

Examples (sct)

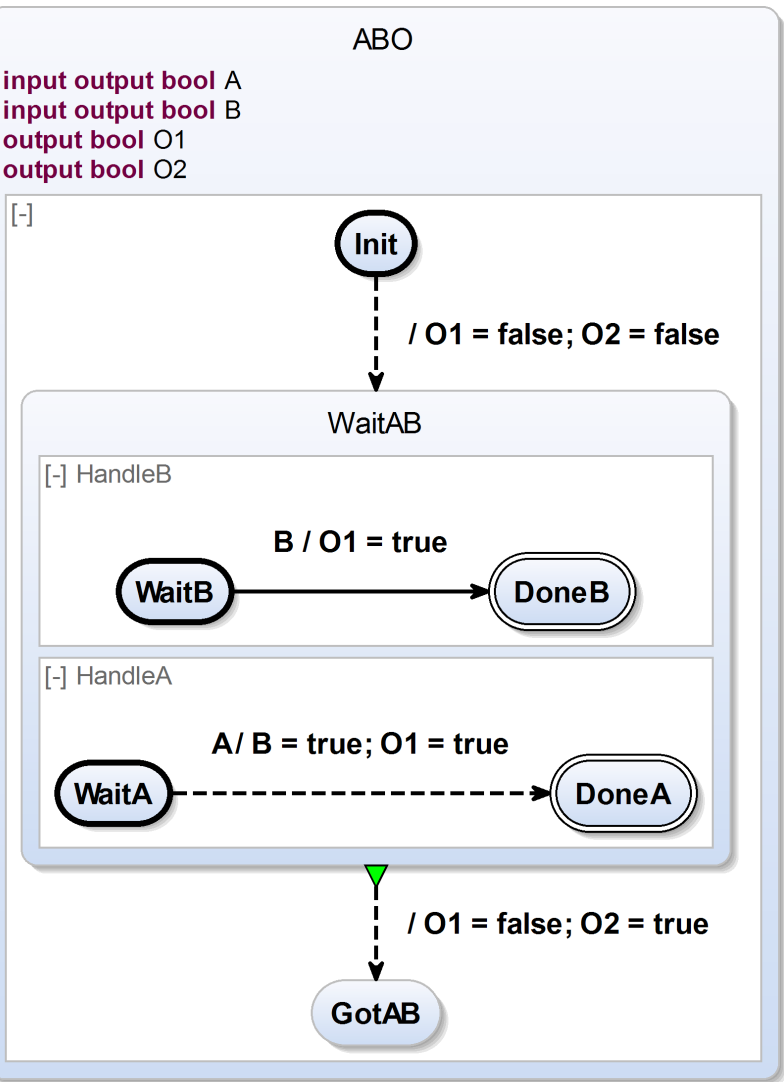
[Download](#)[Online Compiler](#)[Command Line Compiler](#)[Quick Start Guide](#)

SCCharts Examples

A few SCCharts examples...

Example	SCChart (Graphical)	SCChart (Textual SCT)
---------	---------------------	-----------------------

ABO



ABO.sct

```
@VHLayout
scchart ABO {
    input
    output bool
    A;
    input
    output bool
    B;
    output
    bool O1;
    output
    bool O2;
    region:

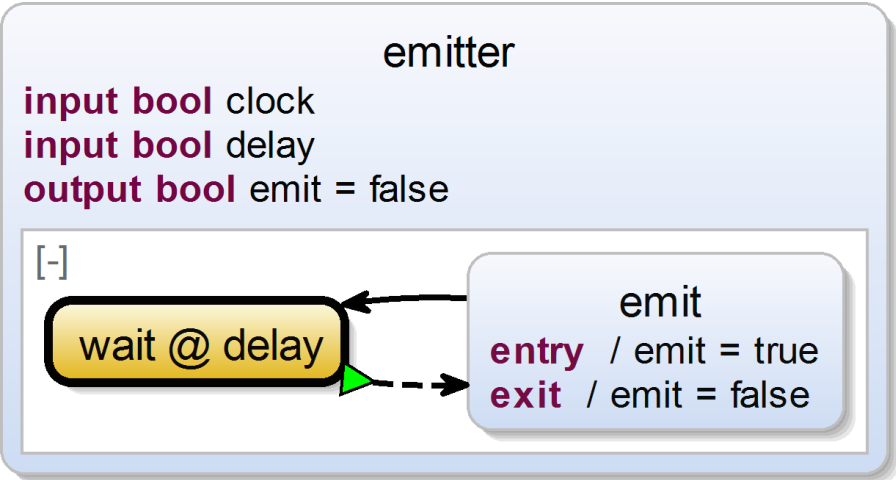
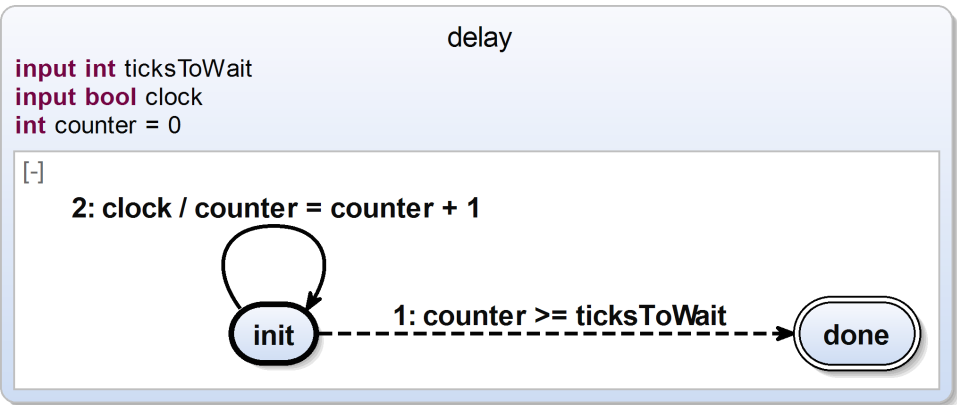
        initial
        state Init
        --> WaitAB
        immediate
        with / O1 =
        false; O2 =
        false;
        state
        WaitAB {
            region
            HandleA:
                initial
                state WaitA
                -->
                DoneA
            immediate
            with A / B =
            true; O1 =
            true;
            final
            state DoneA;
            region
            HandleB:
                initial
                state WaitB
                -->
                DoneB
            with B
            / O1 = true;
            final
            state DoneB;
        }
        --> GotAB
        with / O1 =
        false; O2 =
        true;
        state
        GotAB;
    }
}
```

clock

(Reference
d
SCCharts
Example)

To enable
the
referenced
SCCharts
feature,

activate
the Xtext
nature for
your
project!



shifter3.sct

```
scchart
delay {
    input int
    ticksToWait;
    input bool
    clock;
    int
    counter = 0;
    initial
    state init
    --> done
    immediate
    with
        counter
    >=
    ticksToWait
    --> init
    with
        clock /
        counter =
        counter + 1;
    final
    state done;
}
```

shifter3.sct

```
scchart
emitter {
    input bool
    clock;
    input bool
    delay;
    output
    bool emit =
    false;
    initial
    state wait
    references
    delay
    bind clock
    to clock,

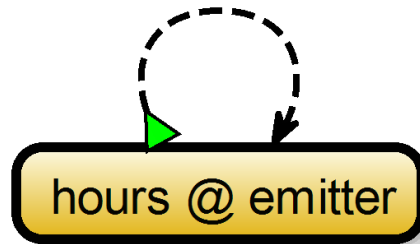
    ticksToWait
    to delay
    --> emit;

    state emit
    {
        entry /
        emit = true;
        exit /
        emit = false;
    }
    --> wait;
}
```

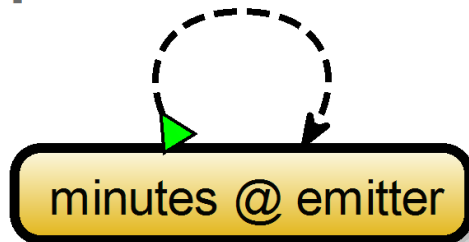
clock

input bool msClock
output bool second
output bool minute
output bool hour
const int SEC = 1000
const int MIN = 60
const int HOUR = 60

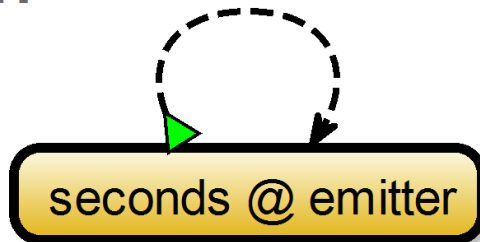
[-] hours



[-] minutes



[-] seconds



shifter3.sct

```
scchart
clock {
  input bool
  msClock;
  output
  bool second;
  output
  bool minute;
  output
  bool hour;
  const int
  SEC = 1000;
  const int
  MIN = 60;
  const int
  HOUR = 60;
  region
  seconds:
    initial
    state seconds
    references
    emitter
    bind clock
    to msClock,

    metrum to
    SEC,
    emit
    to second
    >->
    seconds;

    region
    minutes:
      initial
      state minutes
      references
      emitter
      bind clock
      to second,

      metrum to
      MIN,
      emit
      to minute
      >->
      minutes;

      region
      hours:
        initial
        state hours
        references
        emitter
        bind clock
        to minute,

        metrum to
        HOUR,
        emit
        to hour
        >-> hours;
}
```

SCCharts Evaluation Examples

In the following, you find some examples used for evaluation. Be advised that there currently is no tool integrated evaluation possible. Please refer to the SCCharts superpage for the [project status](#) and [known limitations](#). You may use our [online compiler](#) or the [command line compiler](#) to play around with the following SCCharts. On the right side you'll find the textual SCChart (SCT) description, on the left side you see the equivalent synthesized SCChart diagram for it. You can use our [online SCChart synthesis](#) for rendering SCCharts diagrams from textual SCCharts descriptions.

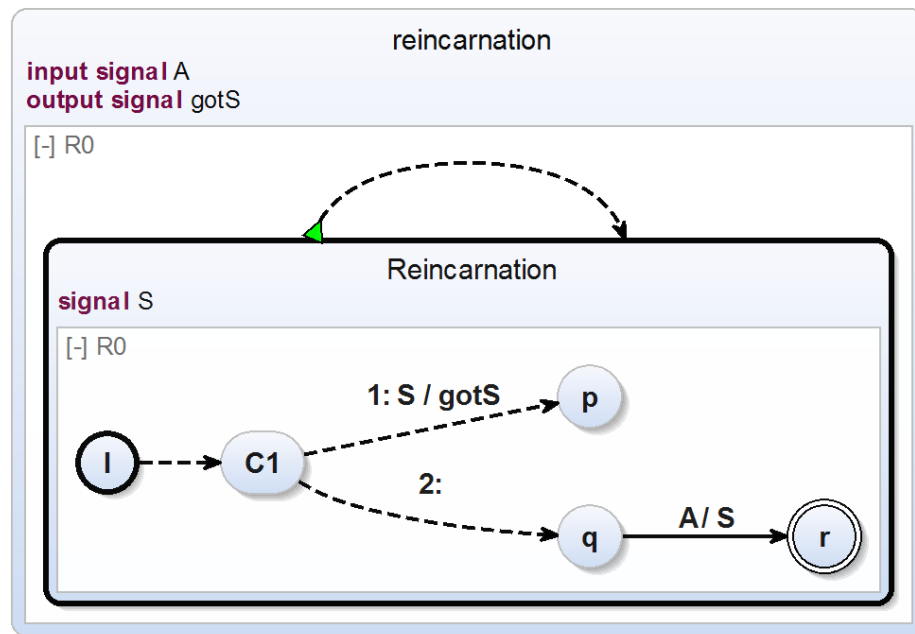
Example	SCChart (Graphical)	SCChart (Textual SCT)
shifter3	The graphical SCChart for 'shifter3' is composed of three regions stacked vertically. Each region contains a single state with a self-loop transition. Region R0 (top) has state I1 with the expression <code>pre(I) / S0(pre(val(I)))</code>. Region R2 (middle) has state I3 with the expression <code>pre(S1) / O(pre(val(S1)))</code>. Region R1 (bottom) has state I2 with the expression <code>pre(S0) / S1(pre(val(S0)))</code>.	<pre> shifter3 .sct scchar t shifte r3 { input signal int I = 1; signal int S0; signal int S1; output signal int O; region R1: initia l state I2 --> I2 with pre (S0) / S1 (pre (val (S0))) ; region R0: initia l state I1 --> I1 with pre (I) / S0(pre (val (I))) ; </pre>

```

region
R2:

  initia
  l
  state
  I3
  -->
  I3
  with
  pre
  (S1)
  / 0
  (pre
  (val
  (S1)))
  ;
  }

```

reincarn
ation

```

reincarnat
sct
    scchar
    t
    reincarnatio
    n {
        input
        signal
        A;

        output
        signal
        gotS;

        region
        R0:

        initia
        l
        state
        Reinca
        rnatio
        n {

        signal
        S;

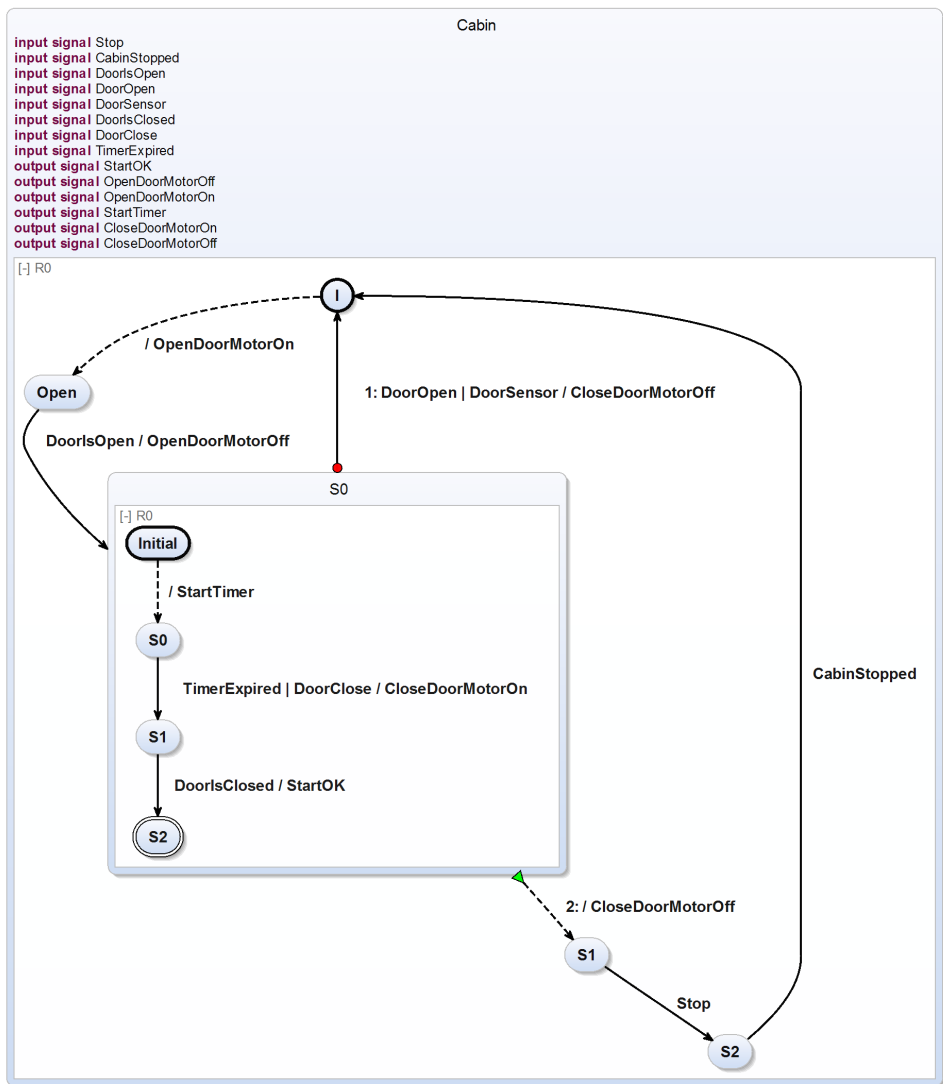
        region
        R0:

        initia
        l
        state
        I
        --
        > C1
        immedi
        ate;

        state
        p;
    }
}

```

			<pre> final state r; state q -- > r with A / S; state C1 -- > p immedi ate with S / gotS -- > q immedi ate; } >-> Reinca rnatio n; } </pre>
cabin			cabin.sct <pre> scchar t Cabin { input signal Stop; input signal CabinS topped ; input signal DoorIs Open; input signal DoorOp en; input signal DoorSe nsor; input signal DoorIs </pre>



```

Closed
;

input
signal
DoorCl
ose;

input
signal
TimerE
xpired
;

output
signal
StartO
K;

output
signal
OpenDo
orMoto
rOff;

output
signal
OpenDo
orMoto
rOn;

output
signal
StartT
imer;

output
signal
CloseD
oorMot
orOn;

output
signal
CloseD
oorMot
orOff;

region
R0:

initia
l
state
I
-->
Open
immedi
ate
with
/
OpenDo
orMoto
rOn;

state
Open
-->
S0
with
DoorIs

```

```
Open
/
OpenDo
orMoto
rOff;

state
S0 {

region
R0:

initia
l
state
Initia
l
--
> S0
immedi
ate
with
/
StartT
imer;

state
S0
--
> S1
with
TimerE
xpired
|
DoorCl
ose /
CloseD
oorMot
orOn;

state
S1
--
> S2
with
DoorIs
Closed
/
StartO
K;

final
state
S2;
}
o->
I
with
DoorOp
en |
DoorSe
nsor
/
CloseD
oorMot
orOff
>->
S1
with
/
CloseD
```

```
oorMot
orOff;

state
S1
-->
S2
with
Stop;

state
S2
-->
I
with
CabinS
topped
;
}
```

```
reactor-
control.
sct

scchar
t
Module
_React
orCont
rol {

output
signal
PullOu
tRods;

output
signal
PushIn
Rods;

input
signal
Start;

input
signal
OverHe
ated;

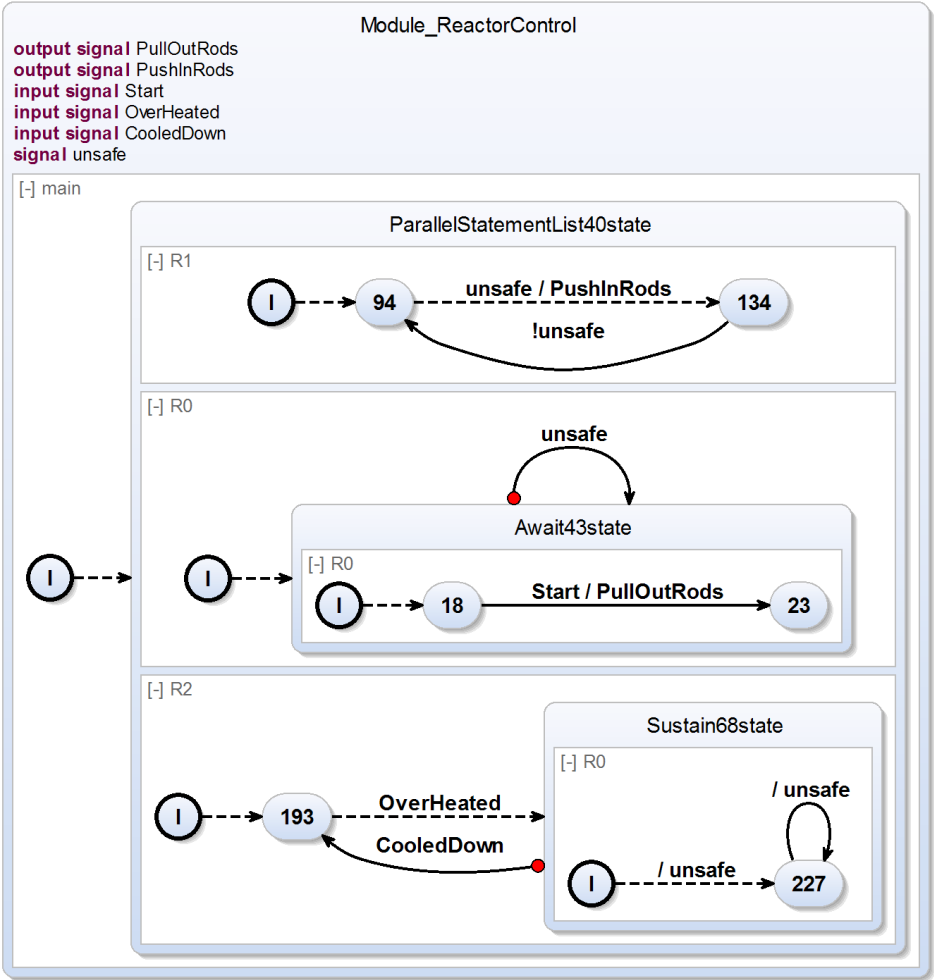
input
signal
Cooled
Down;

signal
unsafe
;

region
main:

initia
l
state
init0
```

reactor
control



```
"I"
-->
ParallelState
elementList40state
immediate;

state
ParallelState
elementList40state {

region
R0:

initial
state
init1
"I"
--
>
Await43state
immediate;

state
Await43state
{

region
R0:

initial
state
init2
"I"

-->
S18
immediate;

state
S18
"18"

-->
S23
with
Start
/
PullOutRods;

state
S23
"23";
}
o-
>
Await43state
```

	with unsafe ; region R1: initia l state init3 "I" -- > S94 immedi ate; state S94 "94" -- > S134 immedi ate with unsafe / PushIn Rods; state S134 "134" -- > S94 with ! unsafe ; region R2: initia l state init4 "I" -- > S193 immedi ate; state S193 "193" -- > Sustai n68sta te immedi ate with OverHe ated; state Sustai n68sta
--	---

```
te {  
  
  region  
  R0:  
  
  initia  
  l  
  state  
  init5  
  "I"  
  
  -->  
  S227  
  immedi  
  ate  
  with  
  /  
  unsafe  
  ;  
  
  state  
  S227  
  "227"  
  
  -->  
  S227  
  with  
  /  
  unsafe  
  ;  
    }  
  o-  
  >  
  S193  
  with  
  Cooled  
  Down;  
    };  
}
```