Lemma 1 Ltd.
c/o Interglossa
2nd Floor
31A Chain St.
Reading
Berks
RG1 2HX

ClawZ
—
Diagram Translator Test Cases

Version:	10.6
Date:	20 June 2003
Reference:	DAZ/INT517
Pages:	54
Prepared by:	R.B.Jones
Tel:	+44 1344 642507
E-Mail:	rbjones@rbjones.com

0 DOCUMENT CONTROL

0.1 Contents

0.2 List of Figures

0.3 Document Cross References

[1] LEMMA1/DAZ/ZED504. *ClawZ - Model Translator Specification.* R.B. Jones, Lemma 1 Ltd., `rbjones@rbjones.com`.

[2] LEMMA1/DAZ/ZED507. *ClawZ - Extending the Z Library.* R.B. Jones, Lemma 1 Ltd., `rbjones@rbjones.com`.

0.4 Changes History

Issue 2.1 First issue.

Issue 5.1 (Phase 3) Final issue to DERA. The integration tests are enhanced to provide testing of the new features in the translator introduced in Phase 3. The DERA Bedford "control" model has been added as a supplementary integration test (it is expected to translate but not to type-check). Tests of the .M file translator have been added.

Issue 6.1 (Real Clawz first delivery) Tests enhanced to cover translation of Matlab .M files.

Issue 7.1 (Real Clawz final delivery) Tests enhanced to cover expression translation and other enhancements.

Issue 7.3 (Clawz extensions final delivery) The document has been simplified and more diagrams have been added.

Issue 7.4 (Clawz extensions II first delivery) Rework of tests to use Mux/Demux synthesis with injected port type information where necessary. Additional tests for improvements to signal inference and synthesis.

Issue 10.2 (Clawz extensions II final delivery) Addition of tests for virtualization simplifications and for virtual loop handling to int517r. Addition of test int517t, which is a Mux/Demux virtualization test from QuinetiQ.

Issue 10.4 JANUARY 2003 - Support for action subsystems. Addition of tests int517u (action subsystems) and int517v (action subsystem errors). Insertion of action subsystems into int517l, int517m, including covering QuinetiQ eg6.

Issue 10.5 MAY 2003 - Support for action subsystems merge synthesis. Updates to tests int517l.mdl, int517m.mdl, int517u.mdl, int517v.mdl.

Issue 10.6 JUNE 2003 - Support for block references to artificial subsystems of libraries. Updates to run scripts for tests int517l.mdl and int517m.mdl. Change to int517v.mdl.

0.5 Changes Forecast

None.

1 GENERAL

1.1 Introduction

This document is one of the deliverables from the Control Law project, placed by DERA Malvern with Lemma1 Ltd. and various subsequent contracts enhancing and extending the resulting ClawZ tool.

1.2 Notation and Conventions

The integration tests in this document consist of Simulink models to be translated by the Control Law diagram translator.

The main purpose of the document is to provide access to the diagrams for the models independent of access the capability to run Simulink, and to describe the purpose and scope of the various tests. Though originally the simulink models were contained in this document and extracted from it, this arrangement proved practically cumbersome and the models are now held as separate ascii files.

2 OVERVIEW

2.1 Testing Methods

The integration testing of the translator is undertaken in the following manner:

Preparation of Models First a small number of Simulink models are prepared or selected for use in the integration tests. For each of the models diagrams are saves as encapsulated postscript from Simulink. The models are also saved in simulink model file format (.mdl). Some of the diagrams are presented in this document. All diagrams and models are available in the project RCS repository.

Translation of Models The integration test makefile feeds the various matlab .m files and simulink .mdl files, together with appropriate library metadata files into the translator. ClawZ is then used to process the files and produce ProofPower-Z specifications suitable for processing by ProofPower.

Automatic Checking of the Translations Two automatic checks are performed on the output of the translator.

Firstly there is a facility in the makefile to perform diffs comparing the output with that of previous runs. This is particularly useful to check that a small change has not had unintended consquences.

Secondly, the output of ClawZ can be automatically type-checked by ProofPower, in the context of appropriate libraries. Some of the tests now involve loading one or more translated .m file or libraries to provide the context for checking the actual model.

Further Assessment of the Translations Once the output from the tool has been shown to be type correct, a further stage of assessment is undertaken, firstly to ensure that the output of the translator conforms to the specification of the translator in [1], and then to assess it in the more general way required for the final report.

Production of Report The assessments are documented in a report which describes the results of the system tests and also includes an assessment of the capabilities and limitations of the translator.

3 MODEL FILES

The following model files are used in the tests:

alfs3v2.mdl	A second version of Alf's model as built using Simulink by Alf.
digclock.mdl	An example used in the User Guide.
fcn_eg.mdl	An example involving an *Fcn* block used in the User Guide.
int517a.mdl	Our own first transcription of Alf's model from a paper copy into Simulink.
int517b.mdl	The digital adder. This is a primarily oriented to testing nested subsystems. It has been adapted to test the output direction facility.
int517c.mdl	The F14 model.
int517d.mdl	The Abs Brakes example.
int517e.mdl	A version of Alf's model edited to test scalar literals.
int517f.mdl	This is a large model previously known as the DERA Bedford control model.
int517g.mdl	A small model introduced for testing vector parameters, and tolerance of one-ended lines.
int517k.mdl	Special model primarily for testing *Fcn* parameter expressions and matlab expressions as block parameters.
int517m.mdl	A model testing library reference.
int517o.mdl	A model supplied by QinetiQ primarily for illustrating their use of libraries and block reference.
int517p.mdl	A small model used to isolate problems during early testing, this is edited as needed and not intended as a stable test.
int517q.mdl	A small model intended to test the name/path translation facilities and the use of matlab variable types in signal inference.
int517r.mdl	A QuinetiQ example concerning propagation of signal names, subsequently augmented for virtaulization.
int517s.mdl	A QuinetiQ example concerning effects of angled brackets in signal names on bus selection.

int517t.mdl	A QuinetiQ example (*clawz112_issue_1*) concerned with ordering of inputs to virtualized Mux blocks.
int517u.mdl	A QuinetiQ example (*eg5*) used to agree details of implementing support for action subsystems.
int517v.mdl	A model for testing the error checks associated with action subsystems, also for errors on artificial subsystems and block references.
int517w.mdl	A model combining two QuinetiQ action subsystem examples (*eg1* and *eg3*).
unitdelay.mdl	An example involving a unitdelay used in the User Guide.

4 LIBRARY FILES

The following library files are used in the tests:

int517l.mdl	A library file which constructed for testing library translation and block reference, and also for testing masked subsystems and artificial subsystems of libraries.
int517n.mdl	A library file supplied by QuinetiQ and referred to by int517o.mdl.

5 .M FILES

The following .m files are used in the tests:

int517i.m	This .m file serves primarily to define the external variables required by the Abs Brakes model (int517d.mdl), but also exercises various features of matlab expressions.
int517j.m	This .m file tests matlab expressions in .m files and defines certain variables which are used in the model int517k.mdl (which exercises some of the special cases identified by DERA as used in their applications).
int517z.m	The is adapted from an example provided by DERA.
mfile_eg.m	To illustrate the use of a translated .M file in conjunction with a Simulink model, a very simple model (fcn_eg.mdl) has been constructed which uses a variable defined in this .M file. The .M file defines the variable *factor* that is used in the parameter expression of the Fcn block. Custom schemas implementing this block and the Display block are given in [2].

6 TESTS

Tests are run with composite metadata files obtained by extracting metadata from one or more files and concatenating the results.

Each test corresponds to one run of ClawZ and hence processes one .mdl file. Each .m file is also translated in a separate run of ClawZ.

When type checking the results it is sometimes necessary to load more than one file, for example when a model depends upon definitions supplied in a .m file, or when it referrs to blocks in a library.

Details of which metadata is used for each test and which combination of files is used for type checking should be sought in the make file.

7 MODELS - NOTES AND DIAGRAMS

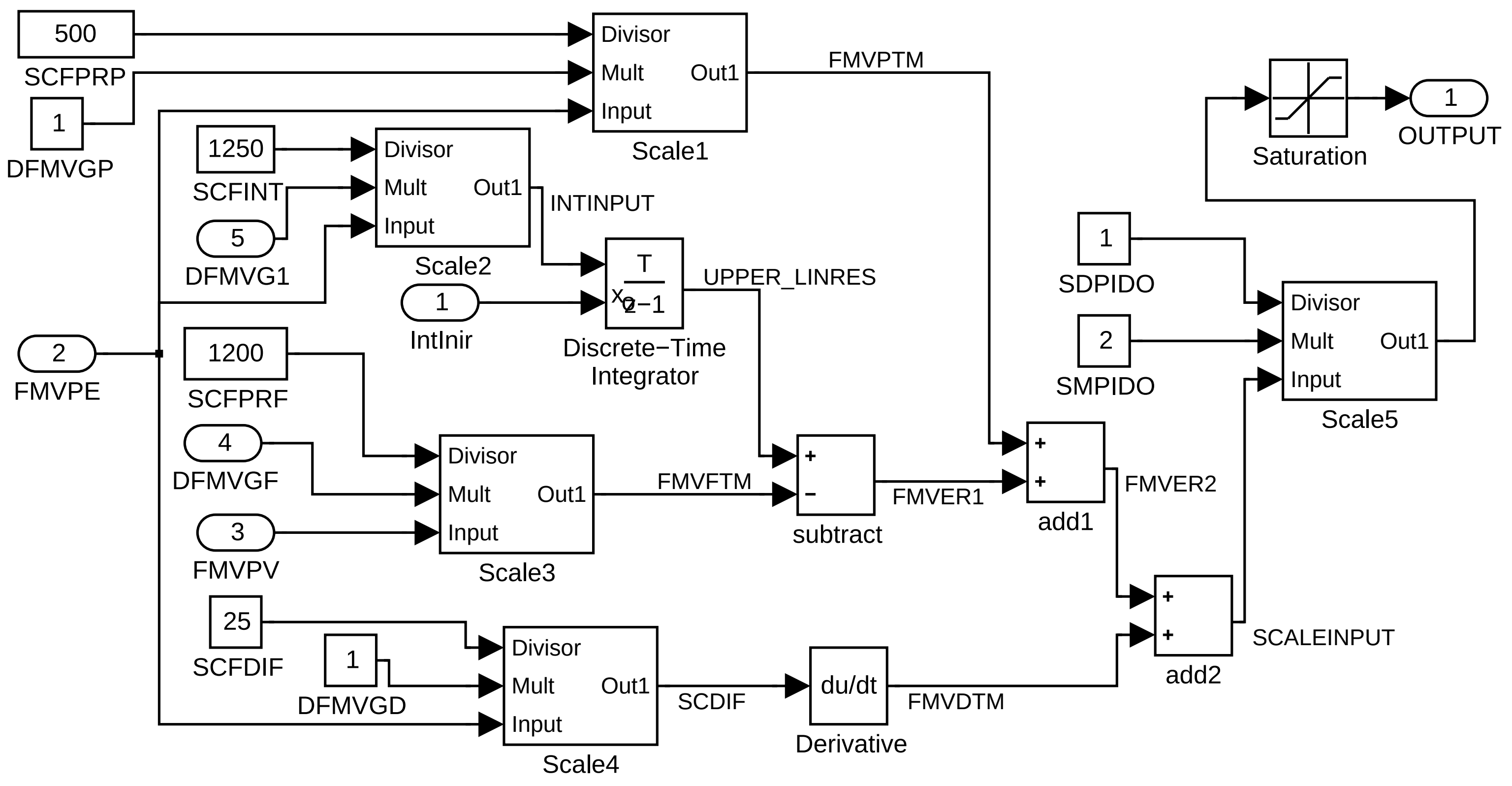
7.1 Alf's Model - RBJ version (int517a)

There are now several versions of Alf's model in use.

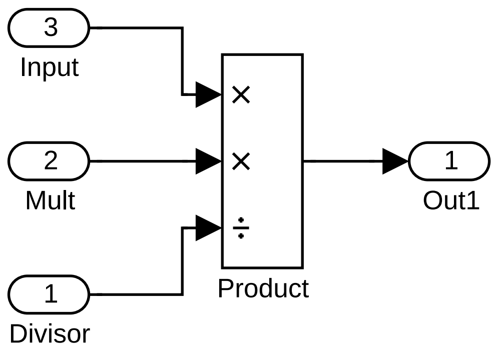
This version was constructed by Roger Jones before Alf did his own Simulink model, and is intended more as a sensible test of the translator than as an accurate Simlink rendition of the original.

The main respects in which it differs from Alf's model are:

1. The discrete integrator in the Simulink library is used. This is less complex than that defined by Alf and also takes some values (e.g. limit values) as parameters rather than inputs.
2. The continuous differentiator from the Simulink library has been used, because no discrete differentiator is provided. It is proposed to translate this *as if it were* a discrete integrator.
3. I originally designed a subsystem to do the scale function and then realised that a block was available. Since this was the only subsystem I decided to leave it as a trivial subsystem, just to make it a slightly better test of the translator.



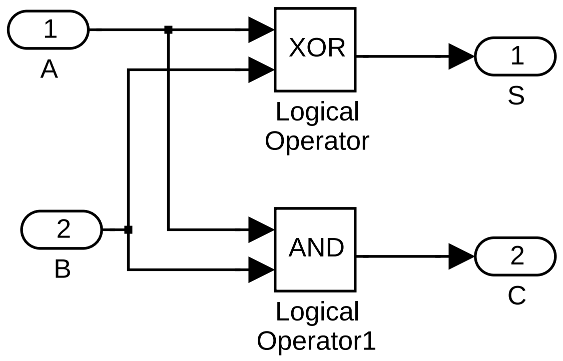
Variant of Alf's Model (int517a1.ep)



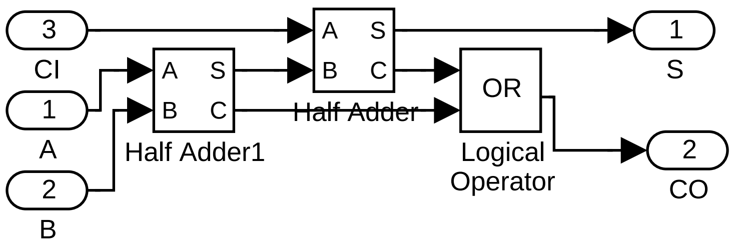
Scale subsystem (int517a2.eps)

7.2 Digital Addition Model (int517b)

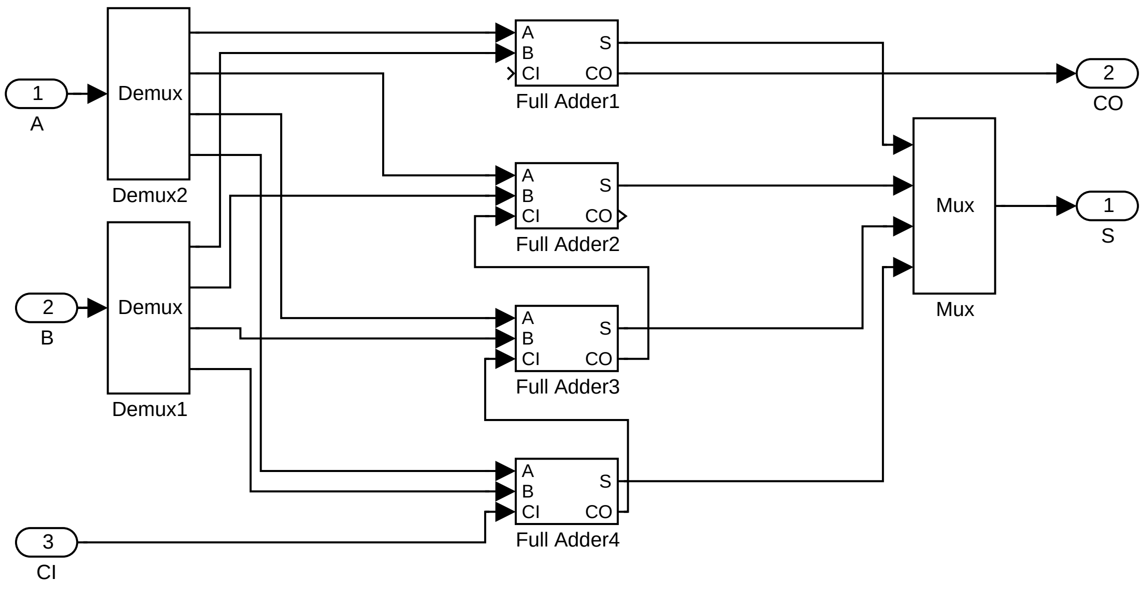
This example is mainly for testing depth and complexity of the diagrams. The library blocks used are straightforward.



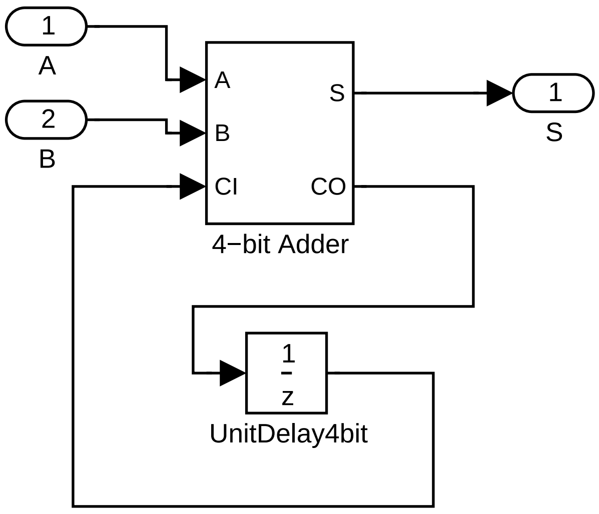
Half Adder (int517b4.eps)



Full Adder (int517b3.eps)



4-bit Adder (int517b2.eps)



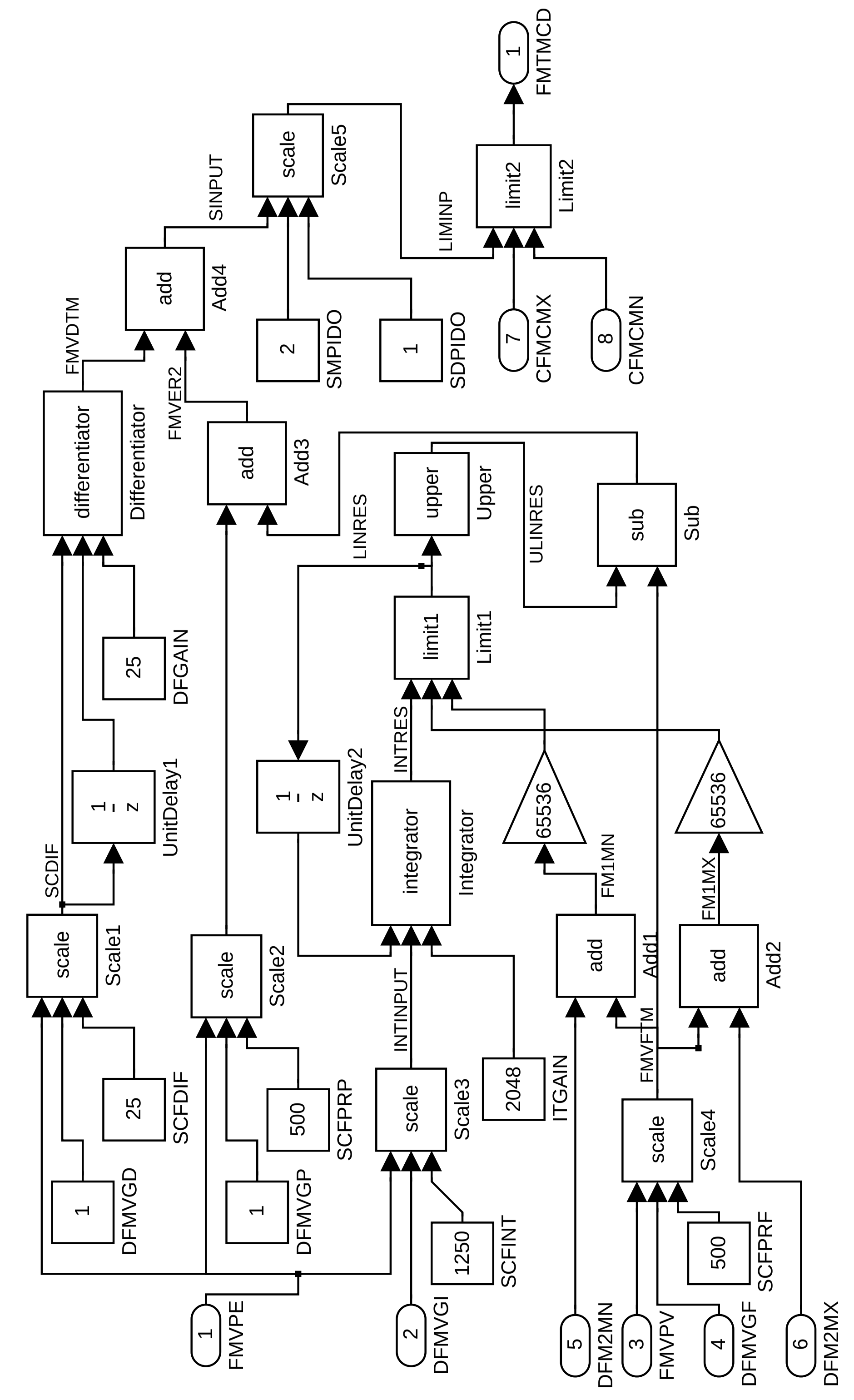
Nibble-serial Adder (int517b1.eps)

7.3 Simulink Demo Models (int517c and int517d)

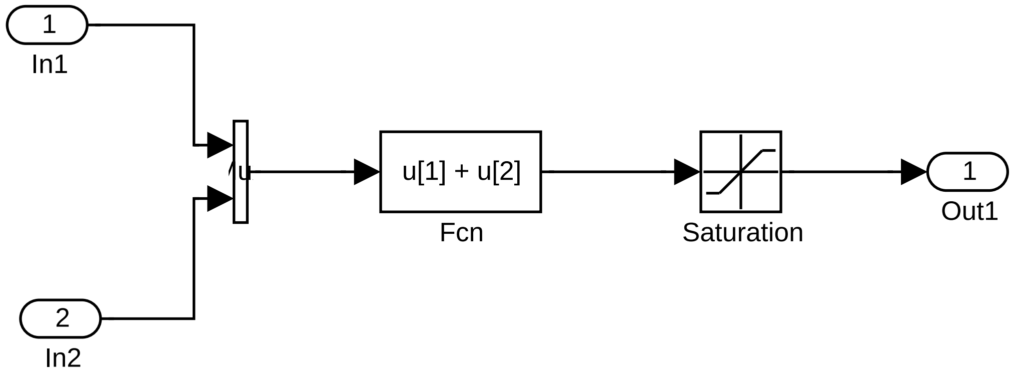
Two of the demonstration models provided with the Simulink software are used in the system tests. They are the F14 flight controller and the ABS brakes simulation. It was not originally expected that the tool would be able to produce good translations of these models, but it was required that the tool behave reasonably and helpfully when asked to translate a model which is beyond its capabilities.

The reader is referred to the Simulink system for the details of these model, which we are not able to supply since the copyright belongs to The MathWorks Inc.

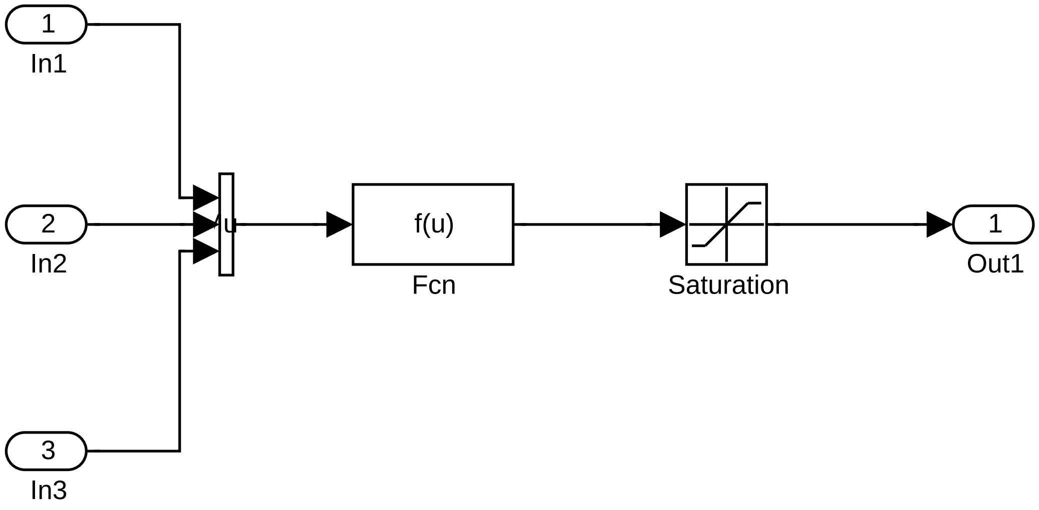
7.4 Alf's Model (int517e)



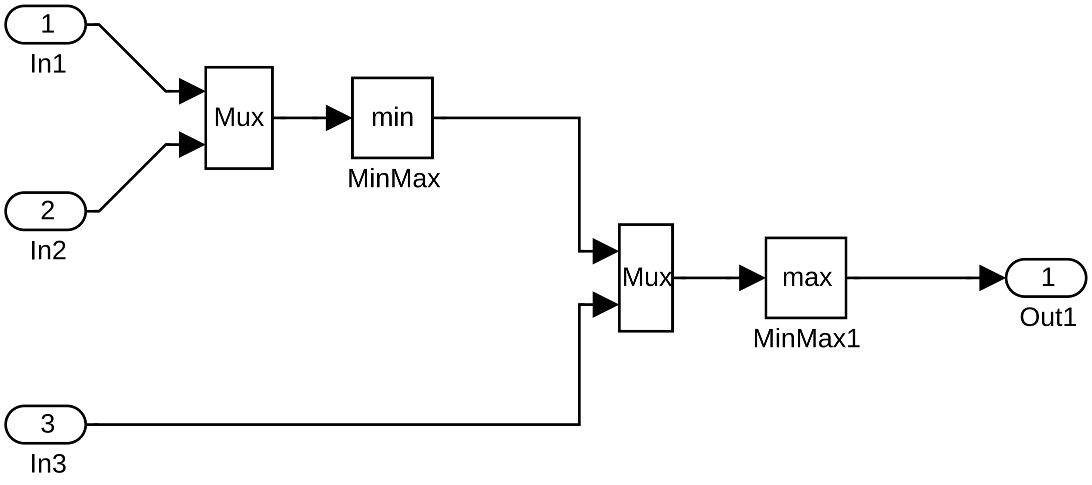
(int517e1.eps)



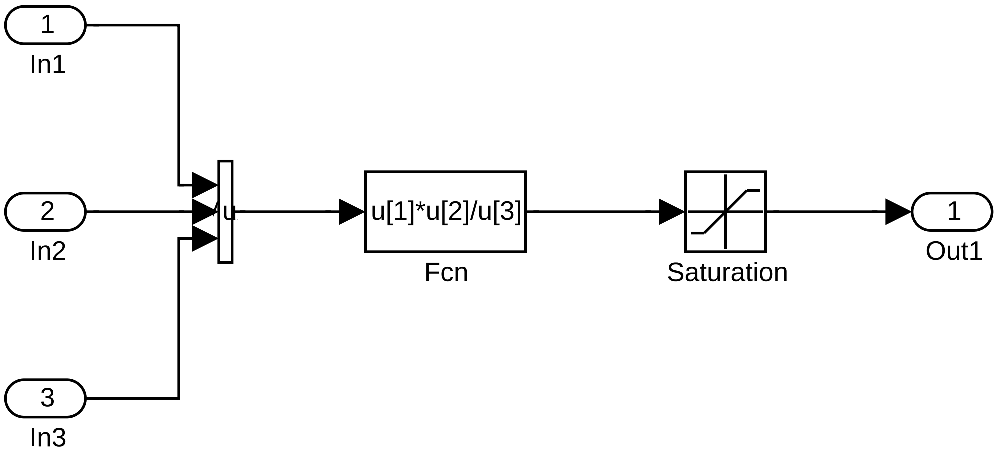
(int517e2.eps)



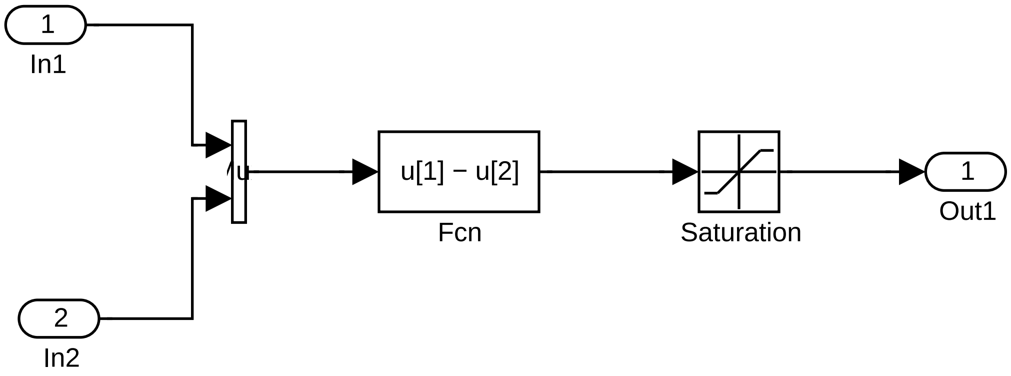
(int517e6.eps)



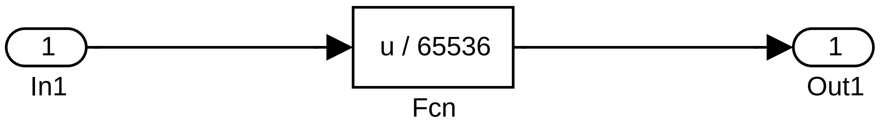
(int517e8.eps)



(int517e10.eps)



(int517e15.eps)



(int517e16.eps)

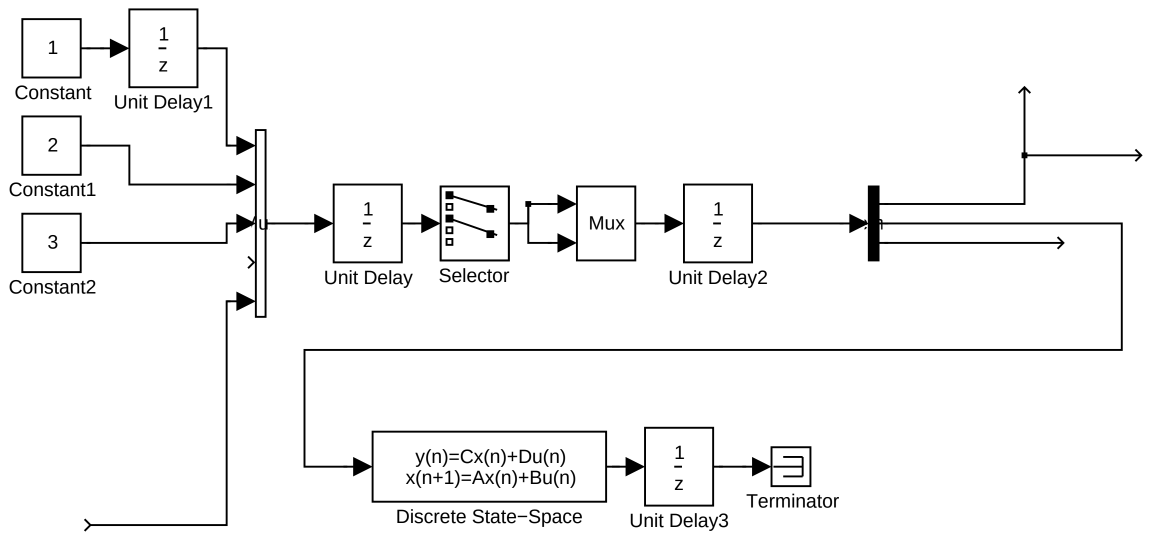
7.5 The DERA Bedford Control Model (int517f)

This is the largest model to which we have access, and the one which is probably most representative of the applications for the ClawZ tool in DERA. Because of its size the diagrams are omitted.

7.6 Loose Ends Model (int517g)

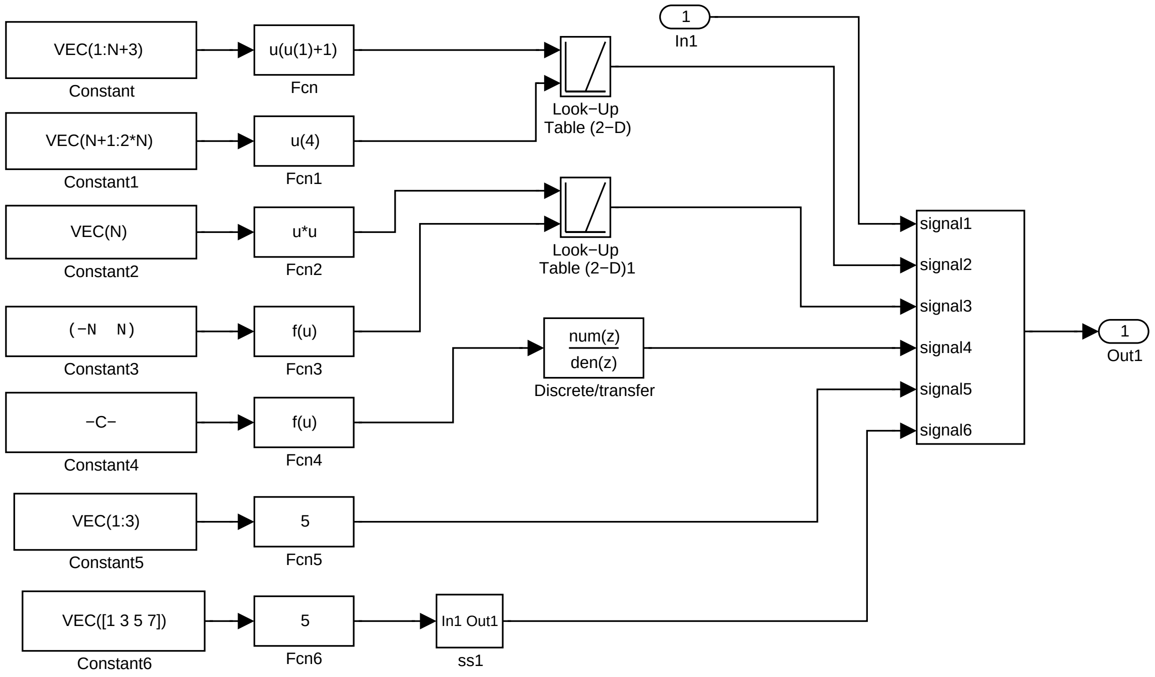
This model is intended to check the following features of the ClawZ translator:

1. Tolerance of one-ended lines
2. Scalar, Vector and Matrix parameter types and their use with overlapping generic specifications.

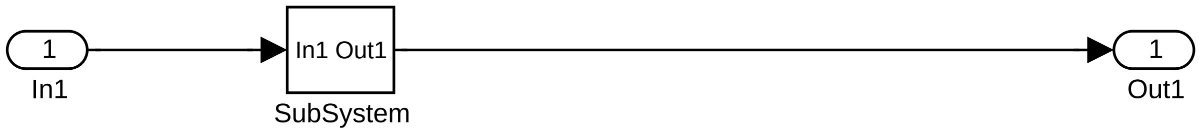


(int517g.eps)

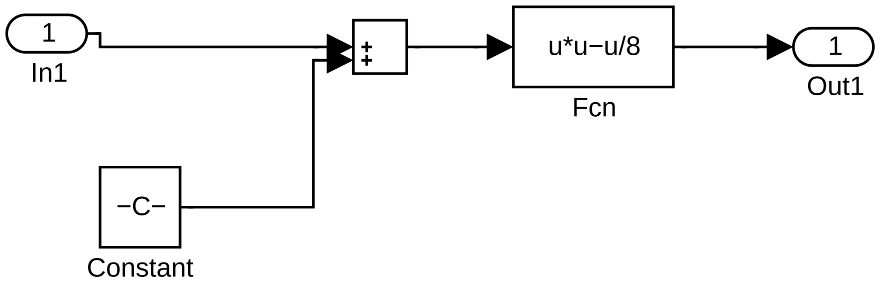
7.7 Fcn Expression Test Model (int517k)



(int517k1.eps)

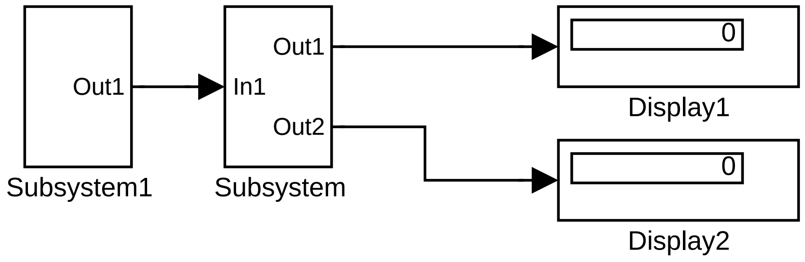


(int517k2.eps)

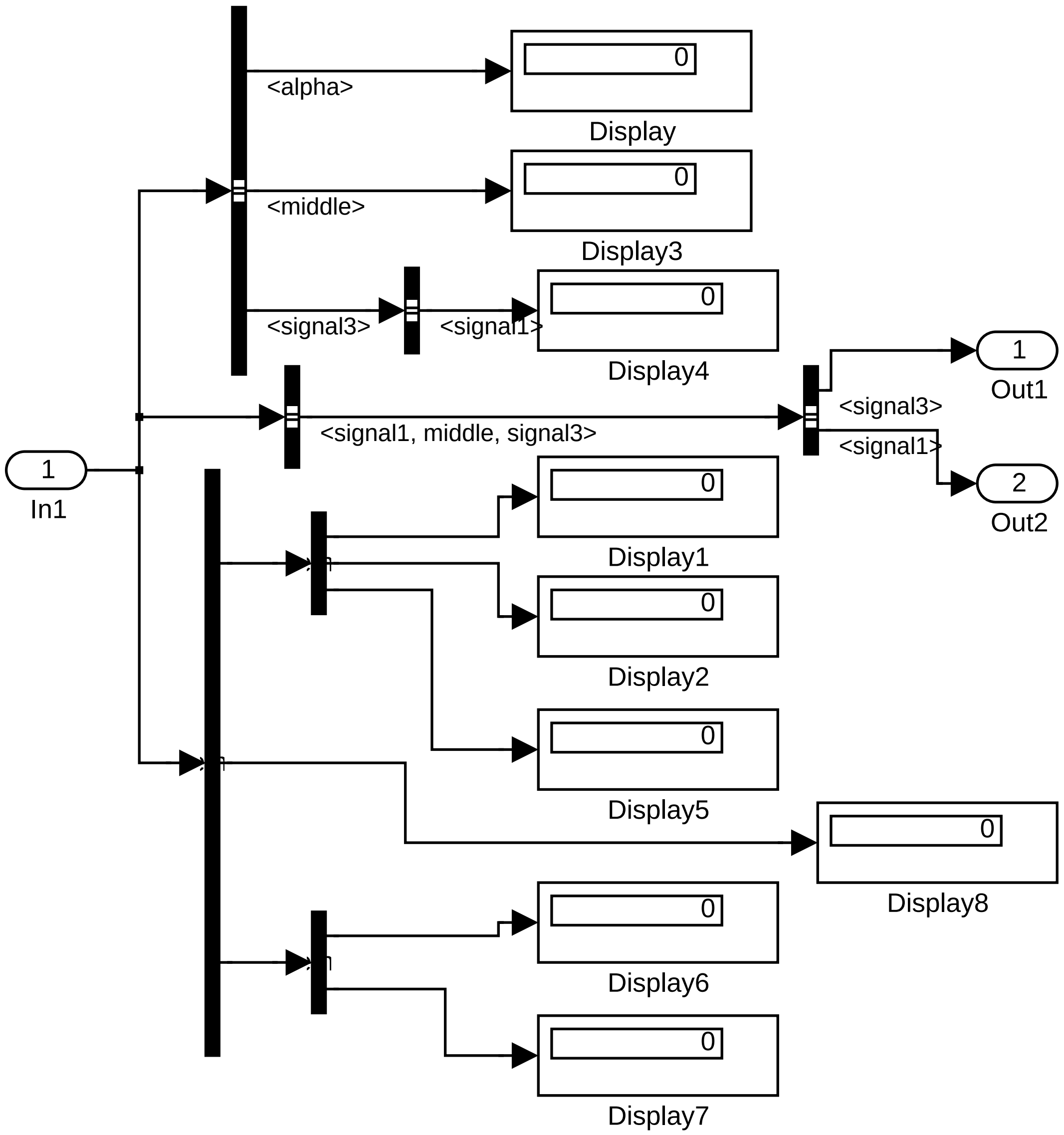


(int517k3.eps)

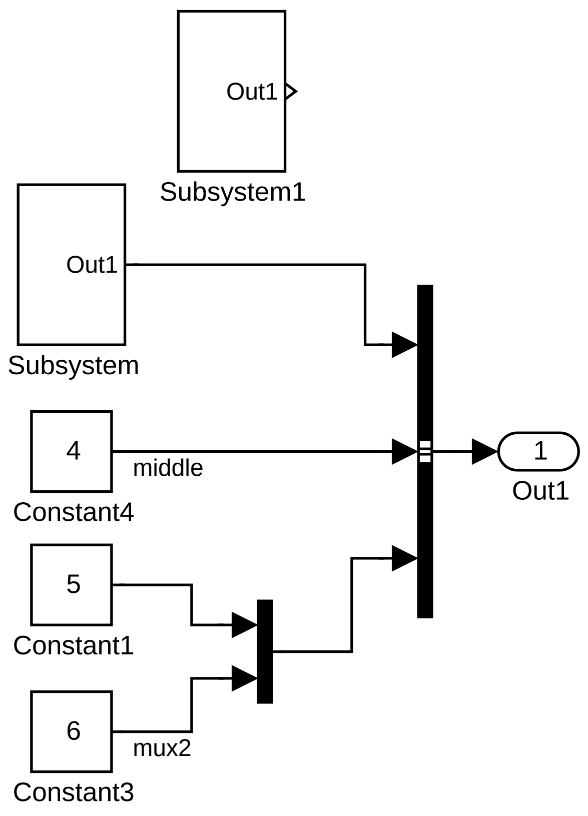
7.8 Library with Masks and Buses (int517l)



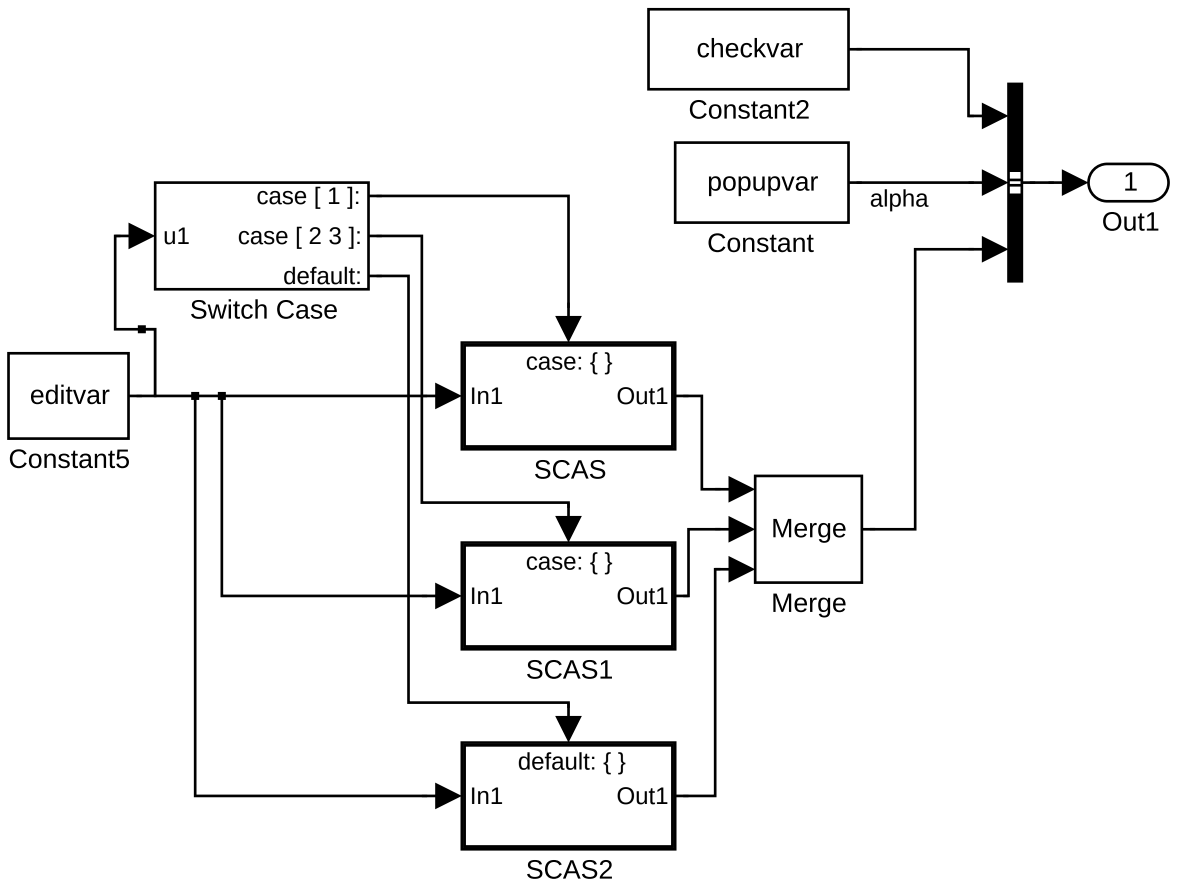
(int517l1.eps)



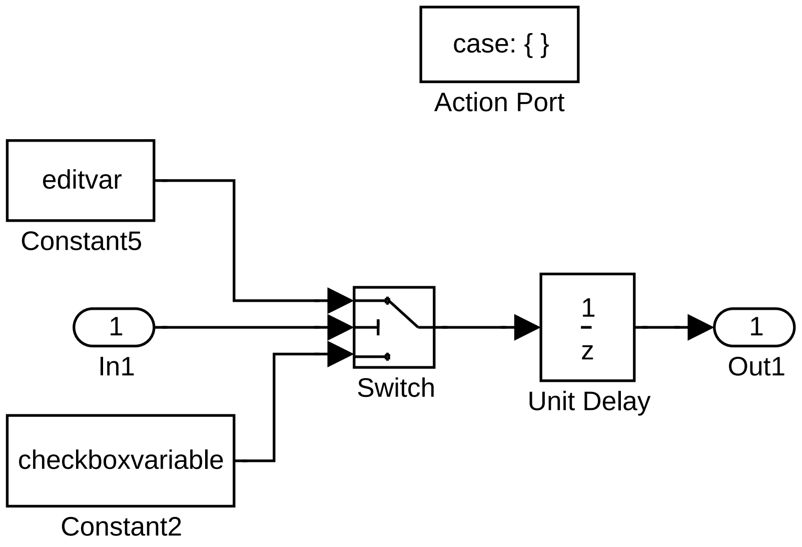
(int517l2.eps)



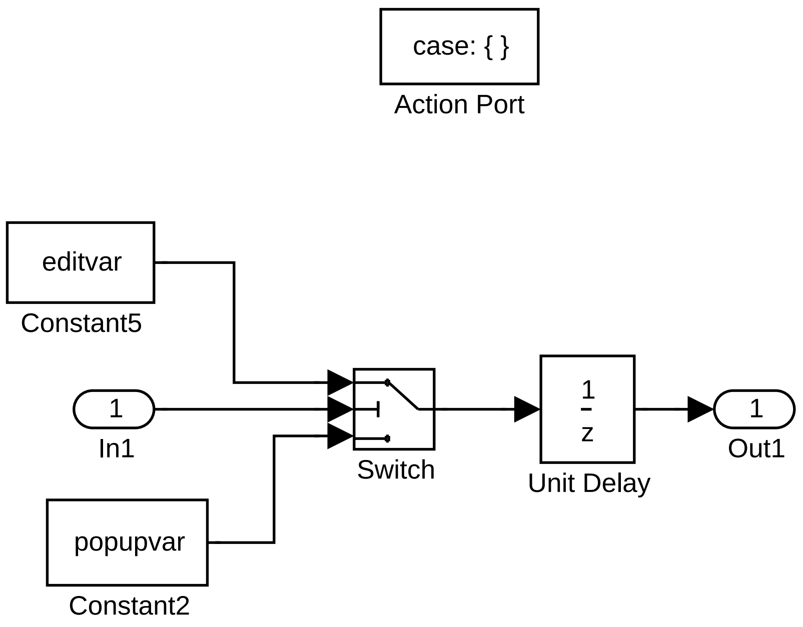
(int517l3.eps)



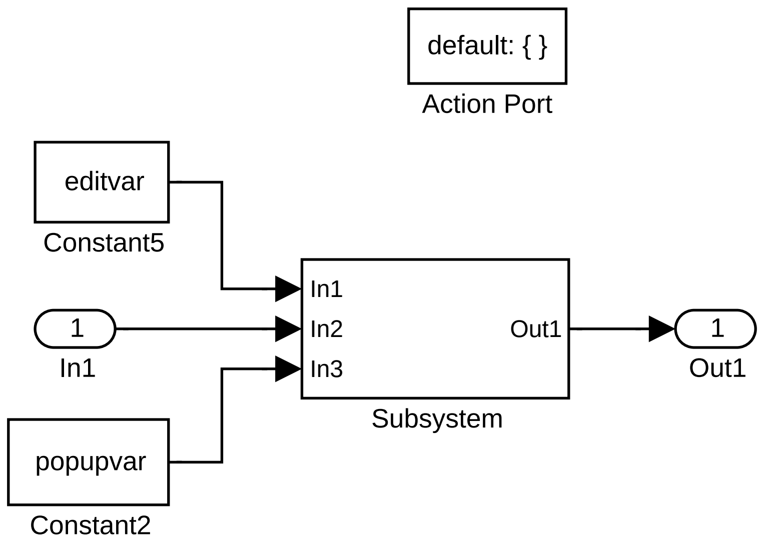
(int517l4.eps)



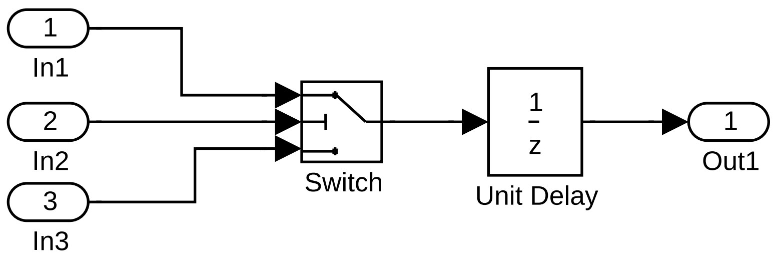
(int517l5.eps)



(int517l6.eps)

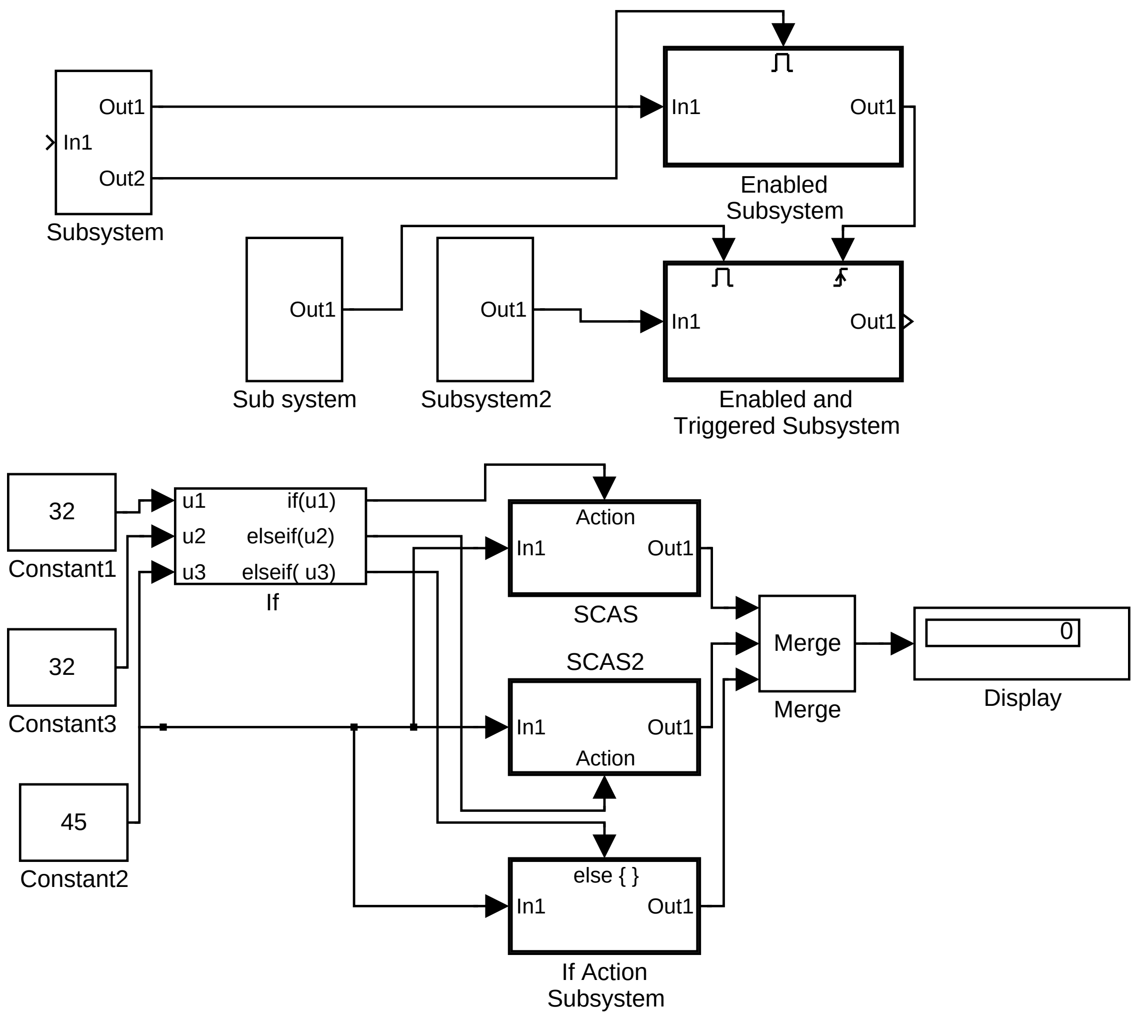


(int517l7.eps)

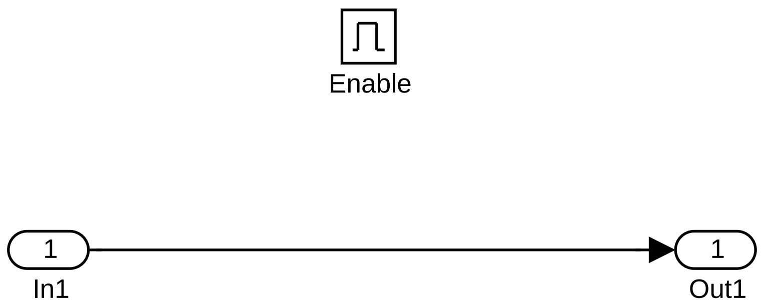


(int517l8.eps)

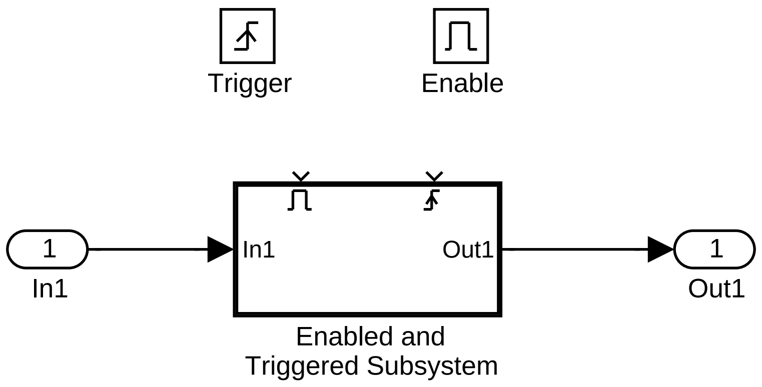
7.9 Block Reference Test Model (int517m)



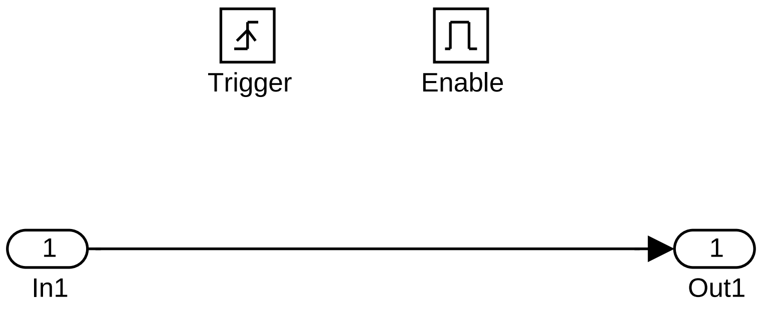
(int517m1.eps)



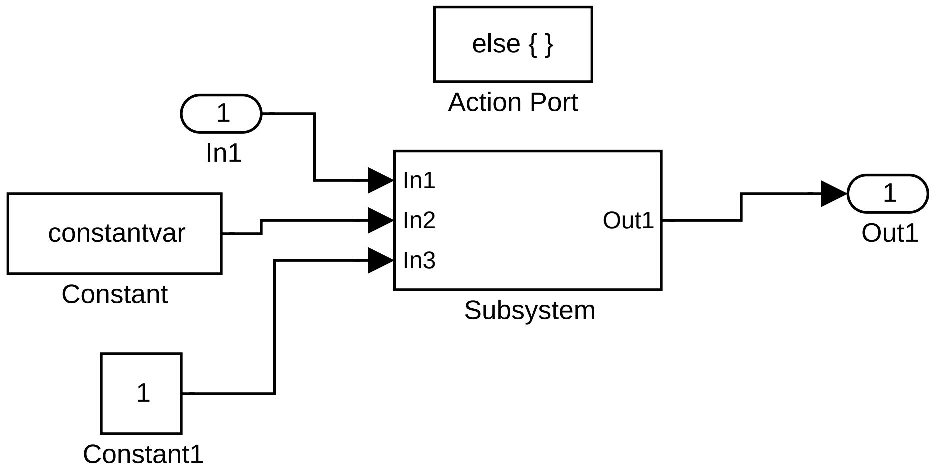
(int517m2.eps)



(int517m3.eps)

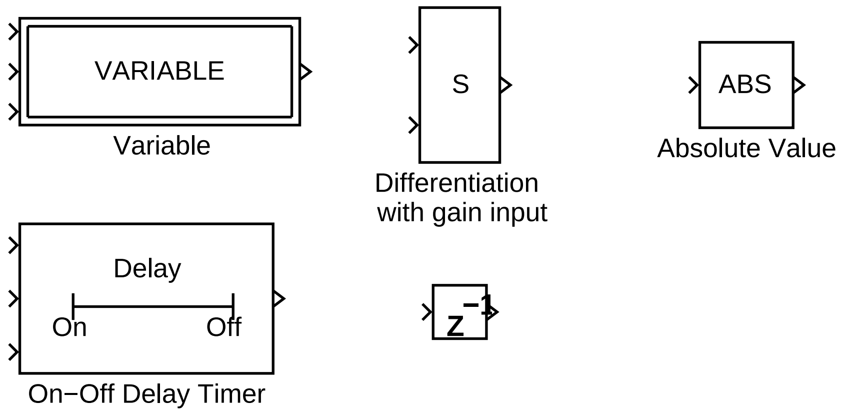


(int517m4.eps)



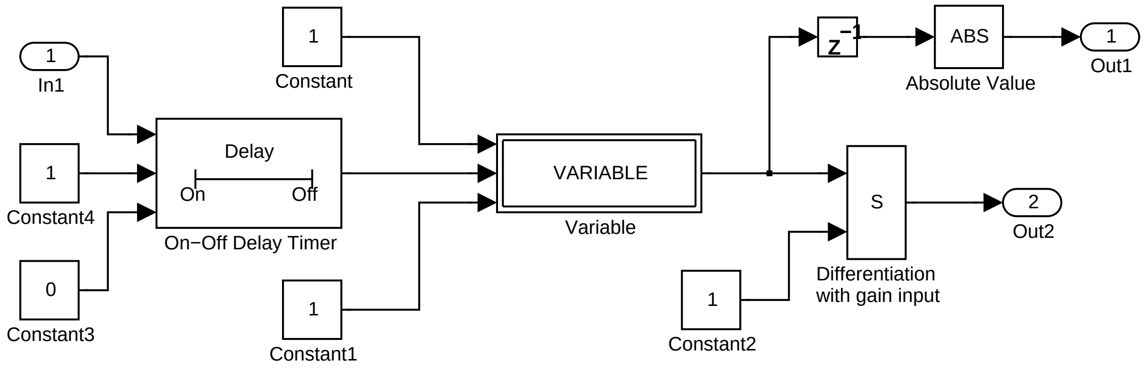
(int517m5.eps)

7.10 QinetiQ Block Reference Example Library (int517n)



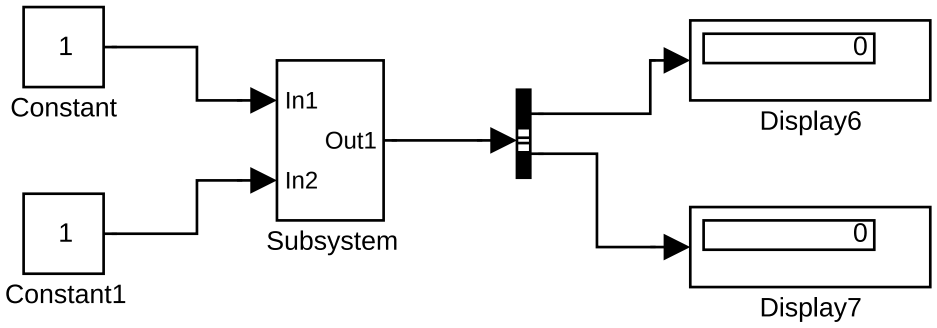
(int517n.eps)

7.11 QinetiQ Block Reference Example Model (int517o)

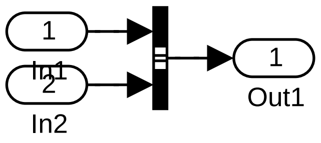


(int517o.eps)

7.12 Diagnostic Model (int517p)



(int517p1.eps)

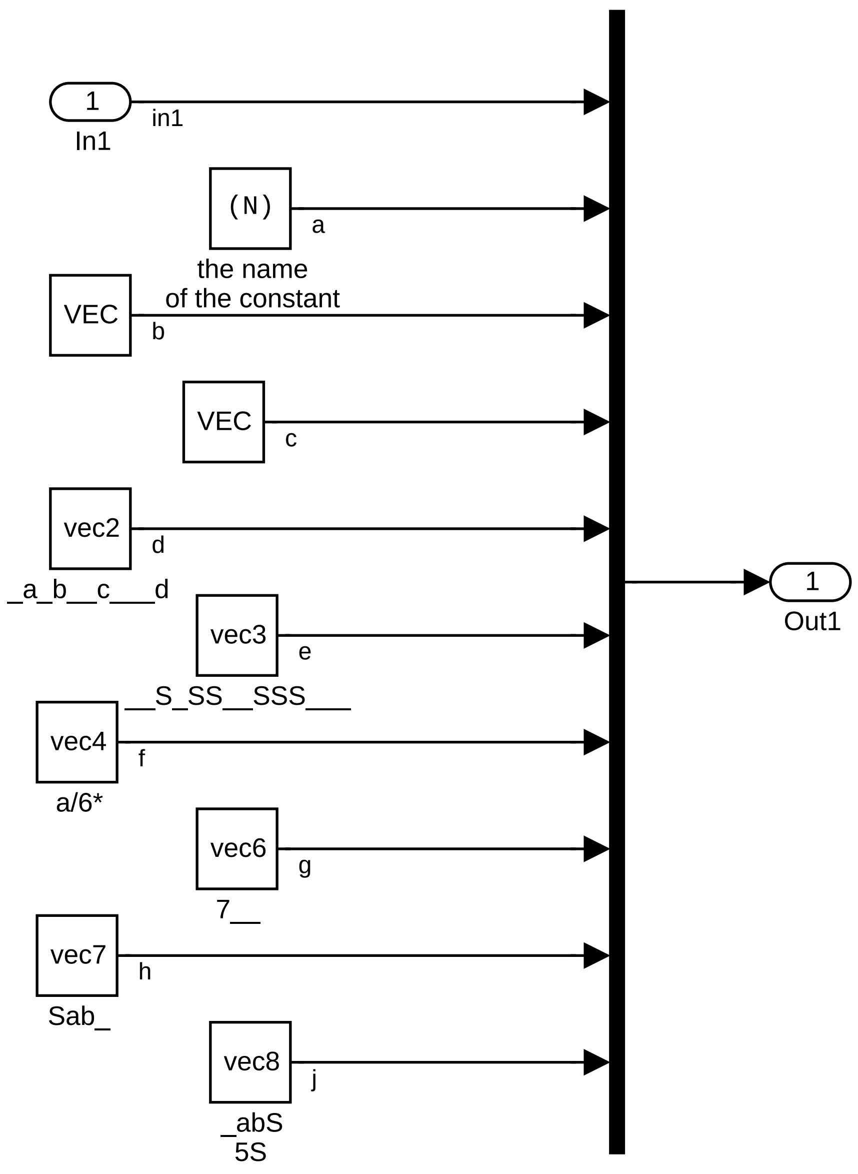


(int517p2.eps)

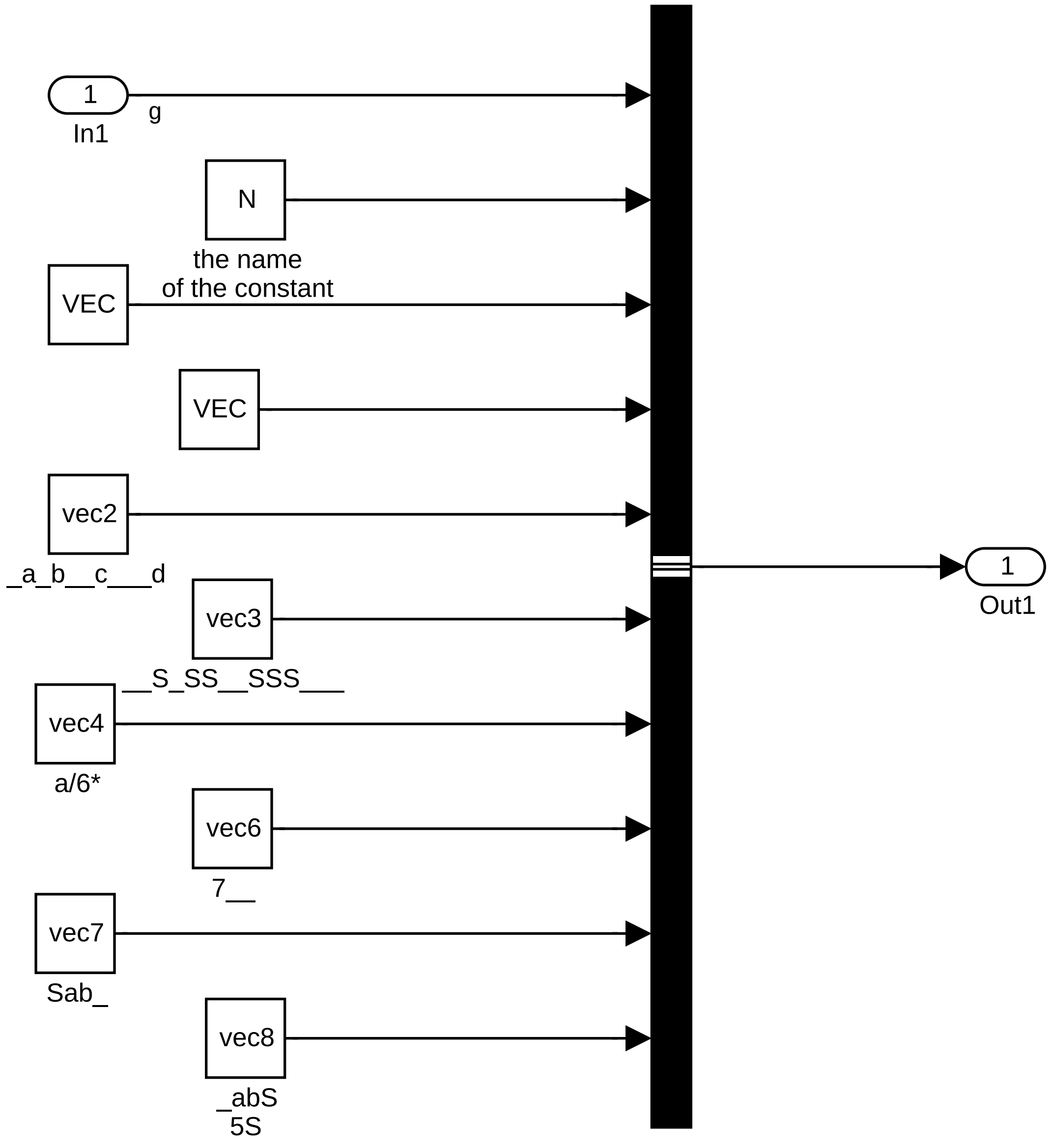
7.13 Name Translation Model (int517q)

This model tests:

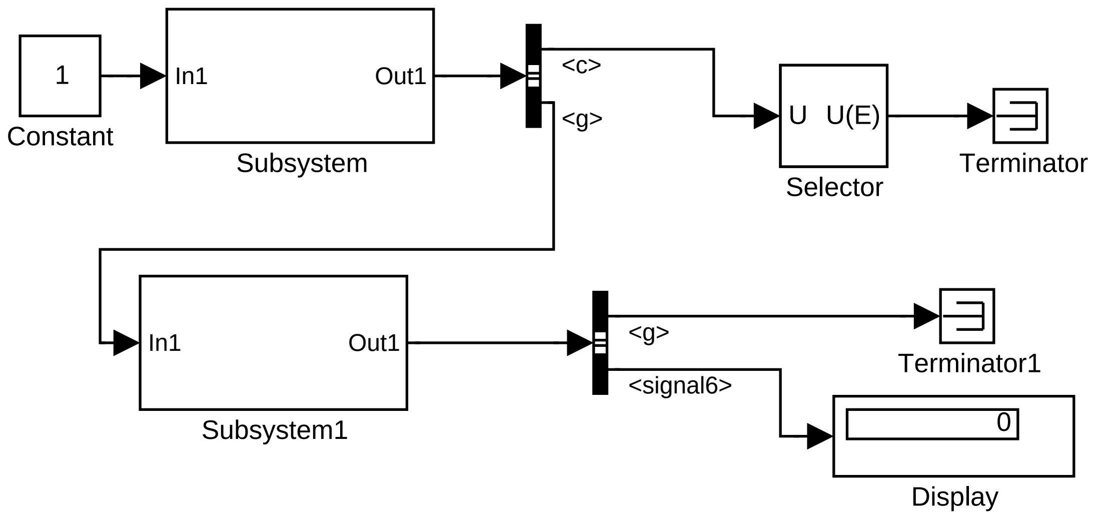
- Name translation
- Signal inference
- Selector and Terminator blocks



(int517q1.eps)



(int517q2.eps)

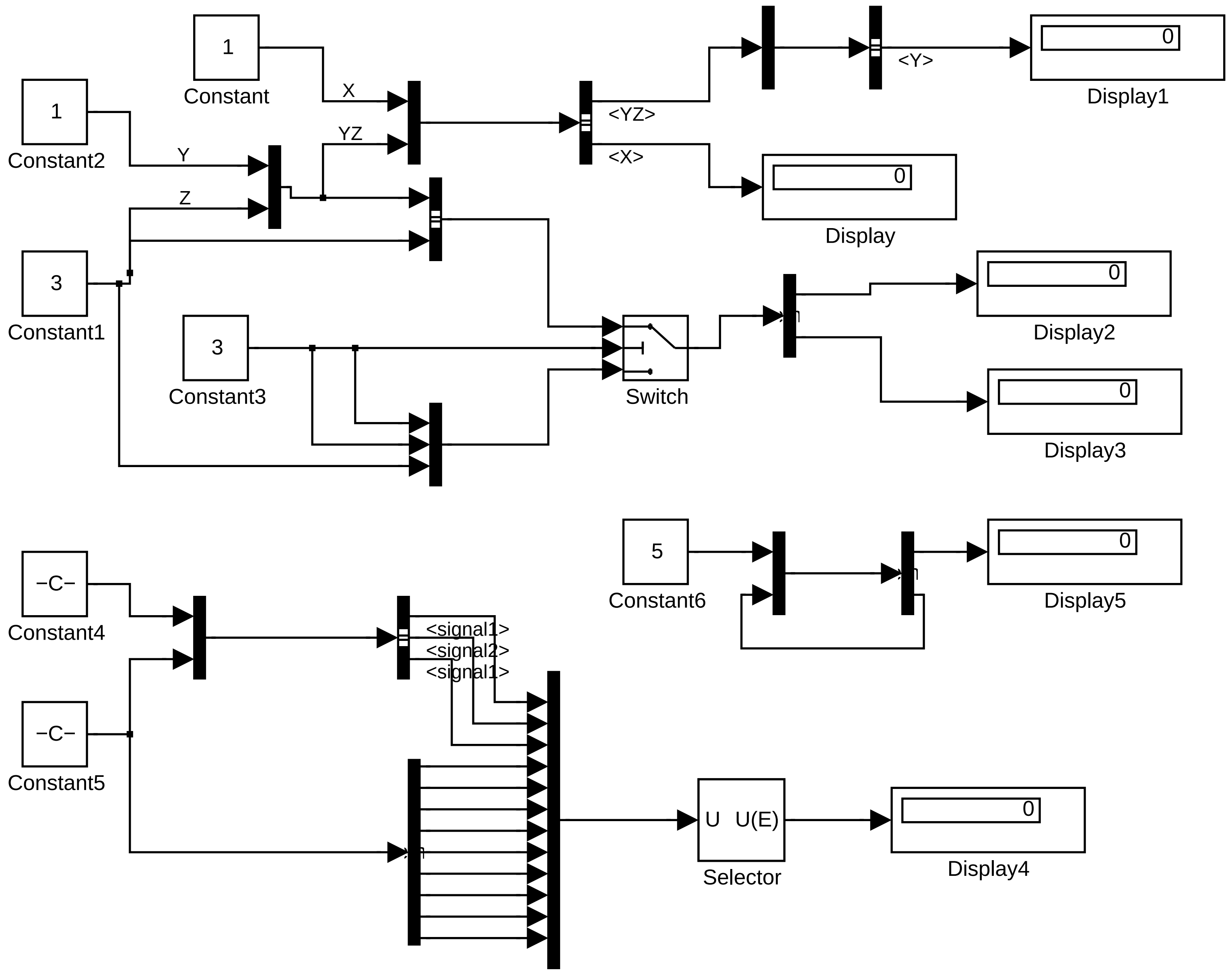


(int517q3.eps)

7.14 Signal block synthesis and virtualization (int517r)

This model tests:

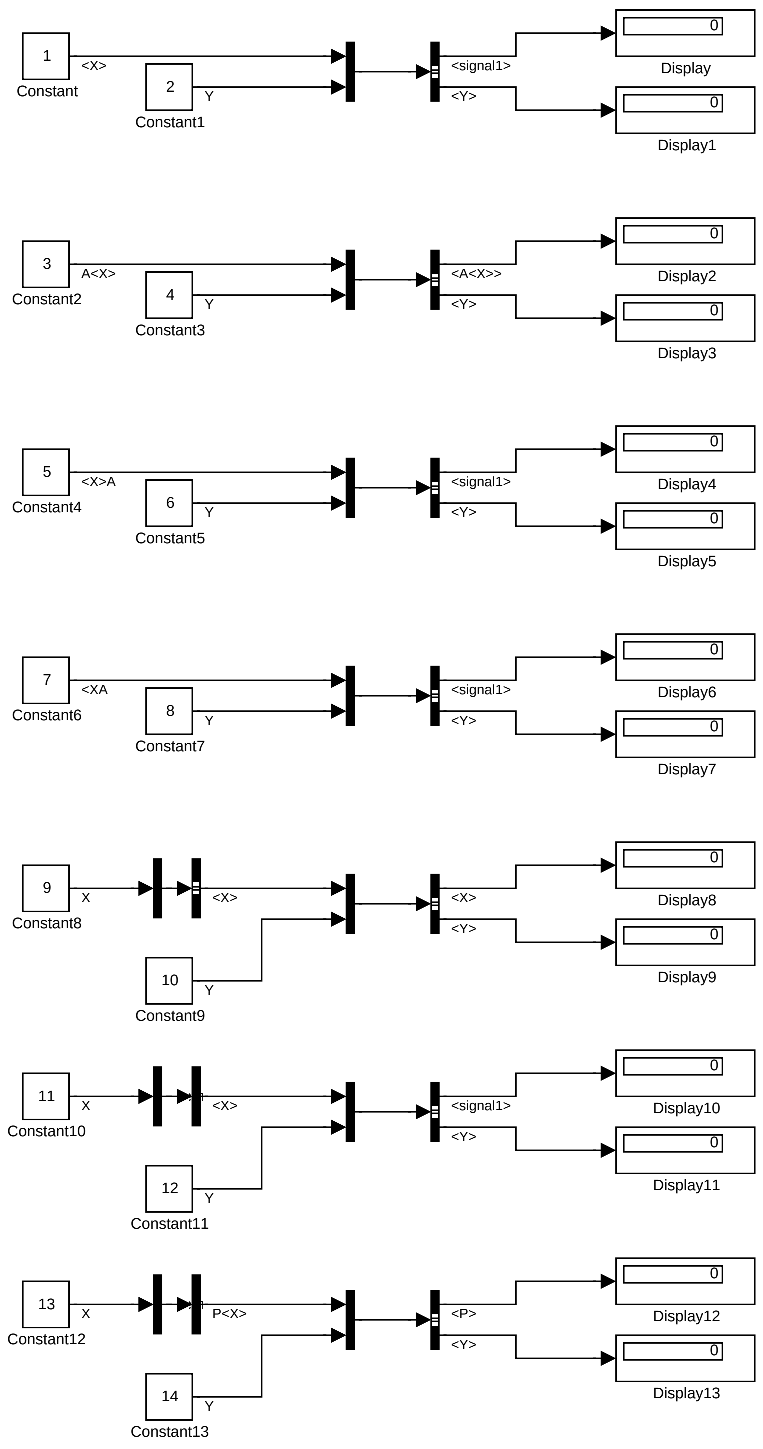
- Signal inference
- Synthesis or Virtualization of Mux, Demux, Bus Creator, Bus Selector blocks.
- Virtualization simplifications.
- Virtualization loop handling.


1
Constant
1
Constant2
3
Constant1
3
Constant3
X
Y
YZ
Z
<YZ>
<X>
<Y>
0
Display1
0
Display
0
Display2
0
Display3
Switch
5
Constant6
0
Display5
−C−
Constant4
−C−
Constant5
<signal1>
<signal2>
<signal1>
U U(E)
Selector
0
Display4

(int517r.eps)

7.15 Name Translation Model (int517s)

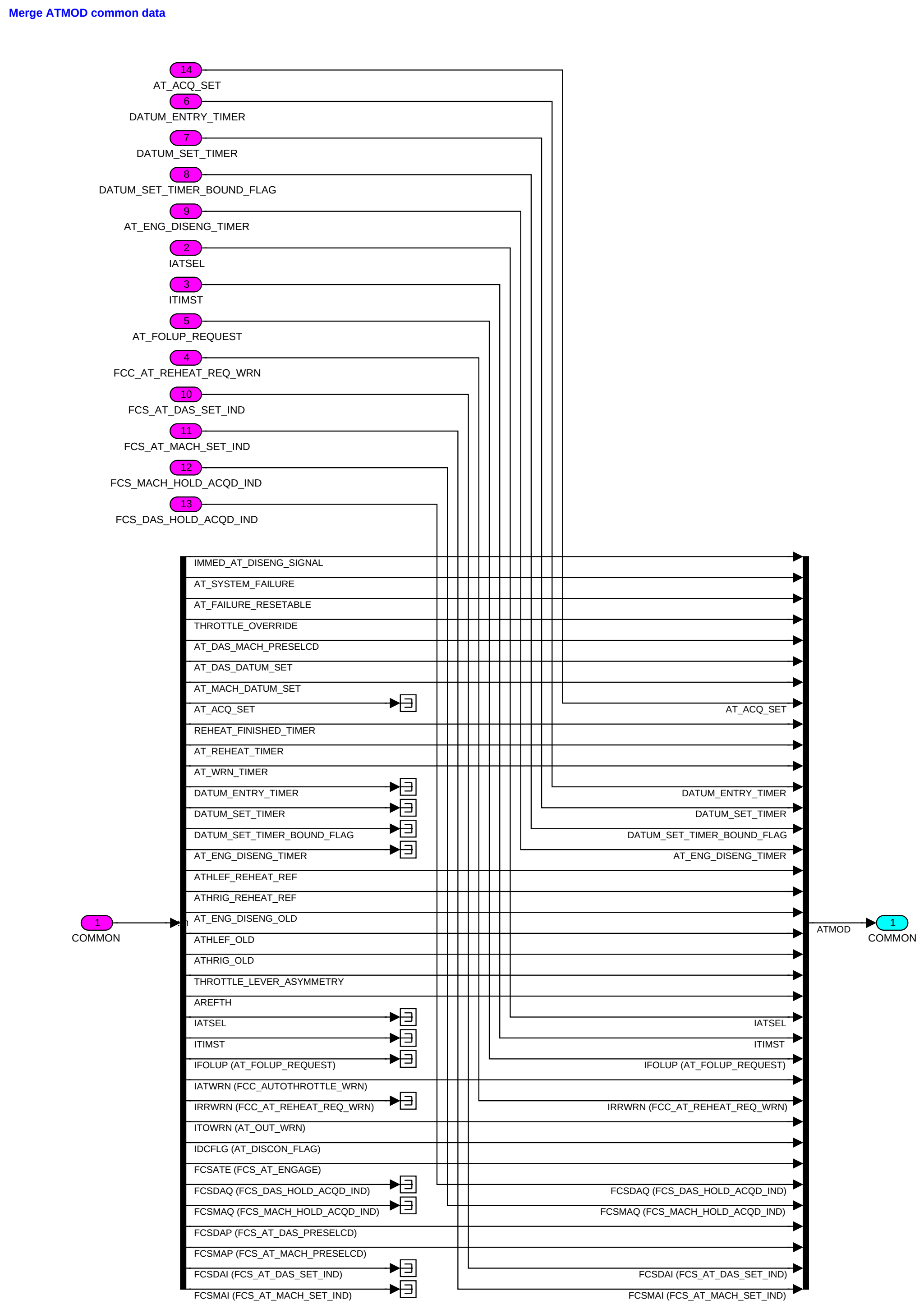
A QuinetiQ example addressing signal name propagation.



(int517s.eps)

7.16 QuintetiQ Mux Virtualization Model (int517t)

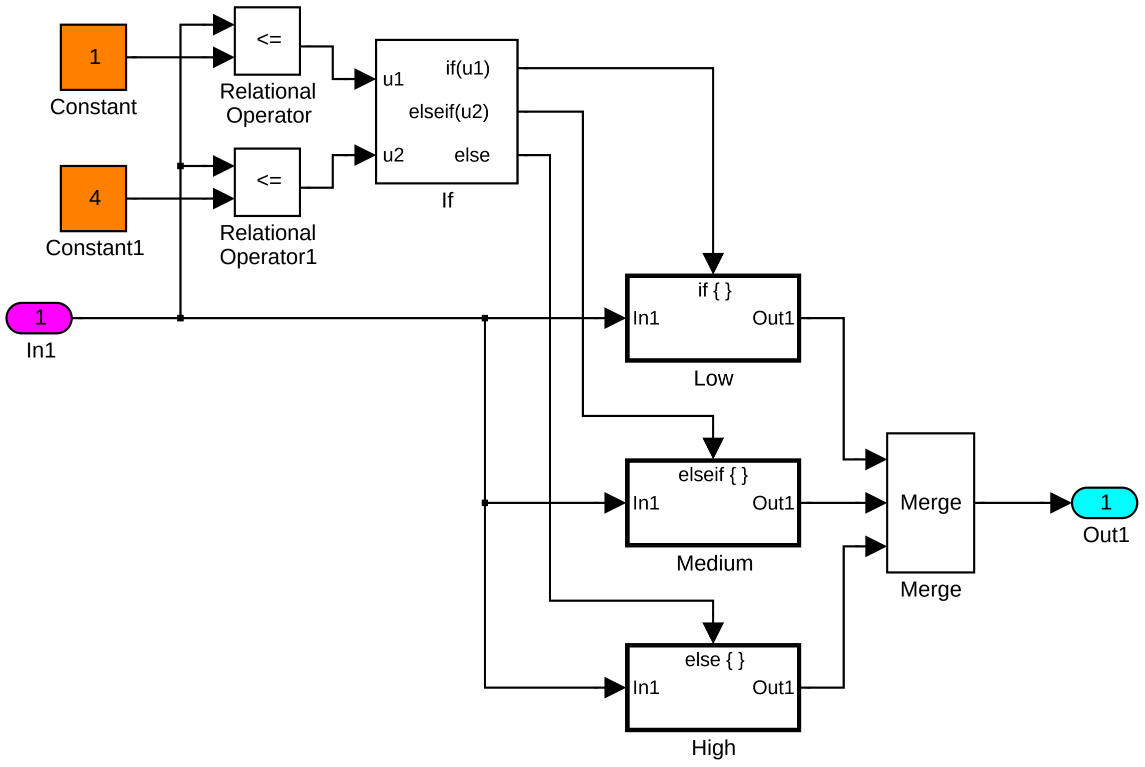
A QuinetiQ example originally showing missordering of elements in a vector display in a completely vitualized Mux/Demux subsystem.



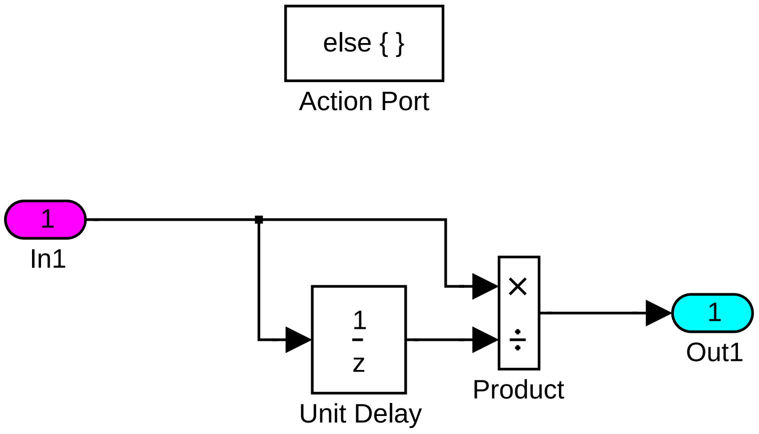
(int517t.eps)

7.17 QuintetiQ Action Subsystem eg5 (int517u)

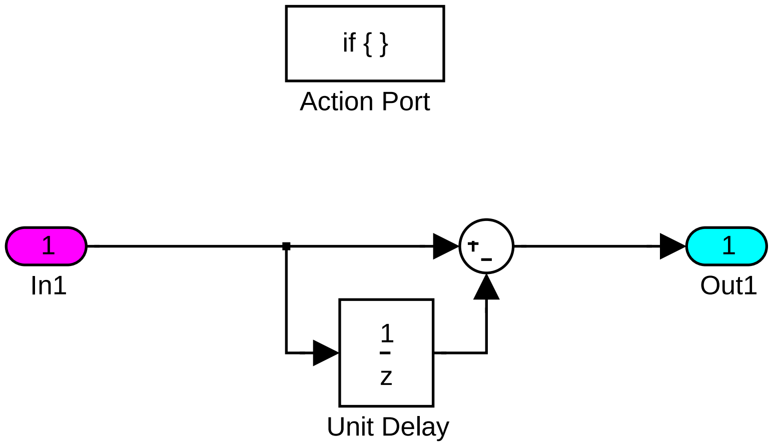
A QuinetiQ example used during the requirements negotiations for support of action subsystems.



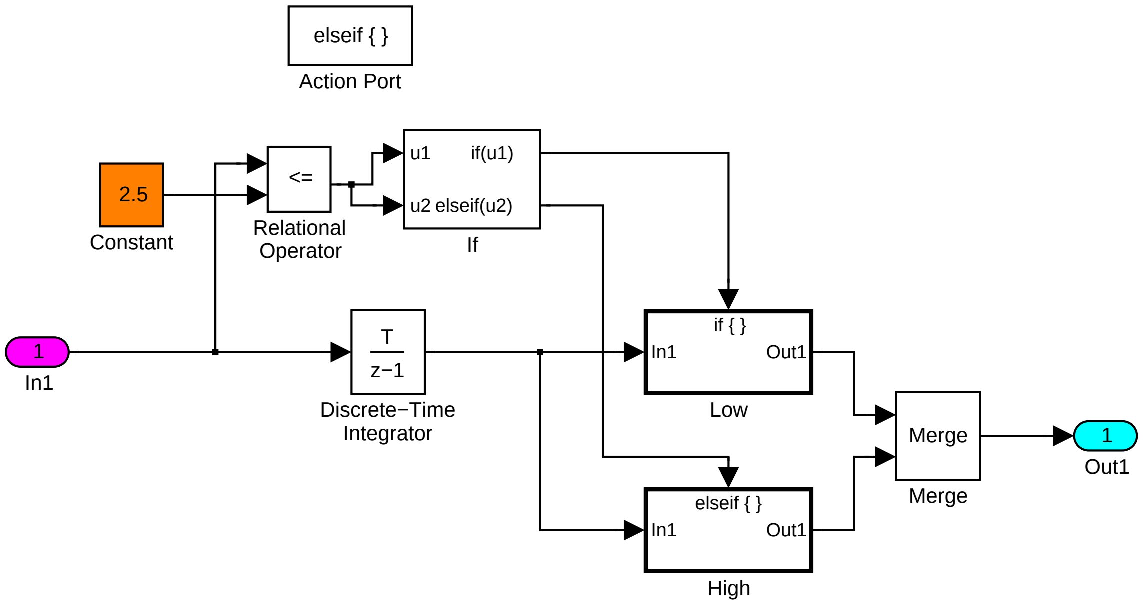
(int517u1.eps)



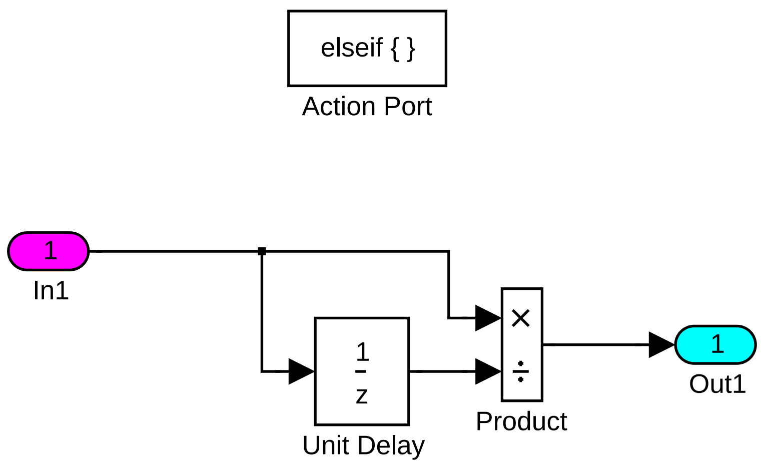
(int517u2.eps)



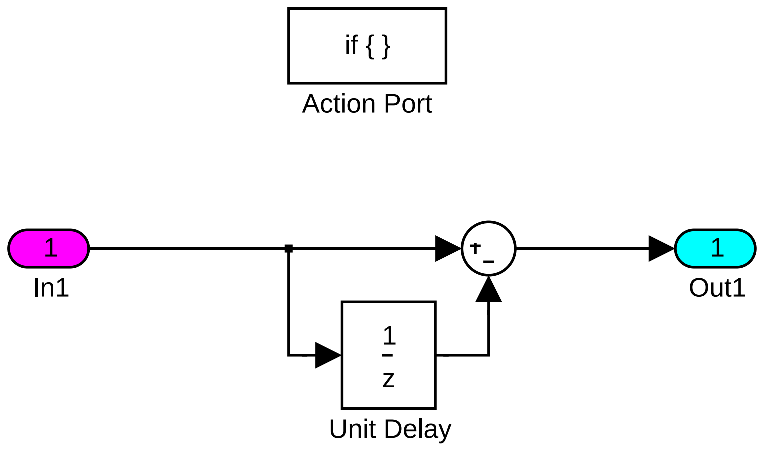
(int517u3.eps)



(int517u4.eps)



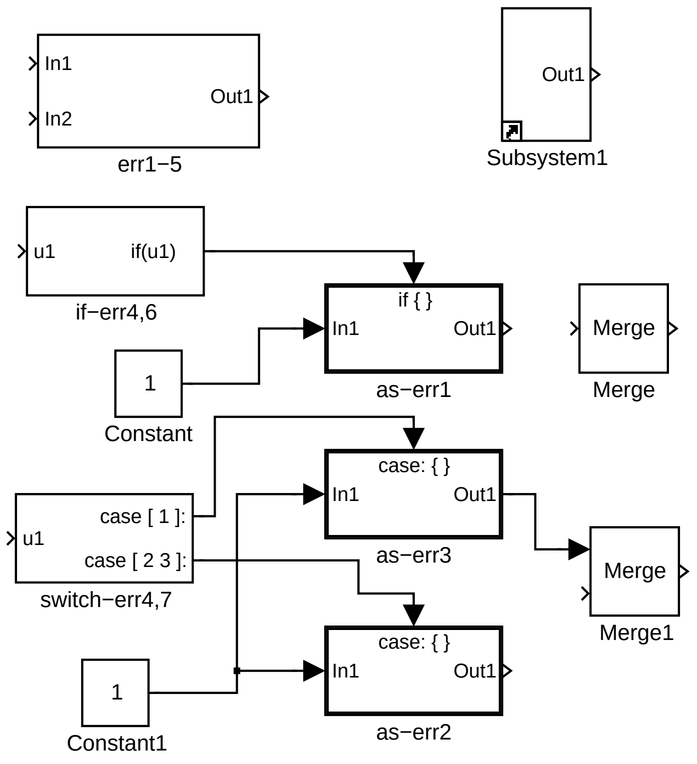
(int517u5.eps)



(int517u6.eps)

7.18 Action Subsystem Error Checks (int517v)

int517v

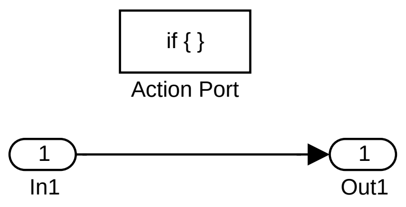


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 1/19

int517v/as−err1

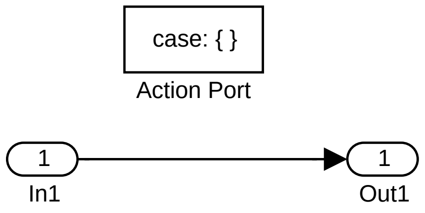


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 2/19

int517v/as−err2

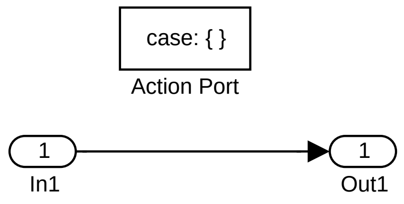


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 3/19

int517v/as−err3

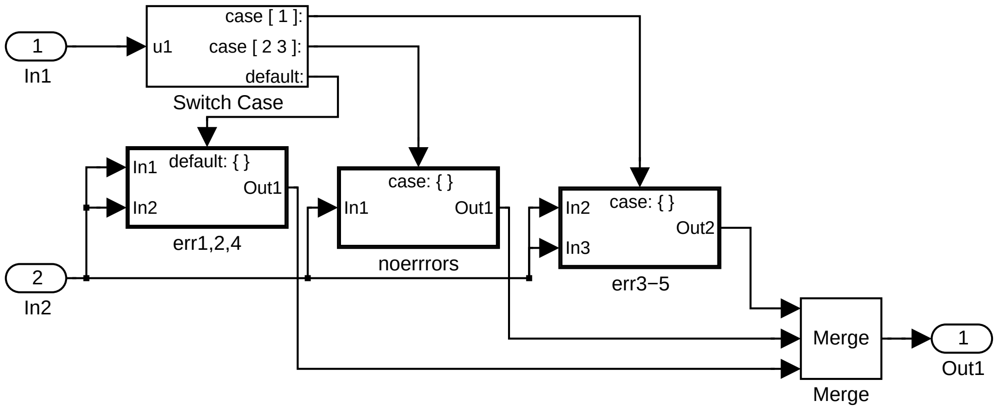


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 4/19

int517v/err1−5

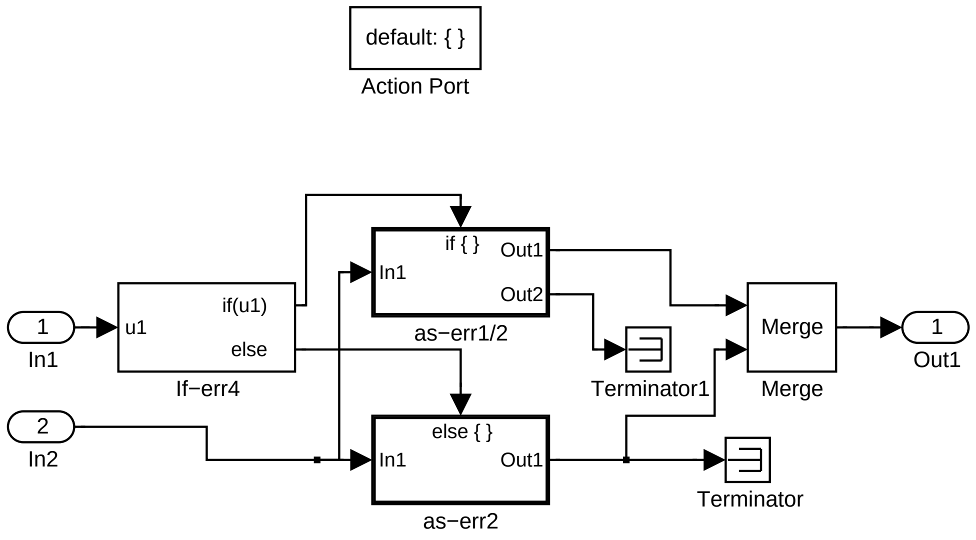


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 5/19

int517v/err1−5/err1,2,4

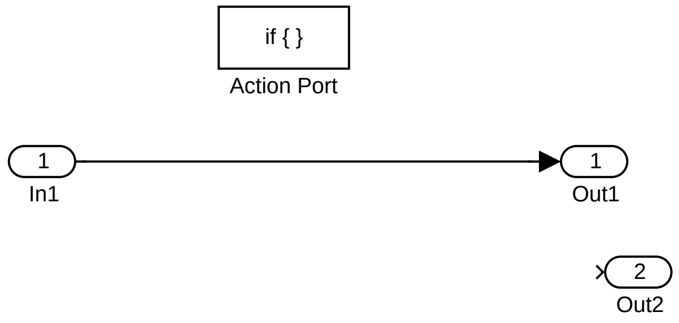


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 6/19

int517v/err1−5/err1,2,4/as−err1//2



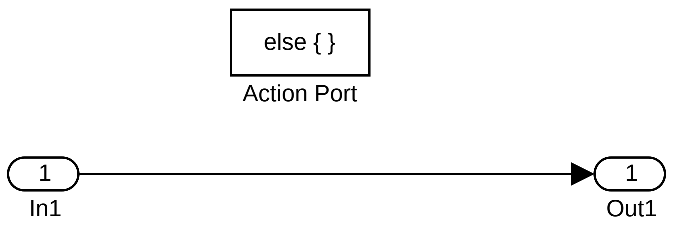
/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 7/19

(int517u7.eps)

int517v/err1−5/err1,2,4/as−err2

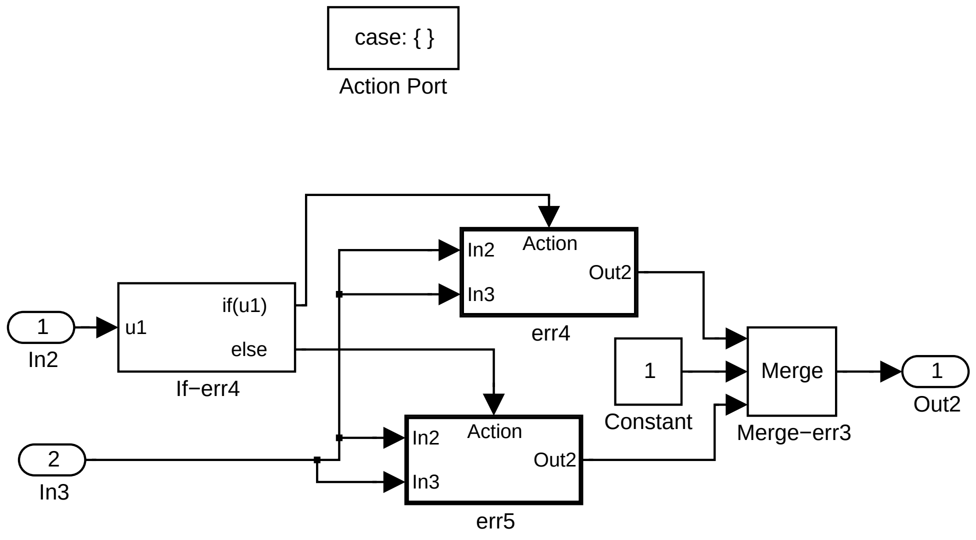


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 8/19

int517v/err1−5/err3−5

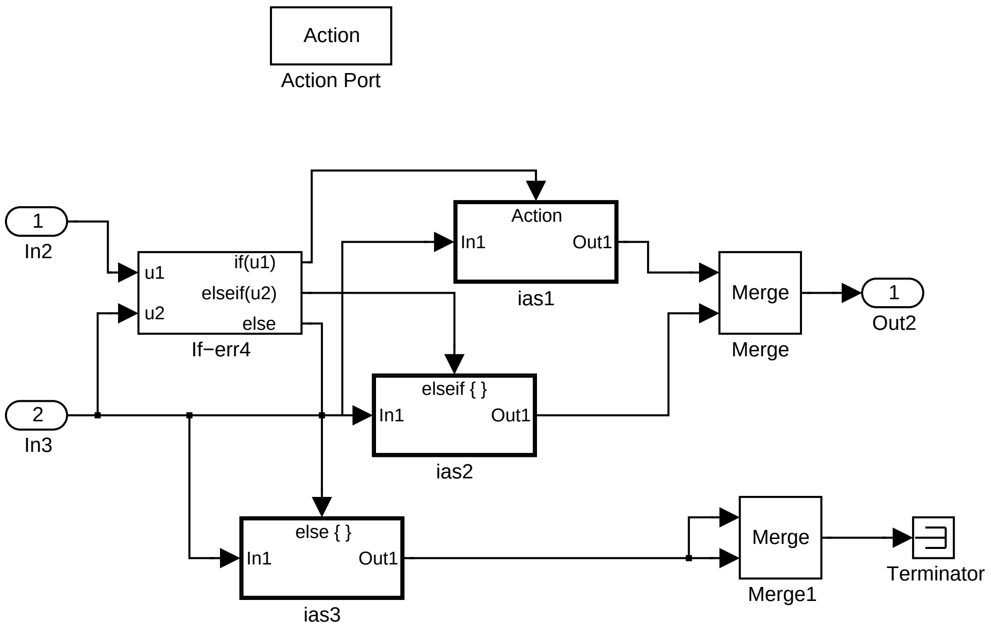


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 9/19

int517v/err1−5/err3−5/err4

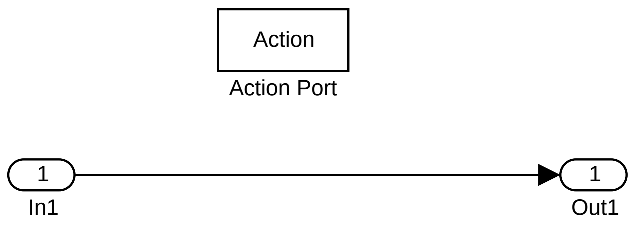


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 10/19

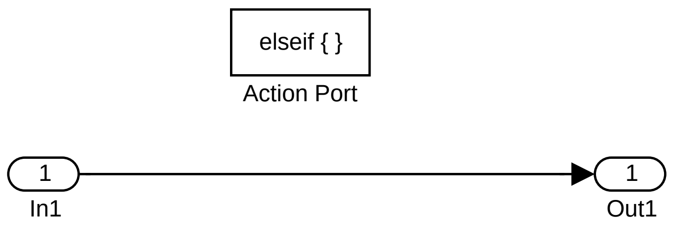
int517v/err1−5/err3−5/err4/ias1



/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17	page 11/19

int517v/err1−5/err3−5/err4/ias2

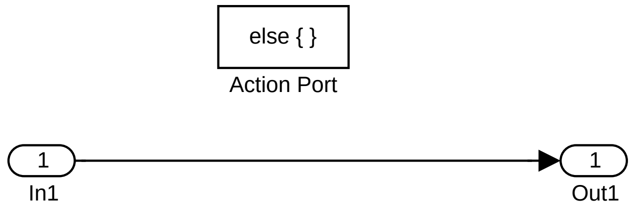


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 12/19

int517v/err1−5/err3−5/err4/ias3

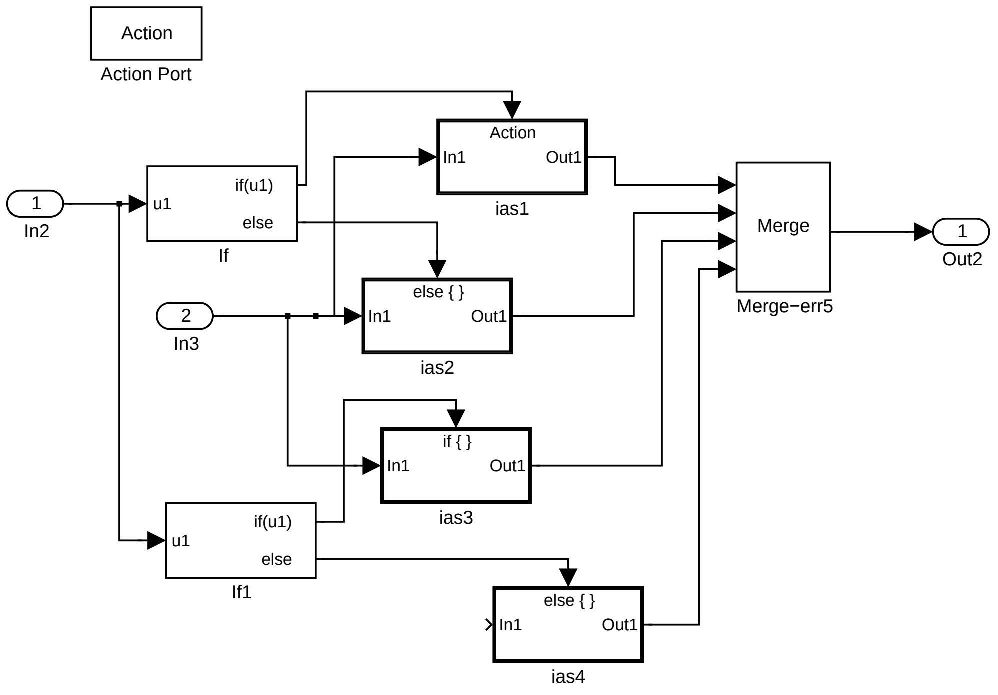


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 13/19

int517v/err1−5/err3−5/err5

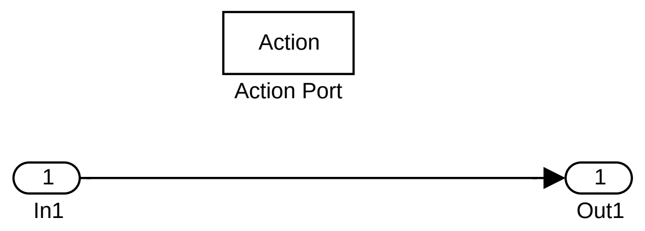


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 14/19

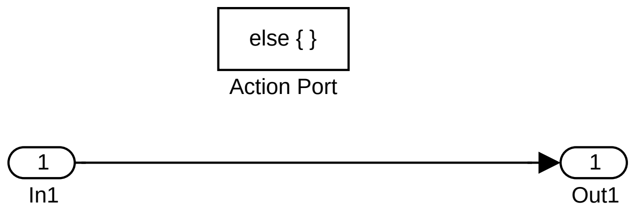
int517v/err1−5/err3−5/err5/ias1



/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17	page 15/19

int517v/err1−5/err3−5/err5/ias2

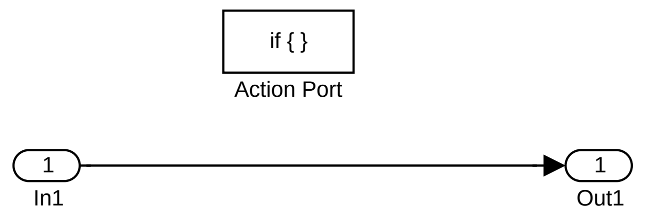


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 16/19

int517v/err1−5/err3−5/err5/ias3

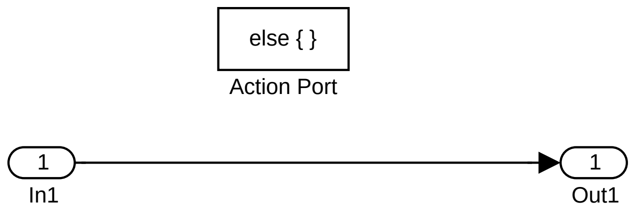


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 17/19

int517v/err1−5/err3−5/err5/ias4

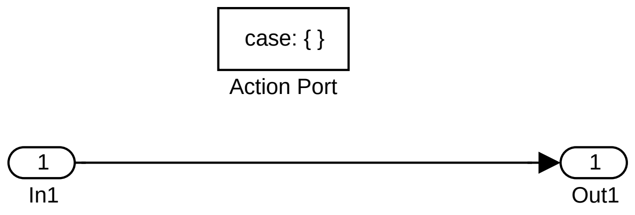


/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

page 18/19

int517v/err1−5/noerrrors



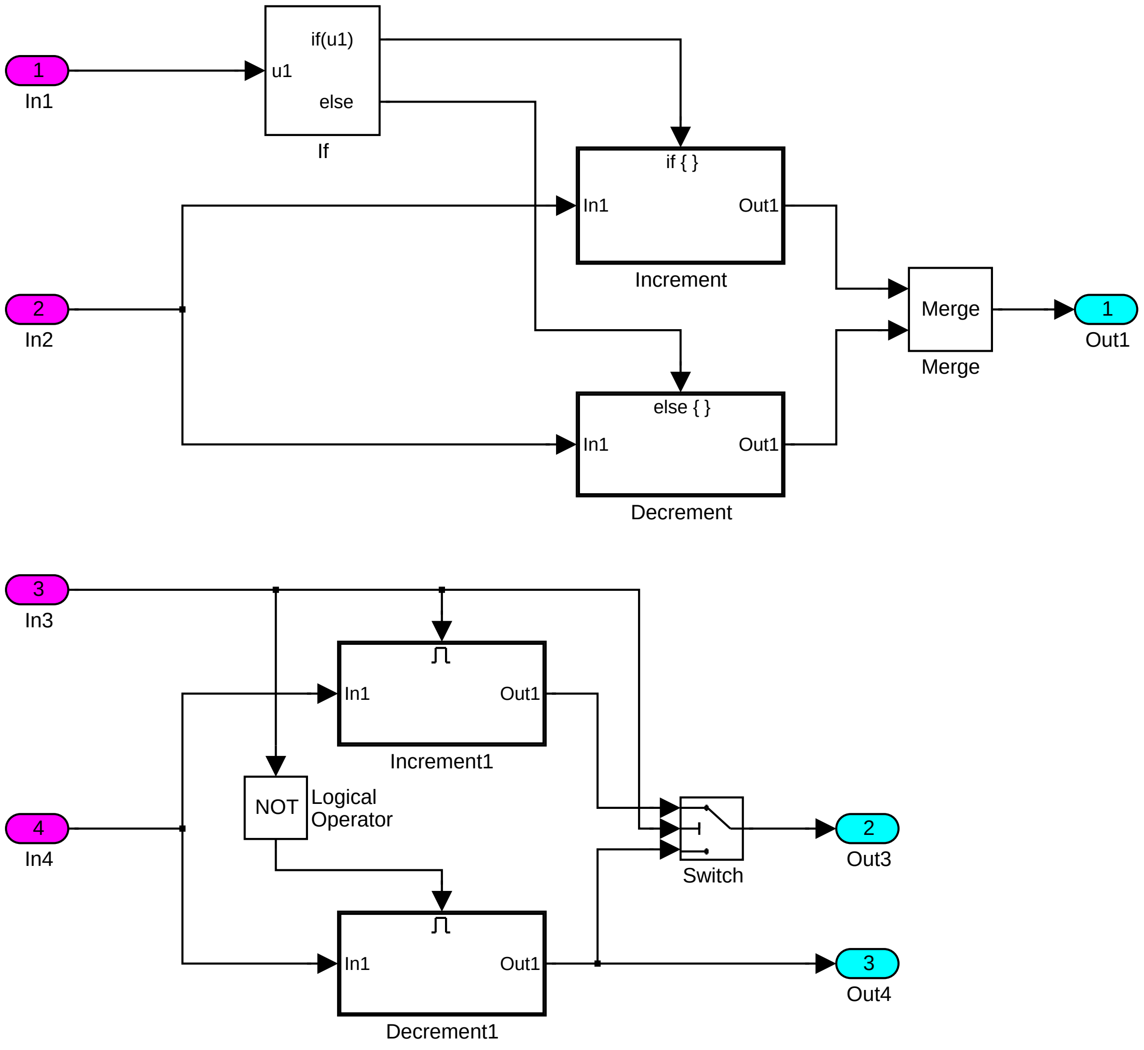
/home/clawz/v1.1.4.rbj.030619/src/int517v.mdl

printed 20−Jun−2003 07:17

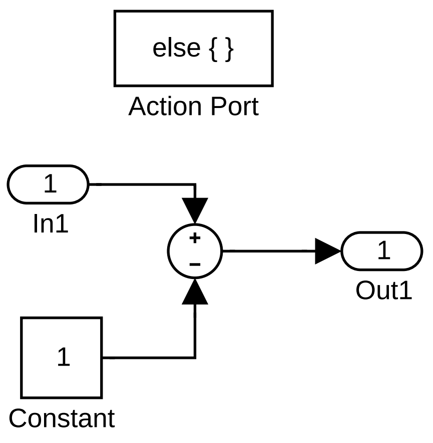
page 19/19

7.19 QuinetiQ Action Subsystems (eg1,eg3) (int517w)

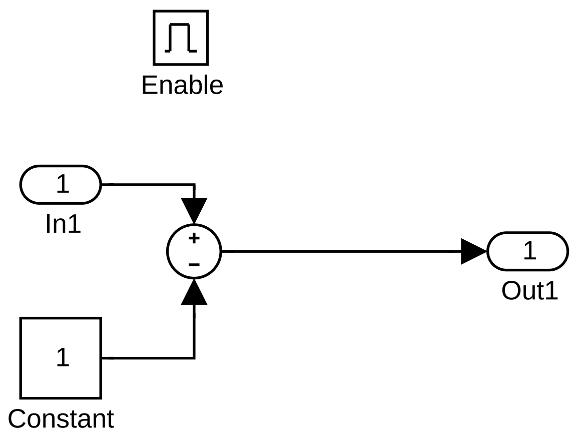
Put together from QuinetiQ eg1 and eg3 examples, also illustrates what happens with enabled subsystems.



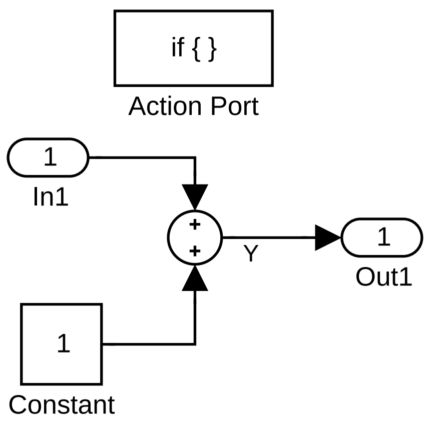
(int517w1.eps)



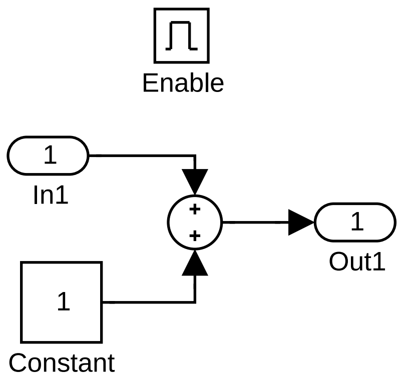
(int517w2.eps)



(int517w3.eps)

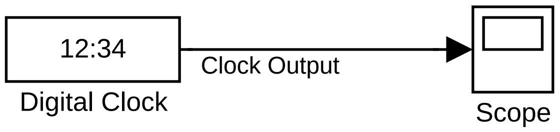


(int517w4.eps)



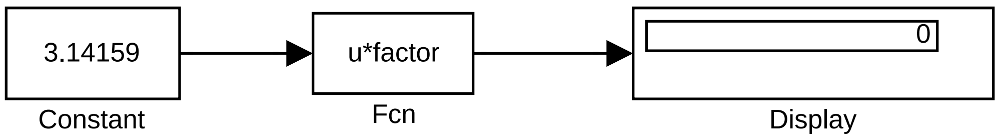
(int517w5.eps)

7.20 Digital Clock Example (digclock)



Digital Clock Example (digclock.eps)

7.21 Fcn Example Model (fcn eg)



Fcn Example ($fcn\_eg.eps$)