

بسم الله الرحمن الرحيم

رحلتي من تغميس الشاي الى البروسيسور

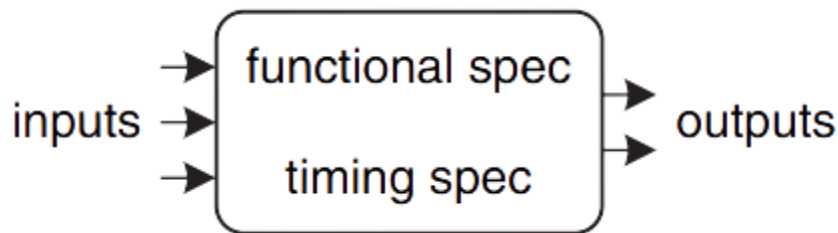
جميع ما هو مكتوب خالي من حقوق الطبع والنشر وتمت كتابته لوجه الله عز وجل

Combinational Logic Design

Introduction :

في الاكترونيات الرقمية الدائرة عبارة عن شبكة تقوم بمعالجة المتغيرات ذات القيمة المنفصلة
discrete-valued variables | ذات القيمة المنفصلة :

صورة لدائرة الرقمية :



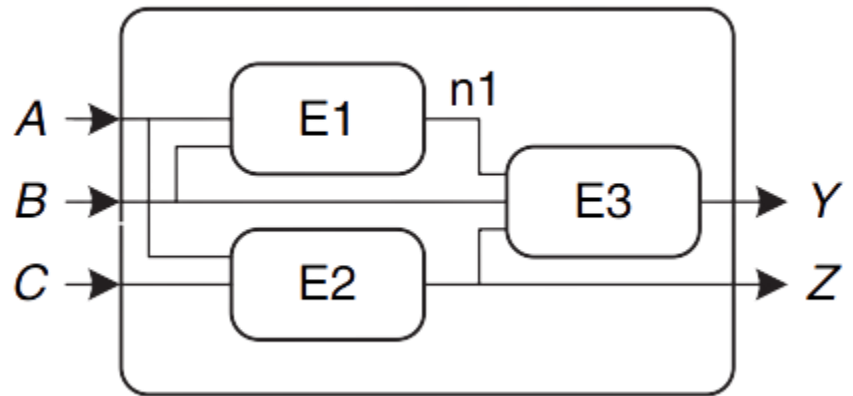
- واحدة او اكثر من القيمة المنفصلة في الادخال
- واحدة او اكثر من القيمة المنفصلة في الاخراج
- functional specification : تصف العلاقة بين المدخلات والمخرجات.
- timing specification : تصف التأخير بين تغيرات الادخال واستجابة المخرجات.

لما تشوف داخل الصندوق بتشوف تتكون الدوائر من

1. nodes : عقد.
2. elements : عناصر .

العناصر هي بحد ذاتها عن دائرة ذات مدخلات ومخرجات و
specification

العقد عبارة عن سلك ينقل الجهد متغيرا ذو قيمة منفصلة يتم
تصنيف العقد على انها مدخلات ومخرجات او داخلية
.Internal



توضح الصورة مكونة من ثلاث عناصر

E1,E2 & E3

و عقد

مدخلات : A ,B & C

مخرجات : Y & Z

والـ

هي عقدة داخلي بين الـ

E1 & E3

يتم تصنيف الدوائر الرقمية على انها متسلسلة

تعتمد مخرجات الدائرة فقط على القيم الحالية للمدخلات بمعنى
اخر فهو يجمع القيم الادخال الحالية لحساب الاخراج

على سبيل المثال البوابة المنطقية

البوابة المنطقية : Logic Gate

هي دائرة تركيبية

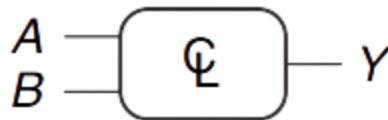
تعتمد مخرجات الدائرة المتسلسلة على القيم للمخرجات بمعنى اخر يعتمد على تسلسل الادخال
الدائرة المركبة

combinational circuit

لا تحتوي على ذاكرة لكن الذاكرة التسلسلة لها ذاكرة

يعتبر عن قيم الاخراج من حيث قيم
functional specification of a combinational circuit:
الادخال الحالية

تتكون من حدود منخفضة وعالية للتأخير :
timing specification of a combinational circuit
من الادخال والخراج



$$Y = F(A, B) = A + B$$

الصورة هاي بتوضح دائرة التركيبية مدخلين ومخرج واحد
على اليسار هيا المدخلات

A & B

والمخرجات على اليمين هيا

Y

والرمز الداخل الصندوق CL

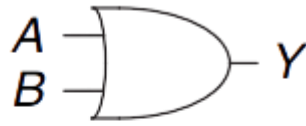
يشير الى انه يتم تنفيذه باستخدام المنطق التركيبي

هذا المثال تم تحديد الدالة او الوظيفة F

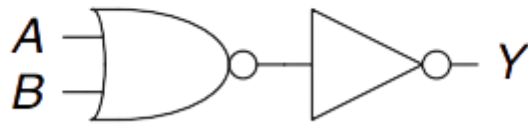
عشان تعمل في منطق OR

$$Y = F(A, B) = A + B;$$

$$Y = A \text{ OR } B;$$

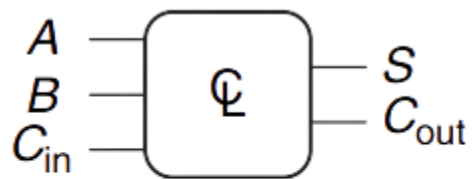


(a)



(b)

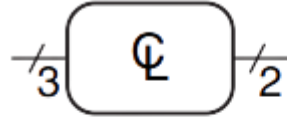
الصورة بتبين تطبيقين محتملين على الدائرة التركيبية



$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + AC_{in} + BC_{in}$$

الصورة هاي بتظهر دائرة تركابية يوجد لها اكثر من مدخل واحد تسمى full adder وراح نرجع لها بعدين



(a)



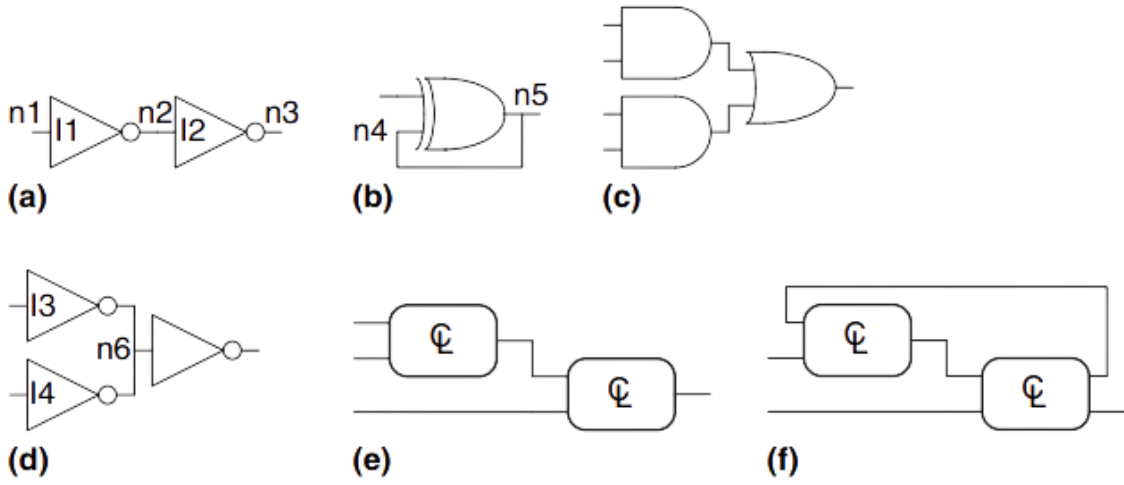
(b)

هذا الرسم ايسط بحيث نستخدم الشرطة المائلة ورقم بجواره للاشارة يحدد الرقم عدد المدخلات 3 والآخر يحدد عدد المخرجات 2 والصورة التي في الاسفل تدل على ان لا يهم وجود عدد البتات او عدد بتات عشوائي من المدخلات والمخرجات

تعتبر دائرة تركيبية اذا كانت تتكون من عناصر مترابطة

- كل عنصر من العناصر الدائرة هو مترابط بحد ذاته
- يتم تعيين كل عقدة في الدائرة كمدخل لدائرة او يتم توصيلها بطرف اخراج واحد
- لا تحتوي المسائرات التراكيبية على مسارات دائرية كل مسار عبر الدائرة يزور كل عقدة في الدائرة مرة واحدة او اكثر

مثال COMBINATIONAL CIRCUITS



اي من الدوائر في هاذي الصورة هي دوائر تركيبية حسب القواعد ؟

(a)

هي دائرة تركيبية مصنوعة من عنصرين تركيبين

I1 & I2

لديها ثلاث عقد

N1, n2 & n3

العقدة الاولى هو مدخل الى الدائرة I2

العقدة الثانية هي عقدة داخلية وهي مخرجات I1 ومدخلات I2

العقدة الثالثة هي عقدة هو مخرج لدائرة I2

(b)

ليست تراكيبة بسبب وجود مسار XOR ويعود الى الادخال

(c)

هي عقدة تركيبية

(d)

ليست عقدة تركيبية بسبب العقدة تتصل بمخرج I3 & I4

(e)

هي تركيبية دائرتين تراكيبة تتصل الى دائرة تركيبية اكبر

(f)

ليست تركيبية لانه المسار خلال عنصرين

BOOLEAN EQUATIONS

المعادلات البولينية تتعامل مع المتغيرات التي تكون اما صحيحة او خاطئة لذا فهي مثالية لوصف المنطق الرقمي

راح نشرح بعض المصطلحات الشائعة الاستخدام في المعادلات البولينية بعدها نوضح كيفية كتابة معادلة منطقية لأي دالة منطقية في جدول الحقيقة الخاص بها

Terminology :

مكمل الـ A هو معكوسه A' المكمل هاذي تسمى literal

على سبيل المثال :

A, A', B & B' يكون اسمها

Literals

الـ A بنسميه

True Form

الـ A' بنسميه

complementary form

لا يعني انه هو يحمل عدد 1 او من هذا القبيل بمعنى ان لا يوجد خط فوقه : True Form

يطلق على AND لواحد او اكثر من الحروف الـ

Literals

تسمى

product or an implicant

$AB' C'$

كلها

Implicants

لوظيفة مكونة من ثلاث متغيرات

Minterm : هو منتج يشتمل جميع مدخلات الوظيفة :

$AB\bar{C}$

: هو الـ مينتيرم للمتغيرات الثلاثة والحد الأدنى لهذا الموضوع هو ان يكون يوجد 3 متغيرات لكن اذا كان AB لا يعتبر مينتيرم

وكمان الـ

OR

فان واحد او اكثر من القيم الحرفية يسمى
Sum

Maxterm : هو مجموع يجمع كافة مدخلات الدالة :

$A + B + C$: Maxterm يسمى هذا

ترتيب المعادلات مهم عند تفسير الـ
Boolean equations

$Y = A + BC$ mean $Y = (A \text{ OR } B)$ or $Y = A \text{ OR } (B \text{ AND } C)$?

في المعادلات البوليانية الاولوية لـ

NOT

وبعدها

AND

وبعدها

OR

يعني الصحيح هو

$Y = A \text{ OR } (B \text{ AND } C)$

اي تعبير بوليني يمكن كتابته باربع صور

Sum-of-Products (SOP)

$$F = ABC + ABC' + AB'C$$

لو تلاحظ مثل ما شرحنا احد رموز الـ AND مش لازم تنكتب بس هي بين الاحرف حاليا A AND B AND C يعني مضربين في بعض عشان هيك اسمها تجميع حواصل الضرب

Canonical SOP

مبدئيا بنرمز لها برمز سيقما (Σ) مجموع يعني

$F(A,B,C) = \Sigma(1,3,4,6)$ تكتب هكذا وهيا minterms تعتبر طيب كم متغير هو في 3 حلو على اي وبدنا نعمل على عدد الثنائي 2 يعني $2^3 = 8$ عشان هيك بدنا 8 ارقام يعني لو تلاحظ

مجموع عددهم مش بنجمع ارقام هم $8 = 2^3$ ارقام يعني راح يبدأ من 0 الى 7 يعني مستحيل نلاقي رقم اكبر من 7 هون

$F(A,B,C) = \Sigma(0,1,3,6)$ يعني الارقام الي مش موجودة هون بتكتبها في الكانوكل الثاني و العكس صحيح شوف مثال في في الكانوكل تجميع حواصل الضرب مافيو 2 حظيناها في حواصل ضرب المجموع الخ.

$$F(A,B,C) = \Pi(0,2,5,7)$$

Product-of-Sum (POS)

مثال : $F = (A+B+C) \& (A+B'+C') \& (A'+B+C')$ هاي اسمها حواصل ضرب المجموع الي بين الاحرف مجموعين بينهم OR وفي الاخر بنضربهم في بعض

canonical POS

نرمز لها برمز باي (Π)

$$2^3 = 8$$

0 الى 7 راح نكتب الارقام الي مش موجود هناك راح نكتبو هون والعكس صحيح

$$F(A,B,C) = \Pi(0,2,5,7)$$

تعال نستنتج الاربع صور هضولا من الـ truth table جدول الحقيقة وكيف من صورة وحدة نجيب منها 3 صور

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

لما نيجي نجيب الـ SoP اول اشي بدنا نشوف صورة الـ F الي خرجو 1

001 = *

011 = *

100 = *

110 = *

عندي يعني 4 من قيم الواحد يعني 4 تيرم عندي

$$F = ABC + ABC + ABC + ABC$$

حاليا الرقم الي عليه 0 او نفي بدنا نخط عليه نفي

$$F = A'B'C + A'BC + AB'C + ABC'$$

لما نيجي نجيب الـ PoS اول اشئ بدنا نشوف صورة الـ F الي خرجو 0

$$000 = *$$

$$010 = *$$

$$101 = *$$

$$111 = *$$

$$F = (A+B+C) \& (A+B+C) \& (A+B+C) \& (A+B+C)$$

النفي هون عكس هون الواحد الي بدنا ننفيه

$$F = (A+B+C) \& (A+B'+C) \& (A'+B+C') \& (A'+B'+C')$$

حاليا جبنا صورهم من جدول الحقيقة

Canonical SOP

نفس الموضوع لكن بدنا نجيب الارقام الثنائية الموجودة في جدول الحقيقة

$$001 \ 011 \ 100 \ 110$$

الان بدنا نحولها لعشري

$$001 \rightarrow 1$$

$$011 \rightarrow 3$$

$$100 \rightarrow 4$$

$$110 \rightarrow 6$$

$$F(A,B,C) = \Sigma(0,1,3,6)$$

Canonical POS

$$000 \ 010 \ 101 \ 111$$

$$000 \rightarrow 0$$

$$010 \rightarrow 2$$

$$101 \rightarrow 5$$

$$111 \rightarrow 7$$

$$F(A,B,C) = \Pi(0,2,5,7)$$

BOOLEAN ALGEBRA

زي ما بتقدر في الجبر العادي انك تبني تعبير يكون ابسط بتقدر كمان تستخدم الجبر البولياني عشان تبني معادلات بسيطة عشان تبسطها تكون اكثر سهولة.

قواعد الجبر البولياني بتشبه كثير قواعد الجبر العادي ولكن في بعض الحالات الجبر البولياني يكون ابسط لان المتغيرات عندها قيمتين محتملين فقط الـ 0 و الـ 1.

بغتمد الجبر البولياني على مجموعة من الـ axioms التي نفترض انها صحيحة

الـ axioms : البديهيات غير قابلة للاثبات بمعنى انه لا يمكن اثبات التعريف ومن هاي البديهيات نثبت جميع نظريات الجبر البولياني.

جدول يبين ما نقصد به :

Axiom		Dual			Name
A1	$B = 0 \text{ if } B \neq 1$	$A1'$	$B = 1 \text{ if } B \neq 0$		Binary Field
A2	$0' = 1$	$A2'$	$1' = 0$		NOT
A3	$0 \text{ AND } 0 = 0$	$A3'$	$1 + 1 = 1$		AND/OR
A4	$1 \text{ AND } 1 = 1$	$A4'$	$0 + 0 = 0$		AND/OR
A5	$0 \text{ AND } 1 = 1 \text{ AND } 0 = 0$	$A5'$	$1 + 0 = 0 + 1 = 1$		AND/OR

Axioms

الجدول هاض بنص على بديهيات الجبر البوليني تحدد هاذي البديهيات الخمسة الـ Dual تاعهم والـ .NOT , AND & OR

الاولى بتنص على انو المتغير البوليني B يساوي 0 اذا لم تكن 1.

تنص البديهيه A1` على انو يكون 1 اذا لم تكن 0 .

A1` & A1 اذا كنا نعمل على حقل منطقي فيو 0 او 1.

الـ A2 AND A2` بتحدد عملية .NOT

والـ A3 الى A5 بتحدد AND والـ A3` الى A5` بتحدد OR.

Theorems of One Variable

Theorem	Dual	Name
T6 $B \cdot C = C \cdot B$	T6' $B + C = C + B$	Commutativity
T7 $(B \cdot C) \cdot D = B \cdot (C \cdot D)$	T7' $(B + C) + D = B + (C + D)$	Associativity
T8 $(B \cdot C) + (B \cdot D) = B \cdot (C + D)$	T8' $(B + C) \cdot (B + D) = B + (C \cdot D)$	Distributivity
T9 $B \cdot (B + C) = B$	T9' $B + (B \cdot C) = B$	Covering
T10 $(B \cdot C) + (B \cdot \bar{C}) = B$	T10' $(B + C) \cdot (B + \bar{C}) = B$	Combining
T11 $(B \cdot C) + (\bar{B} \cdot D) + (C \cdot D) = B \cdot C + \bar{B} \cdot D$	T11' $(B + C) \cdot (\bar{B} + D) \cdot (C + D) = (B + C) \cdot (\bar{B} + D)$	Consensus
T12 $\overline{B_0 \cdot B_1 \cdot B_2 \dots} = (\bar{B}_0 + \bar{B}_1 + \bar{B}_2 \dots)$	T12' $\overline{B_0 + B_1 + B_2 \dots} = (\bar{B}_0 \cdot \bar{B}_1 \cdot \bar{B}_2 \dots)$	De Morgan's Theorem

الجدول هاض بصف نظريات من T6 الى T12 كيفية تبسيط المعادلات التي تتضمن اكثر من متغير واحد. الـ T6 و T7 بعملو مثل الجبر العادي الي هو Commutativity and associativity. التبادل ما باثر ترتيب المدخلات لوظيفة AND او OR على قيمة المخرجات. من خلال العملية التجيمية او الترابطات لا تؤثر المجموعات المحددة على قيمة المخرجات. الـ T8 نظرية التوزيع هاي هي نفسها موجودة في الجبر العادي لكن الـ T8` ليست كذلك.

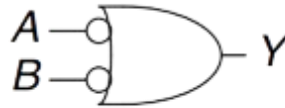
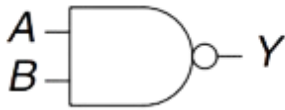
من خلال الـ T8 يتم توزيع الـ AND على الـ OR وفي الـ T8 العكس الـ OR يتم توزيعها على الـ AND. في الجبر التقليدي يوزع الضرب على الجمع ولكن الجمع لا يوزع على الضرب.

الـ T9 الى الـ T12 بتسمح لنا من التخلص من المتغيرات الزائدة عند الحاجة.

نظرية T12 De Morgan's Theorem نظرية قوية في التصميم الرقمي توضح النظرية ان مكمل حاصل ضرب جميع تيرم يساوي مجموع مكمل تيرم والعكس صحيح.

وفقا لنظرية مورجان بوابة الـ NAND بتعادل الـ OR GATE ذات المدخلات المقلوبة. وكمان الـ NOR تعادل بوابة الـ AND ذات المدخلات المقلوبة.

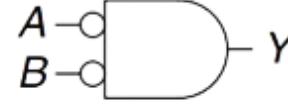
NAND



$$Y = \overline{AB} = \overline{A} + \overline{B}$$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR



$$Y = \overline{A + B} = \overline{A} \overline{B}$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

بنقدر نستخدمها في التبادل.

الدائرة هاي الصغرة بتشوفها في البوابة هاي يكون اسمها bubble فقاعة الي بتادي الي انها تقلب القيمة مثال ممكن قلب من الـ AND الى الـ OR.

مثال في الصورة تتكون بوابة الـ NAND من الـ AND وجود فقاعة الي هي اصلا NOT ممكن تشوفها من الصورة. اذا كانت الـ الفقاعة في الادخال بتكون بتعكس القيمة نفسها واذا كانت في عند الاخراج بتعكس الاخراج.

قواعد اساسية لدفع الفقاعات :

1. Pushing bubbles backward : دفع الفقاعات من المخرج او الى الامام تغيير البوابة من AND الى OR و العكس صحيح.
2. يؤدي دفع الفقاعة من المخرج الى المدخلات الفقاعة تضع على مدخلات البوابة.
3. دفع الفقاعات على بوابات الادخال يوضع الفقاعة على المخرج.

The Truth Behind It All

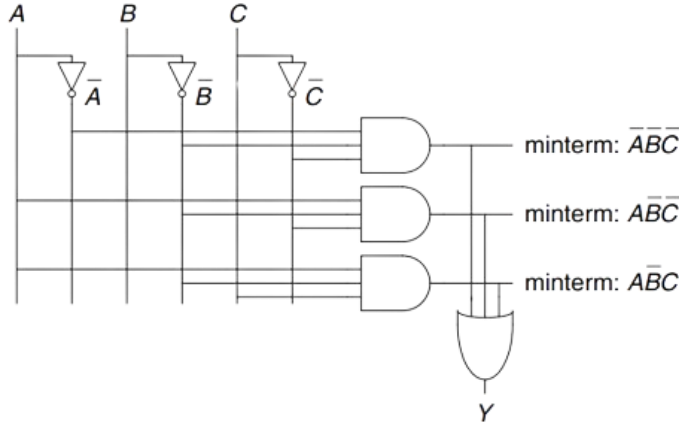
قد يتسأل القارئ الفضولي عن كيفية اثبات النظريات يمكن اثباتها عكس البديهيات الموضوع سهل كيف تثبتها للنظريات انك تحط القيم المحتمله يمكنك اثباتها عن طريق جدول الحقيقة.

Simplifying Equations

النظريات الي شرحناها بتساعدنا على تبسيط الجبر البوليني.

FROM LOGIC TO GATES

الرسم التخطيطي (schematic) هو الـ (diagram) رسم تخطيطي للدوائر الرقمية يوضح العناصر والاسلاك التي تربطها ببعضها البعض.



على سبيل المثال هذا الرسم التخطيطي يوضح.

$$Y = A' B' C' + AB' C' + ABC:$$

ومن خلال رسم المخططات بطريقة متسقة نجعلها اسهل للقراءة وتصحيح الاخطاء.

- الادخلات موجودة في الاعلى او على اليمين
- المخرجات موجودة في الاسفل او على اليسار
- تدفق البوابات من اليسار الى اليمين
- الاسلاك المستقيمة افضل في الاستخدام في من الاسلاك ذات زوايا متعددة
- يتم توصيل دائما الاسلاك عند تقاطع على شكل T
- تشير النقطة التي تتقاطع فيها الاسلاك الى وجود اتصال بين الاسلاك
- الاسلاك المتقاطعة بدون نقاط لا تؤدي الى اي اتصال

التخطيط بشكل منهجي مثل الموجود في الصورة اول اشئ ارسم الاعمدة للمدخلات
حط العاكسات في الاعمدة المجاورة لتوفير complementary للمدخلات اذا لزم الامر

ارسم الـ AND لكل minterms

لكل مخرج ارسم بوابة OR متصلة في minterms المتعلقة بذلك المخرج.

يسمى هذا النمط بالمصفوفة المنطقية القابلة للبرمجة (PLA) programmable logic array
لأن يتم ترتيبها بطريقة منتظمة

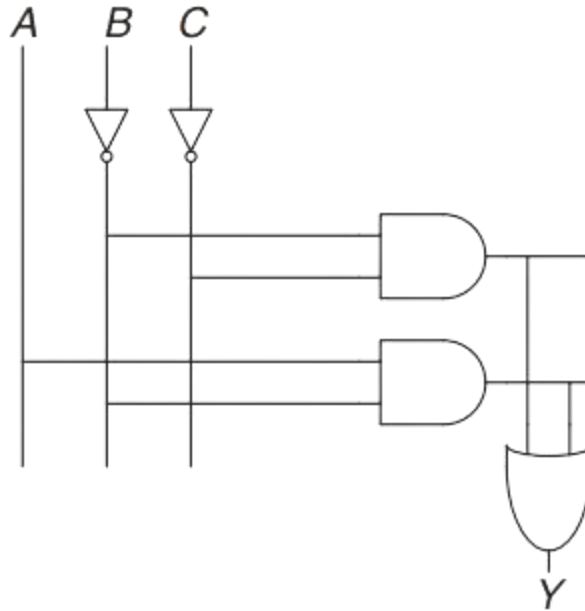


Figure 2.25 Schematic of
 $Y = \overline{B} \overline{C} + A \overline{B}$

الصورة هاي بتظهر تنفيذ المعادلة المبسطة.

يمكننا تقليل عدد البوابات بشكل اكبر ولو عن طريق عاكس واحد من خلال الاستفادة من البوابات المعكوسة (inverting gates)
 لاحظ ان BC عبارة عن AND بمدخلات مقلوبة لنتخلص من العاكس الموجود في C

نتذكر ليش نظريات مورجان مهمة نظرا لنظريته فان AND مع مدخلات مقلوبة تفعلال الـ NOR

اعتمادا على تقنية التنفيذ قد يكون من الارخص استخدام اقل عدد من البوابات او استخدام انواع معينة من الالبواب مثال يتم تفضيل الـ بوابات المعكوسة على العادية مثل NAND NOR على OR AND في الـ CMOS implementations

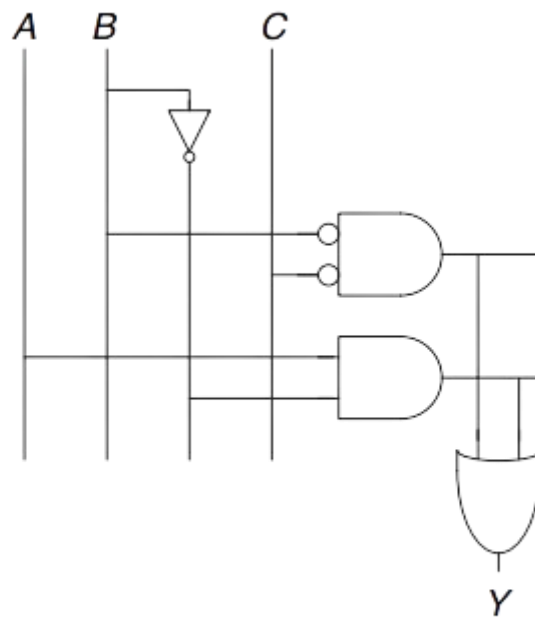


Figure 2.26 Schematic using fewer gates

KARNAUGH MAPS

بعد ما عملنا مع عدة عمليات من ناحية انو نصغر المعادلات البولينية باستخدام الجبر البوليني نفسو وكمان هذا الموضوع حذر اذا صار في اي غلط الموضوع راح يصير معادلة مختلفة بدلا من انها معادلة مبسطة.

Karnaugh maps (K-maps) : هي طريقة رسمية لتبسيط المعادلات البولينية.

تعمل الـ k-maps بشكل جيد مع المشكلات التي تحتوي على ما يصل الى اربعة متغيرات.

والاهم من ذلك انها تعطي نظرة ثاقبة لمعالجة المعادلات البولينية يعني هي بتعطي معلومات اكثر من الـ Truth table .

بتؤخذ شكلين من الـ SoP & PoS هو بشبه الجدول ونقوم بملئها بصفر و الواحد بنظام العد الثنائي.

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

(a)

Y C	AB			
	00	01	11	10
0	1	0	0	0
1	1	0	0	0

(b)

Y C	AB			
	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

(c)

الصورة هاي بتبين الـ K-Maps والـ Truth Table فيها 3 مدخلات.

الصف العلوي من الـ k-maps القيم الاربعة المحتملة للمدخلات A & B والعمود الايسار قيمتين المعتملين لـ C

A & B بمعنى قيمتين 2^2 بكون بساوي 4 يعني

10 11 01 00

والـ C هي قيمة وحدة يعني 2^1 بساوي 2 يعني قيمتين 0 و 1

كل مربع موجود في الـ K-Maps يشبه الـ Truth table ويحتوي على قيمة اخراج Y

على سبيل المثال المربع العلوي على اليسار مع الصف الاول يتوافق مع الـ Truth Table وبشير انو عند الـ $(A,B,C) = (0,0,0)$ يكون الاخراج 1 بزيبط نفس الـ Truth Table كل مربع في الـ k-maps يمثل single minterm

وكل مربع او مينيتيرم يختلف عن المربع المجاور في single variable
هذا يعني ان المربعات المجاورة تشترك في نفس القيم الحرفية باستثناء واحد بشكل True في مربع واحد وبصورة مكملية.
على سبيل المثال المربعات التي تمثيل حدين الـ A BC متجاورين لكن تختلف قيمهم.

ربما لاحظت ان المجموعتين A & B في الصف العلوي مترتبه بترتيب 00 01 10 11 وهاض يكون اسمو الـ Gray code وهاض يختلف عن ترتيب الثنائي العادي الي هو 00 01 10 11 .

واليا بدنا نسوي K-map

$x, y \rightarrow z$

x, y		
0 0	0 1	m_0
1 0	1 1	m_1
0 1	1 0	m_2
1 1	0 1	m_3

بدنا نحولها لخلية للجدول هذا حاليا بدنا نسوي 4 خلايا و 4 خانات مربعات

x \ y	0	1
0	m^0	m^1
1	m^2	m^3

الـ x يكون الها حالتين يا 0 او 1 و الـ y نفس الموضوع
 طيب حاليا صفر وصفر ك مينتيرم كم يكون ؟ صفر
 طيب صفر و واحد راح يكون واحد
 طيب واحد وصفر راح يكون واحد يعني
 طيب واحد و واحد راح يكون كمان واحد وراح نرتبهم ك m^n

حاليا الـ m^2 بتجاور الـ m^3 وتجاور الـ m^0

قيمة الـ m^0 هي 00
 وقيمة الـ m^2 هي 10
 كم الاختلاف بينهم ؟ بختلفو برمز واحد فقط.

نشوف حاليا كمان الـ $m2$ و الـ $m3$ قيمة الـ $m3$ هي 11 كم مقادر الاختلاف بيناتهم ؟ مختلفين في متغير واحد فقط.

$$x, y, z \rightarrow 2^3 = 8 = 0:7$$

$+ yz$	00	01	10	11
00	000	001	010	011
01	100	101	110	111
10	000	001	010	011
11	100	101	110	111

$x \backslash yz$	00	01	10	11
0				
1				

لو تلاحظ في بين الـ 01 و 10 مختلف بمتغيرين يعني الـ 0 هي في الخانه المجاوره 1 والعكس صحيح يعني متغيرين واحد شروط الـ kmap انو يكونو بمختلف متغير واحد فقط عشان هيك راح نستخدم الـ gray code ممكن تتعلمو لحالك مافي اي مشكله المهم تكون عارفو على الاقل المهم راح نقرب الـ 10 و الـ 11 بحيث يكون في اختلاف بمتغير واحد فقط

يعني راح يصير زي هيك :

$$xy \geq \frac{7}{8} \rightarrow 0.7$$

$+ y \geq$	046
0 0 0	
0 0 1	
0 1 0	
0 1 1	
1 0 0	
1 0 1	
1 1 0	
1 1 1	

xy	00	01	11	10
x				
0				
1				

راح نعبي المربعات :

$$xy \geq \frac{7}{8} \rightarrow 0.7$$

$+ y \geq$	046
0 0 0	
0 0 1	
0 1 0	
0 1 1	
1 0 0	
1 0 1	
1 1 0	
1 1 1	

xy	00	01	11	10
x				
0	m^0	m^1	m^3	m^2
1	m^4	m^5	m^7	m^6

وكيفية القراءة مثال الـ

M0 راح تكون لو تشوف عند الـ x هي 0 وعند الـ yz هي 00 يعني راح تكون 000

الـ M1 هي عند الـ x 0 عند الـ yz هي 01 يعني راح تكون 001 وهكذا

تكتب الـ Kmap بطول بالعرض يلي تحبو المهم يكون الترتيب صح وهاي بداية عن الـ kmap

هون شرح لـ kmap افضل بكثير :

<https://www.youtube.com/watch?v=Q8MULkpPFT0>

COMBINATIONAL BUILDING BLOCKS

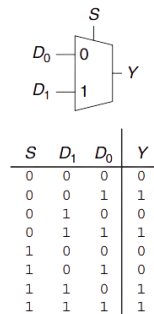
Combinational logic : غالبا ما يتم تجميعه في وحدات بناء اكبر لبناء انظمة اكثر تعقيدا.

يعد هذا الموضوع تطبيقا لمبدأ التجريد حيث يقوم باخفاء التفاصيل غير الضرورية على مستوى الـ gate-level بمعنى هو تصميم الكترونيه تقوم باداء وظائف هي تعتمد فقط على الدوائر الرقمية مثل AND OR NOT لانتاج اشارة مخرجة بناء على تلك الاشارات الداخلة المبدأ الاساسي هنا هو تجميع البوابات الرقمية والـ Combinational logic الى كتل بنائية اكبر واكثر تعقيدا يساعد على تنظيم الدوائر وتقسيمها الى اقسام منفصلة بدل ما كل بوابة بتعمل لحالها هذه الكتل البنائية تكون كأنها وحدات او مكونات اساسية تستخدم لبناء نظم اكبر مثل الـ Bit adders والـ DSP System

Multiplexers

Multiplexers : هي بتكون بتستخدم كثير بين الـ combinational circuits

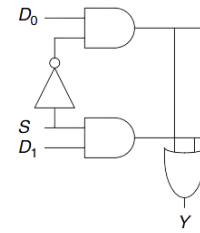
لانهم بتستخدمو مخرج واحد من بين عدة مدخلات محتملة مبنية على قيمة الاشارة ويسمى ايضا MUX كاختصار



الصورة هاهي بتوضح الـ schematic and truth table لـ mux 2:1 مع مدخلي بيانات D0 AND D1 و select input هو S و مخرج واحد الي هو Y
الملتيلكسر بختار بين مدخلي البيانات بناء على الـ if S = 0 ,Y = D0 & if S = 1,Y = D1 ويطلق كمان على الـ S الـ Control Signal or Control Variable لانها هي الي بتتحكم في ما يفعله الـ mux .

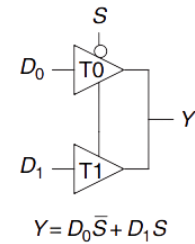
2:1 multiplexer : يمكن انشاؤه من SoP مثل ما راح تشوف الصورة :

		D _{1:0}			
	S	00	01	11	10
Y	0	0	1	1	0
	1	0	0	1	1

$$Y = D_0 \bar{S} + D_1 S$$


يمكن ان تستمدو المعادلة المنطقية لـ MUX باستخدام الـ K-maps او Y is 1 if S = 0 AND D0 is 1 OR if S = 1 AND D1 is 1
$$Y = (S * D1) + (S' * D0)$$

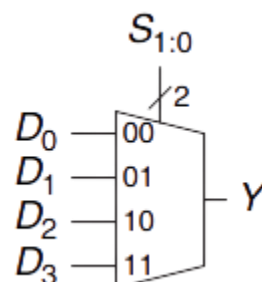
وكمان بتقدر تبني الـ MUX من الـ tristate buffers



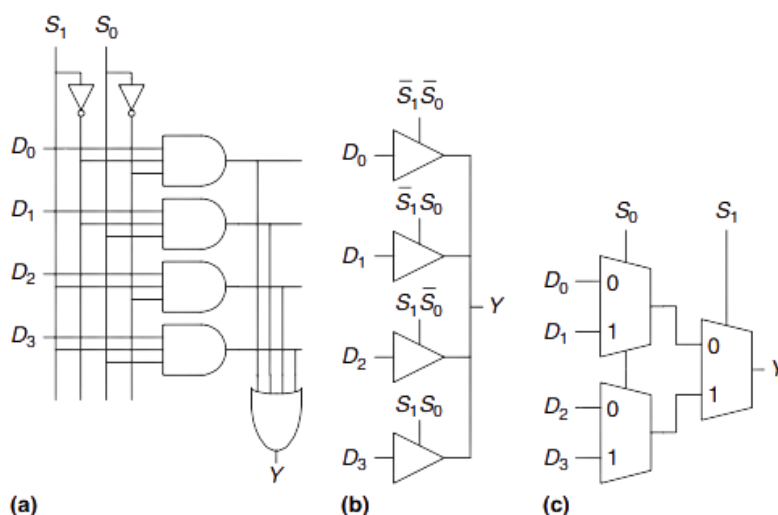
الـ tristate enables : يتم ترتيبها بحيث يكون الـ tristate buffer نشط دائما
لما الـ S = 0 يتم تشغيل tristate T0 مما يسمح لـ D0 بتدفق الى Y .
لما الـ S = 1 يتم تشغيل tristate T1 مما يسمح لـ D1 بتدفق الى Y .

Wider Multiplexers

الـ MUX 4:1 يحتوي على اربعة مداخل بيانات ومخرج واحد



في 2 select signals عند الشحطه لو تلاحظ عشان تختار بين مدخلات البيانات الاربعة. يمكن انشاء الـ MUX من خلال الـ SoP او Logic او multiple 2:1 mux مثل الي في الصورة



الـ product terms يمكن انها تشتغل باستخدام بوابات الـ AND والـ inverters . وكمان من الـ Decoder الي راح نشرحو.

الـ Wider multiplexers مثل الـ 8:1 16:1 mux الخ يمكن بنائهم من خلال الطرق الموجودة في هاي الصورة.

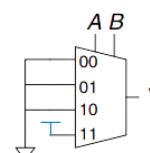
بشكل عام يحتاج الـ MUX N:1 بحتاج الى خيوط بقدر $\log_2 N$ وفي الاخر يعتمد خيار التنفيذ الأفضل على التكنولوجيا المستهدفة.

Multiplexer Logic

يمكن استخدام الـ MUX كجداول (lookup tables) لإنجاز الـ logic functions

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$Y = AB$

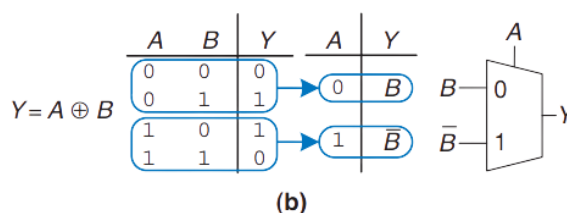
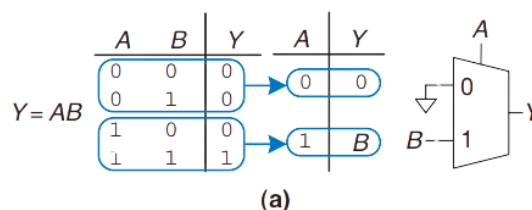


مثل الي في صورة باستخدام MUX يستخدم مدخلين بيانات لتنفيذ الـ AND Gate .
المدخلات A & B مثل الخيوط.

الملتيلكسر مدخلين البيانات الخاصه به يتم توصيله ب 0 او 1 وفقا لصف الي موجود امامه في الـ

Truth Table

الـ 2N-Input MUX يمكن برمجته لأداء اي وظيفة منطقية ذات ادخال N-input logic function من خلال تطبيق الـ 0 و 1 على مدخلات البيانات المناسبة.
تغير مدخلات البيانات يمكن اعادة برمجة الـ MUX ويقوم بانجاز وظيفة اخرى.



الصورة هاي بتوضح وظائف الـ AND & XOR ذات مدخلين يتم تنفيذها من خلال MUX 2:1 .
اول اشئ بدينا في Truth Table ثم نقوم بدمج ازواج من الصفوف لتخلص من الصف الي على أقصى اليمين من خلال انو نحط الـ output terms بس.

مثال في حالة الـ AND عندما يكون الـ A = 0 يكون الـ Y = 0 بغض النظر عن B .
إذا الـ A = 1 إذا الـ Y = 0 يكون الـ Y = 1 في الاخر Y = B بمعنى رياضي افضل :

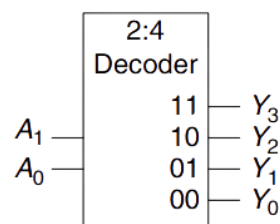
$$A = 1$$

$$Y = 0 \text{ if } (B = 0 \ Y = 0)$$

$$Y = 1 \text{ if } (B = 1 \ Y = B)$$

Decoders

الـ Decoder عندو N Input وبخرج 2N Outputs يؤكد احد المخرجات على مجموعة المدخلات.



A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

2:4 Decoder

عندما :

$$A1:0 = 00$$

$$Y_0 = 1$$

عندما :

$$A1:0 = 01$$

$$Y_1 = 1$$

الخ

المخرج هون اسمو one-hot اسمو Hot لان حرفيا يكون hot يكون في احد حالاتو HIGH في وقت معين.

المثال هذا حلو بستخدم 4 AND Gate بوابات كل بوابة بتعتمد على الـ true او الـ complementary لكل مدخل.

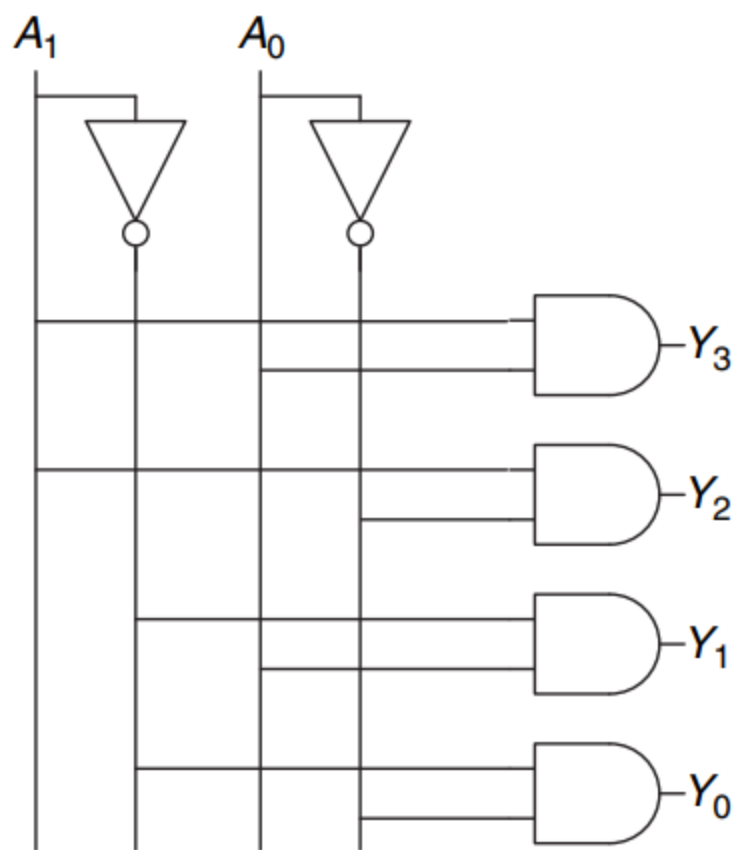
بشكل عام يمكن انشاء Decoder من N:2N يعني N-input و 2N Output تقبل مجموعات مختلفة من المدخلات الـ true او الـ complementary inputs بمعنى 0 او 1.

كل output للـ Decoder هاض بمثل single minterm .

$Y_0 = 1$ اذا كان الادخال 0 0 بمعنى هو يساوي صفر في العشري فلهذا يتم تمثيله

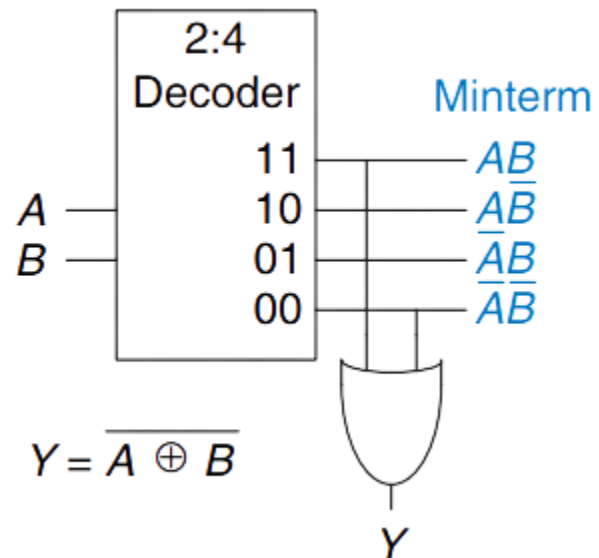
$Y_1 = 1$ لما يكون 01 بمعنى انه واحد في العشري.

الرسم ديغرامى راح يكون بهاض الشكل :



Decoder Logic

يمكن دمج الـ OR Gate لكي تقوم ببناء logic function :



الصورة هاي بتوضح XNOR Gate ذات مدخلين باستخدام 2:4 Decoder لان كل مخرج فكل مخرج يمثل single minterm وتم بناء وظيفة OR لكل الـ minterms in the function . $Y = A' \oplus B'$

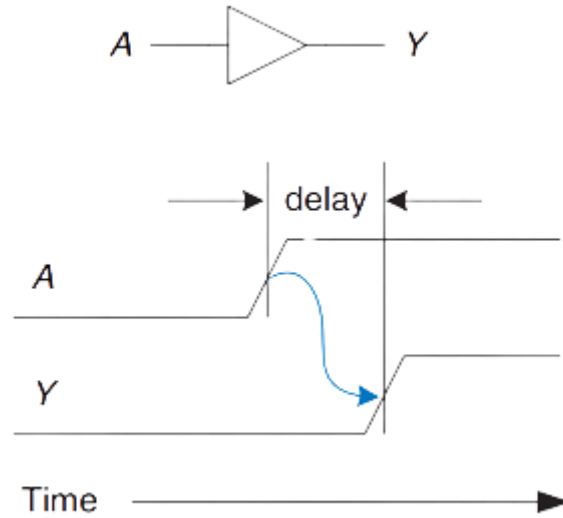
عند استخدام الـ Decoder لبناء logic يكون من السهل تعبير عن الوظائف كـ Truth Table او SoP

TIMING

في الأقسام السابقة كنا مهتمين في المقام الأول بما إذا كانت الدائرة تعمل بشكل مثالي أم لا من الناحية المثالية باستخدام أقل عدد من البوابات. أي مصمم دوائر متمرس أحد المشكلات الأكثر صعوبة في تصميم الدوائر هو التوقيت جعل الدائرة تعمل بسرعة.

الـ Output بوخذ وقت لتغير المدخلات.

الصورة هاي توضح التأخير بين الإدخال والإخراج في الـ buffer gate .



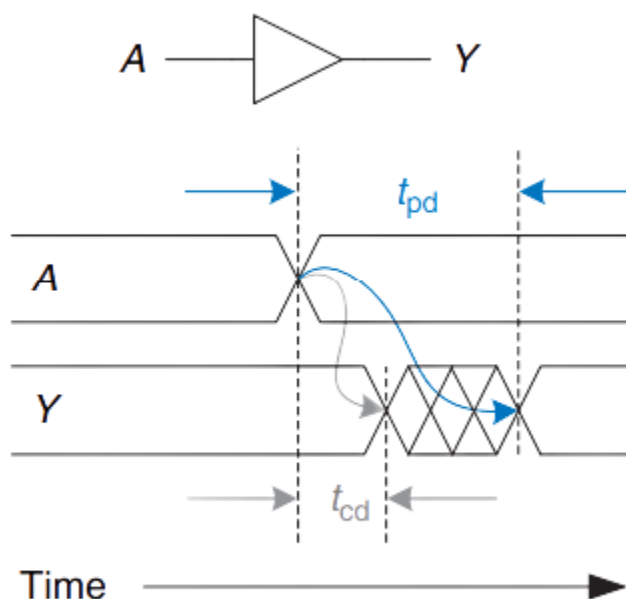
الصورة هاي اسمها timing diagram يقوم بتصوير كيفية الاستجابة buffer circuit عندما يدخل متغير.

الموضوع الي يكون الانتقال بين الـ LOW الى HIGH اسمها rising edge . وبالمثل فان الانتقال الى الاعلى الى المنخفض غير موضح في الصورة الان يسمى falling edge .

السهم الازرق يشير ان الحافة الى rising edge لـ Y ناتجة عن الـ A rising edge . نقوم بقياس التأخير من نقطة 50% من إشارة الإدخال A الى نقطة 50% من إشارة الإخراج Y . نقطة الـ 50% او Point 50% هي النقطة التي يكون فيها الإشارة في منتصف الطريق بين الـ LOW & HIGH أثناء انتقالها (transitions).

Propagation and Contamination Delay

الـ **Combinational logic** يتميز بـ **propagation delay and contamination delay** .
التأخير النشر (propagation delay) الـ t_{pd} هذا هو الحد الأقصى او الـ maximum للوقت منذ تغير المدخلات حتى تصل للمخرجات الى قيمتها النهائية.
تأخير التلوث (contamination delay) الـ t_{cd} هو الحد الأدنى (minimum) من الوقت منذ تغير المدخلات حتى يبدأ اي مخرج بتغير قيمته.



الصورة هاي بتوضح تأخير الـ Buffer من الـ propagation delay and contamination delay

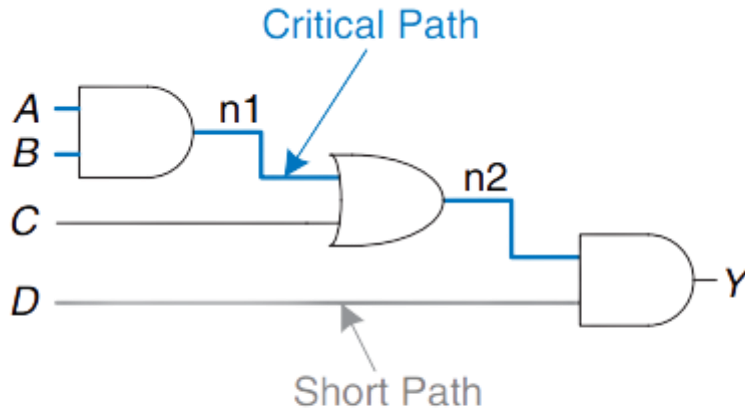
بلون الازرق والرمادي عند الاسهم شكل A بوضح بتكون عالية او منخفضة وتتغير الى الحالة الاخرى في وقت معين ونحن مهتمون بحقيقة انه يتغير لا بقيمته.
الـ Y يتغير في وقت لاحق الاقواس في Y الى ان يبدأ التغير T_{cd} بعد تحولات الـ A وقيمة Y الجديد هي في الـ T_{pd} .

تشمل الاسباب الرئيسية في التأخير في دوائر الوقت لشحن السعة في الدائرة وسرعة الضوء الـ T_{cd} & T_{pd} يختلف لعدة اسباب منها :

1. تأخير الصعود والهبوط مختلفة.
2. مدخلات ومخرجات متعددة بعضها اسرع من الاخر.
3. تتباطأ الدوائر عندما تكون ساخنة وتتسارع عندما تكون باردة.

يتطلب حساب الـ T_{cd} & T_{pd} الخوص في مستويات الادنى من التجريد لن نقوم بشرحه لكن مهندسين الكهرباء بنسبه لهم هذا شيء تافه وفي الشغل المحترم مصممين الدوائر بوفرو تقارير التي تحدد تأخير كل بوابة.

لكن يتم تحديد اشارات التأخير والتلوث من خلال المسار الذي تسلكه الاشارة من الادخال الى الاخراج.



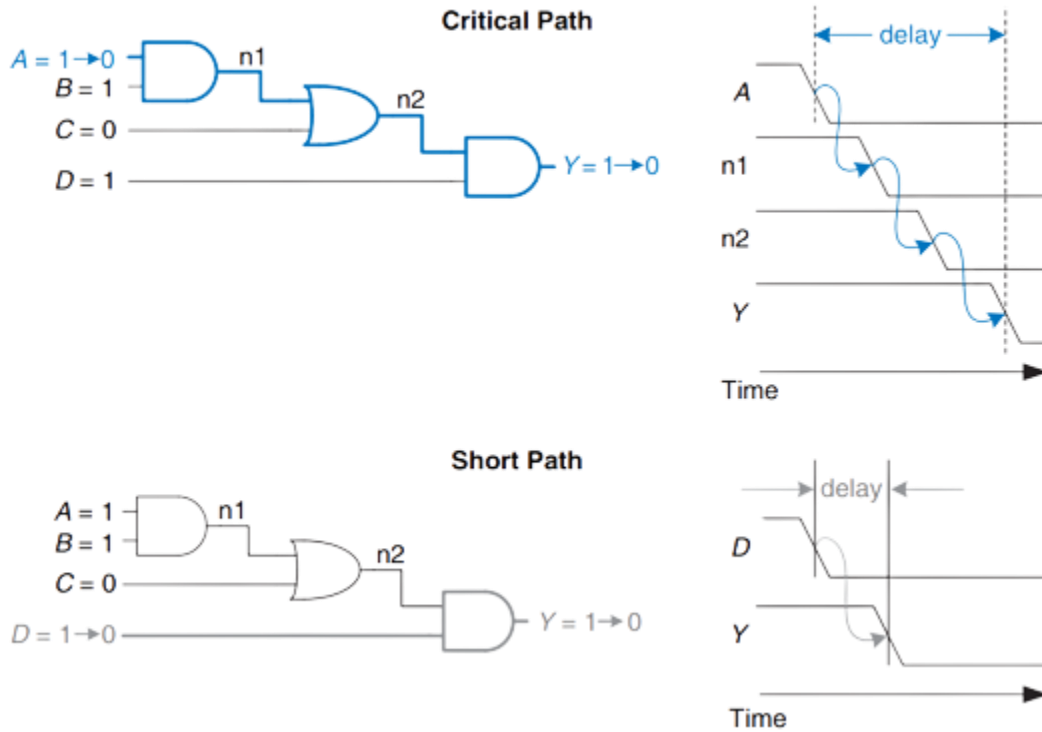
الصورة هاي بتوضح دائرة منطقيه يوجد لها 4 ABCD مدخلات و مخرج واحد Y مسار الدائرة الموضوع بلون الازرق هو مسار الادخال بين الـ A & B الى اخراج Y هو المسار الاطول يعني الابطأ لان المدخلات تنتقل عبر ثلاث بوابات الى مخرج. يعد هذا المسار مهم لانه يحدد من سرعة التي تعمل بها الدائرة.

المسار القصير الذي يسمى (Short Path) موجود في الدائرة باللون الرمادي يعمل من ادخال D الى اخراج Y .

هذا هو المسار الاقصر يعني الاسرع لان الادخال زي ما انتا شايف ينتقل عبر بوابة وحده زي الطياره .

الـ **propagation delay** لدائرة هو اصلا عبارته عن مجموع تاخيرات عبر كل عنصر على مسار الدائرة Critical Path او المسار الحرج يعني اجمع كل المسارات هاي مع بواباتها مع الخبصه كلها بطلع لك الـ **propagation delay**

الـ **contamination delay** لدائرة هو اصلا عبارة عن مجموع تاخيرات التلوث الموجودة خلال كل عنصر موجود على المسار القصير **short path** هنا يتم تحديد .



الصورة هاي بتوضح التاخيرات التلوث خلال كل عنصر على المسار القصير

يتم وصفهم في المعادلات التالية :

$$T_{pd} = 2t_{pd_AND} + T_{pd_OR}$$

$$T_{cd} = t_{cd_AND}$$

Table 2.7**circuit elements**

Gate	t_{pd} (ps)
NOT	30
2-input AND	60
3-input AND	80
4-input OR	90
tristate (A to Y)	50
tristate (enable to Y)	35

Glitches

احنا شرحنا الحالة الي جنب الحيط يعني من الادخال الى الاخراج والسلام عليكم ومع ذلك فمن الممكن ان يؤدي انتقال ادخال واحد الى عدة مخرجات وهذا يسمى **glitches or hazards** صح انو قلنش لا يتسبب بمشاكل لكن من المهم ادراك انها موجودة وحل مشاكلها عند النظر الى الرسوم البيانيه الخاصه بتوقيت.

سُورَةُ الْإِخْلَاصِ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
 قُلْ هُوَ اللَّهُ أَحَدٌ ① اللَّهُ الصَّمَدُ ② لَمْ يَلِدْ وَلَمْ يُولَدْ ③
 وَلَمْ يَكُنْ لَهُ كُفُوًا أَحَدٌ ④

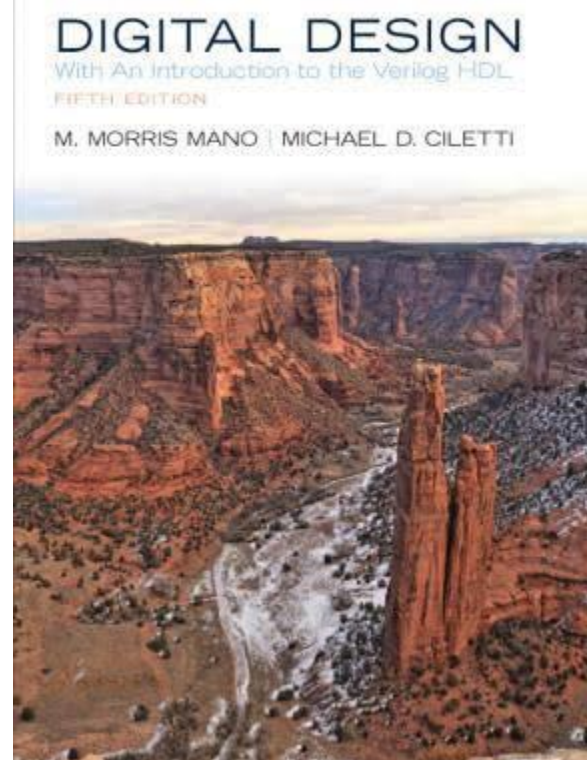
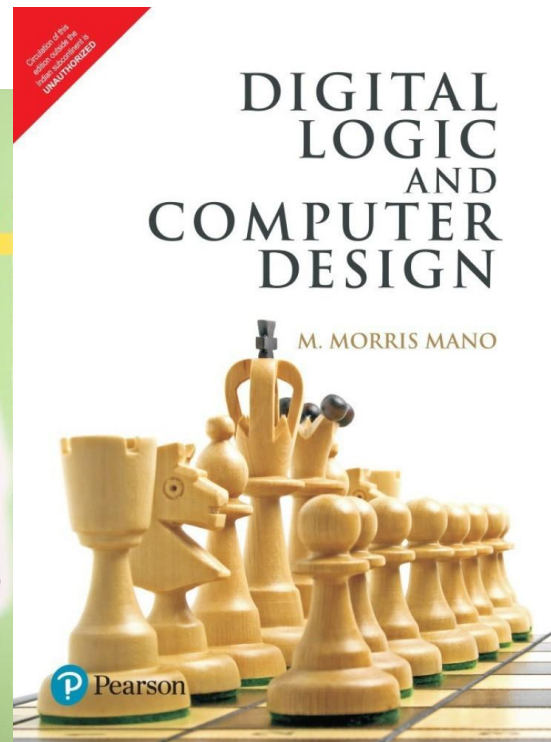
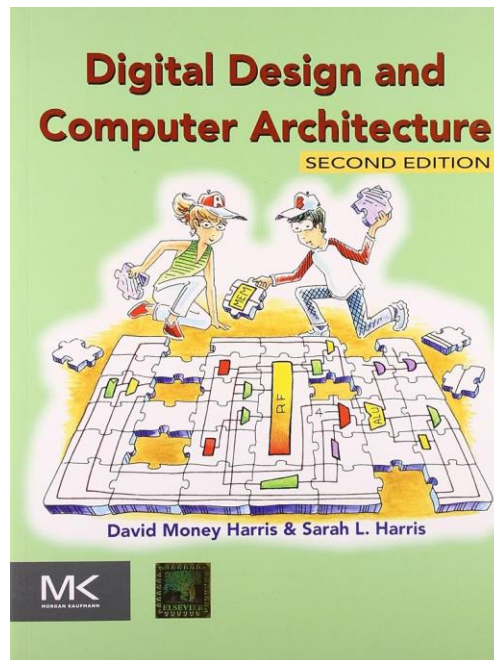
سُورَةُ الْفَلَقِ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
 قُلْ أَعُوذُ بِرَبِّ الْفَلَقِ ① مِنْ شَرِّ مَا خَلَقَ ② وَمِنْ شَرِّ
 غَاسِقٍ إِذَا وَقَبَ ③ وَمِنْ شَرِّ النَّفَّاثَاتِ فِي الْعُقَدِ ④
 وَمِنْ شَرِّ حَاسِدٍ إِذَا حَسَدَ ⑤

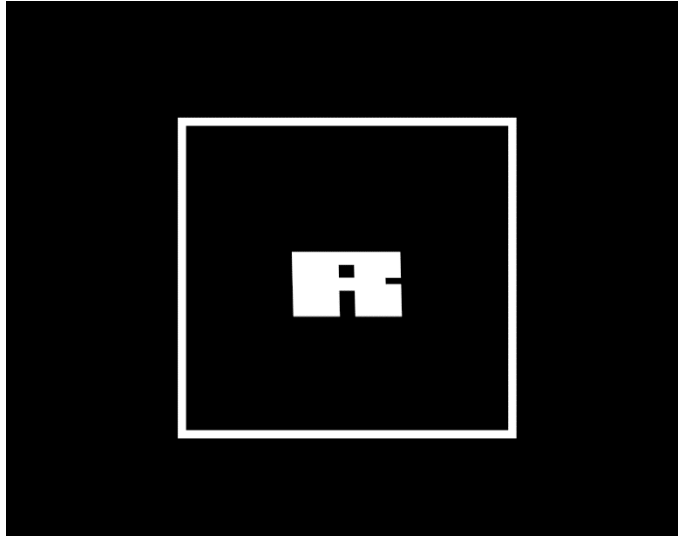
سُورَةُ النَّاسِ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
 قُلْ أَعُوذُ بِرَبِّ النَّاسِ ① مَلِكِ النَّاسِ ② إِلَهِ
 النَّاسِ ③ مِنْ شَرِّ الْوَسْوَاسِ الْخَنَّاسِ ④ الَّذِي
 يُوَسْوِسُ فِي صُدُورِ النَّاسِ ⑤
 مِنَ الْجِنَّةِ وَالنَّاسِ ⑥

المصادر :



Ahmad AlFareed (rETKit)



Twitter : https://twitter.com/dr_retkit

YouTube : <https://www.youtube.com/@retkit1823>

