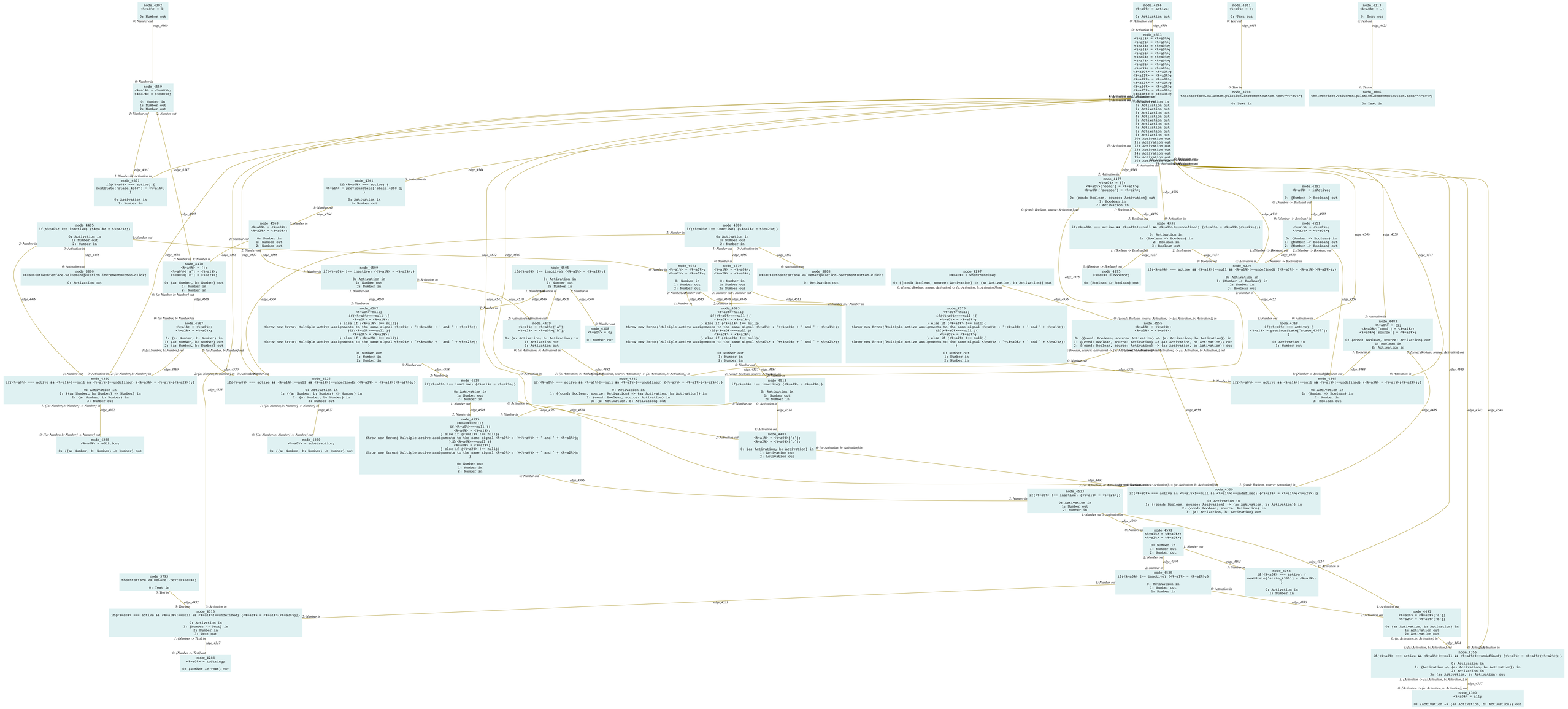
node_4302
<k=a5k> = 1;
0: Number out
edge_4560
0: Number out
node_4559
<k=a1k> = <k=a5k>;
<k=a2k> = <k=a5k>;
0: Number in
1: Number in
2: Number out
1: Number out
2: Number out
edge_4561
edge_4547
edge_4562
edge_4544
node_4371
if(<k=a5k> === active) {
nextState['state_4367'] = <k=a1k>;
}
0: Activation in
1: Number in
node_4361
if(<k=a5k> === active) {
<k=a1k> = previousState['state_4360'];
}
0: Activation in
1: Number out
0: Activation in
1: Number out
node_4563
<k=a1k> = <k=a5k>;
<k=a2k> = <k=a5k>;
0: Number in
1: Number out
2: Number out
0: Number in
edge_4564
node_4495
if(<k=a5k> !== inactive) {<k=a1k> = <k=a2k>;}
0: Activation in
1: Number out
2: Number in
1: Number out
edge_4496
0: Activation in
node_3800
<k=a3k>=theInterface.valueManipulation.incrementButton.click;
0: Activation out
edge_4536
edge_4565
edge_4537
edge_4566
node_4470
<k=a5k> = {}
<k=a5k>['a'] = <k=a1k>;
<k=a5k>['b'] = <k=a2k>;
0: {a: Number, b: Number} out
1: Number in
2: Number in
0: {a: Number, b: Number} out
edge_4572
edge_4540
node_4509
if(<k=a5k> !== inactive) {<k=a1k> = <k=a2k>;}
0: Activation in
1: Number out
2: Number in
0: Activation in
edge_4568
edge_4504
edge_4590
node_4587
<k=a3k>=null;
if(<k=a5k>===null){
throw new Error('Multiple active assignments to the same signal <k=a5k> : '+<k=a5k> + ' and ' + <k=a1k>);
}if(<k=a3k>===null){
<k=a3k> = <k=a2k>;
} else if (<k=a2k> !== null){
throw new Error('Multiple active assignments to the same signal <k=a5k> : '+<k=a5k> + ' and ' + <k=a2k>);
}
0: Number out
1: Number in
2: Number in
0: Number out
node_4567
<k=a1k> = <k=a3k>;
<k=a2k> = <k=a5k>;
0: {a: Number, b: Number} in
1: {a: Number, b: Number} out
2: {a: Number, b: Number} out
1: {a: Number, b: Number} out
2: {a: Number, b: Number} out
edge_4569
edge_4570
node_4320
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {a: Number, b: Number} -> Number} in
2: {a: Number, b: Number} in
3: Number out
1: {{a: Number, b: Number} -> Number} in
edge_4322
0: {{a: Number, b: Number} -> Number} out
node_4288
<k=a5k> = addition;
0: {{a: Number, b: Number} -> Number} out
edge_4588
edge_4535
node_4325
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {{a: Number, b: Number} -> Number} in
2: {a: Number, b: Number} in
3: Number out
1: {{a: Number, b: Number} -> Number} in
edge_4327
0: {{a: Number, b: Number} -> Number} out
node_4290
<k=a5k> = subtraction;
0: {{a: Number, b: Number} -> Number} out
node_4500
if(<k=a5k> !== inactive) {<k=a1k> = <k=a2k>;}
0: Activation in
1: Number out
2: Number in
1: Number out
2: Number in
edge_4503
node_4571
<k=a1k> = <k=a5k>;
<k=a2k> = <k=a5k>;
0: Number in
1: Number out
2: Number out
2: Number out
node_4579
<k=a1k> = <k=a5k>;
<k=a2k> = <k=a5k>;
0: Number in
1: Number out
2: Number out
2: Number out
edge_4580
edge_4578
edge_4586
node_4583
<k=a3k>=null;
if(<k=a5k>===null){
throw new Error('Multiple active assignments to the same signal <k=a5k> : '+<k=a5k> + ' and ' + <k=a1k>);
}if(<k=a3k>===null){
<k=a3k> = <k=a2k>;
} else if (<k=a2k> !== null){
throw new Error('Multiple active assignments to the same signal <k=a5k> : '+<k=a5k> + ' and ' + <k=a2k>);
}
0: Number out
1: Number in
2: Number in
0: Number out
edge_4585
edge_4551
node_4505
if(<k=a5k> !== inactive) {<k=a1k> = <k=a2k>;}
0: Activation in
1: Number out
2: Number in
1: Number out
2: Number in
edge_4542
edge_4530
edge_4589
edge_4506
edge_4508
node_4479
<k=a1k> = <k=a5k>['a'];
<k=a2k> = <k=a5k>['b'];
0: {a: Activation, b: Activation} in
1: Activation out
2: Activation out
0: {a: Activation, b: Activation} in
0: Number out
node_4308
<k=a5k> = 0;
0: Number out
edge_4588
edge_4482
edge_4537
edge_4584
node_4340
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {(cond: Boolean, source: Activation) -> {a: Activation, b: Activation}} in
2: {cond: Boolean, sources: Activation} in
3: {a: Activation, b: Activation} out
3: {a: Activation, b: Activation} in
2: {cond: Boolean, source: Activation} in
node_4513
if(<k=a5k> !== inactive) {<k=a1k> = <k=a2k>;}
0: Activation in
1: Number out
2: Number in
1: Number out
edge_4598
edge_4519
edge_4514
node_4595
<k=a3k>=null;
if(<k=a5k>===null){
<k=a1k> = <k=a1k>;
} else if (<k=a1k> !== null){
throw new Error('Multiple active assignments to the same signal <k=a5k> : '+<k=a5k> + ' and ' + <k=a1k>);
}if(<k=a3k>===null){
<k=a2k> = <k=a2k>;
} else if (<k=a2k> !== null){
throw new Error('Multiple active assignments to the same signal <k=a5k> : '+<k=a5k> + ' and ' + <k=a2k>);
}
0: Number out
1: Number in
2: Number in
0: Number out
node_4487
<k=a1k> = <k=a5k>['a'];
<k=a2k> = <k=a5k>['b'];
0: {a: Activation, b: Activation} in
1: Activation out
2: Activation out
0: {a: Activation, b: Activation} in
edge_4596
edge_4493
node_4523
if(<k=a5k> !== inactive) {<k=a1k> = <k=a2k>;}
0: Activation in
1: Number out
2: Number in
2: Number in
node_4350
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {(cond: Boolean, sources: Activation) -> {a: Activation, b: Activation}} in
2: {cond: Boolean, source: Activation} in
3: {a: Activation, b: Activation} out
1: Number out
edge_4592
node_4591
<k=a1k> = <k=a5k>;
<k=a2k> = <k=a5k>;
0: Number in
1: Number out
2: Number out
2: Number out
edge_4594
edge_4593
edge_4524
node_4529
if(<k=a5k> !== inactive) {<k=a1k> = <k=a2k>;}
0: Activation in
1: Number out
2: Number in
1: Number out
node_4364
if(<k=a5k> === active) {
nextState['state_4360'] = <k=a1k>;
}
0: Activation in
1: Number in
0: Activation in
edge_4530
node_4491
<k=a1k> = <k=a5k>['a'];
<k=a2k> = <k=a5k>['b'];
0: {a: Activation, b: Activation} in
1: Activation out
2: Activation out
0: {a: Activation, b: Activation} in
1: {Activation -> {a: Activation, b: Activation}} in
2: Activation in
3: {a: Activation, b: Activation} out
1: {Activation -> {a: Activation, b: Activation}} in
edge_4337
0: {Activation -> {a: Activation, b: Activation}} out
node_4300
<k=a5k> = all;
0: {Activation -> {a: Activation, b: Activation}} out
node_4246
<k=a3k> = active;
0: Activation out
0: Activation out
edge_4534
node_4533
<k=a1k> = <k=a0k>;
<k=a2k> = <k=a3k>;
<k=a3k> = <k=a8k>;
<k=a4k> = <k=a9k>;
<k=a5k> = <k=a6k>;
<k=a6k> = <k=a7k>;
<k=a7k> = <k=a5k>;
<k=a8k> = <k=a0k>;
<k=a9k> = <k=a3k>;
<k=a10k> = <k=a4k>;
<k=a11k> = <k=a2k>;
<k=a12k> = <k=a1k>;
<k=a13k> = <k=a12k>;
<k=a14k> = <k=a11k>;
<k=a15k> = <k=a9k>;
<k=a16k> = <k=a8k>;
0: Activation in
1: Activation out
2: Activation out
3: Activation out
4: Activation out
5: Activation out
6: Activation out
7: Activation out
8: Activation out
9: Activation out
10: Activation out
11: Activation out
12: Activation out
13: Activation out
14: Activation out
15: Activation out
16: Activation out
15: Activation out
edge_4549
node_4311
<k=a5k> = +;
0: Text out
0: Text out
edge_4415
0: Text in
node_3798
theInterface.valueManipulation.incrementButton.text=<k=a0k>;
0: Text in
node_4313
<k=a5k> = -;
0: Text out
0: Text out
edge_4423
0: Text in
node_3806
theInterface.valueManipulation.decrementButton.text=<k=a0k>;
0: Text in
node_4575
<k=a5k> = {};
<k=a5k>['cond'] = <k=a1k>;
<k=a5k>['source'] = <k=a2k>;
0: {cond: Boolean, source: Activation} out
1: Boolean in
2: Activation in
0: {cond: Boolean, source: Activation} out
edge_4539
edge_4476
node_4292
<k=a1k> = isActive;
0: {Number -> Boolean} out
0: {Number -> Boolean} out
edge_4538
edge_4552
node_4335
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {Boolean -> Boolean} in
2: Boolean in
3: Boolean out
1: {Boolean -> Boolean} out
edge_4537
edge_4454
node_4551
<k=a1k> = <k=a5k>;
<k=a2k> = <k=a5k>;
0: {Number -> Boolean} in
1: {Number -> Boolean} out
2: {Number -> Boolean} out
2: {Number -> Boolean} out
edge_4553
node_4295
<k=a5k> = boolBox;
0: {Boolean -> Boolean} out
edge_4478
0: {Boolean -> Boolean} out
node_4330
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {Number -> Boolean} in
2: Number in
3: Boolean out
1: {Number -> Boolean} out
edge_4452
edge_4554
node_4368
if(<k=a3k> === active) {
<k=a1k> = previousState['state_4367'];
}
0: Activation in
1: Number out
0: Activation in
node_4483
<k=a5k> = {};
<k=a5k>['source'] = <k=a2k>;
0: {cond: Boolean, source: Activation} out
1: Boolean in
2: Activation in
1: Boolean in
0: {cond: Boolean, source: Activation} out
edge_4484
edge_4545
node_4345
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {Number -> Boolean} in
2: Number in
3: Boolean out
edge_4543
edge_4546
node_4297
<k=a5k> = whenTheenElse;
0: {{cond: Boolean, source: Activation} -> {a: Activation, b: Activation}} out
0: {{cond: Boolean, source: Activation} -> {a: Activation, b: Activation}} out
edge_4556
node_4555
<k=a1k> = <k=a3k>;
<k=a2k> = <k=a5k>;
0: {{cond: Boolean, source: Activation} -> {a: Activation, b: Activation}} in
1: {{cond: Boolean, source: Activation} -> {a: Activation, b: Activation}} out
2: {{cond: Boolean, source: Activation} -> {a: Activation, b: Activation}} out
{Boolean, source: Activation} -> {a: Activation, b: Activation}} in
{Boolean, source: Activation} -> {a: Activation, b: Activation}} out
edge_4558
node_4350
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {{cond: Boolean, sources: Activation} -> {a: Activation, b: Activation}} in
2: {cond: Boolean, source: Activation} in
3: {a: Activation, b: Activation} out
edge_4496
edge_4543
edge_4548
node_4355
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {{Activation -> {a: Activation, b: Activation}} in
2: Activation in
3: {a: Activation, b: Activation} out
edge_4494
0: Activation in
node_3793
theInterface.valueLabel.text=<k=a0k>;
0: Text in
0: Text in
edge_4442
edge_4531
node_4315
if(<k=a5k> === active && <k=a1k>!==null && <k=a1k>!==undefined) {<k=a3k> = <k=a1k>(<k=a2k>);}
0: Activation in
1: {Number -> Text} in
2: Number in
3: Text out
1: {Number -> Text} in
edge_4317
0: {Number -> Text} out
node_4286
<k=a5k> = toString;
0: {Number -> Text} out