

ثانيا : Triggered Subsystems

وهى نوع من انواع الانظمة الفرعية والتي يتم تنفيذها عند حدوث ال Trigger وهذه الانظمة الفرعية يكون لها مدخل تحكم واحد ويسمى trigger input والذي يحدد هل يتم تنفيذ النظام ام لا ؟؟

ويوجد ثلاثة انواع لهذا ال Trigger وهما

Rising

وهذه الحالة عندما ترتفع اشارة التحكم من قيمة سالبة او صفر الى قيمة موجبة واذا كانت القيمة الابتدائية سالبة فانه يتم تنفيذ النظام الفرعى عندما ترتفع الى الصفر

Falling

وهذا عكس الحالة السابقة اى عندما تنخفض اشارة التحكم من قيمتها الابتدائية الى قيمة اقل منها

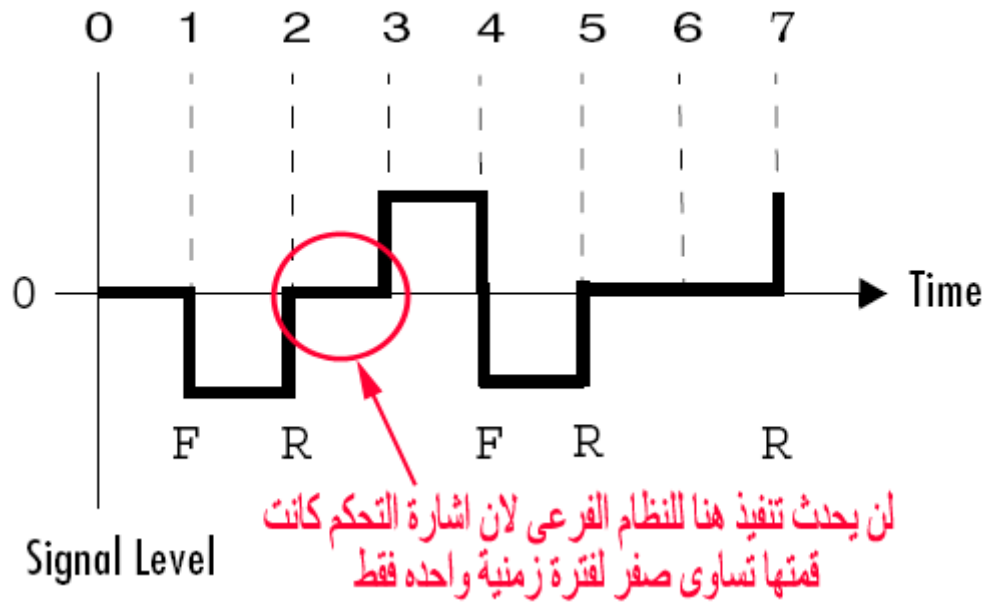
Either

وهذه الحالة تجمع الحالتين السابقتين ويتم تنفيذ النظام الفرعى فى كلتا الحالتين سواء بالارتفاع او الانخفاض .

ملحوظة :

فى حالة الانظمة ذات الزمن المتقطع عندما تتغير الاشارة من الصفر سواء بالارتفاع او الانخفاض فانه يتم تنفيذ حدوث النظام الفرعى عندما تكون قيمة الاشارة صفر لكثر من فترة زمنية واحدة وذلك لتلافى حدوث trigger بسبب عملية التقطيع .

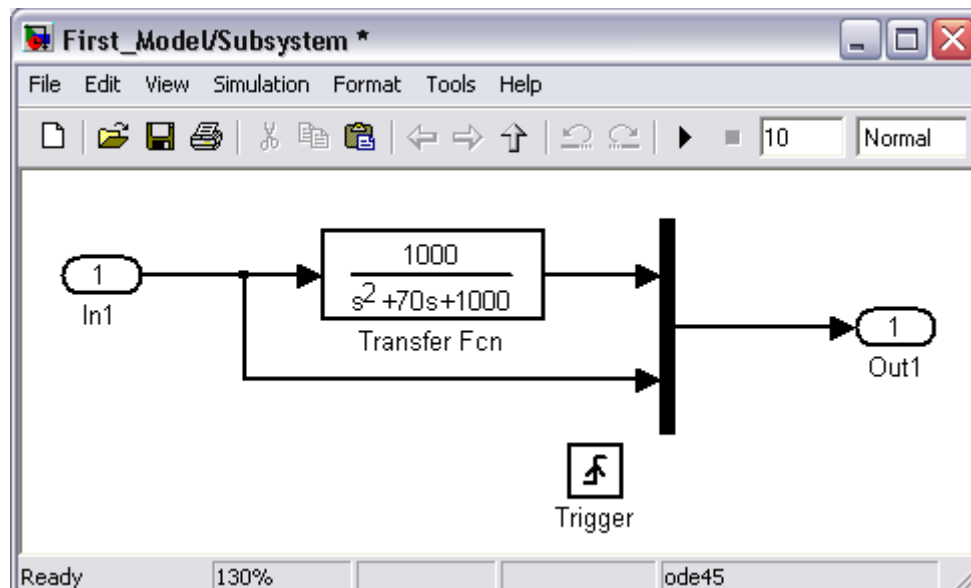
وفى الشكل الاتى نرى تنفيذ لل trigger اما بالارتفاع R او بالانخفاض F ونلاحظ انه فى الفترة الزمنية ٣ لم يحدث trigger لان الاشارة كانت قيمتها صفر لفترة زمنية واحدة ولذلك يجب ان تزيد عن واحد



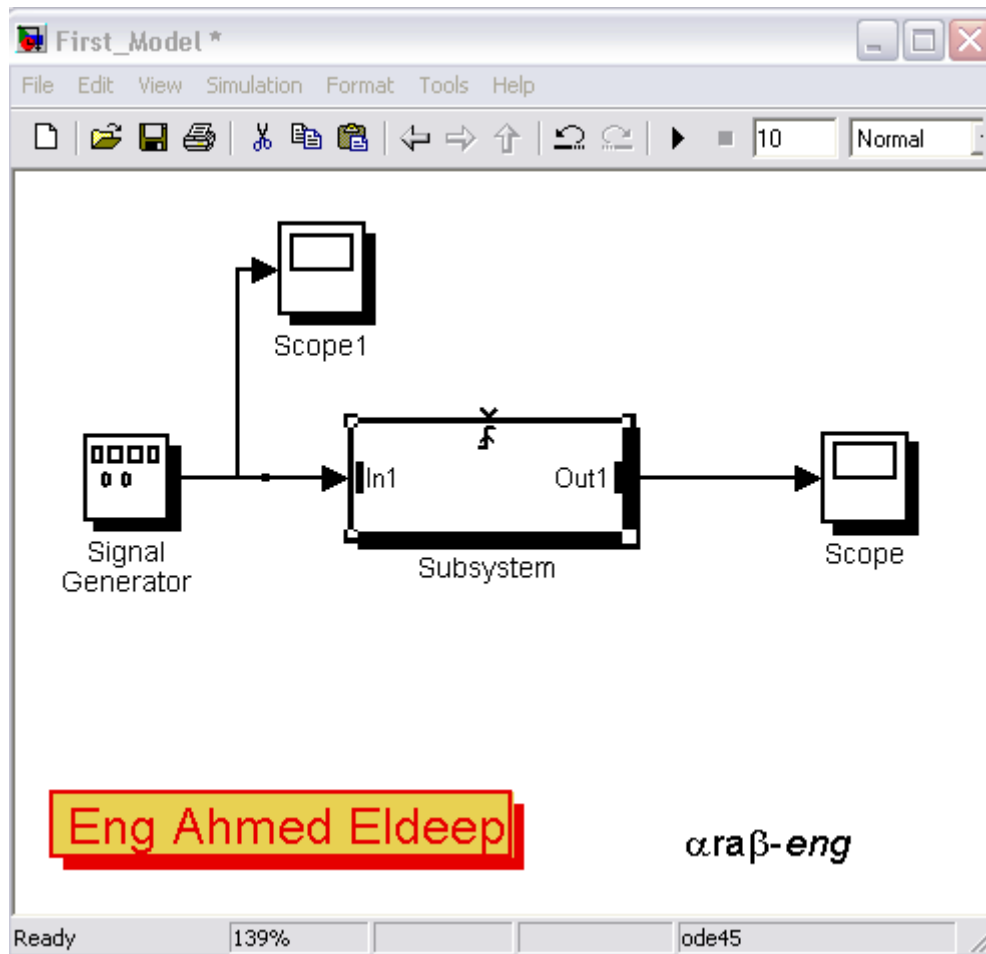
وبتطبيق هذا على المثال الموجد لدينا

قم بالدخول الى قائمة السميولينك ثم الى **subsystems & ports** وقم بسحب بلوك **trigger**

الى النظام الفرعي وسيكون كما يلي :

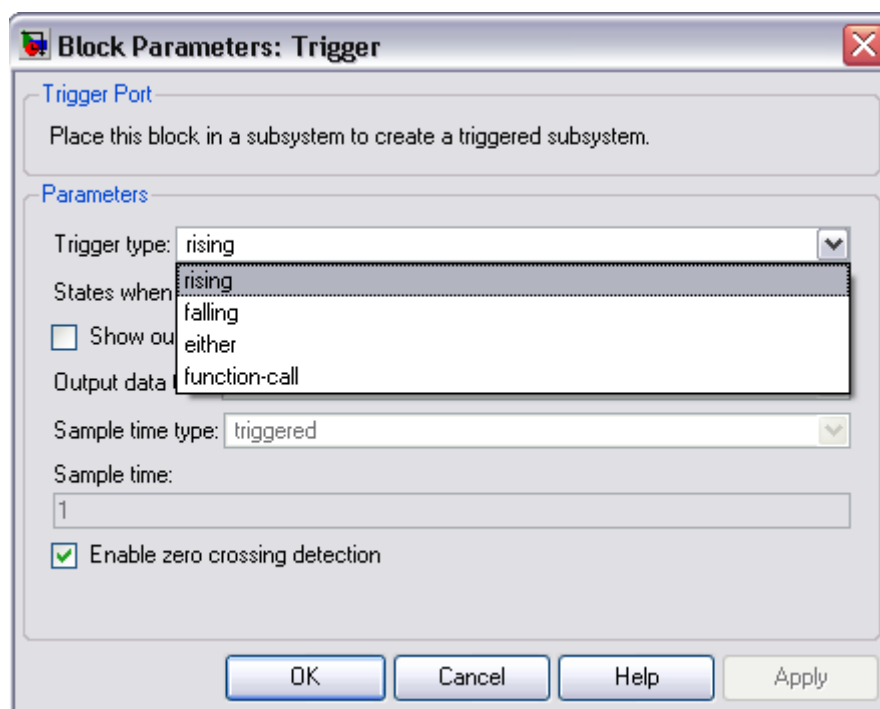


وسيكون شكل النظام الاصلى كما يلى



ولاختيار نوع ال trigger قم بالضغط على البلوك

وستظهر لنا نافذه الاختيارات يمكننا منها اختيار نوعه كما فى الرسم التالى :



ونلاحظ بعد الاختيار بتغيير شكل البلوك بعد كل اختيار كما يلي :

في حالة الارتفاع



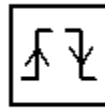
Trigger

وفي حالة الانخفاض



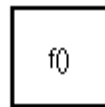
Trigger

اما في كلتا الحالتين



Trigger

وفي حالة استدعاه بأمر



Trigger

ويجب ملاحظة ان triggered subsystems

تختلف عن الانظمة enabled subsystems

حيث انها تقوم بحجز اخر قيمة للخرج ولا يمكننا استعادة الخرج عند قيمة ابتدائية ماعدا الانواع التى يكون استدعاؤها بأمر وهى البلوكات

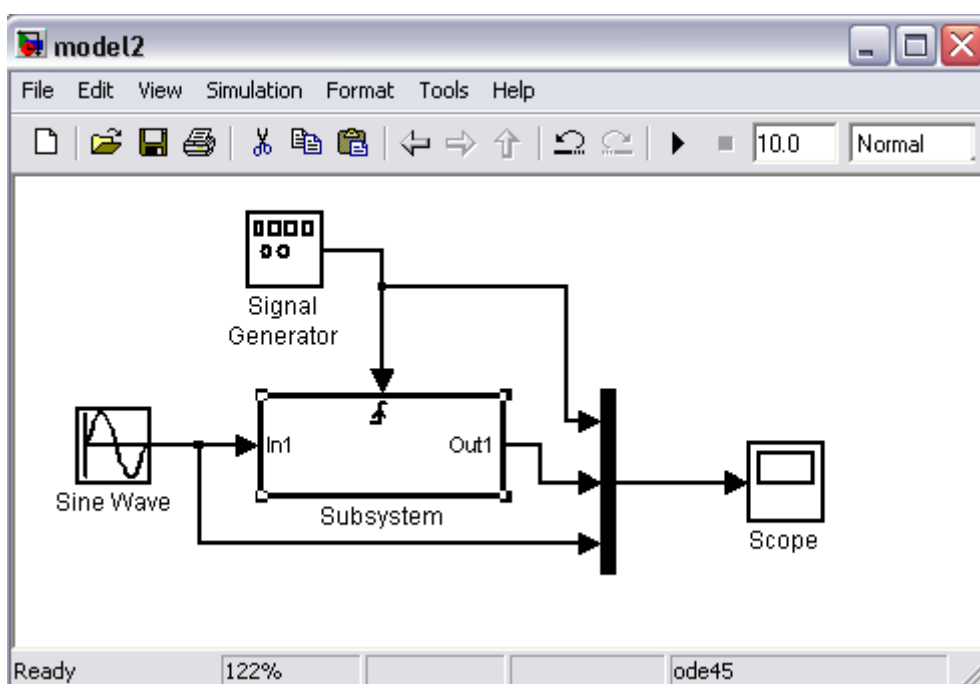
Function-Call Subsystems

ويكون شرط تنفيذ هذه البلوكات مرتبط بأمر منطقى فى ال S-function الخاصة بالبلوك وسوف نتعرض لها لاحقا .

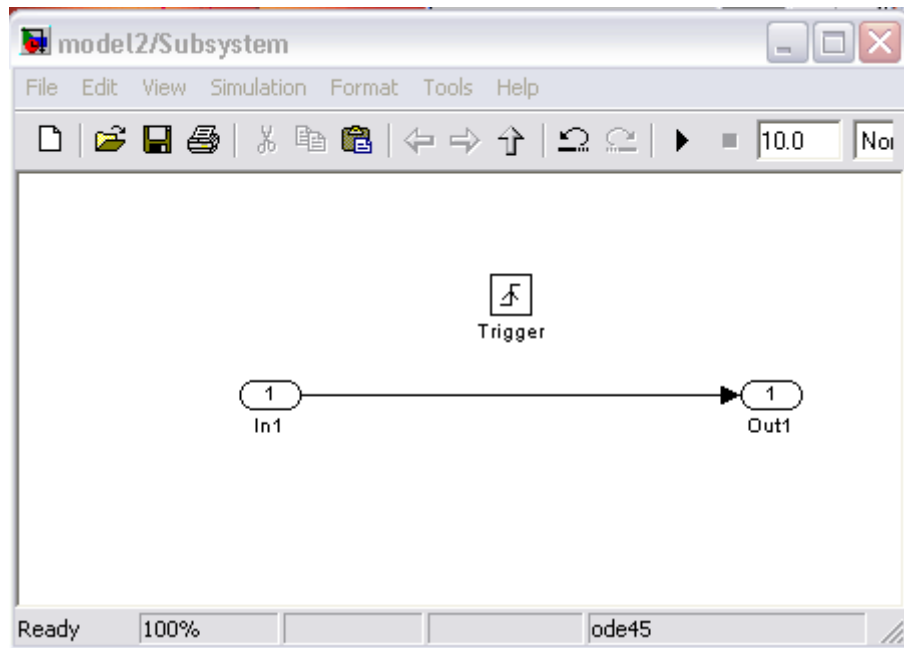
ملحوظة ::

جميع البلوكات فى الانظمة الفرعية التى تستخدم ال Trigger يجب ان يكون زمن تقطيعها متوارث (-1) او مالا نهاية وهذا معناه ان البلوكات الموجودة فى النظام الفرعى يتم تشغيلها عند تشغيل النظام الفرعى اى يجب ان لا يحتوى النظام الفرعى على بلوكات ذات زمن متصل مثل بلوك التكامل.

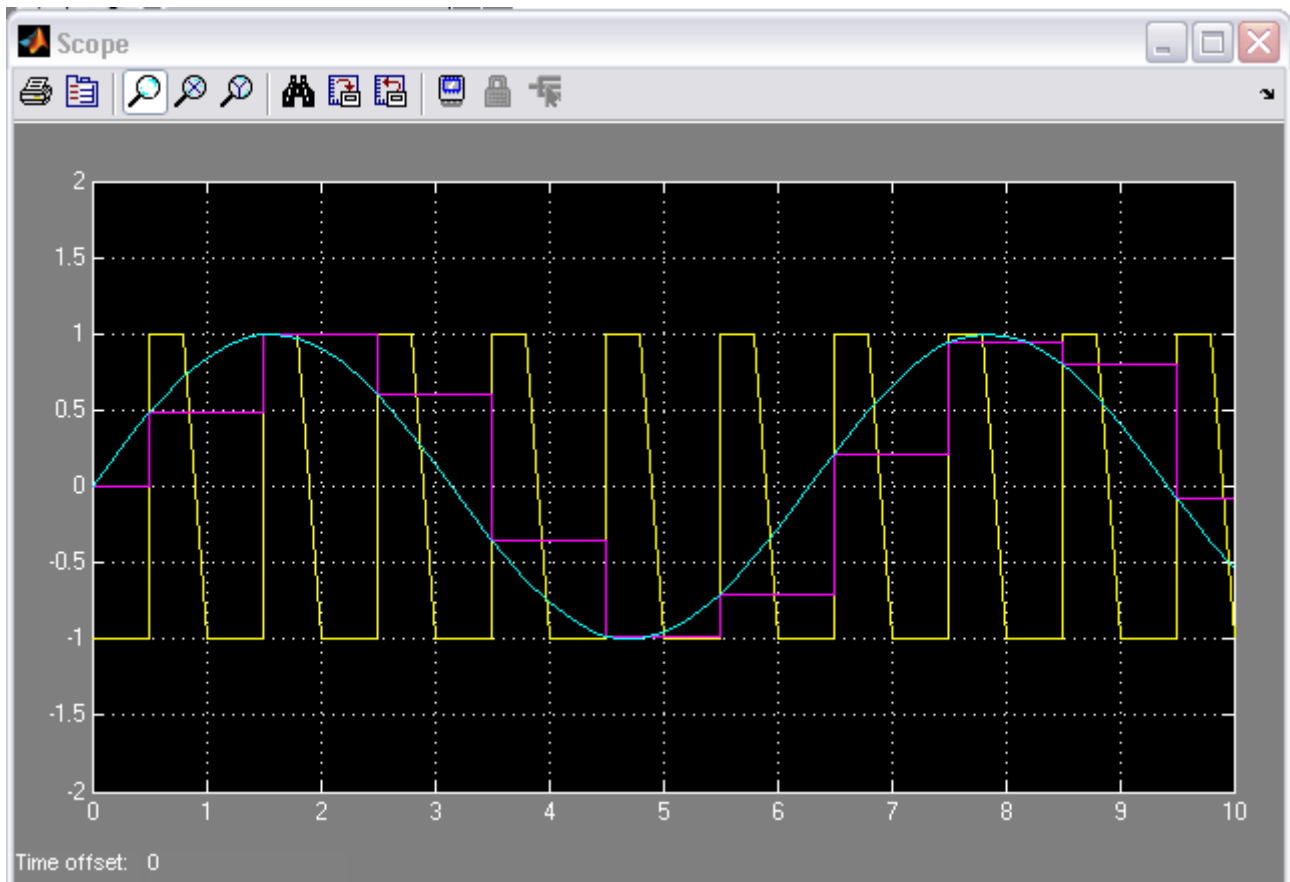
ونقوم الان بعمل هذا المثال البسيط كما يلى ::



والنظام الفرعي



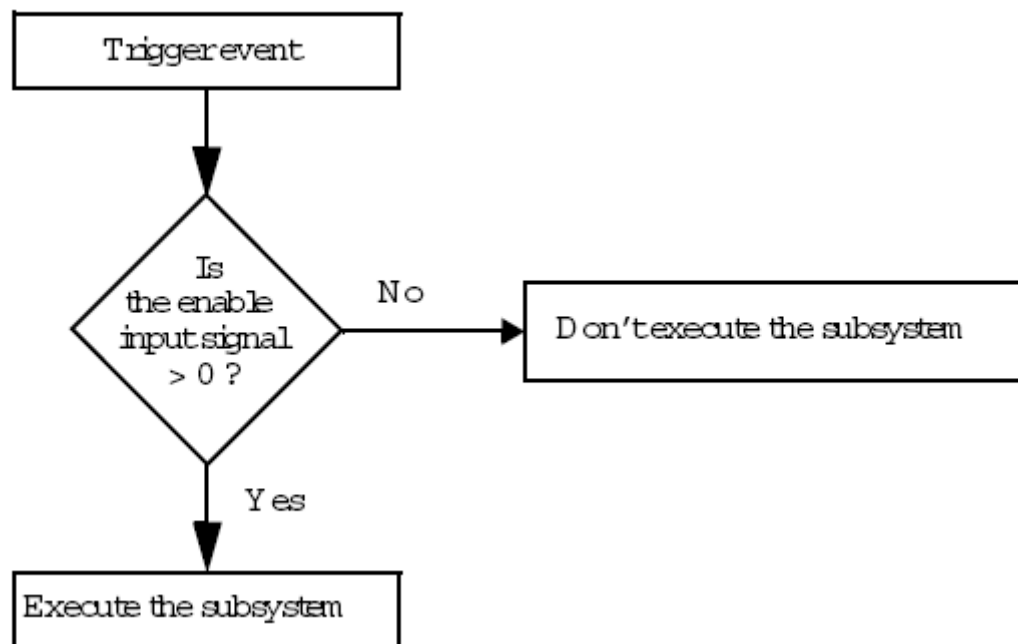
وستكون نتيجة تنفيذ هذا المثال كما يلي :



ومن الشكل السابق نجد الإشارة التي باللون الاصفر ترمز على إشارة التحكم trigger والإشارة التي باللون السماوى هى إشارة الدخل واللون البنفسجى يرمز الى إشارة الخرج ونلاحظ ايضا حدوث تنفيذ البلوك اى حدوث خرج فى حالة ارتفاع إشارة التحكم فقط ثم يتم امساك للقيمة حتى الارتفاع الثانى

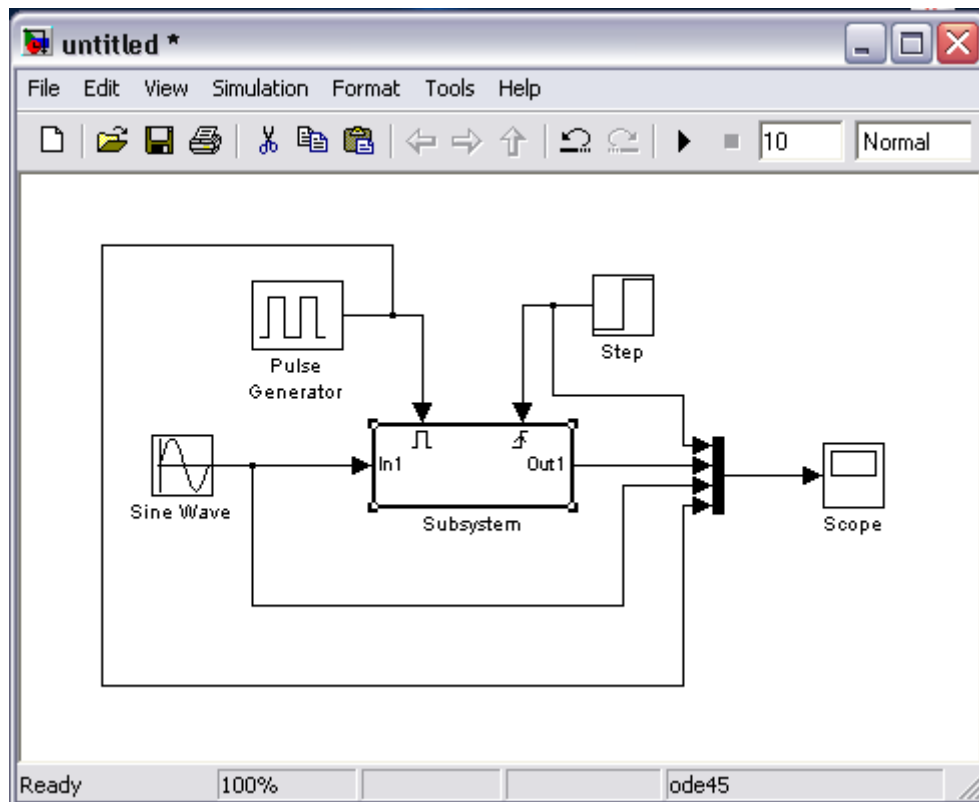
ثالثا :: Triggered and Enabled Subsystems

وهذا النوع الثالث من انواع الانظمة الفرعية وهو يجمع النوعان السابقان حيث يقوم فى البداية بانتظار حدوث ال trigger ثم يقوم بالتعامل معه اذا كانت إشارة التحكم المستخدمة مع الجزء enabled موجودة ويمكن توضيح هذه الخطوات فى الشكل الاتى ::

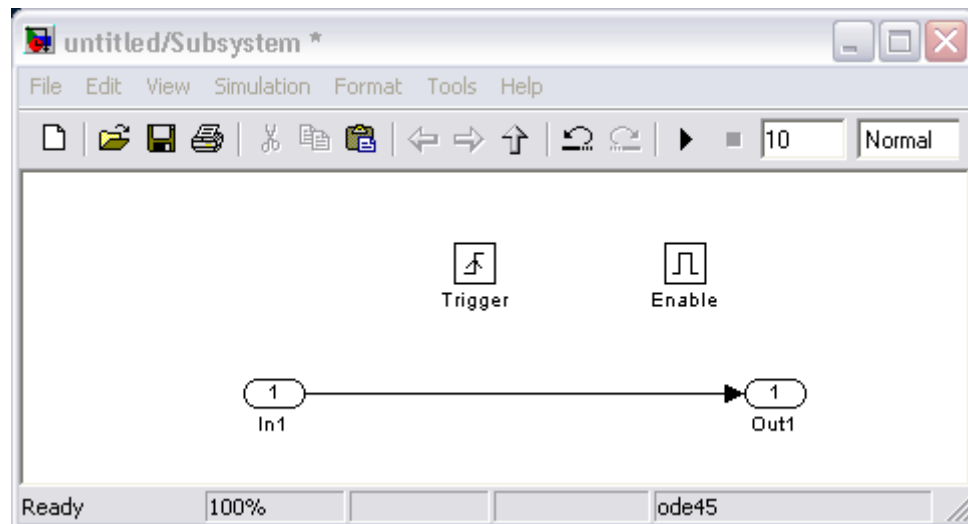


والان سنقوم بعمل مثال بسيط على هذا النوع

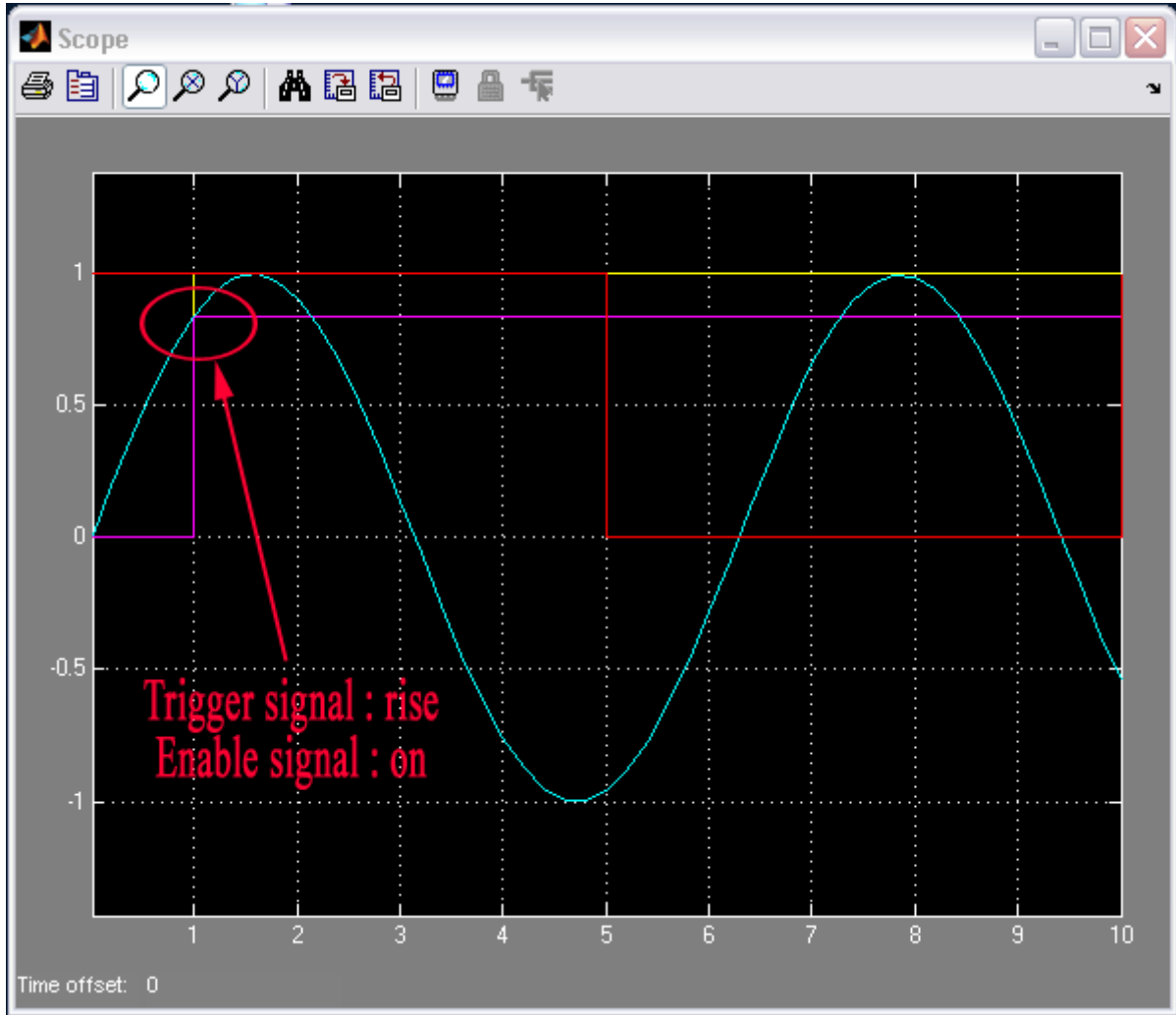
قم بتكوين المثال كما بالرسم الاتي ::



ويكون النظام الفرعي كما في الرسم الاتي



ونتيجة تنفيذ المثال الاتي موجودة في الشكل التالي ::



ومن الشكل السابق نجد ان الإشارة التي باللون السماوي هي إشارة الدخل وهي عبارة عن sine wave اما الإشارة التي باللون الاصفر فهي ال trigger وهي عبارة عن step input وتبدأ عند الزمن واحد ثانية لتصبح قيمتها تساوي واحد وتستمر حتى آخر زمن المحاكاه وهذه الإشارة تستخدم في محاكاة الانظمة الديناميكية والتي تعبر عن تغير مفاجئ في قيم النظام .

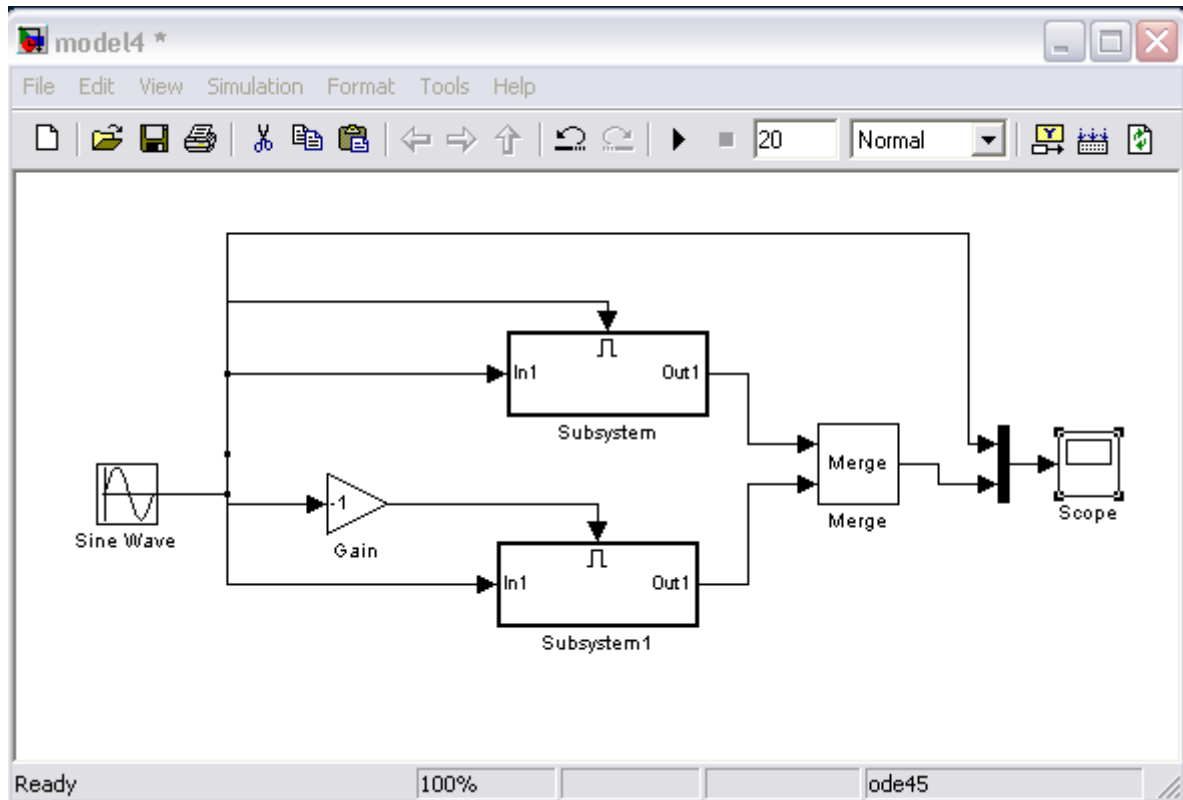
اما الإشارة التي باللون الاحمر فهي عبارة عن pulse مدتها ١٠ ثواني وتكون قيمتها تساوي واحد في ٥٠% من الفترة الزمنية duty cycle وهنا نستخدمها للتعبير عن إشارة التحكم enable

وعند الزمن يساوي واحد يبدأ مخطط تنفيذ النظام عند حدوث ال trigger بالارتفاع وفي هذه النقطة تكون إشارة التحكم موجودة فيحدث تنفيذ للنظام الفرعي وتكون قيمة الخرج مساوية لقيمة ال sine عند الزمن يساوي واحد وبعد ذلك لا نجد ارتفاع اخر لإشارة التحكم trigger ولذلك ستستمر قيمة الخرج ممسكة عند هذه القيمة .

Subsystems Creating Alternately Executing

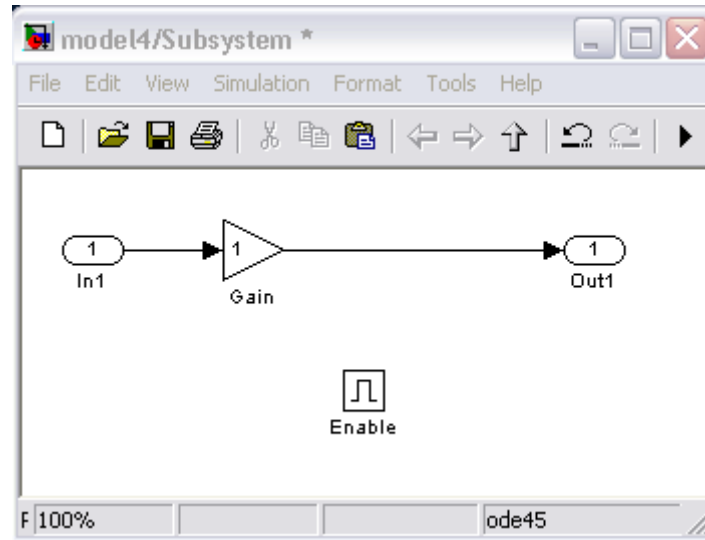
يمكننا استخدام الانظمة الفرعية المتوقف تنفيذها على اشارة تحكم مع بلوك **merge** الذى يستخدم لدمج اشارتين معا لعمل انظمة فرعية يتم تنفيذها بالتبادل ويتم اعتمادها على حالة الموديل

والان سنقوم بتطبيق المثال الاتى وهو موديل لعمل **full-wave rectifier** والتى تستخدم فى تحويل التيار المتردد الى تيار مستمر :

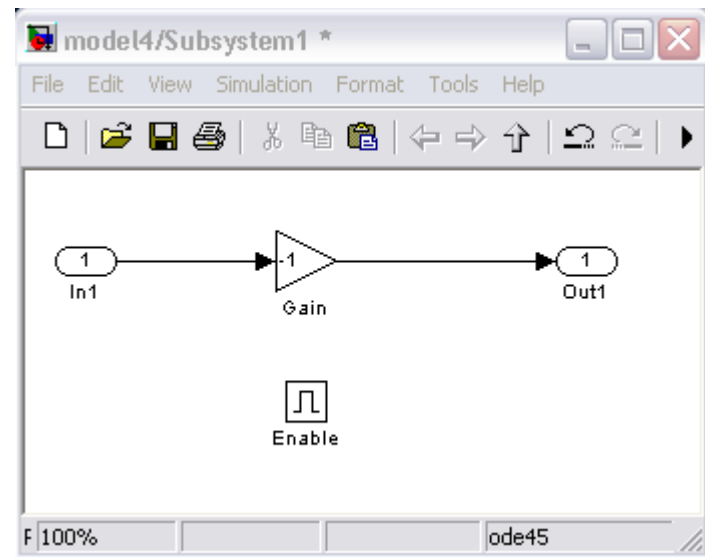


ويتكون من نظامان فرعيان كما يلي

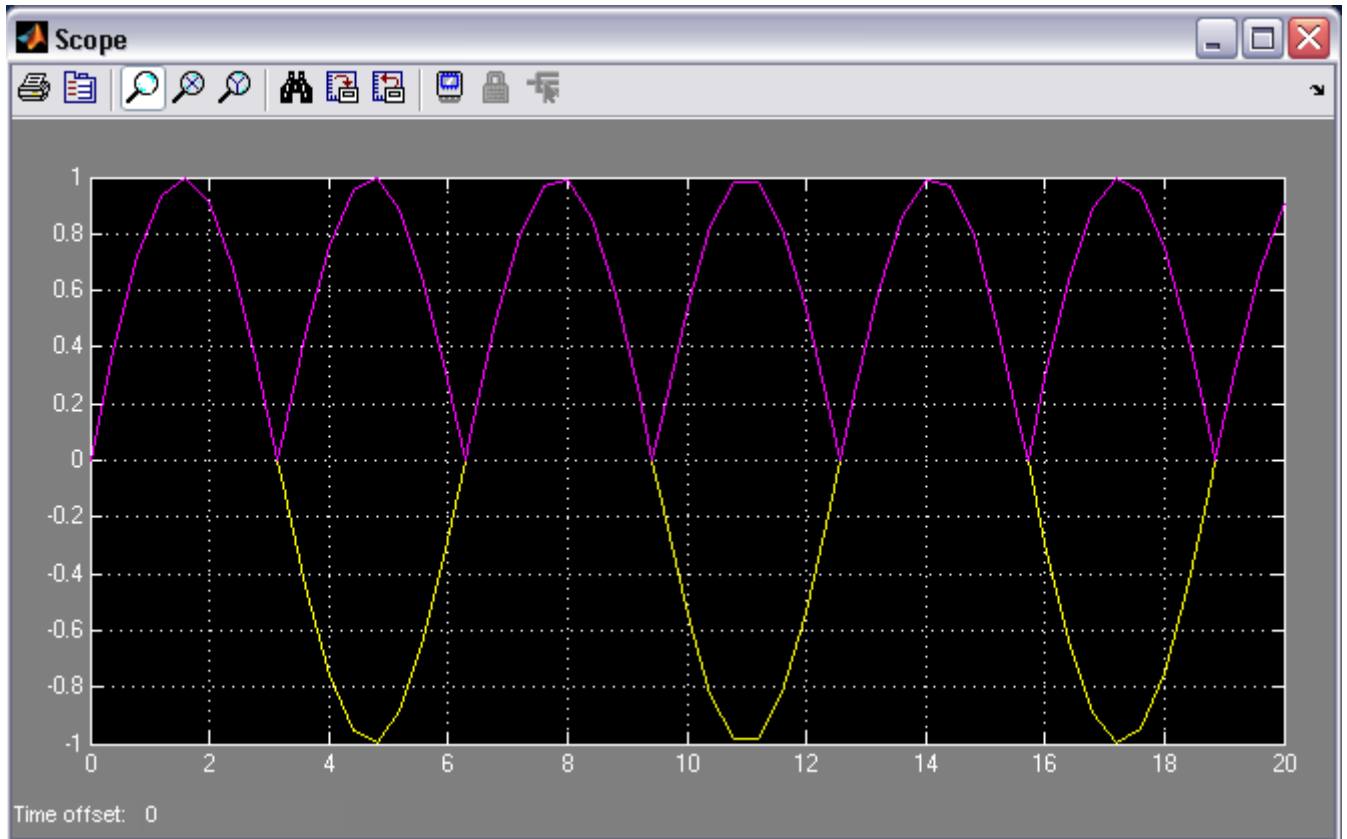
النظام الفرعى الاول



النظام الفرعى الثانى



وتكون نتيجة التنفيذ كما يلي ::

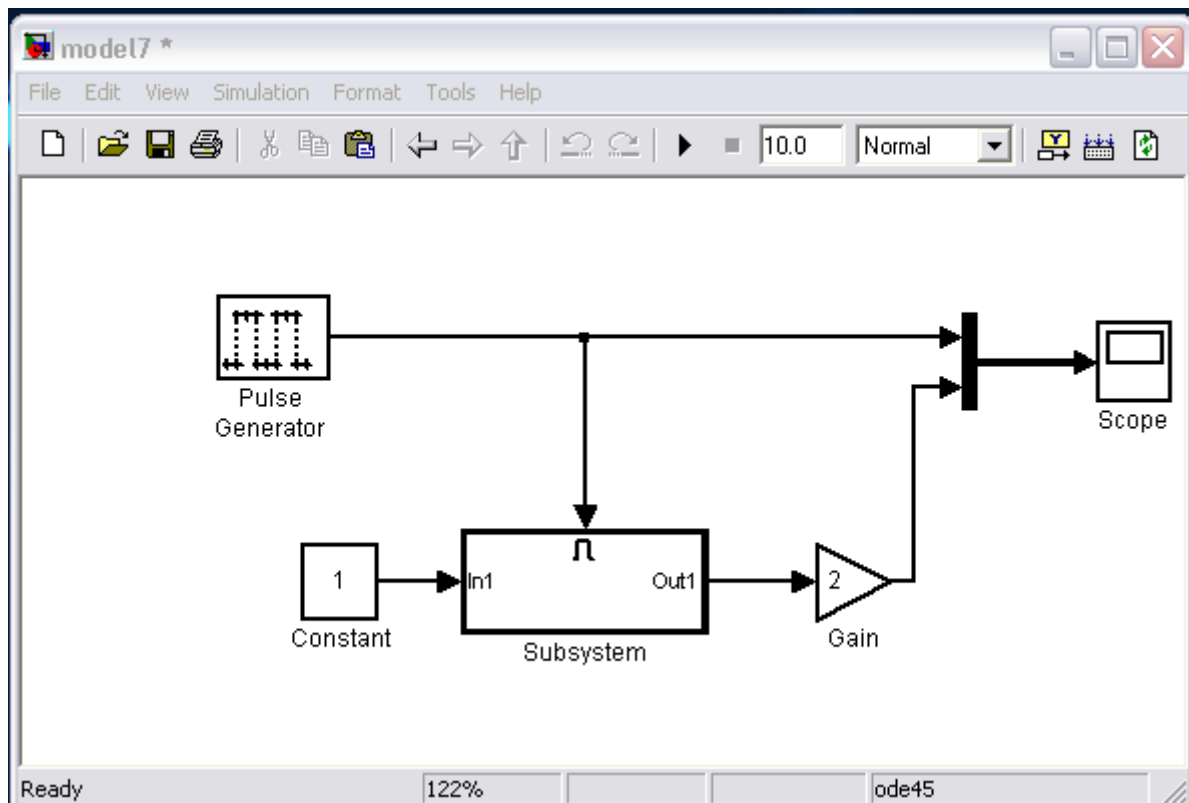


Behavior Conditional Execution

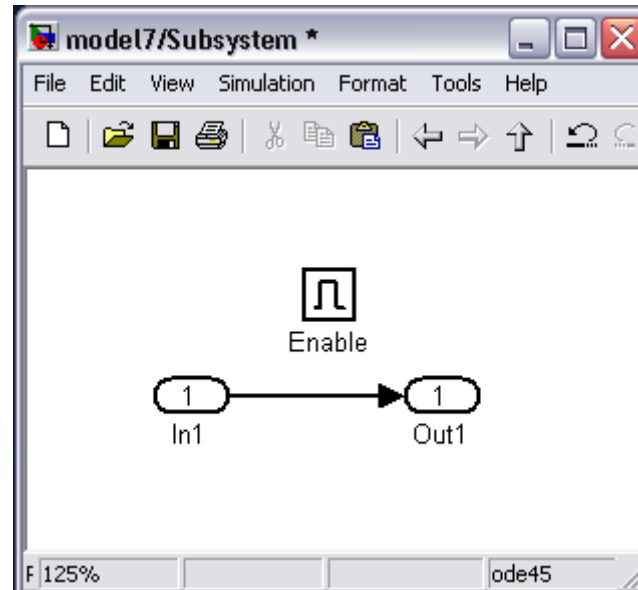
يقوم السميولينك بتجنب البلوكات الغير ضرورية اثناء عملية تنفيذ المحاكاة ومن امثلة هذه البلوكات **Switch** او **Multiport Switch** وهما يستخدمان في تحويل الاشارات وايضا من البلوكات التي من الممكن تكون غير ضرورية الانظمة الفرعية المرتبطة بشرط لتنفيذها وهذا يسمى

conditional execution behavior. (CE)

وفي المثال الاتي سوف نقوم بمعرفة هذه الخاصية :



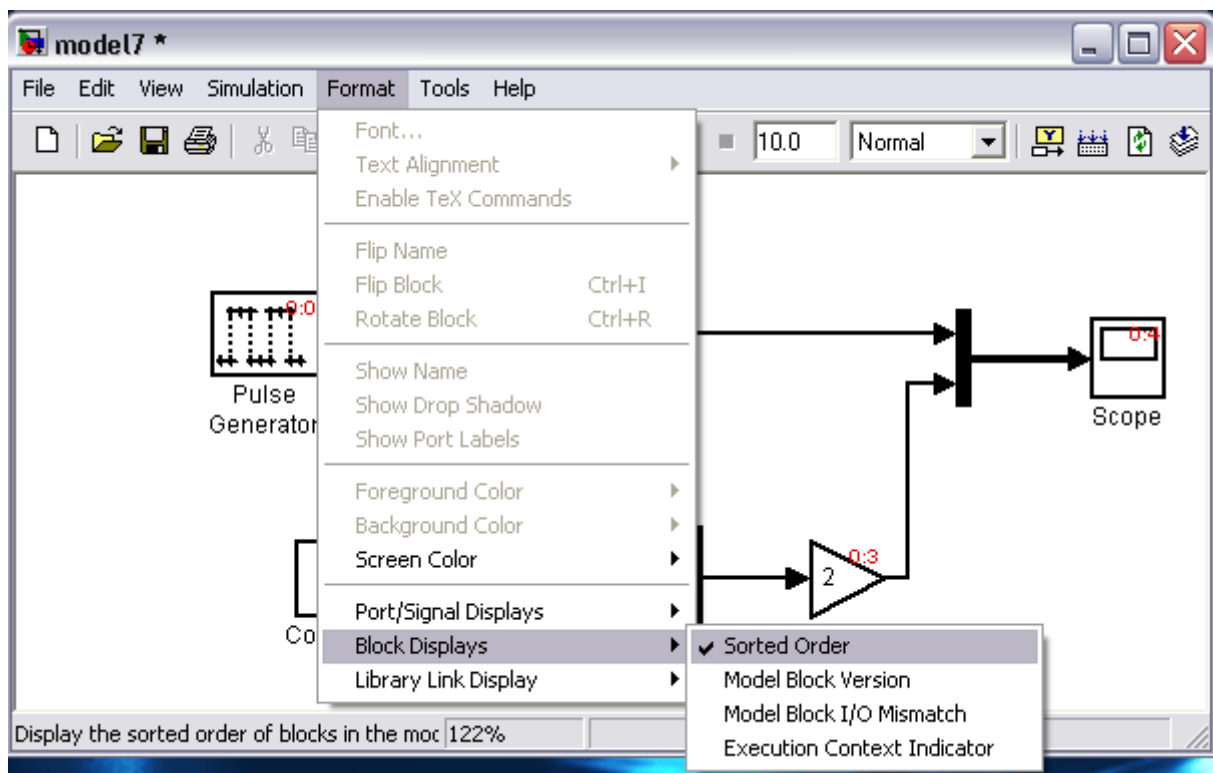
والنظام الفرعي كما في الصورة الاتية :



وقبل عمل ال **simulation** قم بالدخول الى قائمة **format** ثم اختيار

displays Block

ثم **order sorted** كما في الصورة الاتية ::

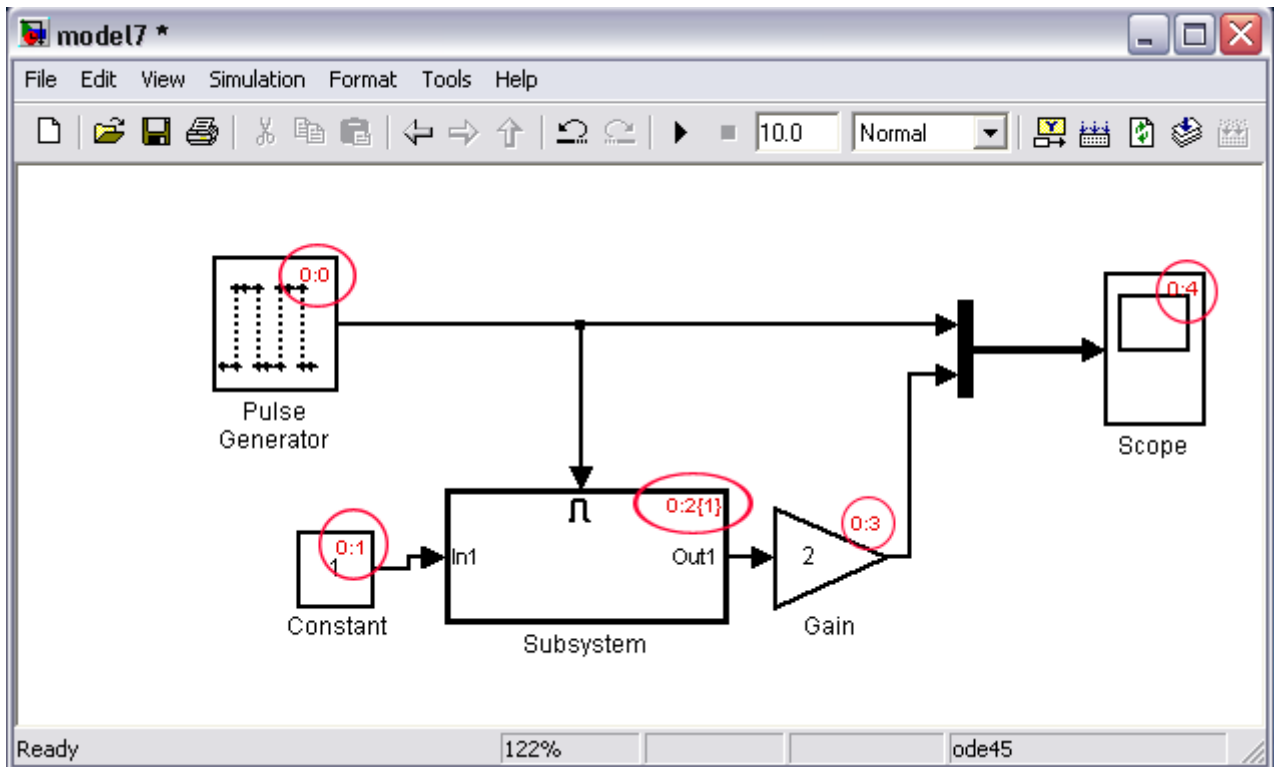


وبعد عمل ال **simulation** نلاحظ وجود ارقام فوق كل بلوك وهذه الارقام تكون على الصورة الاتية :

X:Y

حيث يعبر عن **X** على انها الطبقة **layer** الموجود فيها البلوك فكل البلوكات الموجودة فى الطبقة الاساسية يكون لها **X=0** ال

اما **Y** فترمز الى ترتيب تنفيذ البلوك ففى الشكل الاتى نجد :



بلوكات ال

generator Pulse

و **constant** و **subsystem** و **gain** و **scope** تحتوى على رقم **X** يساوى الى صفر اى انها تنتمى الى الطبقة الاساسية فى الموديل ويكون ترتيب تنفيذها وهو الرقم **Y** كما يلى

generator Pulse -١

constant -٢

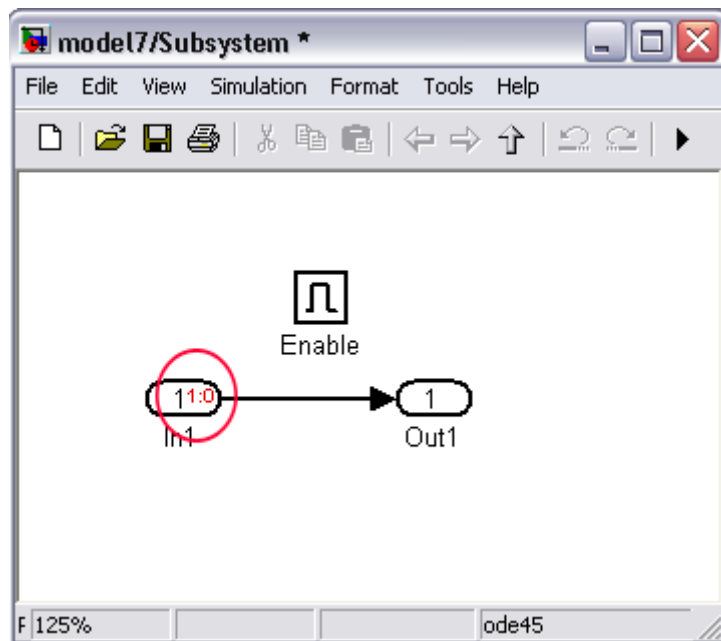
subsystem -٣

gain -٤

scope -٥

ونلاحظ هنا ان بلوك **subsystem** عليه الاتى ١:٠ {١} وهذا معناه انه يحتوى على الطبقة الاولى وهى تاتى بعد الطبقة الاساسية

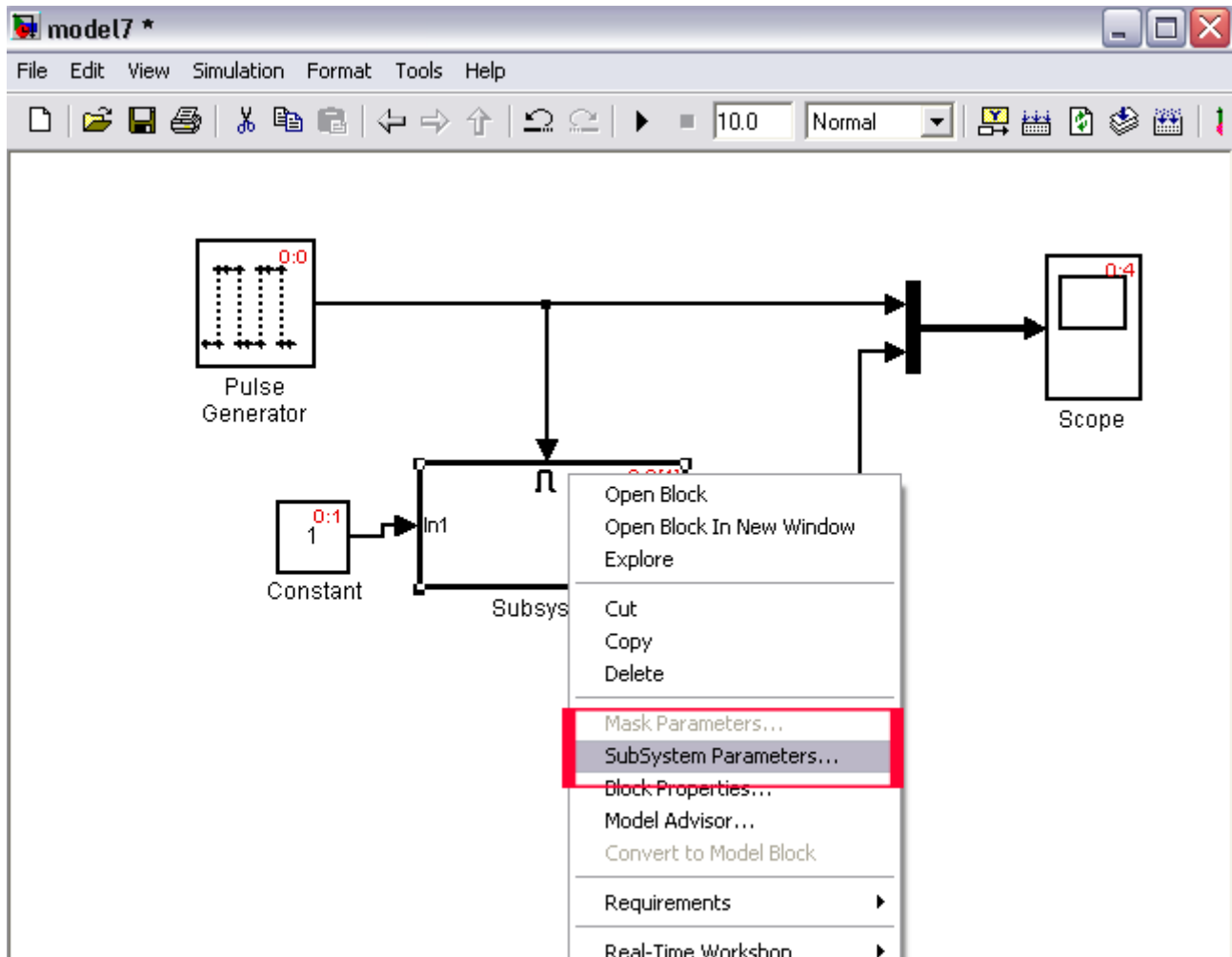
وبالدخول الى النظام الفرعى نجد الاتى ::



ونجد بلوك الادخال يحتوى على الرقم الاتى ١:٠ وهذا معناه انه موجود فى الطبقة الاولى و سيكون تنفيذه فى البداية وبلوك الاخراج لا يحتوى على اى ارقام لانه لم يتم تنفيذه لعدم وجود ما يتم اخراجه

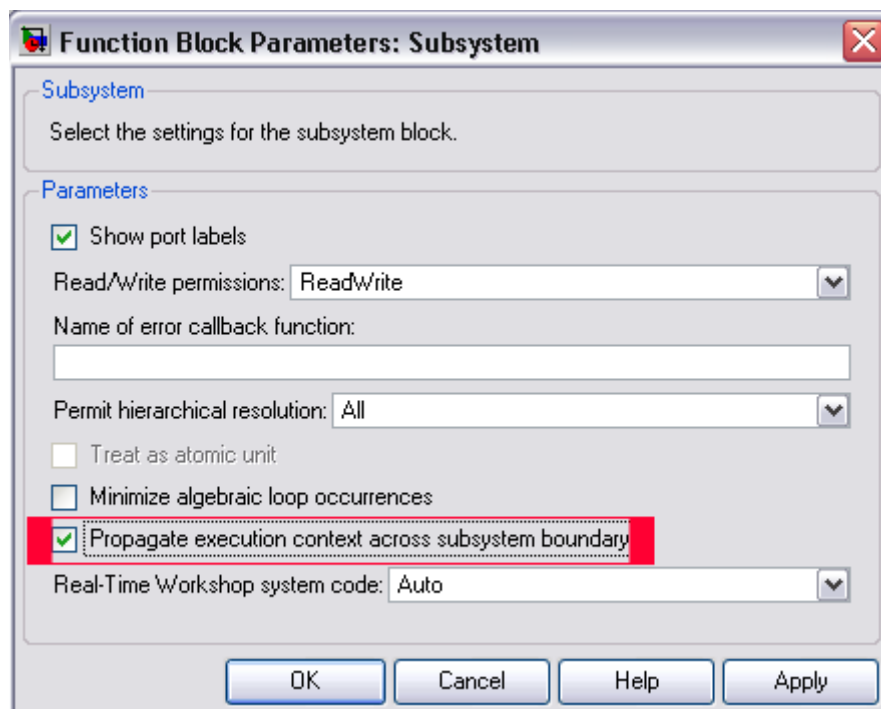
والان قم بالضغط كليك يمين على بلوك **subsystem** واختيار

parameters Subsystem



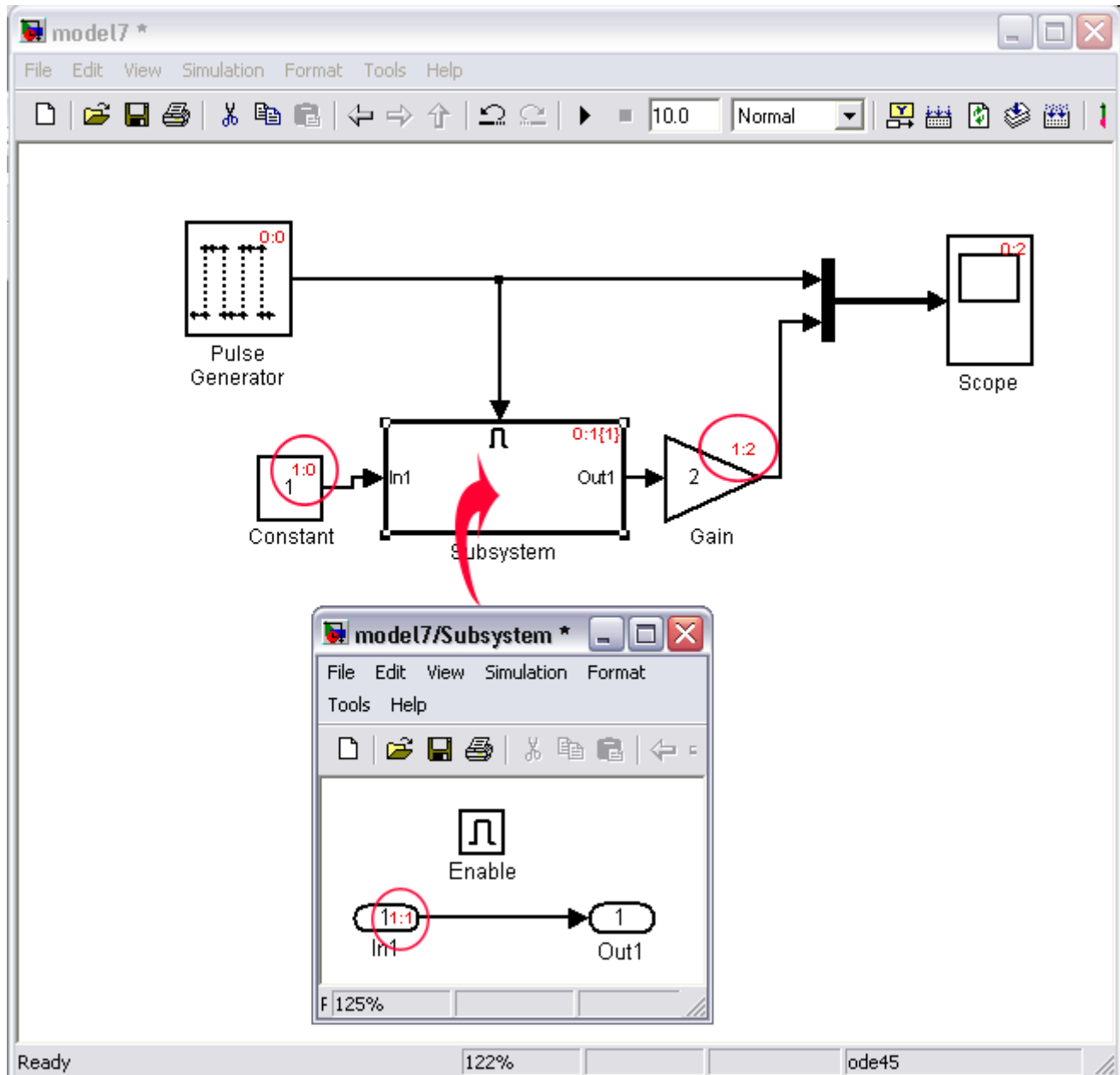
وعندما تفتح لك نافذه الاختيار قم بوضع علامة صح حول

across subsystem boundary Propagate execution context



وهذا فائدته ان يقوم بجعل تنفيذ البلوكات الموجودة حول النظام الفرعى مرتبط بتنفيذ النظام الفرعى وهذا يعمل على زيادة سرعة المحاكاة مثلا لدينا بلوك gain لن نحتاجه الا عندما يتم تنفيذ النظام الفرعى وعند ايقاف النظام الفرعى غير مهم تنفيذ هذا البلوك

والان قم بعمل المحاكاة مرة اخرى والان سنلاحظ الارقام على البلوكات :



والان اصبحت البلوك **constant** والبلوك **gain** ينفذان مع النظام الفرعي فقط على الرغم انهم خارج النظام الفرعي
وموجودين في النظام الاساسي ويكون ترتيب التنفيذ كالآتي

١- **constant**

٢- **subsystem**

٣- **gain**

وهذا ما يعرف بـ **Conditional Execution Behavior** او

behavior CE

Contexts Propagating Execution

عموما برنامج السميوليك يقوم بتعريف ما يسمى بى **execution context** وهذا عبارة عن مجموعة من البلوكات التى تنفذ مرة واحدة خلال زمن معالجة النموذج .

ويقوم ايضا بعمل **execution context** للنظام الاساسى وايضا لكل نظام فرعى .
واى **execution context** هو مجموعة البلوكات التى يحتويها النظام

وعند تنفيذ المحاكاة فان السميوليك يقوم بالبحث عن الشروط الاتية فى كل بلوك:

١- هل خرج البلوك المطلوب فقط لتنفيذ نظام فرعى او دخله يتغير فقط نتيجة تنفيذ نظام فرعى

٢- هل هذا البلوك مرتبط فى تنفيذه مع نظام فرعى وهذه الحالة عندما يكون هذا البلوك على حدود النظام الفرعى

٣- هل خرج البلوك ليس نقطة اختبار **testpoint** وسوف نعرف ما هى نقط الاختبار لاحقا

٤- هل مسموح للبلوك ان يتوارث **execution context**

٥- هل البلوك ليس من النوع **multirate block**

٦- هل زمن التقطيع **sample time** متوارث او ثابت

واذا فعلا وجد ان هذه الشروط محققة ومسموح ان يتم توليد **execution context**

فان السميوليك يقوم بنقل البلوك الى **execution context** الخاص بالنظام الفرعى وهذا للتأكد ان هذا البلوك يتم تنفيذه فقط عندما يتم تنفيذ النظام الفرعى

والتعامل مع بلوك Switch

يقوم السميوليك بمعالجة كل فرع من دخل البلوك بصورة مخفية ويكون لكل فرع

execution context

الخاص به والذى يتم تفعيله عندما يتم اختيار تنفيذ الفرع وعموما فان كل فرع من هذا البلوك يتم تنفيذه فقط عندما يتم اختياره كدخل .

ملحوظة :: لالغاء خاصية conditional execution behavior

قم بالدخول ال Configuration Parameters ثم Optimization وضع علامة صح عند Conditional input branch execution

ملحوظة :: هناك بعض الحالات لا يستطيع البرنامج استخدام خاصية

boundary Propagate execution context across subsystem

وهي عندما يكون النظام الفرعي من النوع triggered subsystem او النظام الفرعي له اكثر من مدخل او مخرج ولها شرط ابتدائي غير الصفر وفي هذه الحالة البلوك الموجود في خرج النظام الفرعي لا يمكنه ان يتوارث execution context subsystem's

ويكمننا اظهار مستطيلات حول المداخل التي لا يستخدم خاصية

Propagate

context execution

وهذا عن طريق الضغط اختيار قائمة format ثم Block Displays ثم

Execution Context Indicator

ويكون شكلها كما يلي

