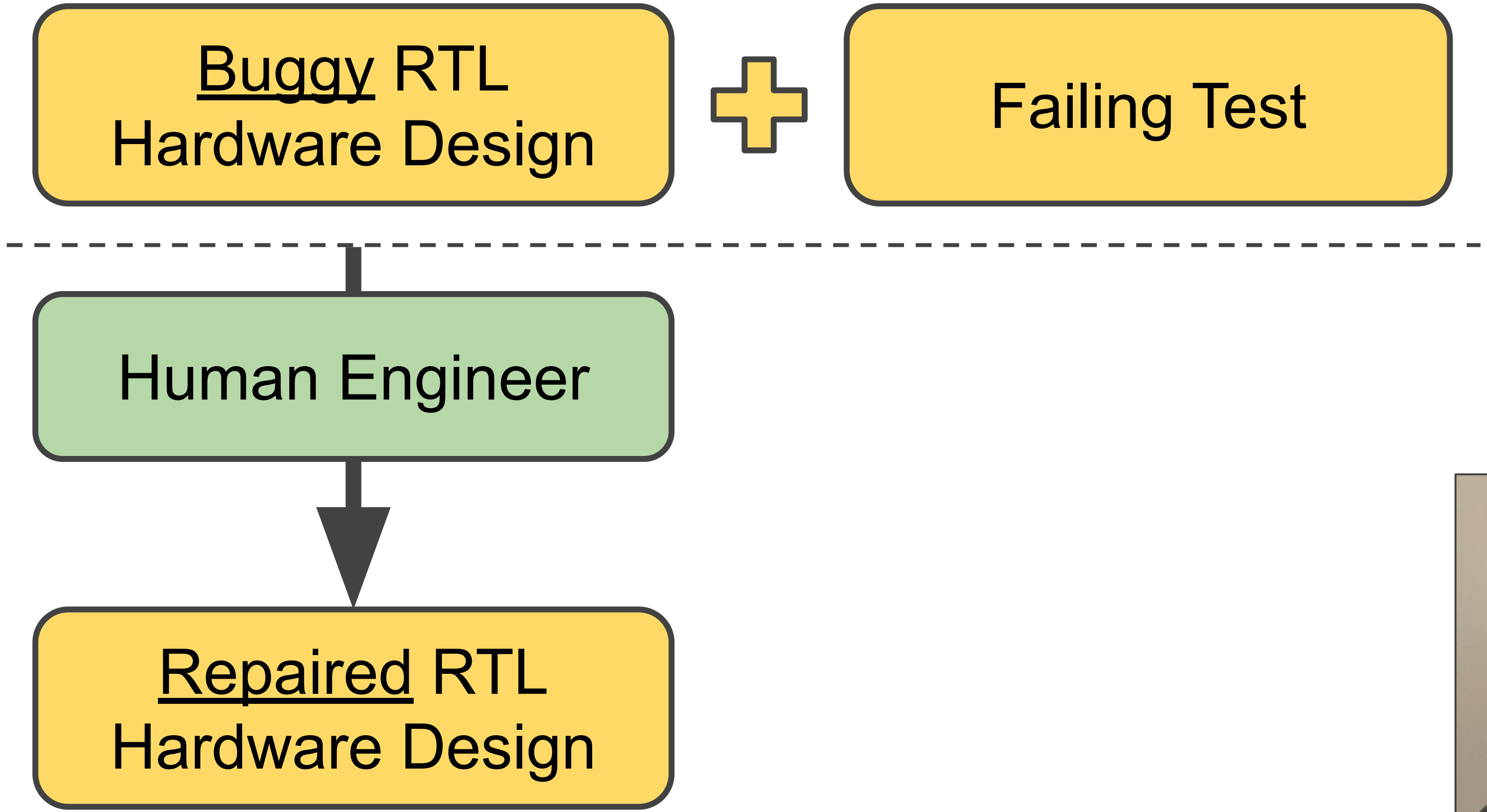
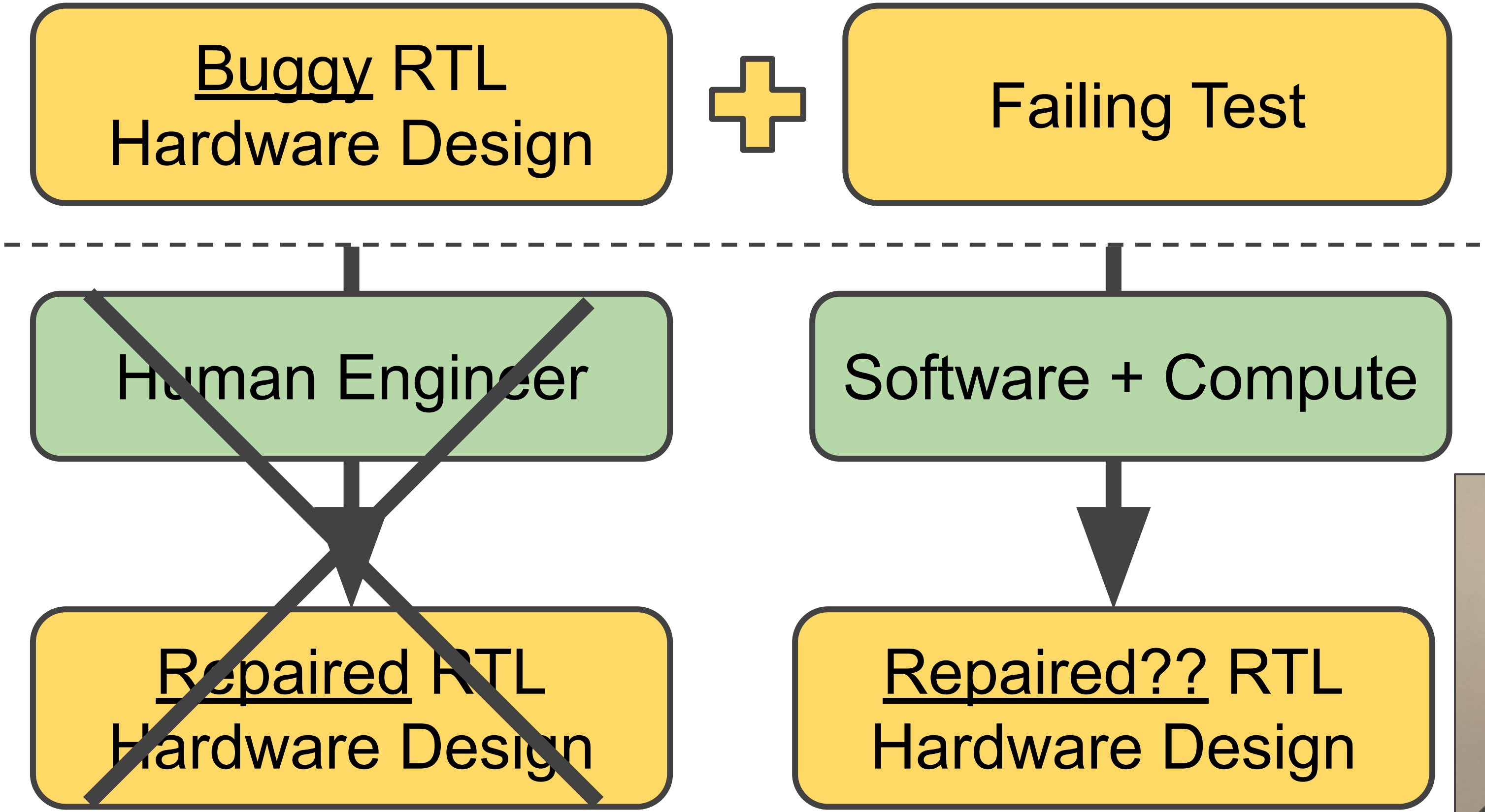
Buggy RTL Hardware Design

+

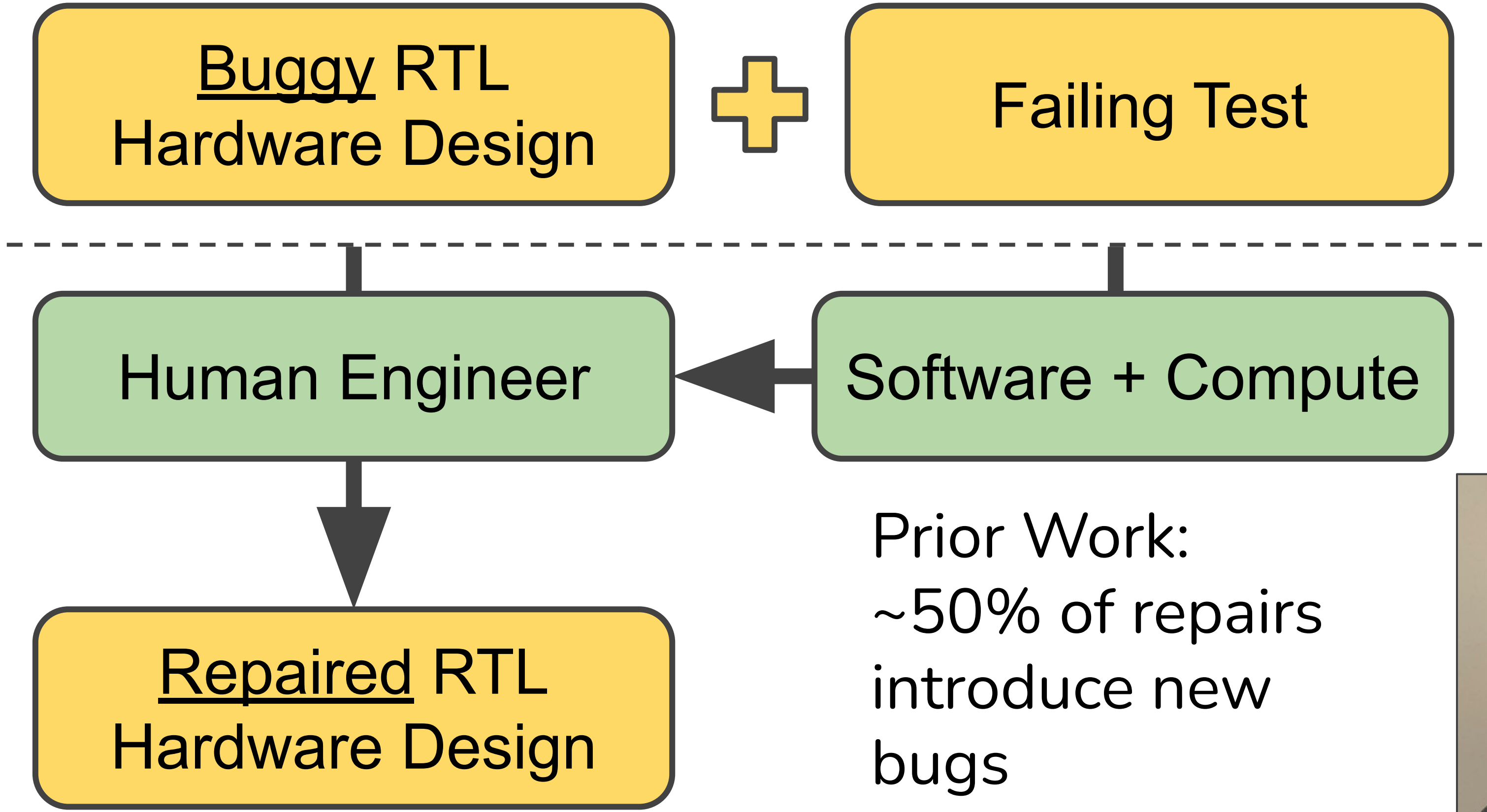
Failing Test


Buggy RTL Hardware Design
Failing Test
Human Engineer
Repaired RTL Hardware Design

## Automated Program Repair

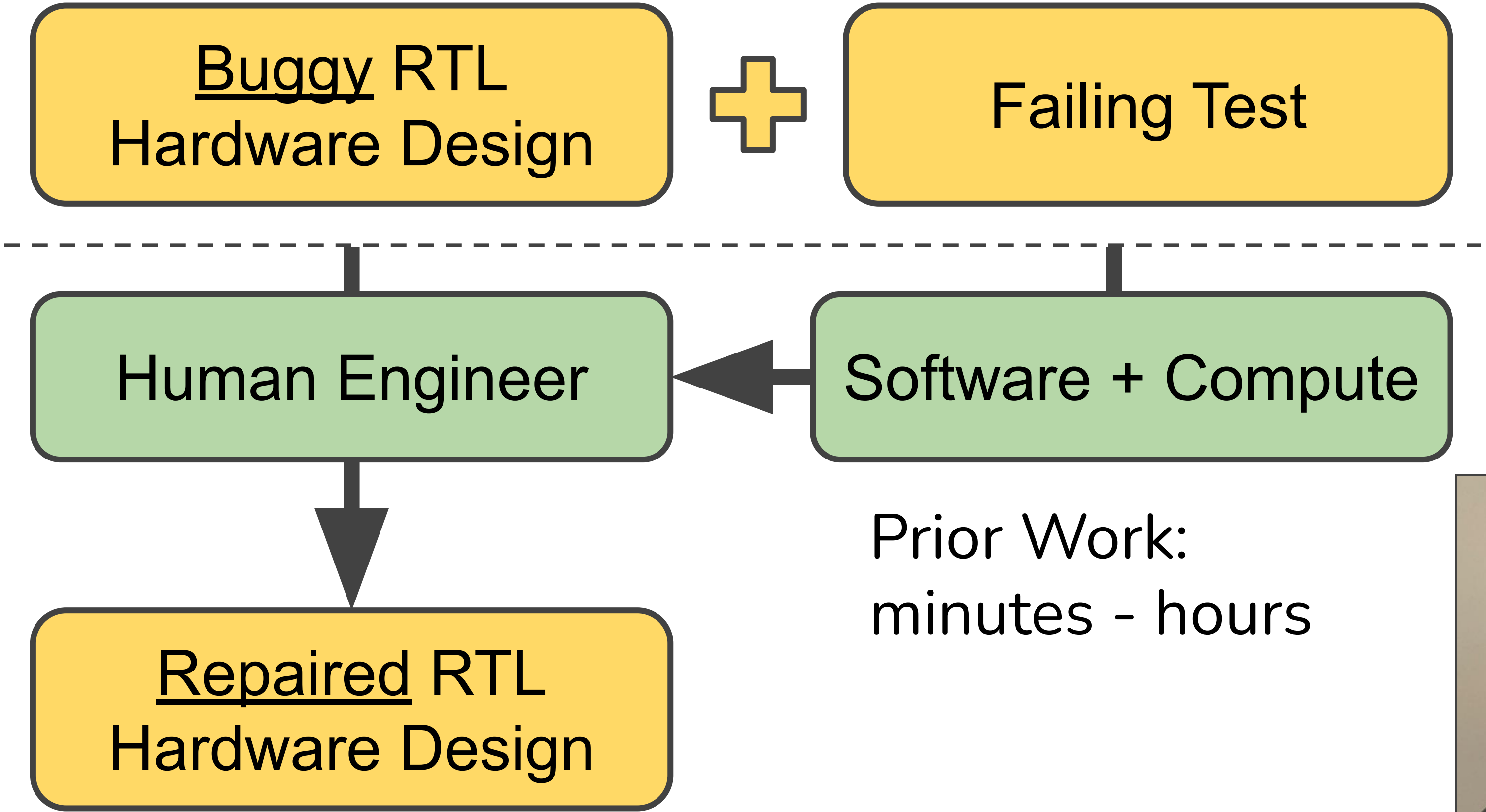

Buggy RTL Hardware Design
Failing Test
Human Engineer
Software + Compute
Repaired RTL Hardware Design
Repaired?? RTL Hardware Design

## Automated Program Repair



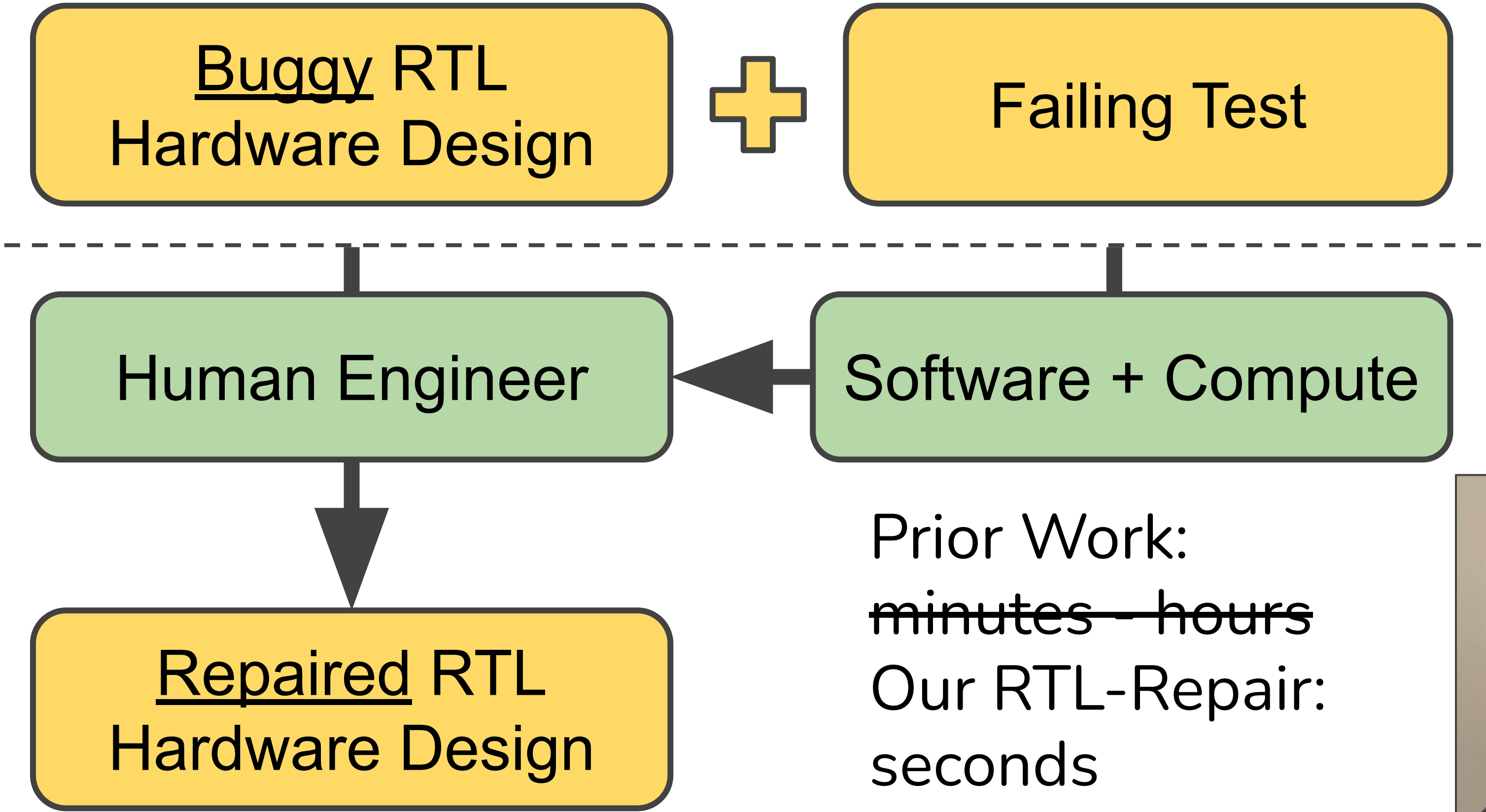
Prior Work:
~50% of repairs introduce new bugs

## Automated Program Repair



Prior Work:
minutes - hours

## Automated Program Repair



Prior Work:
~~minutes - hours~~
Our RTL-Repair:
seconds

## RTL-Repair

```
module counter(....);
always@(posedge clock) begin
 if(reset) begin
   overflow <= 1'b0;
   count <= 4'b0;
 end else if(enable) begin
   count <= count + 1;
 end
 if(count == 'd1) begin
   overflow <= 1'b1;
 end
end
endmodule
```

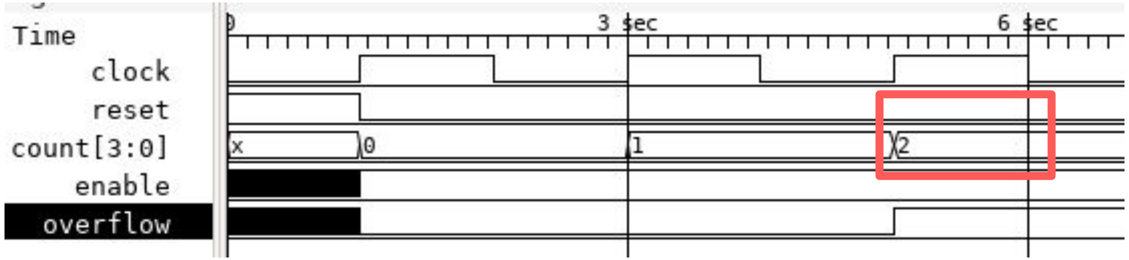
## RTL-Repair

```
module counter(....);
always@(posedge clock) begin
 if(reset) begin
   overflow <= 1'b0;
   count <= 4'b0;
 end else if(enable) begin
   count <= count + 1;
 end
 if(count == 'd1) begin
   overflow <= 1'b1;
 end
end
endmodule
```

| reset | enable | count | overflow |
|---|---|---|---|
| 1 |  | x | x |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |

## RTL-Repair

```
module counter(....);
always@(posedge clock) begin
 if(reset) begin
   overflow <= 1'b0;
   count <= 4'b0;
 end else if(enable) begin
   count <= count + 1;
 end
 if(count == 'd1) begin
   overflow <= 1'b1;
 end
end
endmodule
```

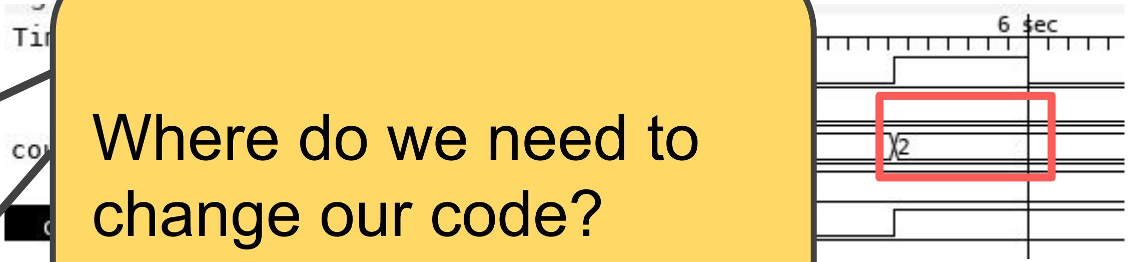


❌ count@3: 2 != 0 (expected)

| reset | enable | count | overflow |
|---|---|---|---|
| 1 | | x | x |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | **0** | 1 |

## RTL-Repair

```
module counter(....);
always@(posedge clock) begin
 if(reset) begin
   overflow <= 1'b0;
   count <= 4'b0;
 end else if(enable) begin
   count <= count + 1;
 end
 if(count == 'd1) begin
   overflow <= 1'b1;
 end
end
endmodule
```



Where do we need to change our code?

xpected)

| reset | enable | count | overflow |
|---|---|---|---|
| 1 |  | x | x |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | **0** | 1 |

## RTL-Repair

```
module counter(....);
always@(posedge clock) begin
 if(reset) begin
   overflow <= 1'b0;
   count <= 4'b0;
 end else if(enable) begin
   count <= count + 1;
 end
 if(count == 'd1) begin
   overflow <= 1'b1;
 end
end
endmodule
```

**After 1s on my laptop:**
*Add the following:*

```
count <= 4'b0;
```

xpected)

| reset | enable | count | overflow |
|---|---|---|---|
| 1 |  | x | x |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | **0** | 1 |

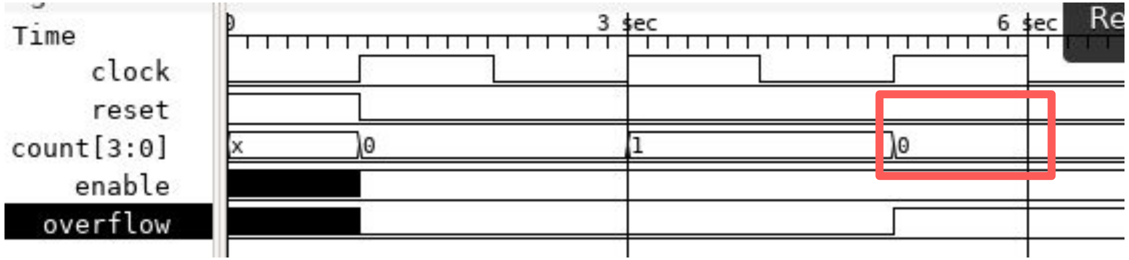
## RTL-Repair

```
module counter(....);
always@(posedge clock) begin
 if(reset) begin
   overflow <= 1'b0;
   count <= 4'b0;
 end else if(enable) begin
   count <= count + 1;
 end
 if(count == 'd1) begin
   overflow <= 1'b1;
   count <= 4'b0;
 end end
endmodule
```

| reset | enable | count | overflow |
|---|---|---|---|
| 1 |  | x | x |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | **0** | 1 |

## RTL-Repair

```
module counter(....);
always@(posedge clock) begin
 if(reset) begin
   overflow <= 1'b0;
   count <= 4'b0;
 end else if(enable) begin
   count <= count + 1;
 end
 if(count == 'd1) begin
   overflow <= 1'b1;
   count <= 4'b0;
 end end
endmodule
```



✓

| reset | enable | count | overflow |
|---|---|---|---|
| 1 | | x | x |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | **0** | 1 |

| | RTL-Repair | | | CirFix [6] | | |
|---|---|---|---|---|---|---|
| | # | median | max | # | median | max |
| ✔ Correct Repairs | 16 | 0.70s | 13.17s | 10 | 2.53min | 14.19h |
| ✖ Wrong Repairs | 2 | 0.51s | 0.68s | 11 | 2.03h | 9.50h |
| ○ Cannot Repair | 14 | 5.64s | 59.81s | 11 | 16.00h | 16.00h |

| | RTL-Repair | | | CirFix [6] | | |
|---|---|---|---|---|---|---|
| | # | median | max | # | median | max |
| ✔ Correct Repairs | 16 | 0.70s | 13.17s | 10 | 2.53min | 14.19h |
| ✖ Wrong Repairs | 2 | 0.51s | 0.68s | 11 | 2.03h | 9.50h |
| ○ Cannot Repair | 14 | 5.64s | 59.81s | 11 | 16.00h | 16.00h |

# RTL-Repair: Fast Symbolic Repair of Hardware Design Code

Kevin Laeufer
laeufer@eecs.berkeley.edu
University of California, Berkeley
Berkeley, CA, USA

Brandon Fajardo*
brfajardo@berkeley.edu
University of California, Berkeley
Berkeley, CA, USA

Abhik Ahuja*
ahujaabhik@berkeley.edu
University of California, Berkeley
Berkeley, CA, USA

Vighnesh Iyer
vighnesh.iyer@eecs.berkeley.edu
University of California, Berkeley
Berkeley, CA, USA

Borivoje Nikolić
bora@eecs.berkeley.edu
University of California, Berkeley
Berkeley, CA, USA

Koushik Sen
ksen@eecs.berkeley.edu
University of California, Berkeley
Berkeley, CA, USA

[Paper PDF]

[Code on Github]

Please join us for our full length talk!