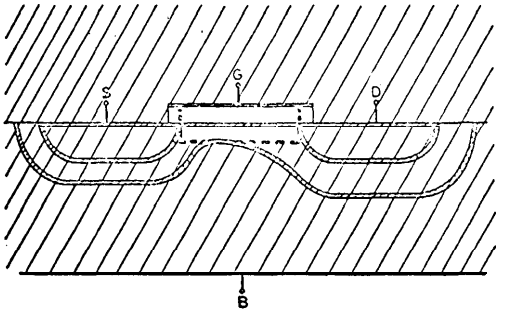


POTENT1 AL CONTOURS

SURFACE POTENTIAL



SIMPLIFIED 2-D SIMULATION



CURRENT EFFORTS

SINGLE CRRRIER + POISSON

AUTOMATIC GRID GENERATION

NON-RECTANGULAR FINITE DIFFERENCE

SIMPLIFIED 2-D ANALYSIS

IMPORTANCE IN VLSI OF :

TWO-DIMENSIONAL DEVICE MODELING

LINKING OF PROCESS & DEVICE MODELING

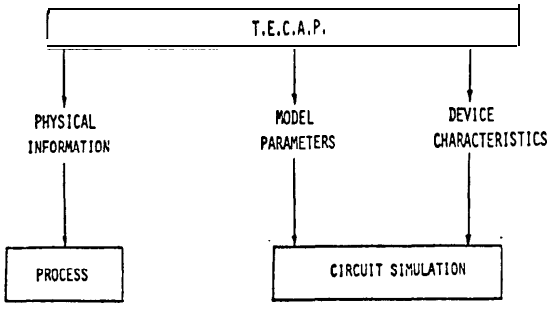
T.E.C.A.P.

TRANSISTOR ELECTRICAL CHARACTERIZATION AND ANALYSIS PROGRAMS

EBRAHIM KHALILY
JULY. 1979

TECAP PRODUCES RELIABLE, THOROUGH, AND ACCURATE INFORMATION ON BIPOLAR AND MOS TRANSISTORS TO SUBSTANTIALLY INCREASE THE IC DESIGNERS UNDERSTANDING OF DEVICE PERFORMANCE AND COMPUTER AIDED DESIGN MODELS.

APPLICATIONS:



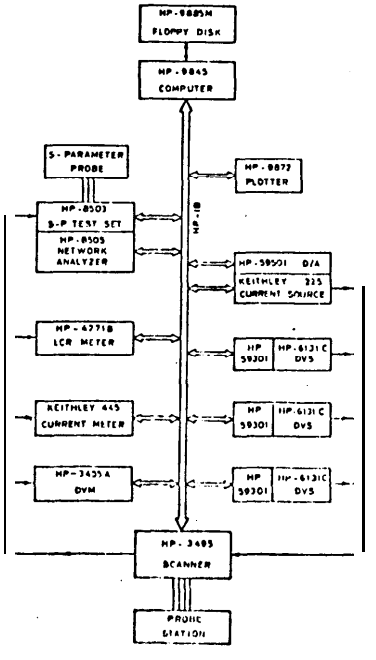
T.E.C.A.P.

THE INTERACTIVE 9845/HP-IB BASED SYSTEM GIVES THE USER A QUANTITATIVE UNDERSTANDING OF THE PROCESS AND DEVICE PERFORMANCE THROUGH FOUR AREAS:

1. INTERACTIVE BASIC ELECTRICAL DC/AC MEASUREMENT
2. CAD MODEL PARAMETER EXTRACTION
3. MODEL/PARAMETER ANALYSIS
4. BASIC STATISTICAL DATA ANALYSIS

FEATURES:

- ACCURATE
- FAST
- INTERACTIVE
- MODULAR
- COMPATIBLE TO CAD PROGRAMS
- EASY TO USE
- EASY TO PROGRAM

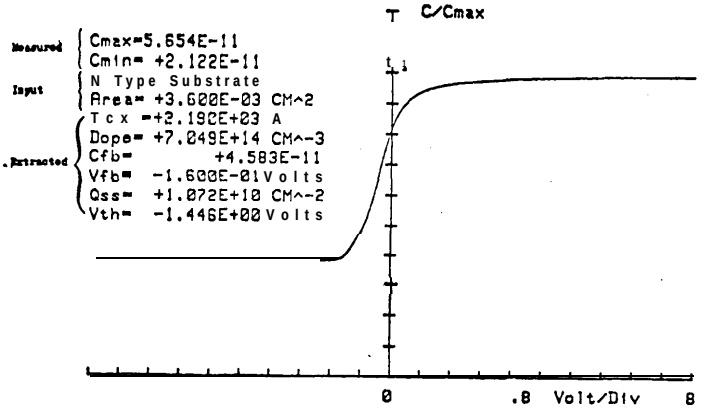


BLOCK DIAGRAM OF THE SYSTEM.

Vb= 0

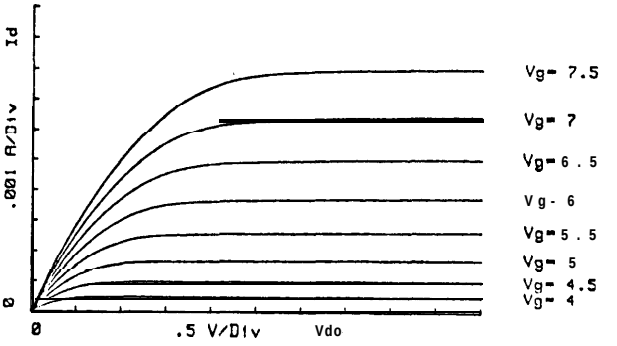
CAPABILITIES

1. MOS DC MEASUREMENT
   - PARAMETER EXTRACTION
   - MODEL/PARAMETER ANALYSIS
2. BJT DC MEASUREMENT
   - PARAMETER EXTRACTION
   - MODEL/PARAMETER ANALYSIS
3. C-V MEASUREMENT
   - P-N JUNCTION CAPACITANCE PARAMETER EXTRACTION
   - MOS GATE CAPACITANCE PARAMETER EXTRACTION
4. S-PARAMETER MEASUREMENT
   - CONVERSION TO Y, H, Z
   - HIGH FREQUENCY MODEL PARAMETER EXTRACTION

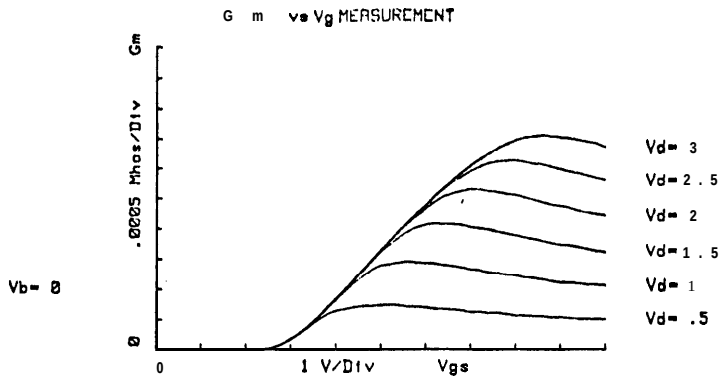


ID#: Dick CS-1/4
E. K
09/11/78 3:09 P M

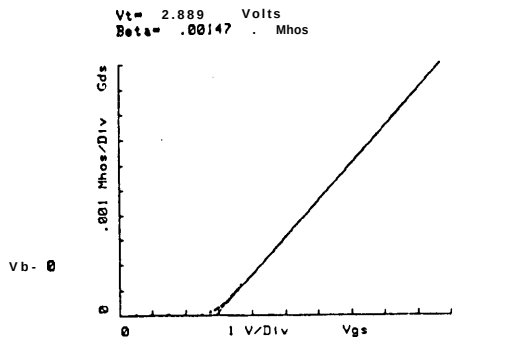
Id vs Vd MEASUREMENT



ID#: 2N-435 1
E.K.
11/30/78 1:19 PM

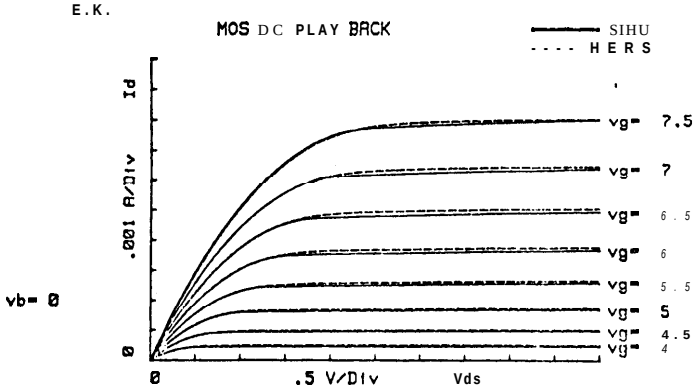


ID+: 2N-4351
E.K.
11/30/78 1:39 PM



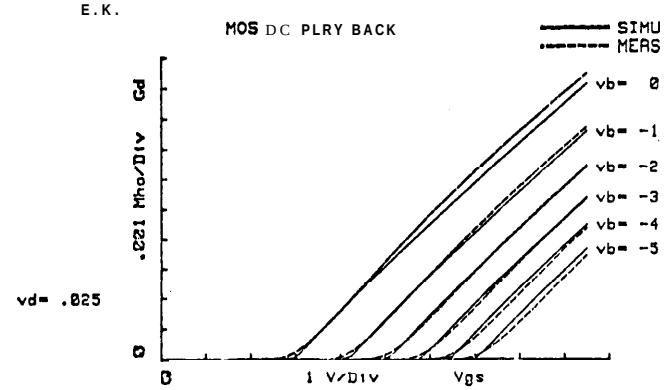
EXTRACTION OF β AND V_{TO} FROM DRAIN-SOURCE CONDUCTANCE.

Date: 12/01/78 1:05 PM
E.K.



Beta0=1.48E-03 ;Vt0=+2.90 ;Phif=0.37 ;Gamma=2.670 ;Theta=0.016
Gleff=1.00E+09 ;Ecrit=1.00E+04 ;Utra=1.00E+05 ;Cox=1.00E-15
Rs= 0 ;Rd= 0 ;L=0.0010 ;Xj=0.00008 ;N Channol

Date: 12/01/78 1:18 PM
E.K.



Beta0=1.48E-03 ;Vt0=+2.90 ;Phif=0.37 ;Gamma=2.670 ;Theta=0.016
Gleff=1.00E+09 ;Ecrit=1.00E+04 ;Utra=1.00E+05 ;Cox=1.00E-15
Rs= 0 ;Rd= 0 ;L=0.0010 ;Xj=0.00008 ;N Channol

A High-performance Raster-Scan Graphics System

Forest Baskett and Andreas Bechtolsheim
Computer Systems Laboratory
Stanford University

Overview

Applications

VLSI project - design automation
TEX project - advanced text processing
Highly interactive program environments

Initial configuration

VAX 11/780, DEC-20, or Foonly host computer
Graphic system connected through parallel I/O port

Future configuration

Personal Computer System (Single User)
Z-8000/68000 Host Processor
Virtual Memory
Hard Disk Storage
Ethernet Connection
Workstation Formfactor

Graphics System Functionality

High Resolution Frame Buffer

1024 * 1024 pixels
can be organized 512 * 512 * 4 for color

Monochrome Display

1024 * 800 pixels visible area
Interlaced or non-interlaced monitors
1024 * 224 pixels invisible area used for
character sets, cursors, symbols, etc.

Color Display

512 * 480 pixels visible area
4 Bits per pixel, mapped into 65536 colors
color map can be changed dynamically

Graphics Processor Characteristics

1) provides high-speed frame buffer access & manipulation
vector drawing rate 1 pixel per microsecond
raster manipulation rate up to 16 pixels per microsecond
fully "soft" character sets
fast image transformations

2) functional interface between host computer and frame buffer
graphics processor handles all frame buffer accesses
performs address mapping and bit shifting
maintains queue of graphics instructions
reduces host load

Graphics Input Device

Tablet or Mouse
under control of host computer

Hardware Overview

Two Modules

Graphics Processor
Frame Buffer

Graphics Processor

microprogrammed, bit slice machine
16 bit data paths, 125 nsec cycle time
1k by 40 Bit Microstore
barrel shifter
can control up to eight frame buffers

Frame Buffer

128 kByte dynamic RAM (1024*1024)
64 Mbit bandwidth
programmable video controller
Color Map with 16 entries
6 bit Red, 6 bit Green, 4 bit Blue
Function Unit
performs raster operations (bit set, clear, xor, etc.)
in a single read modify write cycle

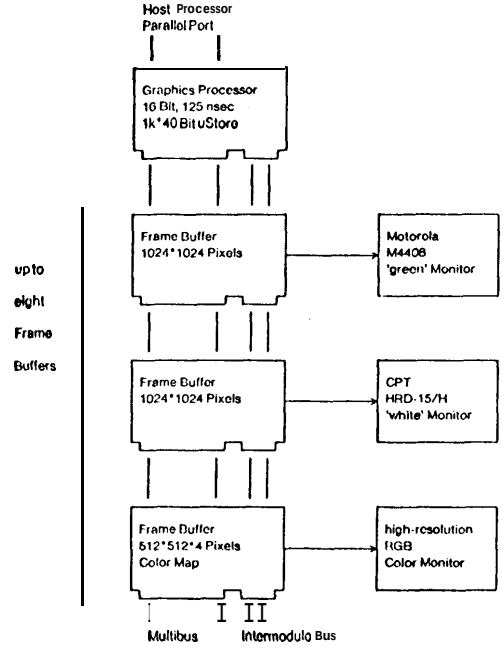
Video Monitors

CPT monitor HRD 15/H
"white", non interlaced, 8 by 10.5 inch
$900

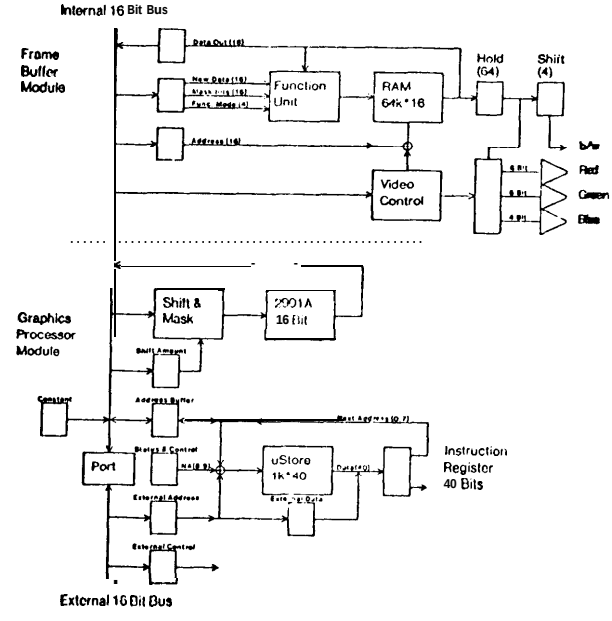
Motorola M4408
"green", interlaced, 8 by 10.5 inch
$475

Hitachi HM-2019
high resolution mask
25 Mhz video bandwidth
$4100

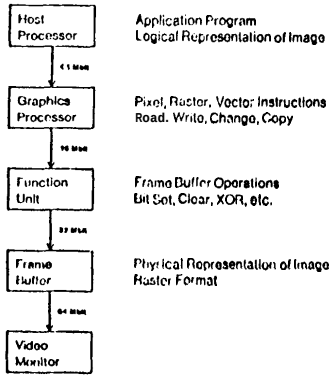
Graphics System... Hardware Modules



Graphics System...Data Paths



Graphics System.. Levels of Functionality



- Unload Host Processor
- Significant overlap can result

- Balance bandwidth, performance, and cost
- Average bandwidth decreases at higher levels

- Understand interface to Frame Buffer
for future VLSI graphics processor.

Graphics Processor Macro Instruction Set

Orthogonal Instruction Set

Frame Buffer Plane
Current Location
Graphical Object
Graphical Descriptor
Graphical Operation

Graphical Objects

Pixels
Rasters
Vectors

Graphical Descriptors

Define Access to Object
Can be used for Automatic Current Location Update
Examples: Rasters (Height, Width), Vectors (dX, dY)

Graphical Operations

Read Source Object
Write Destination Object
Erase Destination Object
Invert Destination Object
Move Source Object to Destination Object
Copy Source Object to Destination Object
BitSet Destination Object with Source Object
BitReset Destination Object with Source Object
BitXOR Destination Object with Source Object

Summary

A raster scan graphics system has been designed that combines
- high speed graphical operations
- high resolution display
- low cost

Applications for this graphics system include
- highly interactive man-machine interfaces
- replacement of vector graphics
- computer-aided design
- advanced text processing

The graphics hardware is partitioned in two modules
- the graphics processor
- the frame buffer

Possible configurations are
- single user workstations
- up to eight frame buffers per graphics processor
- combination color/monochrome displays
- displays with additional bits per pixel
- displays for multiple users

APPENDIX

ATTENDEES

at

FIRST WORKSHOP ON DESIGN AUTOMATION.
Stanford University
Stanford California 94305

July 3 - 4, 1979

Akino, Toshito
Matsushita Electronics Corp.
Kotari- Yakimachi
Nagaoka - Kyo
Kyoto, JAPAN

Alcock, Linda
Fairchild
MS 22 - 130
464 Ellist St.
Mountain View, CA. 94042

Alwin, V. C.
RCA - SSTC
Rt. 202
Somerville, N.J. 08876

Belle Isle, A. P.
General Electric
Electronics Laboratory
Electonics Park - Bldg. 3
Syracuse, New York 13221

Bering, Doug
Lawrence Livermore Laboratory
L-156
Livermore, CA 94550

Bobas, Andrew
BNR
185 Corkstown Rd.
Ottawa, CANADA

Brooksby, Merrill
Hewlett Packard

Brown, Harold
Stanford University
Stanford, CA. 94305

Butner, Steve
BNR Inc.
3174 Porter Drive
Palo Alto, CA. 94304

Casselman, Thomas
Honeywell Corporation
Material Sciences
10701 Lyndale Aves
Bloomington, MN 55420

Chesnut, Randal 1
Amdahl Corporation
1250 E. Arques Avenue
M/S 233 Bldg. E-l Rm. 162
Sunnyvale, CA 94086

Cohen, Ben
Hughes Aircraft Co.
Building 12, MSX103
Culver City, CA 90230

Crippen, Richard
Fairchild
464 Ellis Street
Mountain View, CA 94042

Crowley, Jeri Jane
Texas Instruments, Inc.
M.S. 3907
Dallas, TX 75265

Daniel, Donald D.
Code 3511
u. s. Naval Weapon Center
China Lake, CA 93555

Dewar, Peter A.
Science Research Council
Rutherford Laboratory
Chilton Oxon ENGLAND

El-ziq, Yacoub
Sperry Univac
M.S. 4693/DA & T Dept.
Roseville, MN 55113

Eriksson, L-O
Telefon AB LM Ericsson
TN/X/TdkC
S-12625 Stockholm SWEDEN

Fiduk, Kenneth W.
Texas Instruments, Inc.
M.S. 3907
13500 North Central Expy.
Dallas, TX 75265

Findlay, Pamela J.
Hewlett Packard
Santa Clara Division
5301 Stevens Creek Blvd.
Santa Clara, CA 95050

Fineman, Mark
DEC
58 Squire Drive
Nashua, NH 03060

Fisher, Michael **G.**
Amdahl Corporation
1250 E. Arques
Sunnyvale, CA 94086

Floutier, Denis
Universite' des Sciences
et Techniques du Languedoc
Laboratoire **d'Automatipue**
Place E. Bataitlon
34100 Montpellier FRANCE

Friberg, Tom
LM Ericsson
Stockholm SWEDEN

Fryklund, Jim
Oregon State University
Computer Center
Coruallis, Oregon 97331

Ganpule, Dilip R.
Amdahl
East Arques Ave.
Sunnyvale, CA 94086

Geyer, J. M.
General Electric
Bldg. 3 Rm. 227
Electronics Park
Syracuse, New York 13221

Goheen, Mark
Amdahl
1250 E. Arques
Sunnyvale, CA 94086

Goto, Satoshi
NEC
4-1-1 **Miyazaki,** Takatsu-ku
Kawasaki, JAPAN

Graves, Ken
Amdahl Corp.
1250 E. Arques Ave.
Sunnyvale, CA 94086

Hartenstein, Deiner W.
Kaiserslautern University
Postfach 3049
D-675 Kaiserslautern
F. R. GERMANY

Hartmann, Robert
Signetics
811 E. Arques
Sunnyvale, CA. 94086

Hartshorn, Carl
Four-Phase Systems
10700 N . **DeAnza**
Cupertino, CA 95014

Haydamack, Bill
Hewlett Packard
1501 Page Mill Road
Palo Alto, CA 94304

Hewson, M.
JCL
Wenlock Way
Manchester
M12 **5DR,** UNITED KINGDOM

Hitchcock, Robert **B.** Sr.
IBM
Dept. V34 - Bldg. 610 River Plaza
1701 North Street
Endicott, New York 13760

Holt, Dan
AM1
3800 Homestead Rd.
Santa Clara, CA 95051

Horng, Chi-Song
American Microsystems Inc.
3800 Homestead Rd.
Santa Clara, CA 95051

Hsi, C. G.
IBM
D/84G-83E
Hopewell Junction, New York

Hsueh, Min-Yu
u. c. Berkeley
Berkeley, CA 94720

Jacquemin
University of Montpellier
(Labo. Prof. Lecoy)
Place Eugene Bataillon
34000 Montpellier
FRANCE

Kawanishi, Hiroshi
IC Div. NEC
1753 Shimonumabe
Nakahara-ku, Kawasaki
211 JAPAN

Keys, Phil
BNR Inc.
3174 Porter Drive
Palo Alto, CA 94304

Ko, Danny
Xerox
701 s. Aviation Blvd.
El Segundo, CA 90245

Kusik, Bob
Digital Equipment Corp.
Mail Stop ML 3-5/H33
146 Main Street
Maynard, Mass. 01754

Laf on
Thomson - CSF
1, Rue des Mathurins
92 Bagneux
FRANCE

Lauther
Siemens
Schertlin Str. 8
D 8000 Munchen 70
ZFA FTE3 Aut 222

Leahy, Lee
Intel
M/S 4-428
3065 Bowers Ave.
Santa Clara, CA. 95051

Lee, Gene
Hughes
Centinela Iterale
Culver City, CA 90230

Lipman, Jim
Hewlett Packard
Bldg. 28A
1501 Page Mill Road
Palo Alto, CA. 94304

MacCrisken, John E.
Phoenix Data Systems
516 Greer
Palo Alto, CA

Magnuson, Waldo G. Jr.
Lawrence Livermore Lab
P.O. Box 808, L-156
Livermore, CA 94550

Mallmann, Felix
Siemens AG
Boschetsrieder str. 123
D8000 Munchen 71
GERMANY

McCalla, Bill
HP
1501 Page Mill Rd.
Palo Alto, CA 94304

McEwen, Jeff
Intel
M/S 4-428
3065 Bowers Ave.
Santa Clara CA 95051

McGuf f in, R.
ICL
West Gorton
Manchester ENGLAND

McNamee, L. P.
Hughes Aircraft Co.
Cp-8 M/S T21
Canoga Park, CA. 91304

Moussouris, John
IBM Research
P.O. Box 218
Yorktown Hts., New York 10598

Nomura, M.
NEC
CAD Engineering Department
10, l-Chome, Nisshin-Cho
Fuchu City, Tokyo, 183 JAPAN

Off **erdahl,** Richard
Control Data
4201 N. Lexington
Arden Hills, MN 55112

Opitz
Si **emens** Ag
Schertlinstr.8
ZT ZFA AUT 243
D-8000 Muenchen 70

Organick, Elliott I.
Dept. of Computer Science
Univeristy of Utah
Salt Lake City, UT 84112

Partridge, Doug
Hughes Aircraft Co.
1510 Grissom St.
Thousand Oakes, CA 91360

Pozo, Richard
Hughes Aircraft
Blg 12, MS X103
Culver City, CA 90230

Ramchandani, Tulsi
Mail **A2-05**
Xerox Corp.
701 s. Aviation Blvd.
El Segundo, CA 90245

Raymond, T.C.
IBM Corp Dept. **B19,** Bldg. 951
South Road
Poughkeepsie, New York 12602

Rollenhagen, D. C.
General Electric
Electronics Laboratory
Bld. 3 Rm. 116 Electronics Park
Syracuse, New York 13221

Rosenberg, Larry
RCA M.Z. 177
Route 202
Somerville, NJ 08876

Rozenberg, Donald
IBM Research
Box 218
Yorktown Heights, New York 10598

Rust, Werner
Fairchild
Ellis Street, Bldg. 20
Mailstop 202755
Mountain View, CA. 94042

Sapiro, Steve
AM1
3800 Homestead Rd.
Santa Clara, CA 95051

Sato, Koji
Mitsubishi Electric Corp.
Itami, Hyogo, 664 JAPAN

Sayner, W. V. Jr.
BNR, Inc.
3174 Porter Dr.
Palo Alto, CA 94304

Shectman, Robert M.
Lawrence Livermore Labs
P.O. Box 808 M.S. L-156
Livermore, CA 94550

Shew, E.
Bell-Northern Research
Ottawa, Canada

Shiraishi, Hiroshi
Fujitsu Laboratories Ltd.
1015 Kamikodanaka
Nakahara-Ku, Kawasaki JAPAN

Smith, Merlin G.
IBM T.J. Watson Research Center
Box 218
Yorktown Hts., New York 10598

Suaya, Roberts
Fairchild Corp.
464 Ellis St. MS 17-5904
Mountain View, CA 94042

Suri, Ashok
Fairchild
590 Middlefield Rd.
Ms 17-5904
Mountain View, CA 94042

Tamura, Eiji
University of Waterloo
139 University Ave. W.
Waterloo, Ontario
N2L 3E7 CANADA

Tanaka, Zenichiro
Nikkei-McGraw-Hill
Nikkei Electronics,
Nikkei Bekkan
Tokyo, JAPAN

Thomas, Muth
Ellemtel Utv. AB
Fack 12520 ALVS70
SWEDEN

Torkelsson, Kjell
Telefon AB LM Ericsson
S-126 25 Stockholm
SWEDEN

Tower, Lee
Hughes Aircraft Co.
Bldg. 367 Mail Station C327
El Segundo, CA.

Tsukiyama, Shuji
U.C. Berkeley
1634 Oxford St. #306
Berkeley, CA. 94709

Vasboe, Barry
Bell Labs
11900 Pecos
Denver, CO. 80234

Verdillon, Andre
IMAG
Domaine Universitaire
BP53 38 Grenoble Cedex
FRANCE

Weste, Neil
Bell Labs
Roberts Road
Holmdel, NJ. 07733

Young, Tau K.
811 E. Arques Ave.
Sunnyvale, CA. 94086

Yuan, Shui
RCA Corp.
559 Riverside Dr., East
Princeton, NJ. 08540

Yuan, Steve
RCA
Route 202
Somerville, NJ. 08876