Automatic Synthesis of Comparator Circuit Using Genetic Algorithm

Takayuki NEGISHI, Naoki ARAI, Nobukazu TAKAI, Masato KATO, Hiroaki SEKI, Sumit Kumar BISWAS and Haruo KOBAYASHI

Gunma University, Japan

16th November 2013
OUTLINE

- Background and Objective
- Method of Automatic Synthesis
- Result of Automatic Synthesis
- Summary and Future Work
OUTLINE

- Background and Objective
- Method of Automatic Synthesis
- Result of Automatic Synthesis
- Summary and Future Work
Research Background

「Engineer of Analog Circuit」

Industry Demands
- Short Time to Market
- Cost Reduction

Time required for designing circuit
- Selection: Proper Circuit Topology
- Determination: Circuit Parameter Values
- Accumulation: Knowledge and Experience

Take time to become a good circuit designer

DILEMMA
Research Background

「Automatic Design by Computer」

Industry Demands
- Short Time to Market
- Cost Reduction

Time required for designing circuit
- Programming Time
- Execution Time

It can be designed in a shorter time than engineer

Compatible
Research Objective

Comparator circuit is always used for signal comparison.

- Short Time to Market
- Cost Reduction

Merit of automatic synthesis!

Electrical Characteristics of Comparator

- ✓ Input-offset voltage
- ◯ Input-offset current
- ◯ Input bias current
- ✓ Current consumption
- ◯ CMRR
- ◯ PSRR
- ✓ Response time

We realize automatic synthesis focusing on 3 characteristics!
OUTLINE

Background and Objective

Method of Automatic Synthesis

Result of Automatic Synthesis

Summary and Future Work
Overview of Automatic Synthesis

### Step 1
Create circuit file.
HSPICE input file format
preparation for variable parameters.

### Step 2
Determines proper values
that satisfies the setting condition
using HSPICE optimizing function.

#### Step 3

- **Performance reach the demand (Yes)**
  Determines performance values.

- **Performance don’t reach the demand (No)**
  Circuit topology is considered as “bad”.

  Calculate evaluation function
  as circuit performance figure.

※ Java-language programming
Overview of Automatic Synthesis

- **Step 1**
  - Create circuit file.
  - HSPICE input file format preparation for variable parameters.

- **Step 2**
  - Determines proper values that satisfies the setting condition using HSPICE optimizing function.
  - Application of Genetic Algorithm

- **Step 3**
  - < Performance reach the demand (Yes) >
    - Determines performance values.
  - < Performance don’t reach the demand (No) >
    - Circuit topology is considered as “bad”.
    - Calculate evaluation function as circuit performance figure.

- **Step 4**
  - Calculate Evaluation Function

- **Step 5**
  - Selection of Highest Performance Circuit

※ Java-language programming
Overview of Automatic Synthesis

Step 1
Create circuit file.
HSPICE input file format
preparation for variable parameters.

Step 2
Determines proper values
that satisfies the setting condition
using HSPICE optimizing function.

Step 3
< Performance reach the demand (Yes) >
Determines performance values.

< Performance don’t reach the demand (No) >
Circuit topology is considered as “bad”.

Calculate evaluation function
as circuit performance figure.
Overview of Automatic Synthesis

Step 1
Create Circuit File

Step 2
Optimize Circuit Variables

Step 3
Optimization Completed
- Yes
  - Calculate Performance Values
    - Response Time
    - Current Consumption
    - Input-offset Voltage
  - Calculate Evaluation Function
- No
  - Application of Genetic Algorithm

Step 4
- Increment loop number by 1.
  - < Loop number is less than the specified value >
    - Change another topology using GA.
    - Go to Step 1.
  - < Loop number is equal to the specified value >
    - Go to step 5.

Step 5
- Select the highest performance circuit among all of evaluated circuits.
Overview of Automatic Synthesis

Step 4
Increment loop number by 1.

< Loop number is less than the specified value >
Change another topology using GA.
Go to Step 1.

< Loop number is equal to the specified value >
Go to step 5.

Step 5
Select the highest performance circuit among all of evaluated circuits.
Create Circuit File to Use HSPICE

STEPS: Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance → Application of GA (Selection, Mutation) → Selection of The Highest Performance Circuit

Describe circuit information to run HSPICE

(Ex)

Comparator_001
.lib './contest.lib' model1

.OPTION INGOLD=2 NUMDGT=5 MEASDGT=5

M01 n03 n02 vdd vdd cmosp l=2.41u w=30.8u
M02 n04 inp out vss cmosn l=49.8u w=48.4u
M03 out n03 n04 vss cmosn l=31.2u w=49.3u
M04 n01 vss vdd vdd cmosp l=7.00u w=26.6u
M05 out n02 n01 vdd cmosp l=0.25u w=2.34u
M06 out inm vss vss cmosn l=7.30u w=18.0u
M07 n02 inp vdd vdd cmosp l=14.2u w=3.52u
C08 out n03 1p
R09 n02 vss 100

Vdd vdd gnd dc 1.5
Vss gnd vss dc 1.5
.lib 'Lib_Delay.lib' Delay
.end

Circuit Topology & Circuit Parameter Values

Reading a MOS model file

Reading an analysis method file
How to Optimize

STEPS: Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance → Application of GA (Selection, Mutation) → Selection of The Highest Performance Circuit

Optimization uses response time

All response time parameters to 0.0 second
\[ t_a, t_b, t_c, t_d \]

Output voltage difference (\( V_{o\_dif} \)) to 70% of “\( V_{DD} - V_{SS} \)”

Goal

Target

MOSFET: Gate length and width
Judging Optimization

STEPS: Create Circuit File → **Optimize Circuit Variables** → Calculate Circuit Performance → Application of GA (Selection, Mutation) → Selection of The Highest Performance Circuit

\[ V_{o\_dif} = V_{o\_max} - V_{o\_min} \]

**Judgment Conditions**

\[ (V_{DD} - V_{SS}) \times 50\% \leq V_{o\_dif} \leq (V_{DD} - V_{SS}) \]

\[ V_{o\_max} \leq V_{DD} \]

\[ V_{SS} \leq V_{o\_min} \]

Only circuit that satisfied all conditions → Calculate performance values
Calculate Performance Value

**Steps:**
1. Create Circuit File
2. Optimize Circuit Variables
3. Calculate Circuit Performance
4. Application of GA (Selection, Mutation)
5. Selection of The Highest Performance Circuit

**Input Pulse** ($V_{in}$): 200mV/μs [-0.1V~0.1V]

**Calculated Items**

- **Output Voltage Difference** $: V_{o\_diff}$$
- **Propagation Delay Time** $: (t_a + t_b)/2$
- **Output Voltage Transition Time** $: (t_c + t_d)/2$
- **Current Consumption**
  Effective current value in 1 cycle of input pulse

**Calculated Item**

- **Input-offset voltage** $: V_{out}$

Calculate Evaluation Function

**Steps:**
1. Create Circuit File
2. Optimize Circuit Variables
3. Calculate Circuit Performance
4. Application of GA (Selection, Mutation)
5. Selection of The Highest Performance Circuit

**Represent a Number for Circuit Performance**

**Performance Values:** $s_k$
**Target Values:** $t_k$

- Output voltage difference: $s_1, t_1$
- Propagation delay time: $s_2, t_2$
- Output voltage transition time: $s_3, t_3$
- Current consumption: $s_4, t_4$
- Input-offset voltage: $s_5, t_5$

**Evaluation Function**

**Type 1**

$$ E_1 = \begin{cases} 
\frac{s_1}{t_1} & (s_1 \leq t_1) \\
1 + \log \left( \frac{s_1}{t_1} \right) & (s_1 > t_1)
\end{cases} $$

**Type 2**

$$ E_{2,\ldots,5} = \begin{cases} 
\frac{t_2,\ldots,5}{s_2,\ldots,5} & (s_2,\ldots,5 > t_2,\ldots,5) \\
1 + \log \left( \frac{t_2,\ldots,5}{s_2,\ldots,5} \right) & (s_2,\ldots,5 \leq t_2,\ldots,5)
\end{cases} $$

**Fitness Value:** $F$

$$ F = \prod_{k=1}^{5} E_k $$

**Evaluation Function**
Genetic Algorithm (GA)\cite{6}

STEps : Create Circuit File $\rightarrow$ Optimize Circuit Variables $\rightarrow$ Calculate Circuit Performance $\rightarrow$ Application of GA (Selection, Mutation) $\rightarrow$ Selection of The Highest Performance Circuit

Genetic Algorithm

Construction based on the laws of heredity in the real world.

Change a comparator circuit topology.

「Selection」 $\rightarrow$ Choose preferentially higher $F$ (circuit performance).

「Mutation」 $\rightarrow$ Change circuit topology.

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population($N_p$)</td>
<td>30</td>
</tr>
<tr>
<td>Generation</td>
<td>200</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0%</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>30%</td>
</tr>
</tbody>
</table>

Population($N_p$) :
Circuit topology number generated by the same loop

Generation :
Loop number

< GA conditions for automatic synthesis >

Selection

STEPs: Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance
→ Application of GA (Selection, Mutation) → Selection of The Highest Performance Circuit

Superior gene survives, Inferior gene weeds out.

Fitness Proportion Selection

Choose preferentially higher $F$ (circuit performance).

「Expectation (topology $k$ is selected as next-generation)」

$$\frac{F_k}{\sum_k F_k / N_p} = \frac{F_k}{\bar{F}} \quad k: 1,\ldots, N_p$$

$N_p$: Population (Circuit topology)

$\bar{F}$: Mean of fitness value

(Ex) Fitness@N-th generation

$F_1=1, F_2=3, F_3=5, F_4=7, F_5=9$

$\bar{F} = 5\left(=\frac{1+3+5+7+9}{5}\right)$

Expectation selected@(N+1)-th generation

$F_1=1/5, F_2=3/5, F_3=1, F_4=7/5, F_5=9/5$

High performance ($F_4, F_5$) is easy to survive.
Consider 3 type mutations to change circuit topology.

- **Split into two nodes (A, A').**
- **Create a new node (D, E).**
- **Node-F connect to another node (G).**

Change topology if such a connection exists.

※ Take care of not producing floating node.
Mutation - Case 1

STEPs: Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance → Application of GA (Selection, Mutated) → Selection of The Highest Performance Circuit

Before

There is an H-type node

Node-A is H-type

Split into two nodes (A, A')

After
Mutation - Case 2

Steps: Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance → Application of GA (Selection, Mutation) → Selection of The Highest Performance Circuit

Before

There are two H-type nodes

After

Create new nodes (D, E).
**Mutation - Case 3**

**Before**

If current path split into two

**After**

Node-F connected to another node (G).

**Steps:**

1. Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance
2. Application of GA (Selection, **Mutation**) → Selection of the Highest Performance Circuit
Initial Circuit

Required initial circuit in this study.

Evolve into high-performance topology using GA.

basic comparator circuit.

Supply Voltage

\[ V_{DD} = 1.5V \]
\[ V_{SS} = -1.5V \]

Bulk

PMOS : \( V_{DD} \)
NMOS : \( V_{SS} \)

Technology

TSMC 180nm CMOS Process

< Initial Circuit >
Transition of the Maximum $F$

Fitness upgrades: Until 25 generations

PC Specification:
- Clock frequency: 1.6GHz
- Memory: 8GB
- Number of Cores: 4

Runtime: 20 hours
Best Performance Circuit

**Supply Voltage**
- $V_{DD} = 1.5V$
- $V_{SS} = -1.5V$

**Technology**
- TSMC 180nm CMOS Process

**Bulk**
- PMOS: $V_{DD}$
- NMOS: $V_{SS}$

**Steps**
- Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance → Application of GA (Selection, Mutation) → Selection of The Highest Performance Circuit
**Input & Output Waveforms**

**STEPS:**
1. Create Circuit File
2. Optimize Circuit Variables
3. Calculate Circuit Performance
4. Application of GA (Selection, Mutation)
5. Selection of The Highest Performance Circuit

**Enlarged view of rising part**
Input & Output Waveforms

STEPS: Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance → Application of GA (Selection, Mutation) → Selection of The Highest Performance Circuit

Enlarged view of falling part
# Performance Comparison

<table>
<thead>
<tr>
<th>Evaluation Item</th>
<th>Value Type</th>
<th>Initial Circuit</th>
<th>Final Circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output voltage difference</td>
<td>Performance</td>
<td>2.29V</td>
<td>2.72V</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>1.06</td>
<td>1.13</td>
</tr>
<tr>
<td>Propagation delay time</td>
<td>Performance</td>
<td>168ns</td>
<td>73.7ns</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>1.0</td>
<td>1.36</td>
</tr>
<tr>
<td>Output voltage transition time</td>
<td>Performance</td>
<td>144ns</td>
<td>272ns</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>1.0</td>
<td>0.72</td>
</tr>
<tr>
<td>Consumption current</td>
<td>Performance</td>
<td>5.65mA</td>
<td>1.48mA</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>1.0</td>
<td>1.58</td>
</tr>
<tr>
<td>Input-offset voltage</td>
<td>Performance</td>
<td>59.6mV</td>
<td>3.0mV</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>1.0</td>
<td>2.3</td>
</tr>
<tr>
<td>Fitness value (product of all evaluation)</td>
<td></td>
<td><strong>1.06</strong></td>
<td><strong>4.06</strong></td>
</tr>
</tbody>
</table>
Summary & Future Work

- Develop automatic synthesis program.

- HSPICE optimizing function determines circuit parameters.

- GA determines the circuit topology.

- Synthesized circuit performance is better than initial circuit.

- Realize automatic synthesis of comparator from its faulty initial circuit.

- Silicon prototype of synthesized circuit.
Appendix

Change Reference Voltage

Change Input Pulse

MOSFET’s Operating State
Change Reference Voltage

\[ V_{\text{ref}} = 1.4V \]

Input Pulse \((V_{\text{in}})\)

\[ 200\text{mV/\mu s} \ [1.3V \sim 1.5V] \]
Change Reference Voltage

\[ V_{\text{ref}} = -0.6 \text{V} \]

Input Pulse\( (V_{\text{in}}) \)

\[ 200 \text{mV/\mu s [\(-0.7\text{V} \sim -0.5\text{V}\)]} \]

Operating Range : \(-0.6\text{V} \leq V_{\text{ref}} \leq 1.4\text{V} \)
Change Input Pulse

Input Pulse ($V_{in}$)
200mV/μs [$-0.1V \sim 0.1V$]
20mV/μs [$-0.01V \sim 0.01V$]

Operation Check : Overdrive 10mV
MOSFET’s Operating State

<table>
<thead>
<tr>
<th>MOS</th>
<th>Pulse Low</th>
<th>Pulse High</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Linear</td>
<td>Linear</td>
</tr>
<tr>
<td>M2</td>
<td>Linear</td>
<td>Saturation</td>
</tr>
<tr>
<td>M3</td>
<td>Saturation</td>
<td>Saturation</td>
</tr>
<tr>
<td>M4</td>
<td>Saturation</td>
<td>Saturation</td>
</tr>
<tr>
<td>M5</td>
<td>Saturation</td>
<td>Linear</td>
</tr>
<tr>
<td>M6</td>
<td>Saturation</td>
<td>Saturation</td>
</tr>
<tr>
<td>M7</td>
<td>Linear</td>
<td>Linear</td>
</tr>
<tr>
<td>M8</td>
<td>Cutoff</td>
<td>Linear</td>
</tr>
<tr>
<td>M9</td>
<td>Linear</td>
<td>Saturation</td>
</tr>
</tbody>
</table>

Steps: Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance → Application of GA (Selection, Mutation) → Selection of The Highest Performance Circuit
Faulty initial circuit
Mutation - Case 4

STEPs: Create Circuit File → Optimize Circuit Variables → Calculate Circuit Performance → Application of GA (Selection, Mutation) → Selection of The Highest Performance Circuit

Change probability is 50%

PMOS  \[\xrightarrow{\text{Mutation}}\] NMOS