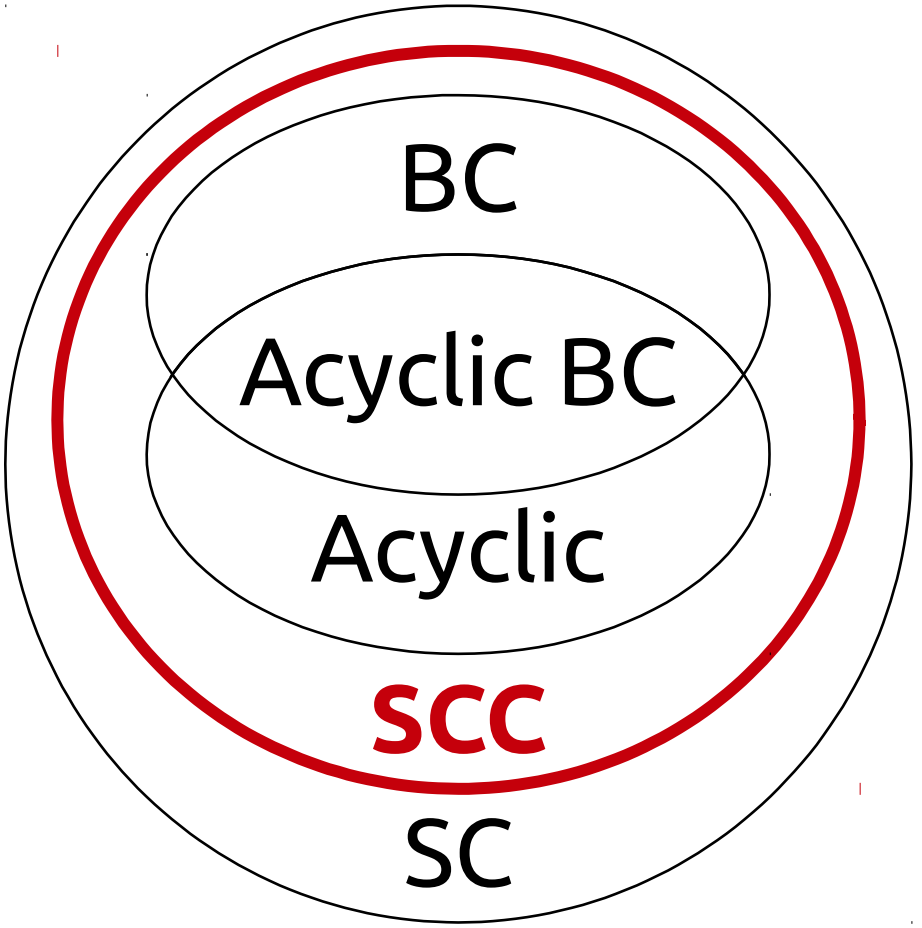
A Sequentially Constructive Circuit Semantics for Esterel

Alexander Schulz-Rosengarten,
Reinhard von Hanxleden, Kiel University
and Michael Mendler, Bamberg University

Program Classes

Berry Constructive (BC) Esterel

- Circuit semantics
  → Delay insensitive circuits



Sequentially Constructive Circuit (SCC) Semantics

Overview

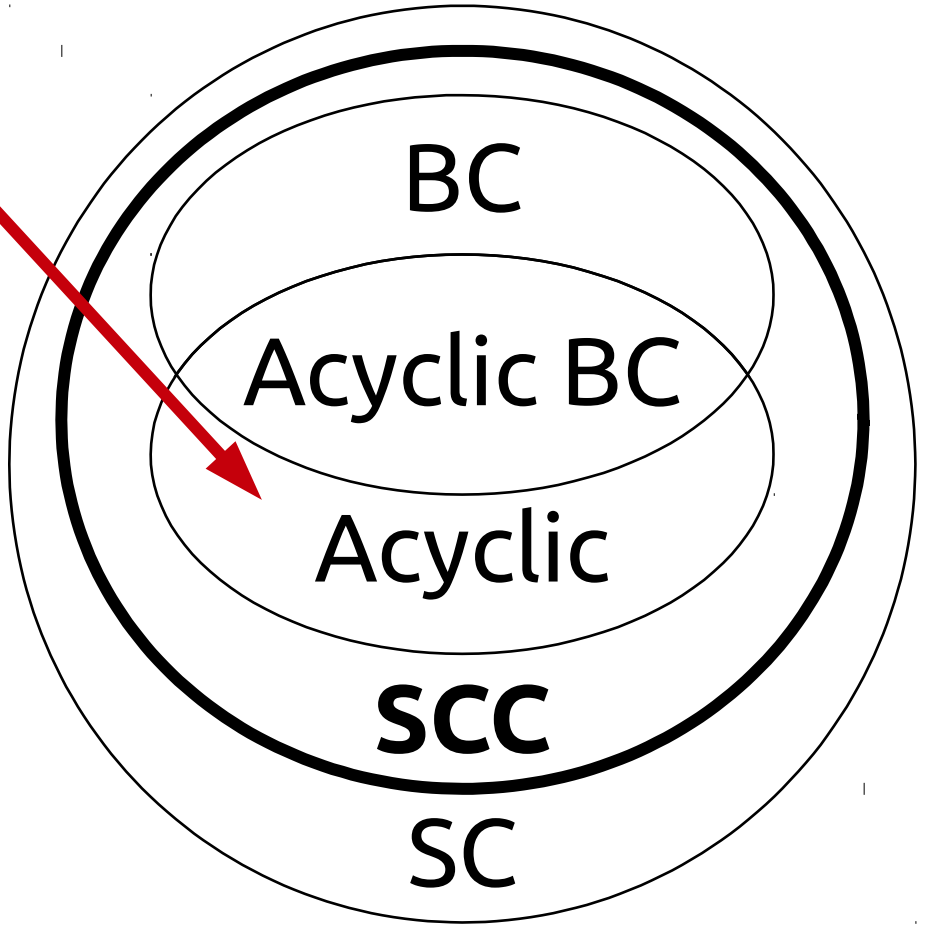
 Visibility

 Visibility in circuits

emit S_0 Visibility on source level

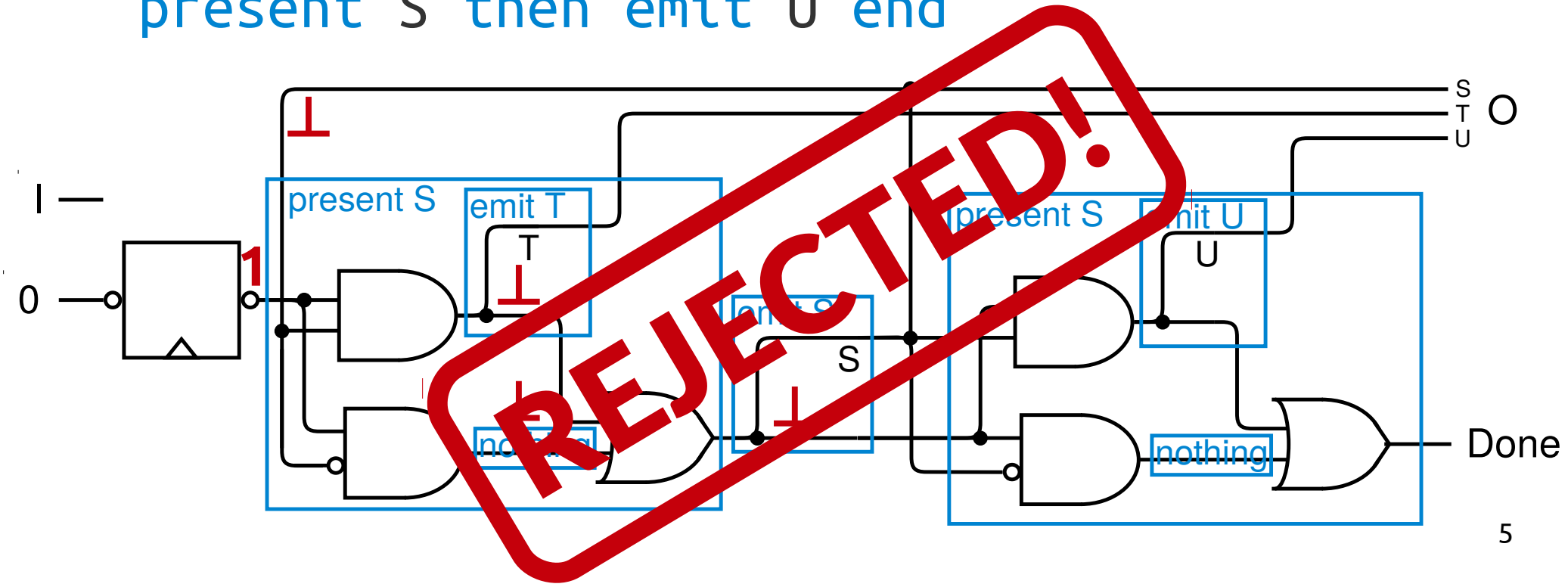
Program Classes

```
module OffOn:
```



Esterel Constructive Circuits

```
module OffOn:
output S, T, U;
present S then emit T end;
emit S;
present S then emit U end
```



Esterel vs. Java / C / SCL

```
module OffOn:
output S, T, U;
present S then
  emit T end;
emit S;
present S then
  emit U end
```

```
bool S = T = U = false;
if S {
  T = true };
S = true;
if S {
  U = true };
```

Signals encoded as booleans

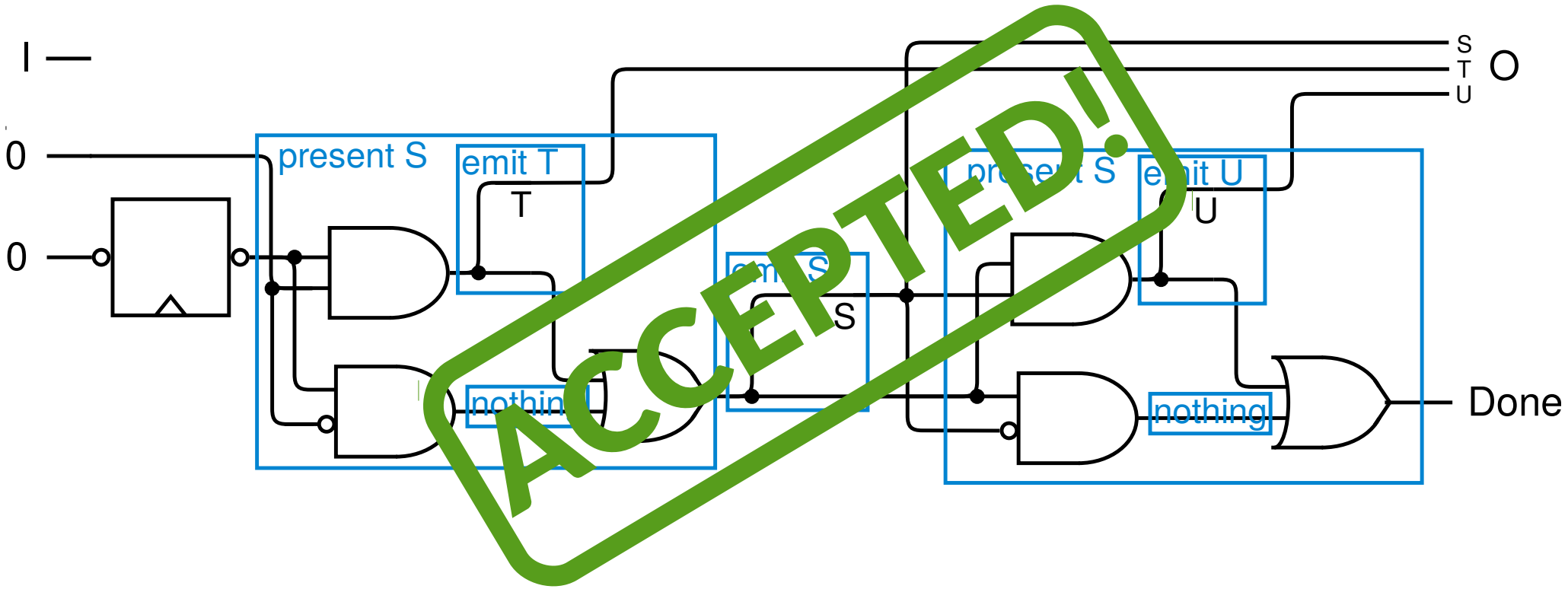
Proposal

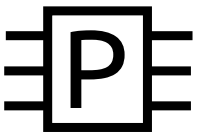
Constructive coherence law:
A signal is present/absent iff it must/cannot be emitted

Sequentially Constructive Coherence Law:
A signal is present/absent iff it must/cannot be emitted
concurrently or sequentially preceding

We say that an emit E is **SC-visible** to a present test P if:
1) E is concurrent to P or
2) E sequentially precedes P

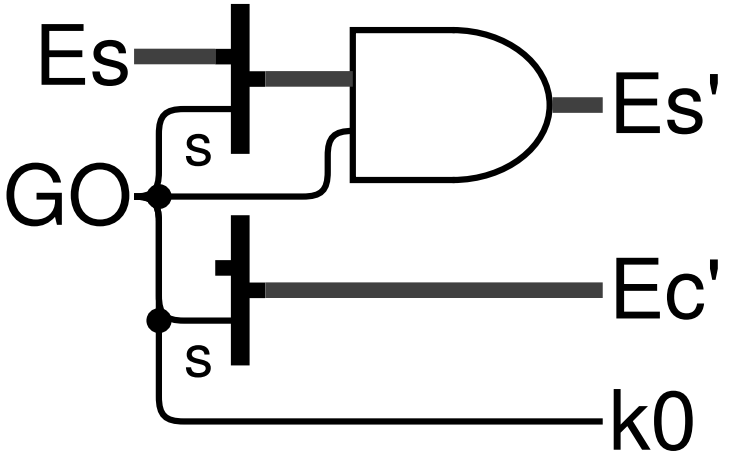
SC-Visibility in Circuits





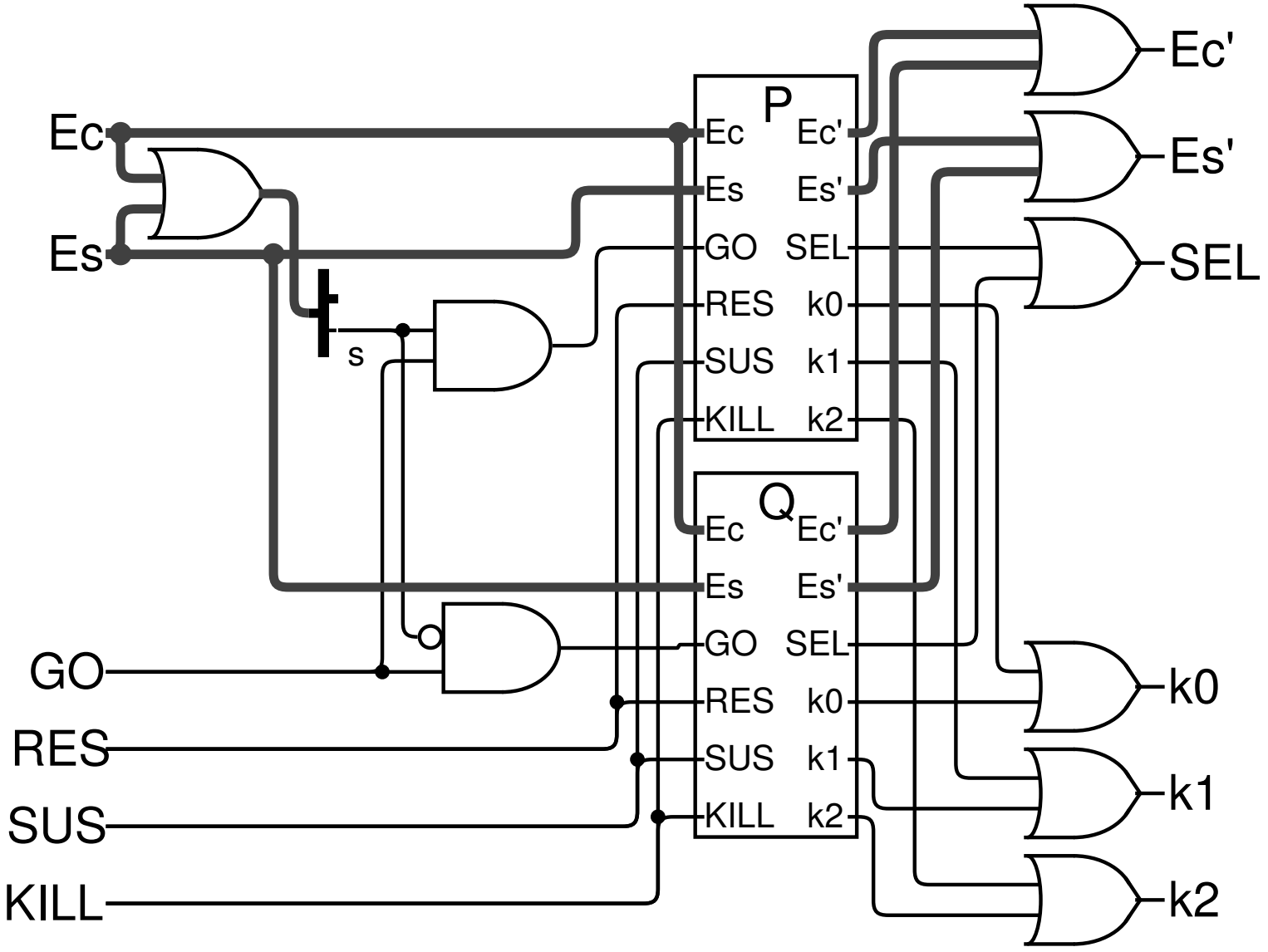
SCC Emit

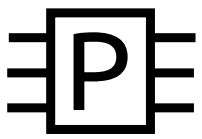
!s



SCC Present Test

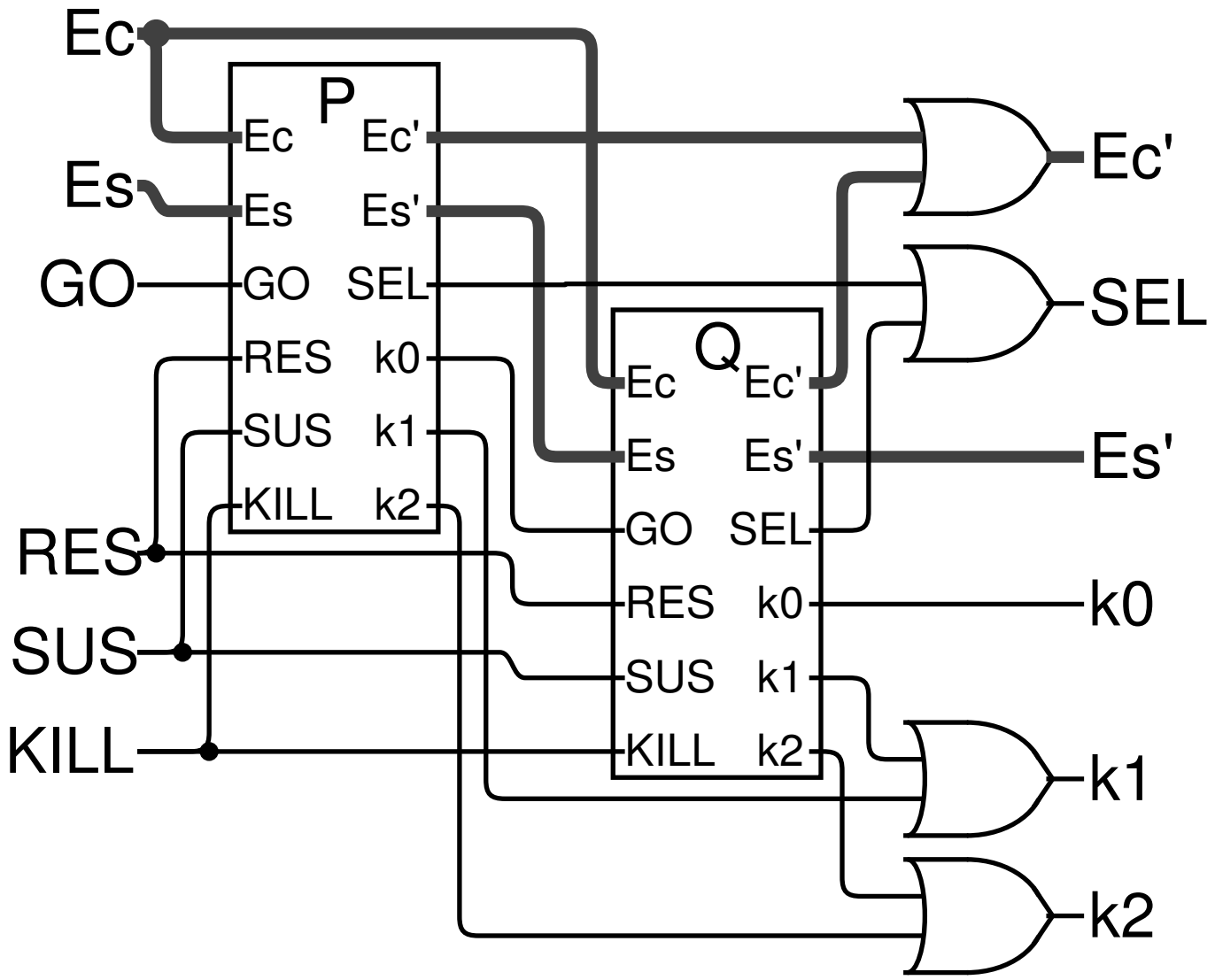
s ? P , Q

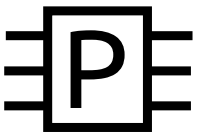




SCC Sequence

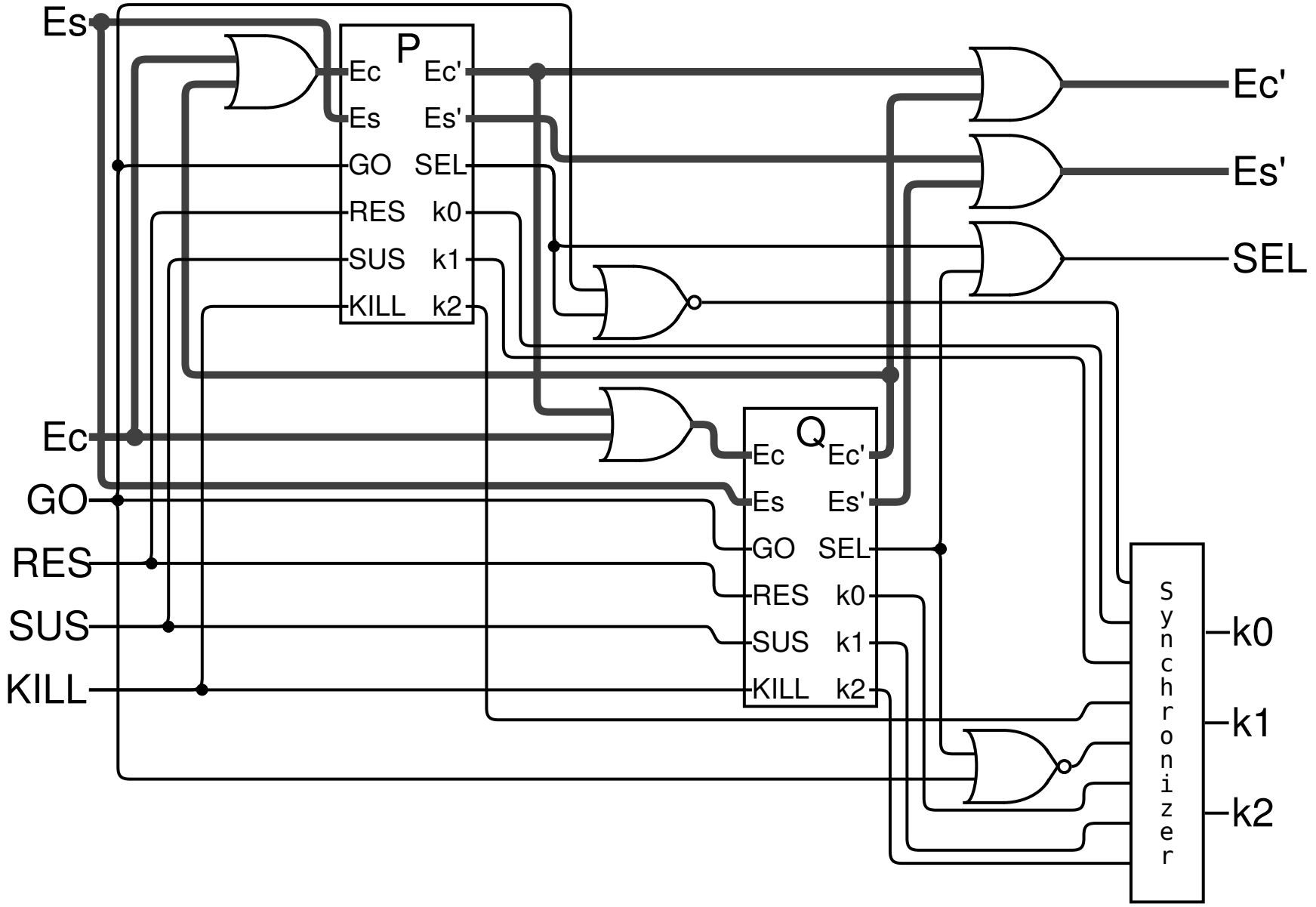
P ; Q

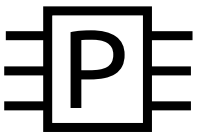




SCC Parallel

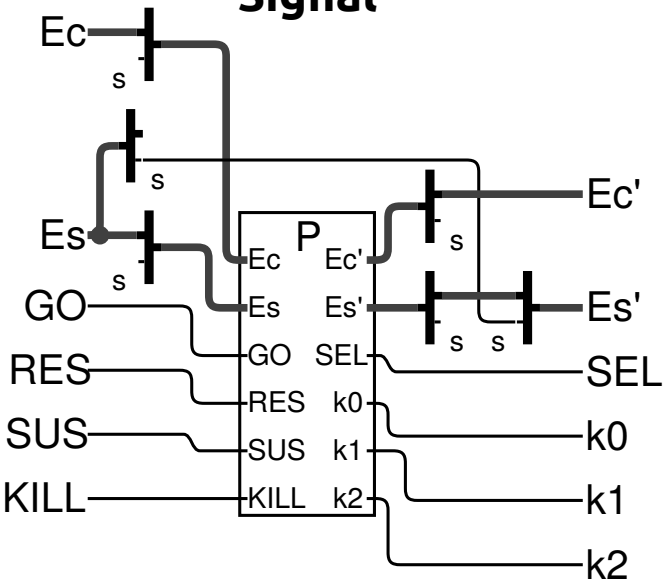
P || Q



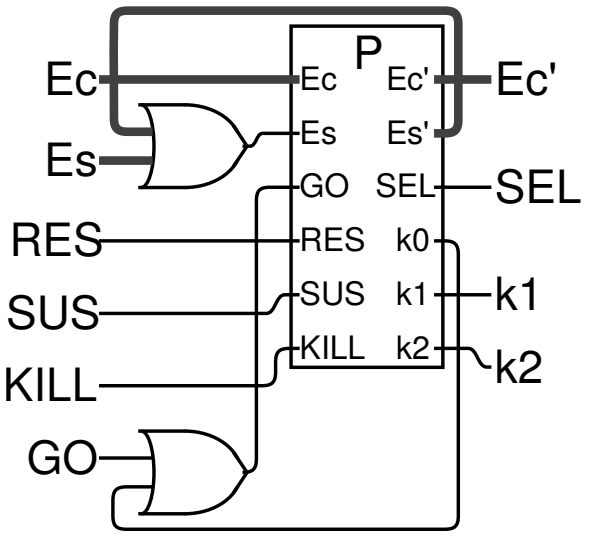


Further SCC Rules

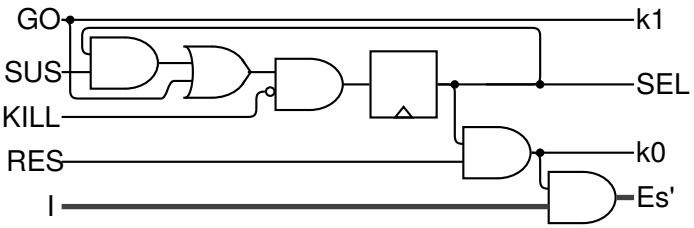
Signal



Loop



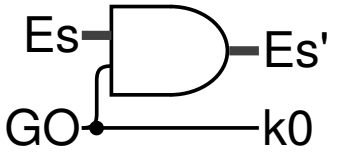
Pause



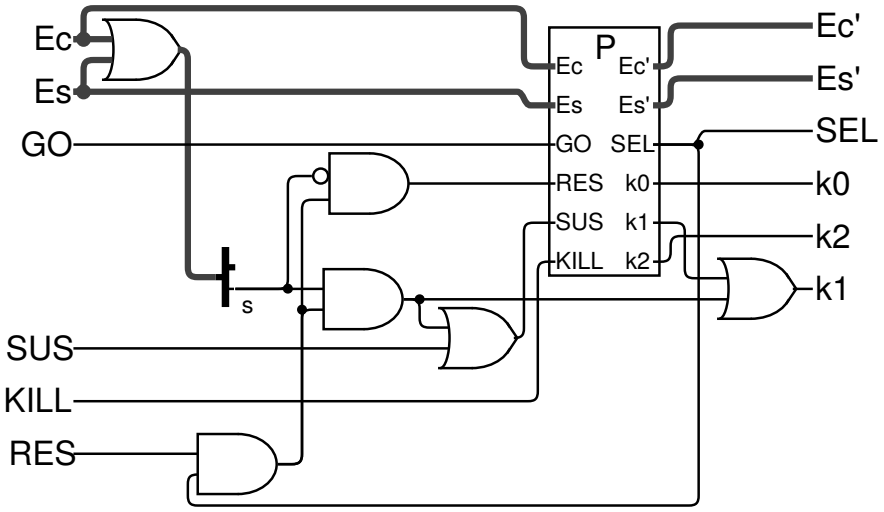
Exit

GO–k2

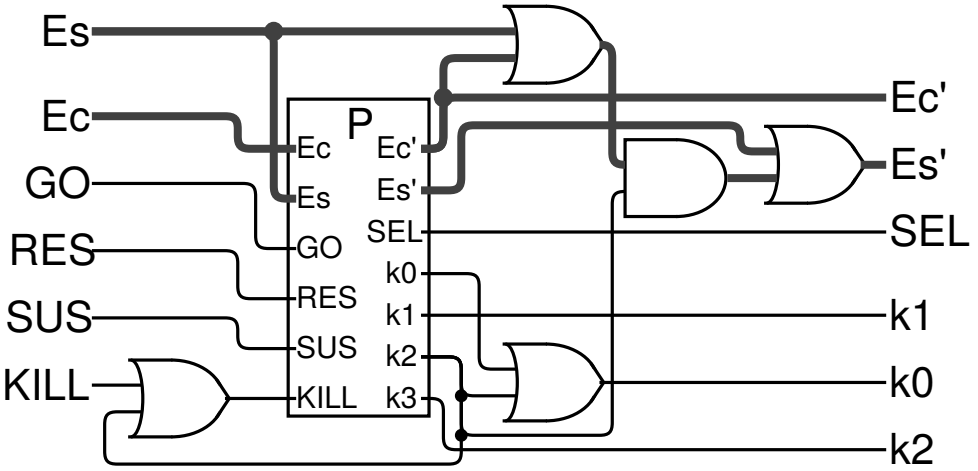
Nothing

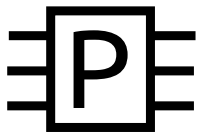


Suspend



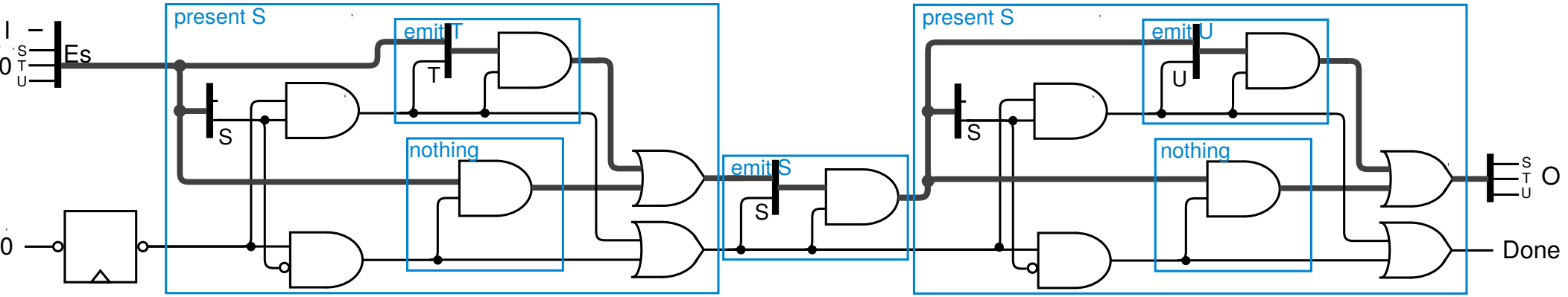
Trap





OffOn with SCC

```
module OffOn:
output S, T, U;
present S then emit T end;
emit S;
present S then emit U end
```

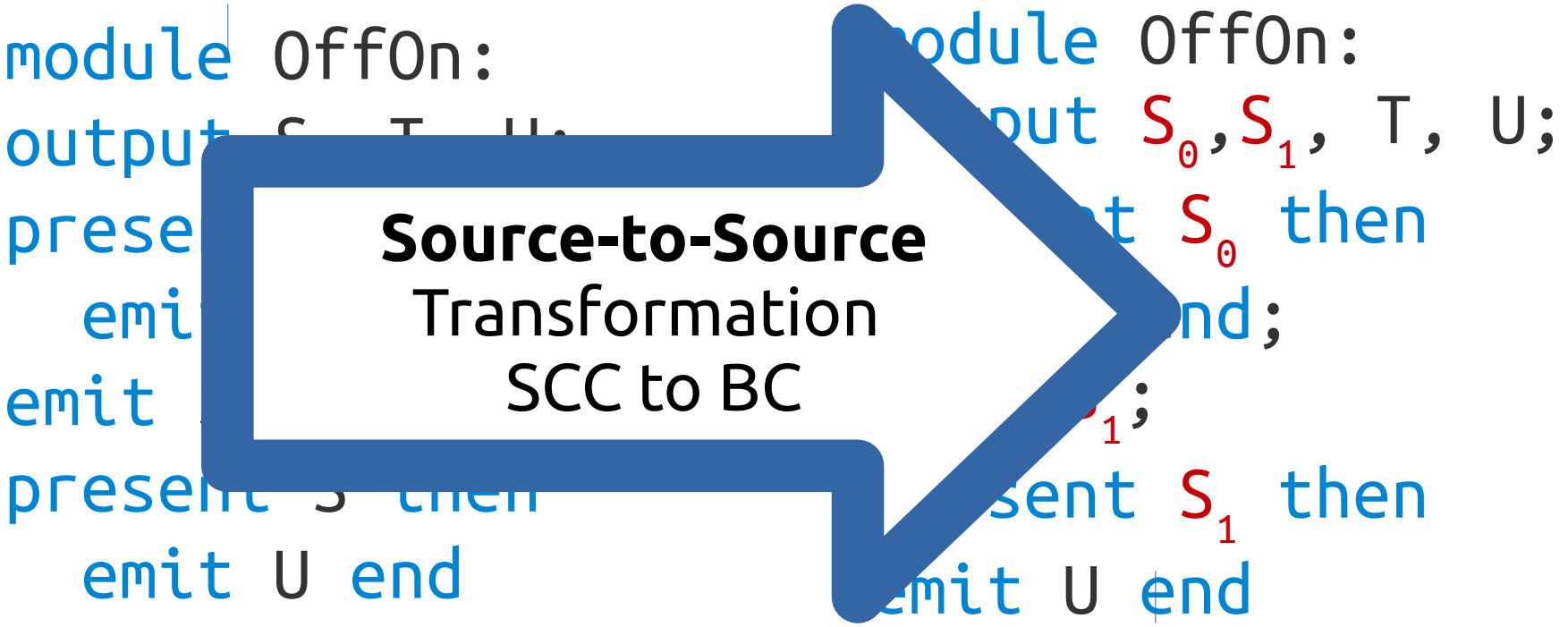


emit S_0

Back to Source Level

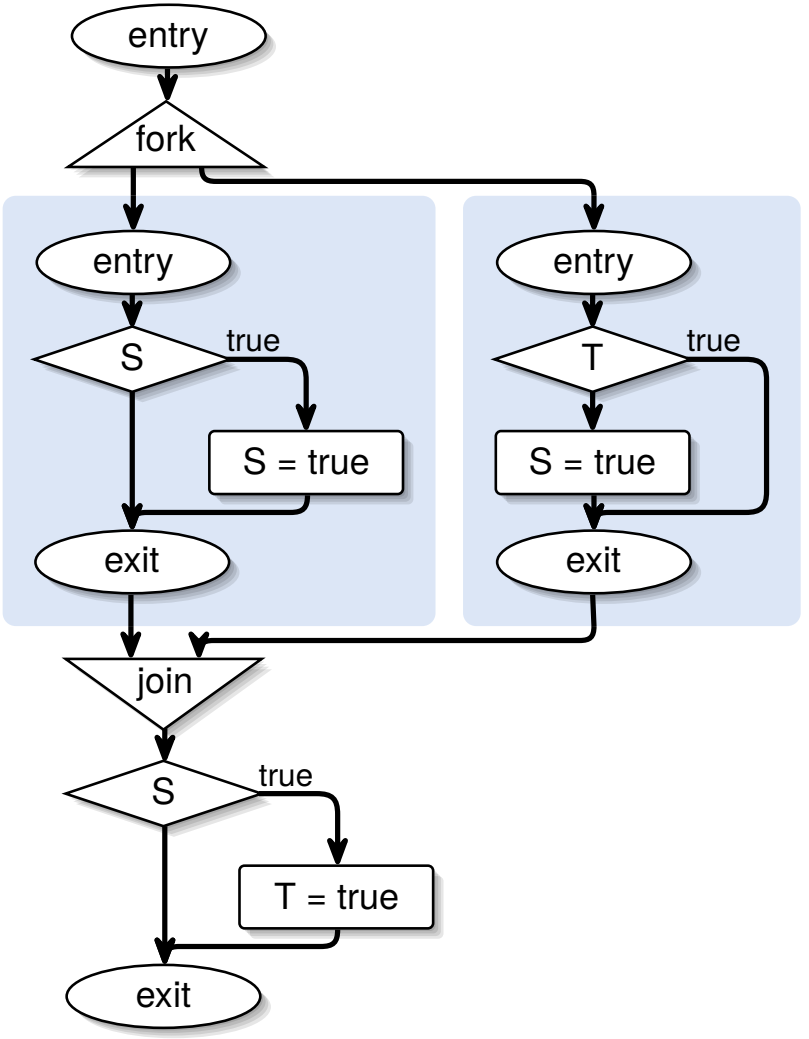
Applying SC-visibility to Esterel on source level.

Static Single Assignment Form



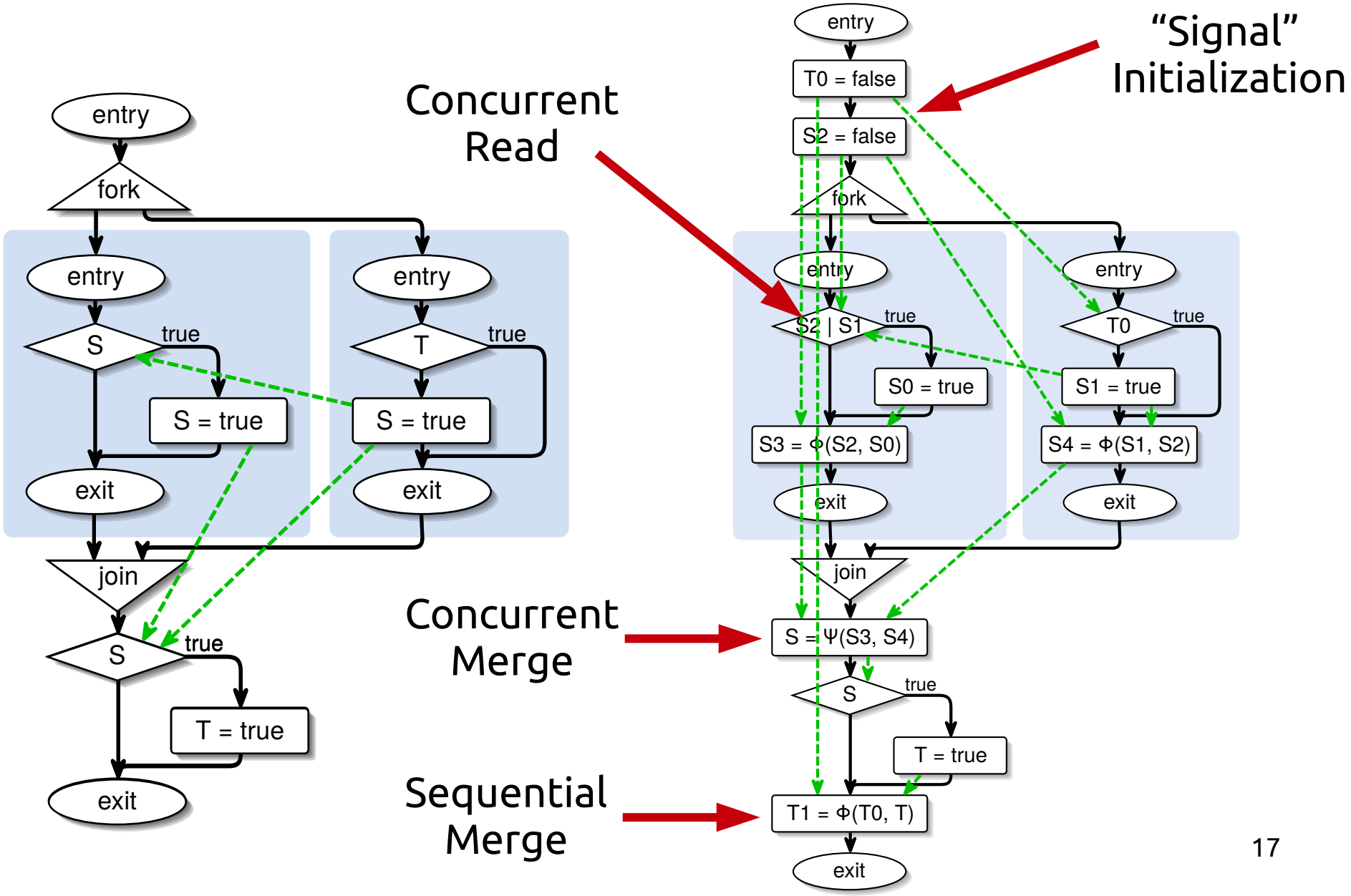
Concurrent SSA for Esterel

```
module ST:
output S, T;
[
  present S then
    emit S end
||
  present T else
    emit S end
];
present S then
    emit T end
```



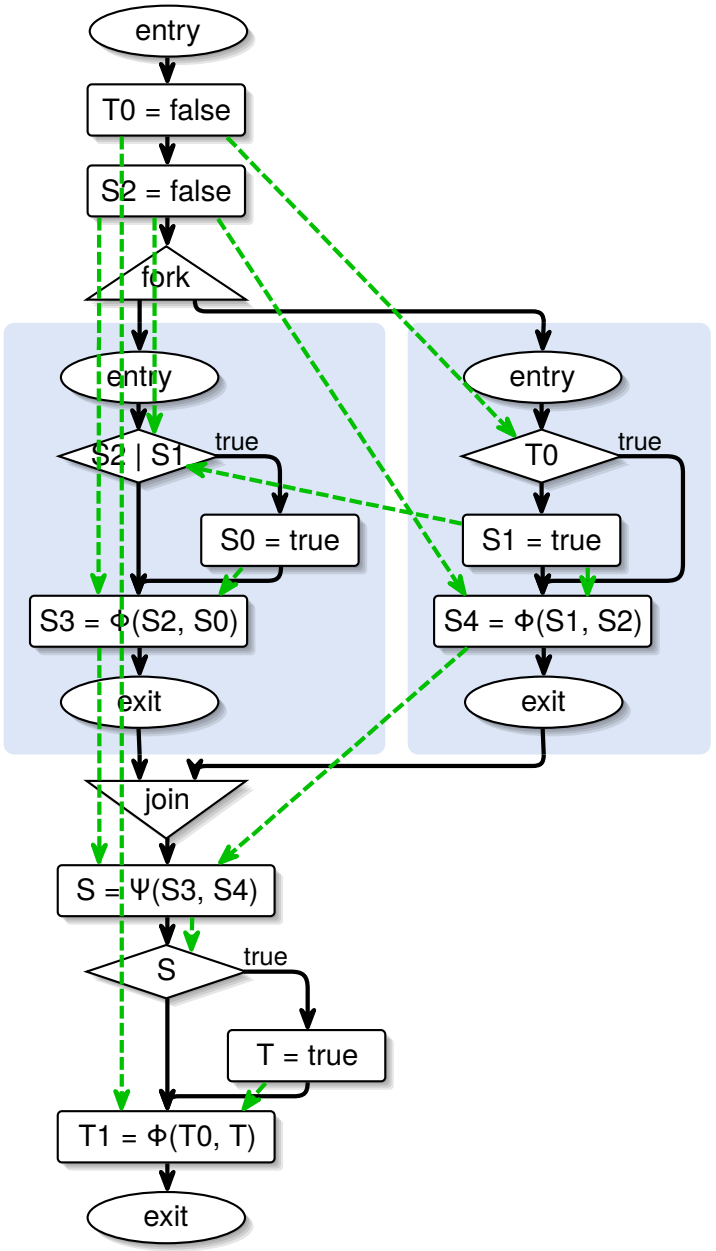
emit S_0

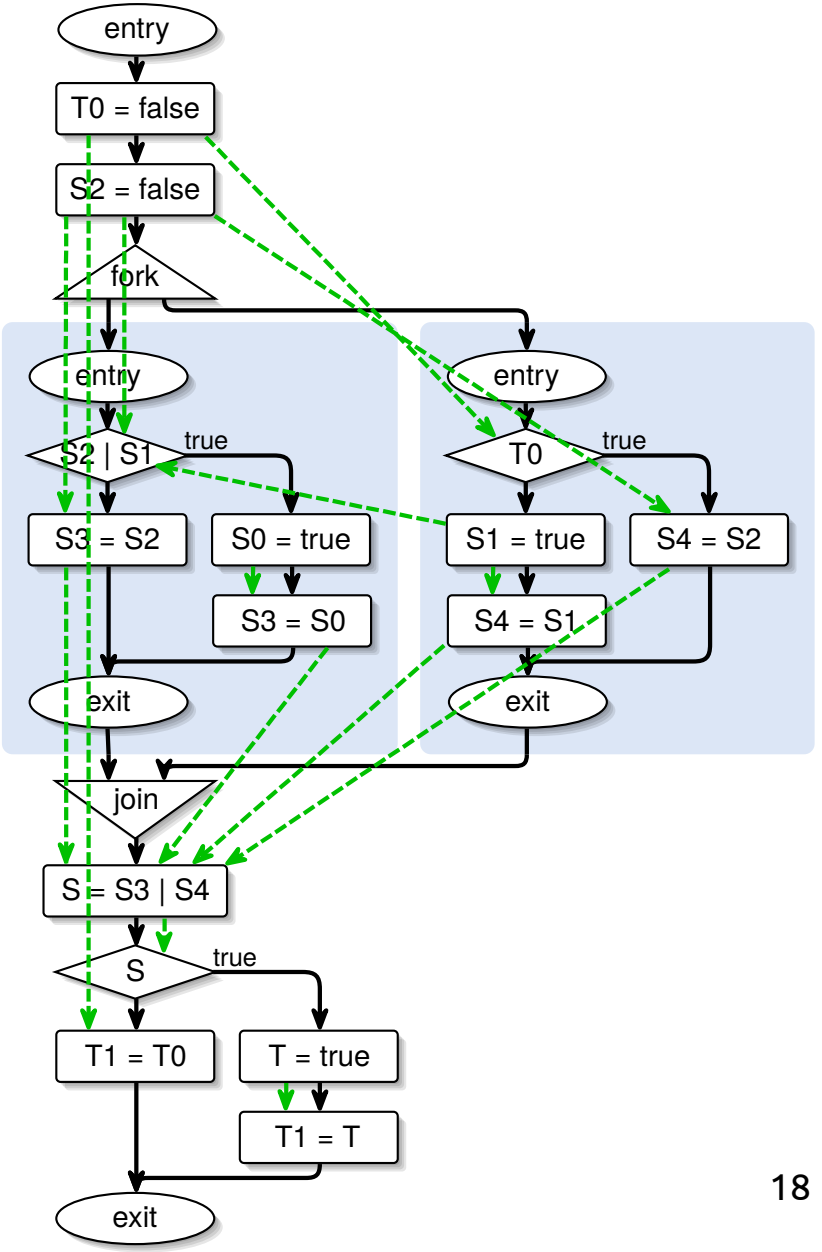
Concurrent SSA for Esterel



emit S_0

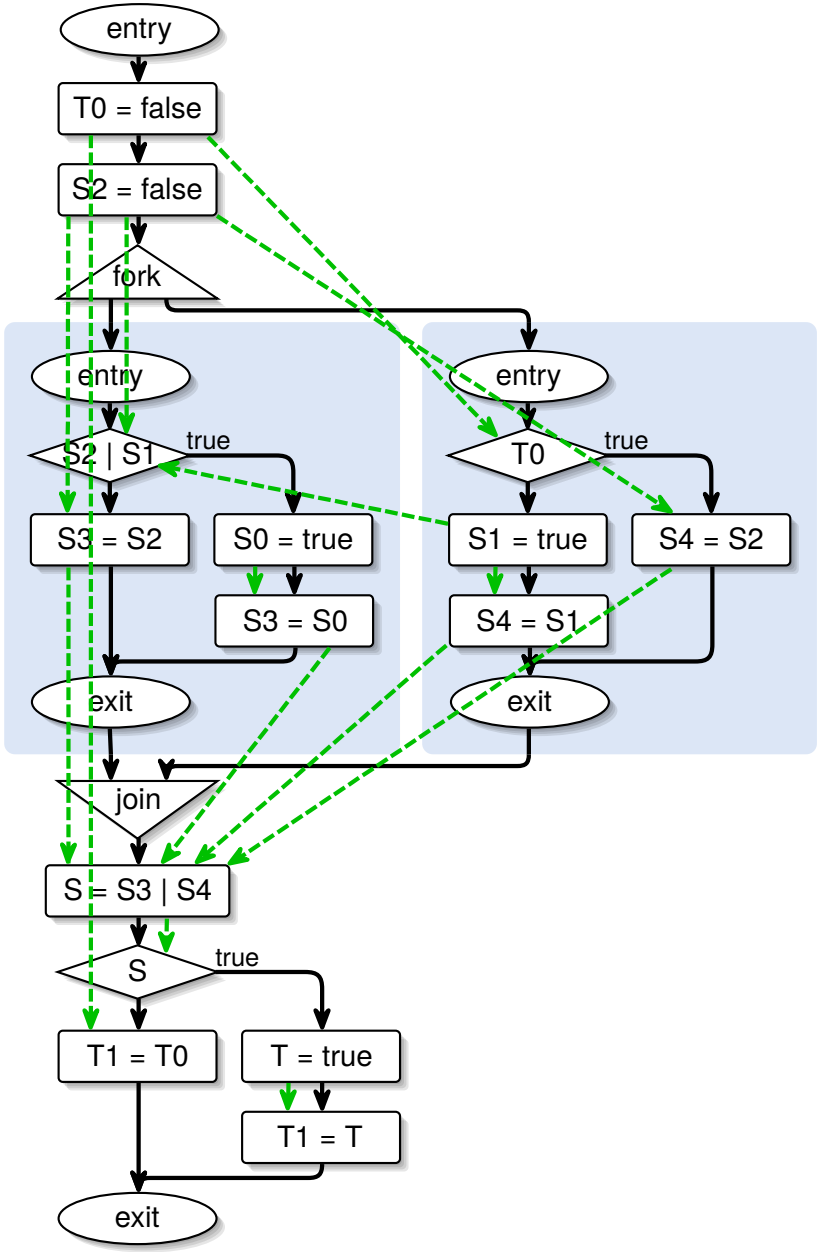
Concurrent SSA for Esterel

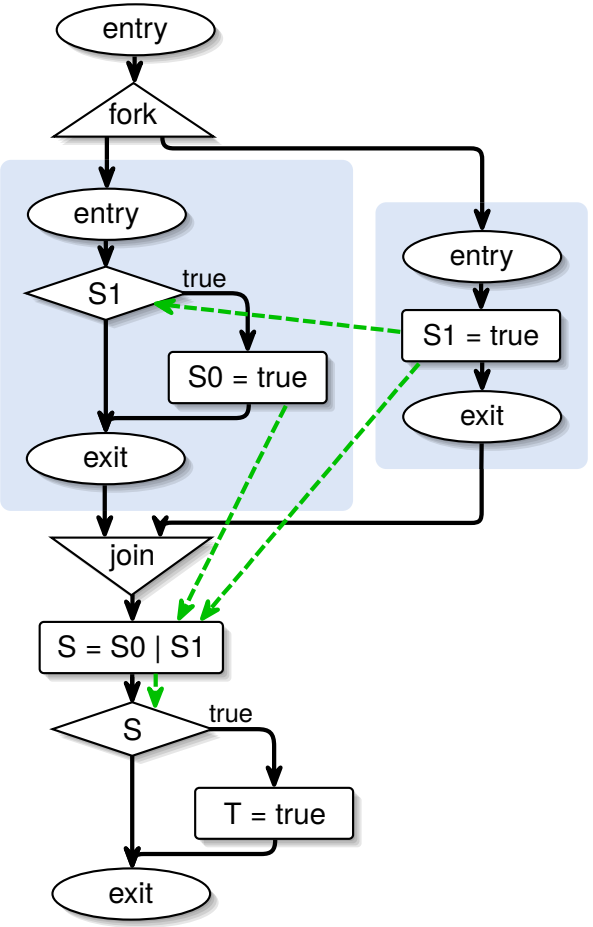




emit S_0

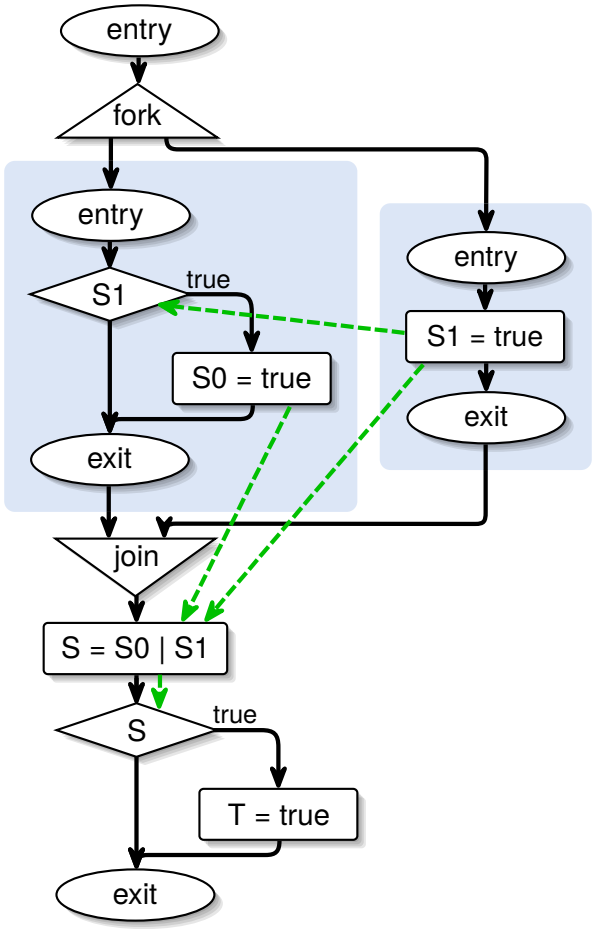
Concurrent SSA for Esterel





emit S_0

Concurrent SSA for Esterel




```
module ST-SSA:
output S, T;
signal S0, S1 in
[
   present S1 then
      emit S0 end
||
   emit S1
];
present S0 or S1 then
      emit S end;
present S then
      emit T end
```

Summary

- Sequentially Constructive Circuit (SCC) semantics for Esterel
- Source-to-source transformation from SCC programs into equivalent Berry constructive programs
- Claim conservativeness
  - BC => SCC
  - Preserving semantics
  - Proof sketch (WIP)

The End