

**LAPORAN PROYEK AKHIR DASAR  
PEMROGRAMAN  
AUTOMATIC TEST PATTERN GENERATION**



**Samuel Arta Raju Manik\_13323016**

**Janter Hugo Purba\_13323039**

**Agita Rahel Panjaitan\_13323047**

**PROGRAM STUDI DIII TEKNOLOGI KOMPUTER  
FAKULTAS VOKASI INSTITUTE TEKNOLOGI DEL  
LAGUBOTI 2023**

# DAFTAR ISI

<b>1 LATAR BELAKANG.....</b>	<b>3</b>
<b>2 DESKRIPSI PROYEK .....</b>	<b>4</b>
<b>3 IMPLEMENTASI .....</b>	<b>5</b>
<b>4 PENUTUP.....</b>	<b>8</b>

# 1 LATAR BELAKANG

Test Pattern Generation (ATPG) adalah metode atau teknologi desain elektronik yang digunakan untuk menemukan urutan input (atau tes) yang, ketika diterapkan pada rangkaian digital, memungkinkan peralatan tes otomatis untuk membedakan antara perilaku sirkuit yang benar dan perilaku sirkuit yang salah yang disebabkan oleh cacat. ATPG digunakan untuk menguji perangkat semikonduktor setelah pembuatan, atau untuk membantu menentukan penyebab kegagalan (analisis kegagalan).

ATPG dapat digunakan untuk menguji rangkaian kombinasi dan rangkaian sekuensial. ATPG dapat diukur efektivitasnya dengan jumlah cacat yang terdeteksi dan jumlah pola yang dihasilkan. ATPG juga mempertimbangkan efisiensi ATPG yang dipengaruhi oleh model cacat yang dipertimbangkan, jenis sirkuit yang diuji, dan tingkat abstraksi yang digunakan untuk menghasilkan pola tes. ATPG dapat dilakukan dengan tiga cara yaitu metode manual, metode algoritmik (dengan atau tanpa heuristik), dan metode pseudo-random.

Salah satu metode yang digunakan dalam membangkitkan pola uji pada rangkaian kali ini yaitu dengan menggunakan metode algoritma exhaustive, yang mana metode ini merupakan sebuah teknik pencarian solusi secara brute force untuk masalah yang melibatkan pencarian elemen dengan sifat khusus. Metode ini bekerja dengan cara mendeklarasikan semua input yang ada pada rangkaian, setiap  $n$ -input akan dihasilkan sebanyak  $2^n$  pola uji. Pada penerapannya metode ini digunakan untuk input rangkaian yang tidak terlalu besar dan cenderung digunakan pada rangkaian yang sederhana maka dari itu penulis menggunakan metode pengujian ini dikarenakan rangkaian yang akan diujikan cukup sederhana, yaitu kurang dari 15 input dan metode ini juga dapat membangkitkan seluruh pola pengujian yang diharapkan dapat mempropagasikan seluruh nilai stuck-at-fault pada setiap node rangkaian dengan optimal.

## **2 DESKRIPSI PROYEK**

Pada Proyek Automatic Test Pattern Generation ini , terdapat 5 input dan 32 output dalam Gerbang Logika dan Menggunakan seluruh gerbang logika (AND, NAND, OR, NOR, XOR, XNOR, NOT) terdapat beberapa menu ini, yaitu ;

### **1. Multipleksers**

Pada menu ini , Multipleksers digunakan untuk menggabungkan sinyal-sinyal input menjadi satu sinyal output.

### **2. Demultipleksers**

Pada menu ini, Demultipleksers digunakan untuk memisahkan satu sinyal input menjadi beberapa sinyal output.

### **3. Rancangan Sistem**

Pada menu ini, Sistem yang akan dibuat dalam tugas besar ini yaitu sistem pengujian yang dapat menemukan test vector minimum dan test fault coverage maksimum pada rangkaian combinational.

### **4. Algoritma Sistem**

Pada menu ini, Algoritma dari sistem melakukan pengujian stuck-at-fault dalam tugas besar gerbang logika.

### 3 IMPLEMENTASI

Dalam program Automatic Test Pattern Generation yang kami buat memiliki fitur, yaitu:

#### 1. Gerbang Logika Dasar

Digunakan untuk mengolah input bilangan biner menjadi output bilangan biner

- OR: Menghasilkan output 1 jika salah satu input adalah 1, dan 0 jika semua input adalah 0.
- AND: Menghasilkan output 1 jika semua input adalah 1, dan 0 jika salah satu input adalah 0.

NOT: Memegalikan input

NAND: Menghasilkan output 0 jika semua input adalah 1, dan 1 jika salah satu input atau lebih adalah 0.

NOR: Menghasilkan output 1 jika semua input adalah 0, dan 0 jika salah satu input atau lebih adalah 1.

XOR: Menghasilkan output 1 jika salah satu input adalah 1, tetapi bukan keduanya, dan 0 jika keduanya sama.

XNOR: Menghasilkan output 0 jika salah satu input adalah 1, tetapi bukan keduanya, dan 1 jika keduanya sama.

```

0,0,0,0,0
0,0,0,0,1
0,0,0,1,0
0,0,0,1,1
0,0,1,0,0
0,0,1,0,1
0,0,1,1,0
0,0,1,1,1
0,1,0,0,0
0,1,0,0,1
0,1,0,1,0
0,1,0,1,1
0,1,1,0,0
0,1,1,0,1
0,1,1,1,0
0,1,1,1,1
1,0,0,0,0
1,0,0,0,1
1,0,0,1,0
1,0,0,1,1
1,0,1,0,0
1,0,1,0,1
1,0,1,1,0
1,0,1,1,1
1,1,0,0,0
1,1,0,0,1
1,1,0,1,0
1,1,0,1,1
1,1,1,0,0
1,1,1,0,1
1,1,1,1,0
1,1,1,1,1

```

### 5 Input.

- Kode tersebut dimulai dengan membuka dua file, input.txt dan output.txt. File input.txt berisi data input, yaitu nilai dari lima variabel input. File output.txt berisi hasil output dari program, yaitu nilai dari variabel logika a dan b.

A	B	C	D	E	A/0	A/1	B/0	B/1	C/0	C/1	D/0	D/1	E/0	E/1	a/0	a/1	b/0	b/1	c/0	c/1	d/0	d/1	e/0	e/1	f/0	f/1	g/0	g/1	h/0	h/1	i/0	i/1	op1	op2	
0	0	0	0	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	0
0	0	0	0	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	1
0	0	0	1	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	0
0	0	0	1	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	1
0	0	1	0	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	0
0	0	1	0	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	1
0	0	1	1	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	0
0	0	1	1	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	1
0	1	0	0	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	0
0	1	0	0	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	1
0	1	0	1	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	0
0	1	0	1	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	1
0	1	1	0	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	0
0	1	1	0	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	1
0	1	1	1	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	0
0	1	1	1	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	1
1	0	0	0	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	0
1	0	0	0	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	1
1	0	0	1	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	0
1	0	0	1	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	1
1	0	1	0	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	0
1	0	1	0	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	1
1	0	1	1	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	0
1	0	1	1	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	1
1	1	0	0	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	0
1	1	0	0	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	1
1	1	0	1	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	0
1	1	0	1	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	1
1	1	1	0	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	0
1	1	1	0	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	0	1
1	1	1	1	0	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	0
1	1	1	1	1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	1	1

### 32 Output.

#### 1. Header

| A | B | C | D | E | A/0 | A/1 | B/0 | B/1 | C/0 | C/1 | D/0 | D/1 | E/0 | E/1 | a/0 | a/1 | b/0 | b/1 | c/0 | c/1 | d/0 | d/1 | e/0 | e/1 | f/0 | f/1 | g/0 | g/1 | h/0 | h/1 | i/0 | i/1 | op1 | op2 |

- Kolom A, B, C, D, dan E: Mewakili nilai lima input integer.

- Kolom A/0, A/1, B/0, B/1, dst.: Menunjukkan status valid/invalid dari masing-masing input. **1. "V": Valid (input = 1)**

## 2. " ": Invalid (input = 0)

- Kolom a, aa, b, c, dst.: Mewakili nilai variabel logika internal yang digunakan untuk menghitung output.
- Kolom op1 dan op2: Mewakili dua output akhir yang dihitung berdasarkan logika yang ditentukan dalam kode.

## 2. Baris Output:

Setiap baris dalam file output sesuai dengan satu baris data input. Baris tersebut berisi informasi berikut:

- | %2d | %2d | %2d | %2d | %2d | : Nilai lima input integer.
- \*\* %c | %c | %c | %c | %c | \*\* : Status valid/invalid dari lima input integer, dipisahkan oleh spasi.
- \*\* %c | %c | %c | %c | %c | %c | %c | %c | %c | %c | \*\* : Nilai variabel logika internal.
- \*\* %c | %c | %d | %d | \*\* : Nilai dua output akhir dan spasi.

## 3. Interpretasi Output:

- Kolom A-E menunjukkan nilai input, di mana "V" berarti valid dan spasi berarti invalid.
- Kolom a-h menunjukkan nilai variabel logika internal yang digunakan dalam perhitungan.
- Kolom op1 dan op2 menunjukkan dua output akhir yang dihitung berdasarkan logika yang ditentukan dalam kode.

## 4 PENUTUP

Dengan proyek kami telah berhasil membuat sebuah program Dictionary yang bertujuan untuk membuat bahwa ATPG merupakan metode desain elektronik yang digunakan untuk menemukan urutan input yang dapat digunakan untuk menguji perangkat elektronik, seperti perangkat ketimbangan digital dan semikonduktor, dengan efisien dan efektif. ATPG membantu dalam mengurangi waktu pengujian dan memastikan bahwa perangkat dibuat secara seimbang. Beberapa metode yang digunakan dalam ATPG meliputi metode manual, metode algoritmik (dengan atau tanpa heuristik), dan metode pseudo-random.

### **ATPG memiliki beberapa keuntungan, seperti:**

1. Mengurangi waktu pengujian: ATPG memungkinkan untuk cepat menemukan pola tes yang sesuai dengan kebutuhan perangkat, sehingga waktu pengujian dapat disebutkan lebih singkat.
2. Memastikan efisiensi dan efektivitas tes: ATPG membantu mencapai ketersediaan tes yang tinggi dengan memastikan bahwa semua kemungkinan kesalahan diidentifikasi dan diperbaiki.
3. Membantu dalam analisis kegagalan: ATPG dapat digunakan untuk mengidentifikasi penyebab kegagalan dan membantu dalam analisis kegagalan (analisis kegagalan).