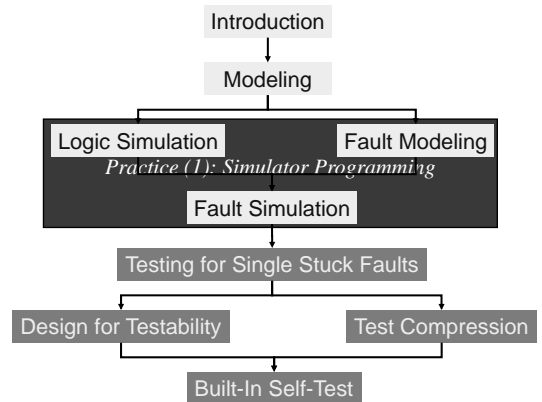# VLSI Test

Tsung-Chu Huang

Department of Electronic Engineering
National Changhua University of Education
Email: tch@cc.ncue.edu.tw

2007/11/02

Syllabus & Chapter Precedence

Outline

- Internal Model and Data Structure
- Programming the Simulator
- Practice on SynTest TurboFault.
- Explanation for Project about Logic/Fault Simulation.
- Practice on SynTest TurboScan

Project 1

1. Write a set of C (or C++) programs to read the ISCAS85 benchmark.
2. Construct an internal model (data structure).
3. Do functional logic simulation in the internal model.
4. Add a bit for fault insertion in each gate and do serial fault simulation.
5. Write a deductive fault simulator.

Example for Proj#1/ (1)

ISCAS85 Benchmark TDL Example

c17.tdl

```
MODULE      : C17;
INPUTS      :
  I1gat,
  I2gat,
  I3gat,
  I6gat,
  I7gat;
OUTPUTS     :
  I22gat,
  I23gat;
DESCRIPTION : TDL file created by Carafe of Hemlock
USE         :
DEFINE      :
  aoi21s_0(q=I7) = aoi21s(a1=I10,a2=I7gat,b=I13);
  ai2s_3(q=I10) = ai2s(a=I3gat,b=I6gat);
  ai2s_2(q=I22gat) = ai2s(a=I8,b=I12);
  i1s_1(q=I23gat) = i1s(a=I7);
  i1s_0(q=I13) = i1s(a=I12);
  ai2s_1(q=I12) = ai2s(a=I2gat,b=I10);
  ai2s_0(q=I8) = ai2s(a=I1gat,b=I3gat);
END         : MODULE;
```

Example for Proj#1/ (2)

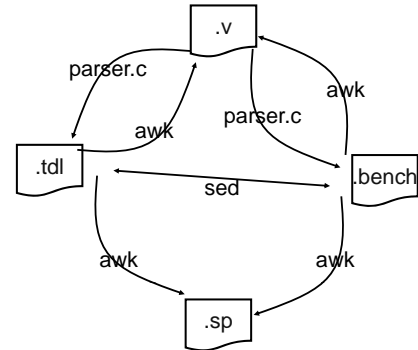**ISCAS85 Benchmark Verilog Example**

c17.v

```
module C17(I1gat, I2gat, I3gat, I6gat, I7gat);
input I1gat, I2gat, I3gat, I6gat, I7gat;
output I22gat, I23gat;
wire I7, I8, I10, I12, I13;
// ISCAS85 C17 Circuits in Verilog

aoi21s aoi21s_0(.q(I7), .a1(I10), .a2(I7gat), .b(I13));
ai2s   ai2s_3 (.q(I10), .a(I3gat), .b(I6gat));
ai2s   ai2s_2 (.q(I22gat), .a(I8), .b(I12));
i1s    i1s_1  (.q(I23gat), .a(I7));
i1s    i1s_0  (.q(I13), .a(I12));
ai2s   ai2s_1 (.q(I12), .a(I2gat), .b(I10));
ai2s   ai2s_0 (.q(I8), .a(I1gat), .b(I3gat));
endmodule;
```
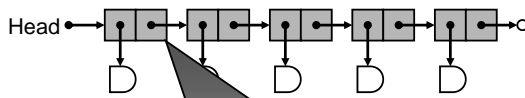
Usual Treatments of the Netlists

- Automatic Word Kits:
  1. Search some message in the file:
     - UNIX: grep, sort, wc
     - DOS: find
  2. Substitution
     - e.g. `sed 's/old/new/g' file`
  3. Line-by-line processor
     - e.g. `awk 'BEGIN{}{if($1=key) print $0}END{}' file`
- Flexible and Complete Processor
  1. Programming (using C or C++)
- Controlling All Job Sequence including Tools
  1. C shell, Perl, system( ) call in C, (Batch in DOS)

Example: Format Transformation

CONS


```
typedef struct _glist GateList;
struct _glist {
        Gate         *gate;
        GateList     *next;
}
```

Objective Structure in C

```
typedef enum _gtype GateType;
enum _gtype {OR, AND};

typedef enum _ginv GateInversion;
enum _ginv {POS, NEG};

typedef enum _gval GateValue;
enum _gval {0, 1, D, E}; // E=Dbar

typedef struct _gate GATE;
strcut _gate {
      GateList          *in, *out;
      GateType          type;
      GateInversion     inv;
      GateValue         value;
}
```

Typical Netlist Parsing

1. Reading the Netlist to a List
   - Checking redundancy and memory allocation
   - Maintain a Hash and Put into the Hash
   - Keeping unlinked notations (symbols)
   - Initializing each value
2. Linking
   - Hashing and Linking All Fan In/Out
3. Levelization
   - Sorting the Netlist by Objective Levels
4. Processing
   - Processing the Netlist according to Requirements
5. Output the Objective Message or Netlists

Example for Proj#1/ (3)
TDL Parser and Internal Model

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <alloc.h>
#define  CALLOC(n,x) ((x *)calloc(n,sizeof(x)))

typedef struct line_struct LineType;
struct line_struct {
    char  *line, id[10], ip[6][10];
    int    im;         /* max input lead */
    char   valid;
    int level;
    LineType *next; };

LineType *Head, *Tail;       /* Head and Tail of  */

typedef struct IDNode IDnode;
struct IDNode {
    char  *id;
    IDnode *next; };

IDnode  *I[10000];
```

Example for Proj#1/ (4)
TDL Parser and Internal Model

```
main(argc, argv)
  int   argc;
  char  *argv[];
{
  FILE *tdlf;
  char firstname_tdl[20], str[100],s[30],*tok,*temp, ok, Y, F;
  LineType  *t,*p,*q;
  int   i;

  if(argc != 2) {printf("Usage: tdlr file<.tdl>\n"); exit(0);}
  strcpy(firstname_tdl,argv[1]); strcat(firstname_tdl,".tdl");
  if((tdlf=fopen(firstname_tdl,"r"))==NULL)
    { printf("Open file error!\n"); exit(1);}

  do{ fgets(str,100,tdlf);
      printf("%s",str);
      sscanf(str,"%s ",s);
  }while( strcmp(s,"MODULE") );

  do{ fgets(str,100,tdlf);
      printf("%s",str);
      tok=strtok(str," ,;: ");
  }while(strcmp(tok,"INPUTS"));
```

Example for Proj#1/ (5)
TDL Parser and Internal Model

```
fgets(str,100,tdlf);
  printf("%s",str);
  tok=strtok(str," ,;: ");
  Head=Tail=NULL;

  while(strcmp(tok,"OUTPUTS")){
     t=CALLOC(1,LineType);
     t->im=0;
     t->valid=1;
     t->level=0;
     t->next=NULL;
     strcpy(t->id,tok);
     if(Head==NULL) Head=Tail=t;
     else {Tail->next=t; Tail=t;}
     fgets(str,100,tdlf);
       printf("%s",str);
     tok=strtok(str," ,;: ");
  }
```

Example for Proj#1/ (6)
TDL Parser and Internal Model

```
fgets(str,100,tdlf);
printf("%s",str);
tok=strtok(str," ,;: ");
while(strcmp(tok,"DESCRIPTION")){
   fgets(str,100,tdlf);
   printf("%s",str);
   tok=strtok(str," ,;: ");
}

do{fgets(str,100,tdlf);
   printf("%s",str);
   tok=strtok(str," ,;: ");
}while(strcmp(tok,"DEFINE"));

fgets(str,100,tdlf);
temp=((char *)calloc(1,strlen(str)+1));
strcpy(temp,str);
tok=strtok(str,"(=: ");
```

Example for Proj#1/ (7)
TDL Parser and Internal Model

```
while(strcmp(tok,"END")){
   t=CALLOC(1,LineType);
   t->next=NULL;
   t->im=t->valid=t->level=0;
   t->line=temp;
   tok=strtok(NULL,"=)");
   tok=strtok(NULL,"=)");
   strcpy(t->id,tok);
   tok=strtok(NULL,"(");
   tok=strtok(NULL,",); ");
   while(tok!=NULL) {
      if(tok[0]==')'||tok[0]==';'||tok[0]=='\n') break;
      strcpy(t->ip[t->im++],tok);
      tok=strtok(NULL,",); ");
   }
   Tail->next=t;
   Tail=Tail->next;
   fgets(str,100,tdlf);
   temp=((char *)calloc(1,strlen(str)+1));
   strcpy(temp,str);
   tok=strtok(str,"(=: ");
}
}
```

Example Programs

- Ask an account of vlsi4 from the administrator
- vlsi4.eedept.ncue.edu.tw (120.107.171.184) or vlsi13.eedept.ncue.edu.tw (120.107.171.193)
- Familiarize textedit, gcc
- Benchmark and examples are put in ~tch/VLSITest
- Exercise by putty or telnet to vlsi4