



سیزدهمین کنفرانس مهندسی برق و الکترونیک  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران



# طراحی و بهینه سازی مدارات منطقی ترکیبی با استفاده از الگوریتم ژنتیک

سیدمحسن موسوی و دانیال خشابی  
دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر  
{ moosavi.sm,d.khashabi } @gmail.com

## سرفصل ها

- ▶ طراحی مدارات منطقی ترکیبی
  - روش های تکاملی طراحی مدارات منطقی
- ▶ بهینه سازی تکاملی
- ▶ بحثی بر معیارهای طراحی
- ▶ روش اول معرفی شده
- ▶ روش دوم معرفی شده
- ▶ نتایج
- ▶ جمع بندی



سیزدهمین کنفرانس دانشجویی مهندسی برق  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران

## طراحی مدارات منطقی ترکیبی



- [1] Arturo Hernández Aguirre et al, "Using Genetic Programming and Multiplexers for the Synthesis of Logic Circuits", *Engineering Optimization*, Vol. 36, No. 4, pp. 491--511, August 2004.
- [2] Sushil J. Louis, Gregory J.E. Rawlins: "Designer Genetic Algorithms: Genetic algorithms in StructureDesign", *Procs of the Fourth International Conference on Genetic Algorithm*, pages 53-60, 1991
- [3] J. R. Koza, "Genetic Programming; On the Programming of Computers by Means of Natural Selection", MIT Press, 1992.
- [4] C.A.C. Coello et al, "Toward Automated Evolutionary Design of Combinational Circuits", Department of Computer Science, Tulane University, New Orleans, USA, 1999.

## طراحی مدارات منطقی ترکیبی

- ▶ طراحی از مدارات منطقی ترکیبی ◀ با داشتن جدول درستی
- ▶ روش های متداول (فقط دو طبقه):
  - جدول کارنو
  - الگوریتم Quine-McClusky

[1] Arturo Hernández Aguirre et al, "Using Genetic Programming and Multiplexers for the Synthesis of Logic Circuits", *Engineering Optimization*, Vol. 36, No. 4, pp. 491--511, August 2004.

[2] Sushil J. Louis, Gregory J.E. Rawlins: "Designer Genetic Algorithms: Genetic algorithms in StructureDesign", *Procs of the Fourth International Conference on Genetic Algorithm*, pages 53-60, 1991

[3] J. R. Koza, "Genetic Programming; On the Programming of Computers by Means of Natural Selection", MIT Press, 1992.

[4] C.A.C. Coello et al, "Toward Automated Evolutionary Design of Combinational Circuits", Department of Computer Science, Tulane University, New Orleans, USA, 1999.

## طراحی مدارات منطقی ترکیبی

- ▶ طراحی از مدارات منطقی ترکیبی ◀ با داشتن جدول درستی
- ▶ روش های متداول (فقط دو طبقه):
  - جدول کارنو
  - الگوریتم Quine-McClusky
- ▶ طراحی با استفاده از روش های تکاملی ◀ سخت افزار تکاملی<sup>5</sup> (EHW)
- ▶ تاریخچه:
  - اولین سعی برای استفاده از الگوریتم های تکاملی برای بهینه سازی مدارها ◀ توسط Fridman [1]
  - اولین استفاده از الگوریتم ژنتیک برای طراحی مدارات منطقی ◀ توسط S. J. Louis [2]

[1] Arturo Hernández Aguirre et al, "Using Genetic Programming and Multiplexers for the Synthesis of Logic Circuits", *Engineering Optimization*, Vol. 36, No. 4, pp. 491--511, August 2004.

[2] Sushil J. Louis, Gregory J.E. Rawlins: "Designer Genetic Algorithms: Genetic algorithms in StructureDesign", *Procs of the Fourth International Conference on Genetic Algorithm*, pages 53-60, 1991

[3] J. R. Koza, "Genetic Programming; On the Programming of Computers by Means of Natural Selection", MIT Press, 1992.

[4] C.A.C. Coello et al, "Toward Automated Evolutionary Design of Combinational Circuits", Department of Computer Science, Tulane University, New Orleans, USA, 1999.

## طراحی مدارات منطقی ترکیبی

- ▶ طراحی از مدارات منطقی ترکیبی ◀ با داشتن جدول درستی
- ▶ روش های متداول (فقط دو طبقه):
  - جدول کارنو
  - الگوریتم Quine-McClusky
- ▶ طراحی با استفاده از روش های تکاملی ◀ سخت افزار تکاملی<sup>5</sup> (EHW)
- ▶ تاریخچه:
  - اولین سعی برای استفاده از الگوریتم های تکاملی برای بهینه سازی مدارها ◀ توسط Fridman [1]
  - اولین استفاده از الگوریتم ژنتیک برای طراحی مدارات منطقی ◀ توسط S. J. Louis [2]
  - برنامه ریزی ژنتیک ◀ Koza با تاکید بر بدست آوردن جواب و طراحی [3]
  - پیاده سازی بهینه سازی ژنتیک بصورت باینری (BGA) و اعداد صحیح (N-cardinal یا NGA): Carlos Coello [4]
  - پیاده سازی چند هدفه (MGA) توسط Carlos Coello [4]
  - کار های دیگر برای بهینه سازی تکاملی روی مدارات منطقی
    - ACO ، PSO ، GA+SA

[1] Arturo Hernández Aguirre et al, "Using Genetic Programming and Multiplexers for the Synthesis of Logic Circuits", *Engineering Optimization*, Vol. 36, No. 4, pp. 491--511, August 2004.

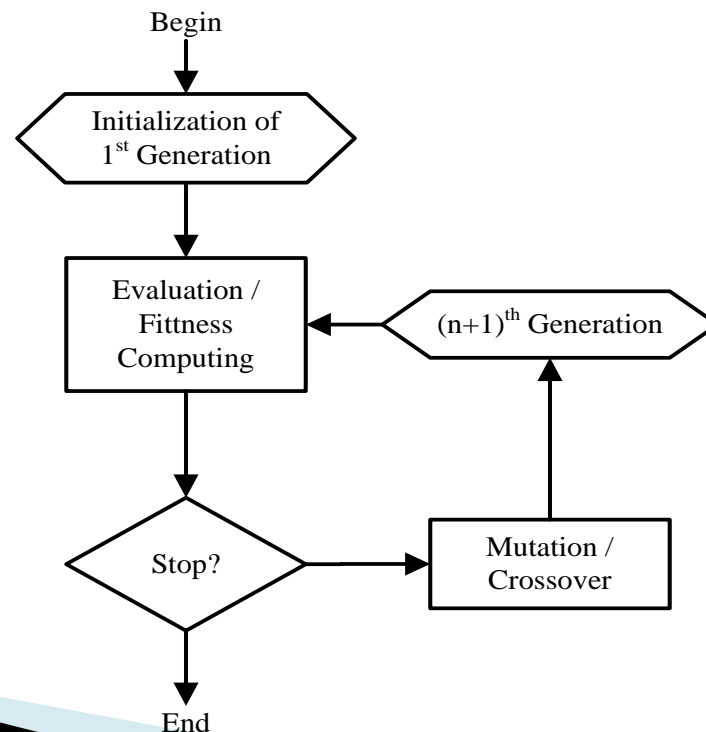
[2] Sushil J. Louis, Gregory J.E. Rawlins: "Designer Genetic Algorithms: Genetic algorithms in StructureDesign", *Procs of the Fourth International Conference on Genetic Algorithm*, pages 53-60, 1991

[3] J. R. Koza, "Genetic Programming; On the Programming of Computers by Means of Natural Selection", MIT Press, 1992.

[4] C.A.C. Coello et al, "Toward Automated Evolutionary Design of Combinational Circuits", Department of Computer Science, Tulane University, New Orleans, USA, 1999.

## بهینه سازی تکاملی

- ▶ الگوریتم های ژنتیک ◀ الگوریتم تکاملی ◀ با الهام از الگوی طبیعی گذار نسل ها و نظریه ی انتخاب طبیعی داروین
- ▶ جمعیتی از جواب ها ◀ بهبود در هر نسل
- ▶ تابع برازندگی<sup>7</sup>، میزان انطباق کروموزم (میزان مناسب بودن جواب متناظر) را در مقایسه با مقادیر خواسته شده نمایش می دهد.





سیزدهمین کنفرانس دانشجویی مهندسی برق  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران



# بحثی بر معیارهای طراحی





سیزدهمین کنفرانس مهندسی برق  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران



# بحثی بر معیارهای طراحی

معیارها: ►

◦ تعداد گیت های بکاررفته ◄ گیت کمتر، زندگی بهتر!



سیزدهمین کنفرانس دانشجویی  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران



## بحثی بر معیارهای طراحی

► معیارها:

- تعداد گیت های بکاررفته ◀ گیت کمتر، زندگی بهتر!
- نوع ترانزیستورهای بکاررفته ◀ ساختاربرخی گیت ها ساده تر است!



سیزدهمین کنفرانس دانشجویی  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران



## بحثی بر معیارهای طراحی

► معیارها:

- تعداد گیت های بکاررفته ◀ گیت کمتر، زندگی بهتر!
- نوع ترانزیستورهای بکاررفته ◀ ساختاربرخی گیت ها ساده تر است!
- تعداد سطوح طراحی مدار ◀ سطوح بیشتر موجب تاخیر است!



سیزدهمین کنفرانس مهندسی برق  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران



## بحثی بر معیارهای طراحی

► معیارها:

- تعداد گیت های بکاررفته ◀ گیت کمتر، زندگی بهتر!
- نوع ترانزیستورهای بکاررفته ◀ ساختاربرخی گیت ها ساده تر است!
- تعداد سطوح طراحی مدار ◀ سطوح بیشتر موجب تاخیر است!
- پیچیدگی الگوریتم ◀ الگوریتم با پیچیدگی کمتر مطلوب است.



## بحثی بر معیارهای طراحی

### معیارها:

- تعداد گیت های بکاررفته ◀ گیت کمتر، زندگی بهتر!
- نوع ترانزیستورهای بکاررفته ◀ ساختاربرخی گیت ها ساده تر است!
- تعداد سطوح طراحی مدار ◀ سطوح بیشتر موجب تاخیر است!
- پیچیدگی الگوریتم ◀ الگوریتم با پیچیدگی کمتر مطلوب است.
- ▶ اجزای متغیر در افراد ◀ گیت ها، ورودی گیت ها
- ▶ باید مطمئن بود ساختار مورد نظری تواند تمامی حالات را شامل شود ◀ کامل بودن
  - 1- {AND, OR, NOT}
  - 2- {AND, NOT}
  - 3- {OR, NOT}
  - 4- {NAND}
  - 5- {NOR}



## بحثی بر معیارهای طراحی

### ► معیارها:

- تعداد گیت های بکاررفته ◀ گیت کمتر، زندگی بهتر!
- نوع ترانزیستورهای بکاررفته ◀ ساختاربرخی گیت ها ساده تر است!
- تعداد سطوح طراحی مدار ◀ سطوح بیشتر موجب تاخیر است!
- پیچیدگی الگوریتم ◀ الگوریتم با پیچیدگی کمتر مطلوب است.
- اجزای متغیر در افراد ◀ گیت ها، ورودی گیت ها
- باید مطمئن بود ساختار مورد نظری تواند تمامی حالات را شامل شود ◀ کامل بودن
  - 1- {AND, OR, NOT}
  - 2- {AND, NOT}
  - 3- {OR, NOT}
  - 4- {NAND}
  - 5- {NOR}
- افزایش مجموعه گیت ها :



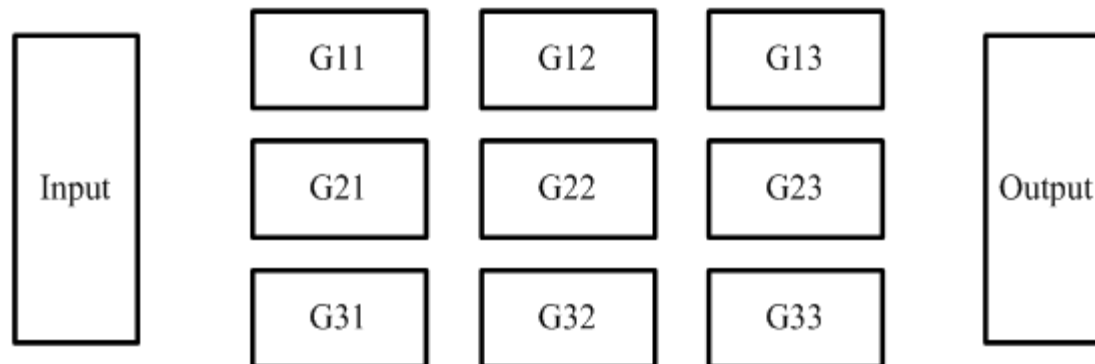
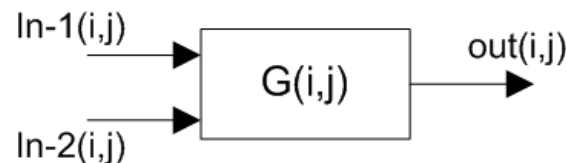
## بحثی بر معیارهای طراحی

### معیارها:

- تعداد گیت های بکاررفته ◀ گیت کمتر، زندگی بهتر!
- نوع ترانزیستورهای بکاررفته ◀ ساختاربرخی گیت ها ساده تر است!
- تعداد سطوح طراحی مدار ◀ سطوح بیشتر موجب تاخیر است!
- پیچیدگی الگوریتم ◀ الگوریتم با پیچیدگی کمتر مطلوب است.
- ▶ اجزای متغیر در افراد ◀ گیت ها، ورودی گیت ها
- ▶ باید مطمئن بود ساختار مورد نظری تواند تمامی حالات را شامل شود ◀ کامل بودن
  - 1- {AND, OR, NOT}
  - 2- {AND, NOT}
  - 3- {OR, NOT}
  - 4- {NAND}
  - 5- {NOR}
- ▶ افزایش مجموعه گیت ها :
  - موجب افزایش فضای جستجو
  - امکان کوچک تر شدن مدارات

## کدینگ اول

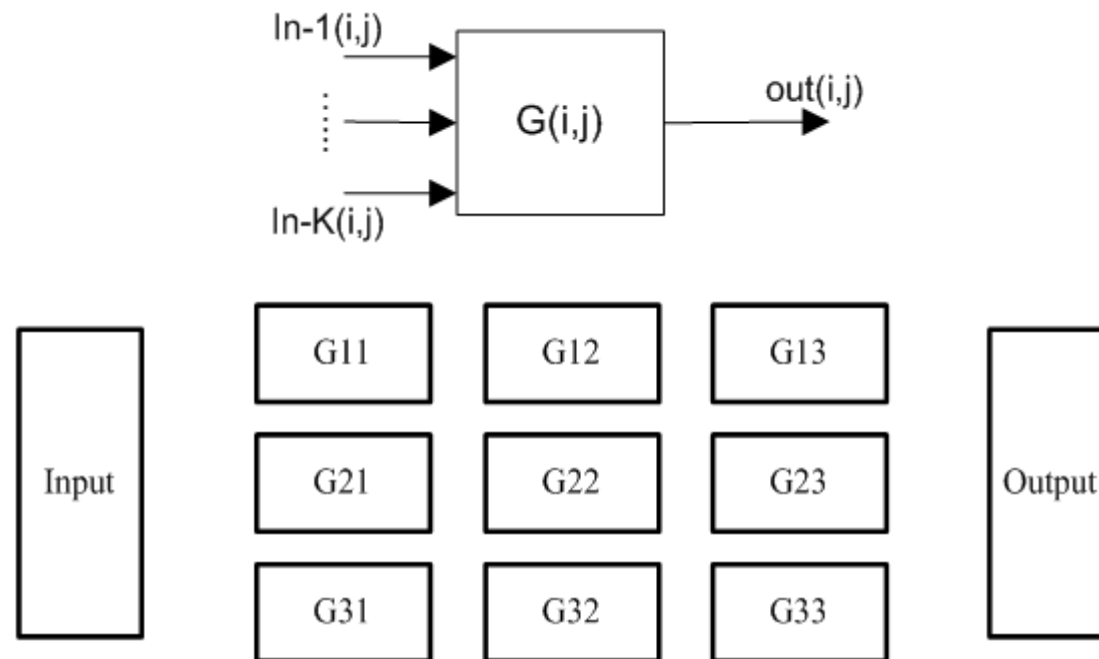
- ▶ هدف: طراحی مدار با گیت های دو ورودی
- ▶ مبدا ورودی ها : تمامی طبقات قبلی از جمله ورودی
- ▶ مجموعه گیت ها: XOR ، NOR ، NAND ، OR ، AND ، NULL





## کدینگ دوم

- ▶ هدف: طراحی مدار منطقی با گیت های چندین ورودی
- ▶ ورودی گیت ها: تمامی طبقات قبل از جمله ورودی
- ▶ مجموعه گیت ها: XOR و NOR ، NAND ، OR ، AND ، NULL





سیزدهمین کنفرانس دانشجویی  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران

## سایر مشخصات



دانشگاه تربیت مدرس  
دانشکده مهندسی برق و کامپیوتر



## سایر مشخصات

$$f = \frac{w_{match} \times N_{match} + w_{null} \times N_{null}}{w_{match} + w_{null}}$$

▶ تابع برازندگی:

- N-Null: تعداد گیت های Null
- N-Match: تعداد خروجی های منطبق بر خروجی مطلوب
- نتایج مناسب به ازای  $\frac{w_{match}}{w_{null}} \cong 10$



## سایر مشخصات

$$f = \frac{w_{match} \times N_{match} + w_{null} \times N_{null}}{w_{match} + w_{null}}$$

▶ تابع برازندگی:

◦ N-Null: تعداد گیت های Null

◦ N-Match: تعداد خروجی های منطبق بر خروجی مطلوب

◦ نتایج مناسب به ازای  $\frac{w_{match}}{w_{null}} \cong 10$

▶ جمعیت اولیه: تصادفی: افزایش پراکندگی جواب ها



## سایر مشخصات

$$f = \frac{w_{match} \times N_{match} + w_{null} \times N_{null}}{w_{match} + w_{null}}$$

▶ تابع برازندگی:

◦ N-Null: تعداد گیت های Null

◦ N-Match: تعداد خروجی های منطبق بر خروجی مطلوب

◦ نتایج مناسب به ازای  $\frac{w_{match}}{w_{null}} \cong 10$

▶ جمعیت اولیه: تصادفی: افزایش پراکندگی جواب ها

▶ عملگر ژنتیکی ترکیب: روش یک نقطه ای: با انتخاب اندیس تصادفی [ به عنوان سطح،

مدارات دو سمت اندیس را دو به دو به هم متصل کرده و مدار جدیدی را بدست می دهد.



## سایر مشخصات

تابع برازندگی: 
$$f = \frac{w_{match} \times N_{match} + w_{null} \times N_{null}}{w_{match} + w_{null}}$$

◦ N-Null: تعداد گیت های Null

◦ N-Match: تعداد خروجی های منطبق بر خروجی مطلوب

◦ نتایج مناسب به ازای  $\frac{w_{match}}{w_{null}} \cong 10$

▶ جمعیت اولیه: تصادفی: افزایش پراکندگی جواب ها

▶ عملگر ژنتیکی ترکیب: روش یک نقطه ای: با انتخاب اندیس تصادفی  $\lfloor$  به عنوان سطح،

مدارات دو سمت اندیس را دو به دو به هم متصل کرده و مدار جدیدی را بدست می دهد.

▶ عملگر ژنتیکی جهش: خانه ای به طور تصادفی انتخاب کرده و مقادیر آن را به طور تصادفی عوض می کند.



## سایر مشخصات

تابع برازندگی: 
$$f = \frac{w_{match} \times N_{match} + w_{null} \times N_{null}}{w_{match} + w_{null}}$$

◦ N-Null: تعداد گیت های Null

◦ N-Match: تعداد خروجی های منطبق بر خروجی مطلوب

◦ نتایج مناسب به ازای  $\frac{w_{match}}{w_{null}} \cong 10$

▶ جمعیت اولیه: تصادفی: افزایش پراکندگی جواب ها

▶ عملگرژنتیکی ترکیب: روش یک نقطه ای: با انتخاب اندیس تصادفی  $\lfloor$  به عنوان سطح،

مدارات دو سمت اندیس را دو به دو به هم متصل کرده و مدار جدیدی را بدست می دهد.

▶ عملگرژنتیکی جهش: خانه ای به طور تصادفی انتخاب کرده و مقادیر آن را به طور تصادفی عوض می کند.

▶ تضمین همگرایی برای هر تابعی: با استفاده از شکستن به دو قسمت



## سایر مشخصات

تابع برازندگی: 
$$f = \frac{w_{match} \times N_{match} + w_{null} \times N_{null}}{w_{match} + w_{null}}$$

◦ N-Null: تعداد گیت های Null

◦ N-Match: تعداد خروجی های منطبق بر خروجی مطلوب

◦ نتایج مناسب به ازای  $\frac{w_{match}}{w_{null}} \cong 10$

▶ جمعیت اولیه: تصادفی: افزایش پراکندگی جواب ها

▶ عملگر ژنتیکی ترکیب: روش یک نقطه ای: با انتخاب اندیس تصادفی  $\lfloor$  به عنوان سطح،

مدارات دو سمت اندیس را دو به دو به هم متصل کرده و مدار جدیدی را بدست می دهد.

▶ عملگر ژنتیکی جهش: خانه ای به طور تصادفی انتخاب کرده و مقادیر آن را به طور تصادفی عوض می کند.

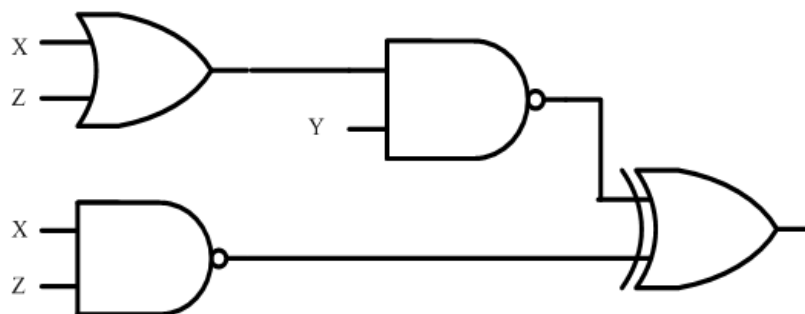
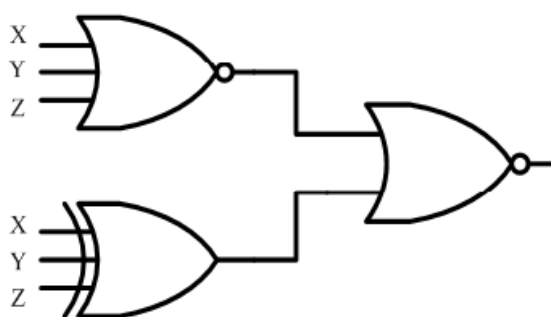
▶ تضمین همگرایی برای هر تابعی: با استفاده از شکستن به دو قسمت

▶ روش انتخاب: چرخ رولت: موجب پراکندگی اعضای نسل بعدی



## نتایج

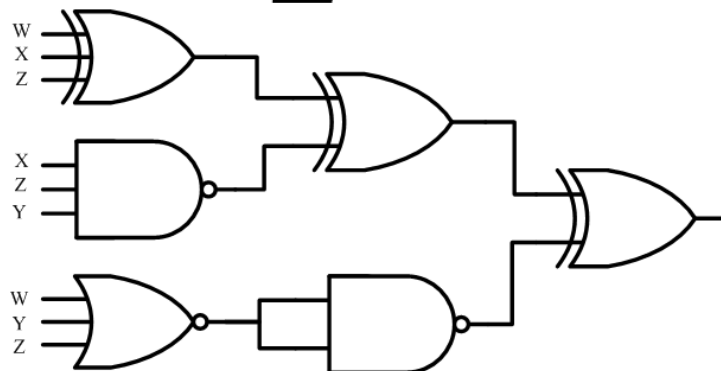
▶ مثال 1:  $f_1(X,Y,Z) = \sum (0,0,0,1,0,1,1,0)$



تعداد سطوح	تعداد گیت ها	طراح
۳	۴ گیت ( ۱ XOR ، ۱ OR ، ۲ NAND )	روش اول ارائه شده
۲	۳ گیت ( ۱ XOR (سه ورودی)، ۲ NOR (سه ورودی) )	روش اول ارائه شده
۳	۴ گیت ( ۱ XOR ، ۱ OR ، ۲ AND )	روش MGA
۴	۵ گیت ( ۱ AND ، ۱ OR ، ۲ XOR ، ۱ NOT )	روش NGA
۳	۵ گیت ( ۲ AND ، ۱ OR ، ۲ XOR )	طراحی انسانی

## نتایج

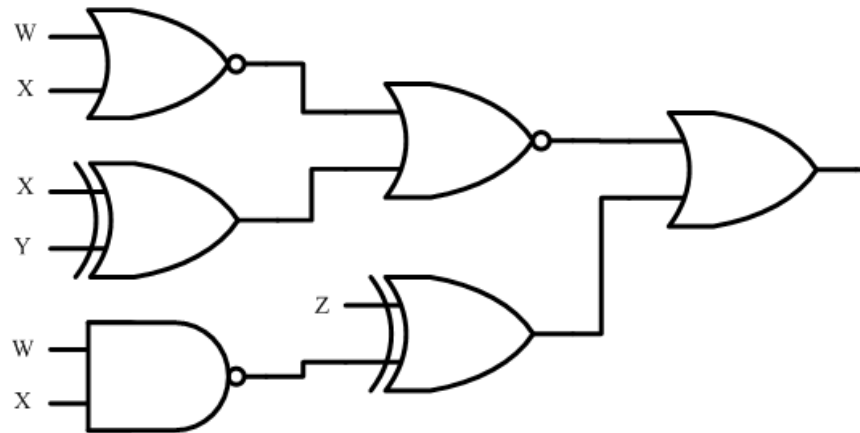
مثال ۲:  $f_2(W, X, Y, Z) = \sum (1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0)$



طراح	تعداد گیت ها	تعداد سطوح
روش دوم ارائه شده	۶ گیت ( ۳ XOR ، ۲ NAND ، ۱ NOR )	۳
روش MGA	۷ گیت ( ۱ AND ، ۲ OR ، ۳ XOR ، ۱ NOT )	۵
روش NGA	۱۰ گیت ( ۲ AND ، ۳ OR ، ۳ XOR ، ۲ NOT )	۶
روش BGA	۸ گیت ( ۱ AND ، ۳ OR ، ۳ XOR ، ۱ NOT )	۵
روش <u>Saso</u>	۱۲ گیت ( ۳ XOR ، ۲ AND ، ۴ NOT ، ۳ NOR )	۴
روش عسگریان	۶ گیت ( ۲ NAND ، ۲ NOR ، ۲ XOR )	۳
طراحی انسانی	۱۱ گیت ( ۴ AND ، ۱ OR ، ۲ XOR ، ۴ NOT )	۴

## نتایج

▶ مثال 3:  $f_3(W, X, Y, Z) = \sum (1,0,1,0,1,0,1,1,1,1,0,0,1,1,1)$

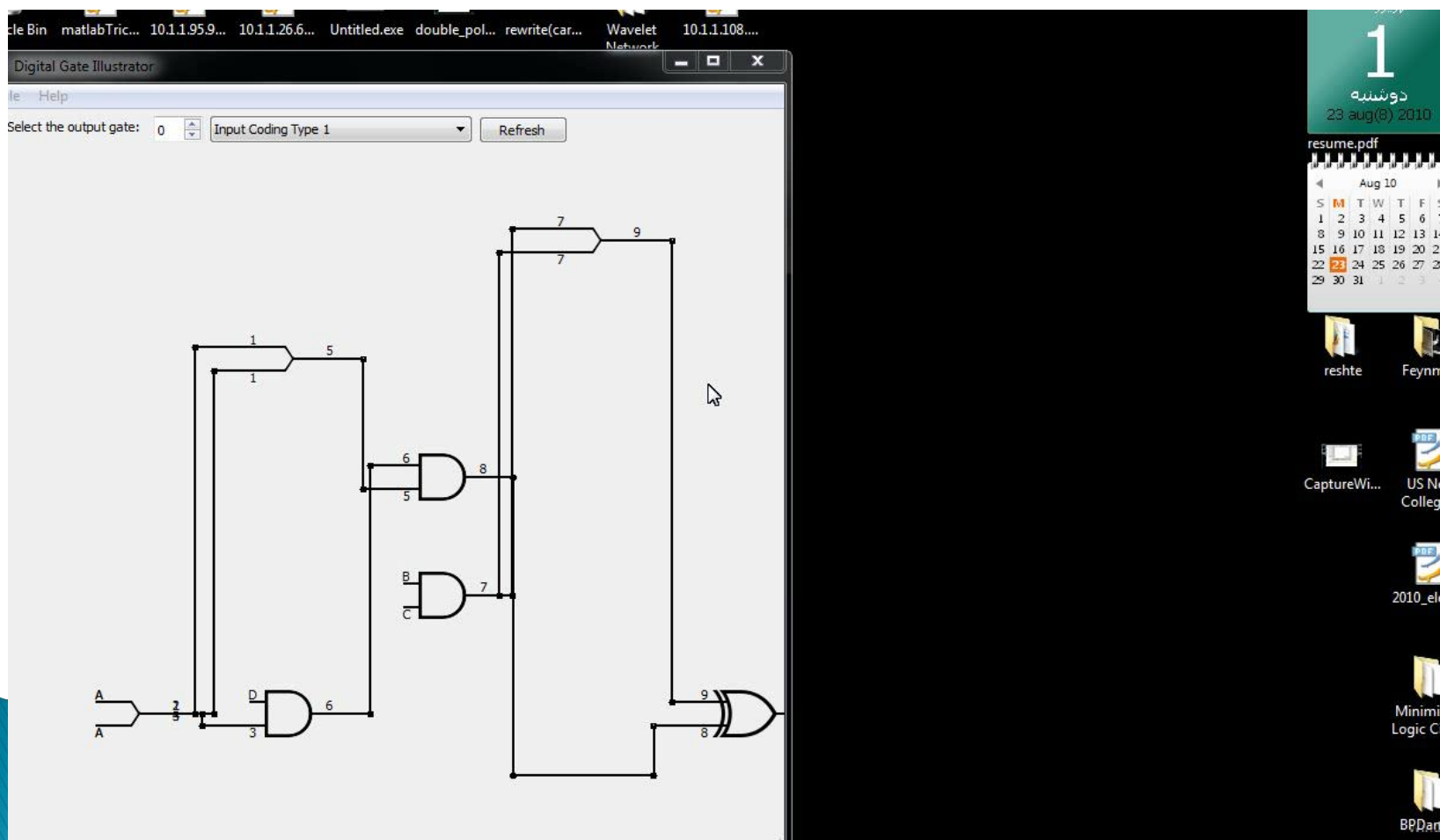


تعداد سطوح	تعداد گیت ها	طراح
۳	۶ گیت (OR ۱, NAND ۱, XOR ۲, NOR ۲)	روش اول ارائه شده
۴	۷ گیت (NOT ۱, AND ۲, OR ۲, XOR ۲)	روش NGA
۴	۸ گیت (NOT ۱, OR ۲, AND ۳, XOR ۲)	روش BGA

▶ بهینه سازی تحت MATLAB انجام می گیرند.

▶ نمایش دهنده ای با استفاده از کتابخانه ی Qt 4.6 طراحی شده است!

▶ کدهای بهینه سازی در این آدرس قابل دسترسی است: <http://ele.aut.ac.ir/~khashabi/wp/?p=82>



The screenshot displays the 'Digital Gate Illustrator' application window. The main workspace shows a logic circuit diagram with various components labeled with numbers (1-9). The circuit includes a large multiplexer at the top, several AND gates, and an OR gate at the bottom. The interface also features a menu bar (File, Help), a toolbar with 'Select the output gate' and 'Input Coding Type 1' dropdowns, and a 'Refresh' button. The Windows taskbar at the bottom shows the date '23 aug(8) 2010' and several desktop icons including 'reshte', 'Feynm', 'CaptureWi...', 'US Ne College', '2010\_ele', 'Minimiz Logic Cir', and 'BQDate'.



سیزدهمین کنفرانس دانشجویی مهندسی برق  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران

# جمع بندی





سازمان کنفرانس دانشجویی  
مهندسی برق ایران  
دانشگاه تربیت مدرس

سیزدهمین کنفرانس دانشجویی مهندسی برق  
ایران



## جمع بندی

- ▶ پیچیدگی طراحی مدارات منطقی چند سطحی ◀ استفاده از روش های بهینه سازی تکاملی
- ▶ دو روش ◀ طراحی مدارات منطقی ◀ با استفاده از الگوریتم ژنتیک.
- ▶ با توجه به مقایسه ی نتایج ◀ نتایج بهتری ارائه شد.



## جمع بندی

- ▶ پیچیدگی طراحی مدارات منطقی چند سطحی ◀ استفاده از روش های بهینه سازی تکاملی
- ▶ دو روش ◀ طراحی مدارات منطقی ◀ با استفاده از الگوریتم ژنتیک.
- ▶ با توجه به مقایسه ی نتایج ◀ نتایج بهتری ارائه شد.
- ▶ مزیت:
  - طراحی مدارات با سطوح بیشتر
  - بهینه سازی چندین تابع بطور همزمان

## جمع بندی

- ▶ پیچیدگی طراحی مدارات منطقی چند سطحی ◀ استفاده از روش های بهینه سازی تکاملی
- ▶ دو روش ◀ طراحی مدارات منطقی ◀ با استفاده از الگوریتم ژنتیک.
- ▶ با توجه به مقایسه ی نتایج ◀ نتایج بهتری ارائه شد.
- ▶ مزیت:
  - طراحی مدارات با سطوح بیشتر
  - بهینه سازی چندین تابع بطور همزمان
- ▶ عیب:
  - عدم کارامدی عملی برای تعداد متغیر های زیاد





## جمع بندی

- ▶ پیچیدگی طراحی مدارات منطقی چند سطحی ◀ استفاده از روش های بهینه سازی تکاملی
- ▶ دو روش ◀ طراحی مدارات منطقی ◀ با استفاده از الگوریتم ژنتیک.
- ▶ با توجه به مقایسه ی نتایج ◀ نتایج بهتری ارائه شد.
- ▶ مزیت:
  - طراحی مدارات با سطوح بیشتر
  - بهینه سازی چندین تابع بطور همزمان
- ▶ عیب:
  - عدم کارامدی عملی برای تعداد متغیر های زیاد
- ▶ کار های پیشنهادی برای آینده:
  - تمرکز روی بهینه سازی چند منظوره
  - ترکیب بهینه سازی تکاملی با روش های ابتکاری (Heuristic):
    - نیاز به بدست آوردن جواب های با خروجی دقیق و شبه بهینه روی اندازه ی گیت
    - افزایش سرعت و توانایی برای بهینه سازی با تعداد متغیر های بالا (مثلا 50 متغیر)

سوال / پیشنهاد / انتقاد؟  
با تشکر از توجه شما!