2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

Time-Travel Debugging for High-Level Synthesis

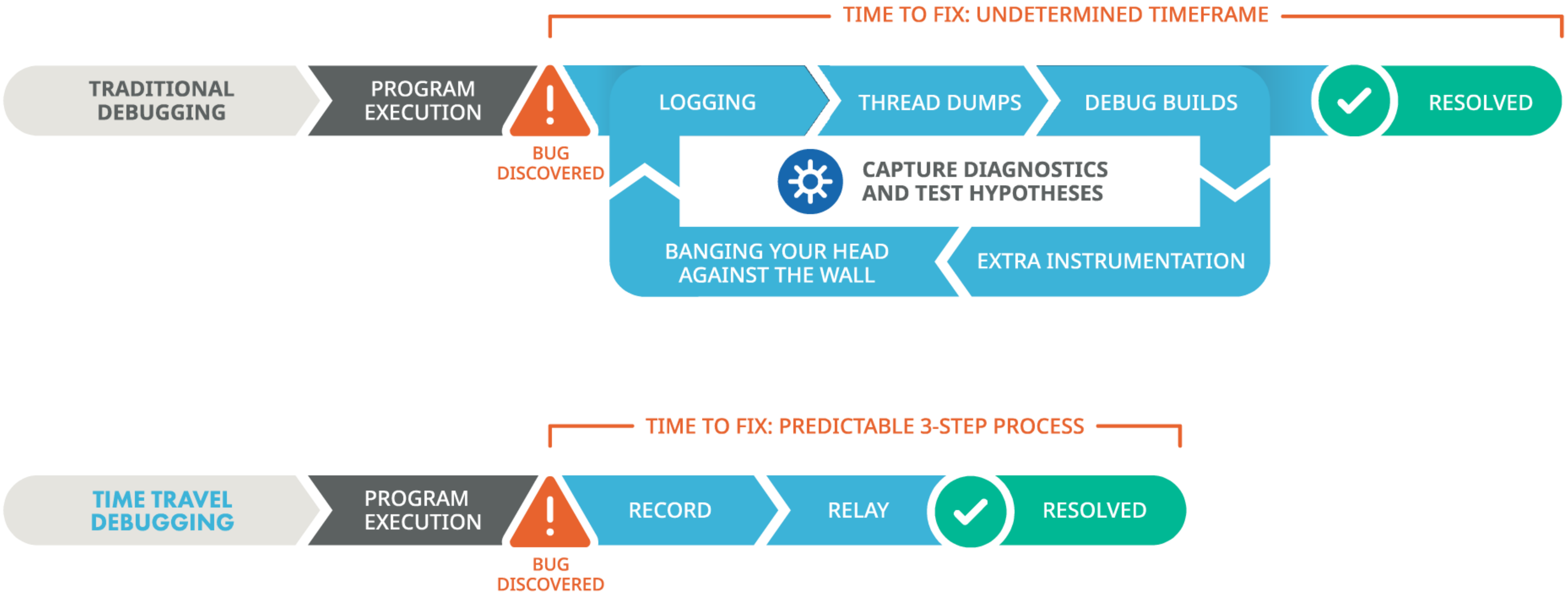
Greg Law, Undo

Jonathan Bonsor-Matthews, LightBlue Logic

undo LIGHTBLUE LOGIC

accellera SYSTEMS INITIATIVE

Why Time-Travel Debugging?



Time-Travel Debugging Commands

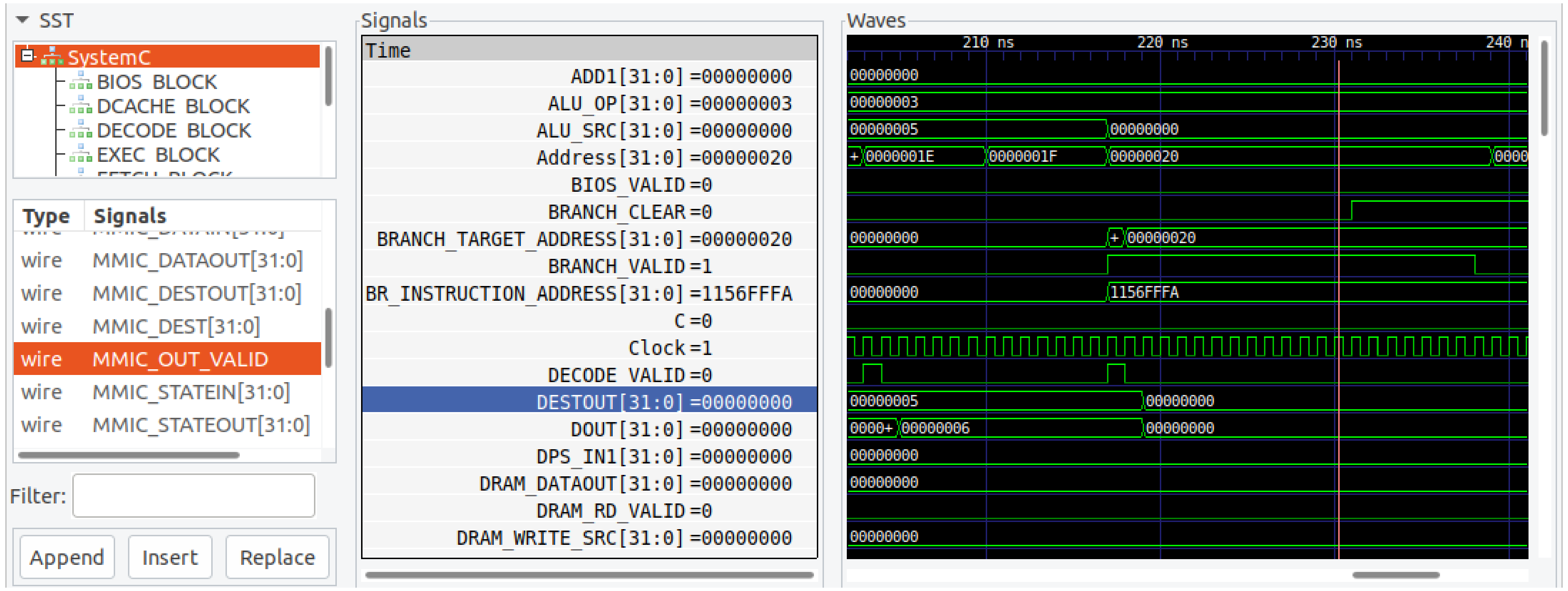
Command	Forward Function	Reverse Function
Step	Step into next function	Step into previous function
Next	Execute next line	Execute previous line
Finish	Return from function	Execute until just before function called
Break (condition)	Stop execution at location in code (optional condition)	Same
Watch	Stop execution when certain variable/memory changes	Same
Continue	Execute forwards	Execute backwards
Last	-	Jump to last time data changes

Best Practice

- Build **time-travel debugging** into **regression flow**
  - **Record** the failing test, **simply replay** the recording
- Design for debug
  - **Identify failures** in the executable to use last command
  - Use **assertions** widely
  - Add **intermediate variables**, for conditional breakpoints
- Thread Fuzzing
  - **Challenge mode** to provoke **concurrency issues** (race conditions, deadlocks)

Waveforms -> Debugger

- **Extract waveforms** from recording *without re-running*
- **Click on transition** to load same point in debugger



Type	Signals
wire	MMIC_DATAOUT[31:0]
wire	MMIC_DESTOUT[31:0]
wire	MMIC_DEST[31:0]
wire	MMIC_OUT_VALID
wire	MMIC_STATEIN[31:0]
wire	MMIC_STATEOUT[31:0]

```
Time
ADD1[31:0] =00000000
ALU_OP[31:0] =00000003
ALU_SRC[31:0] =00000000
Address[31:0] =00000020
BIOS_VALID =0
BRANCH_CLEAR =0
BRANCH_TARGET_ADDRESS[31:0] =00000020
BRANCH_VALID =1
BR_INSTRUCTION_ADDRESS[31:0] =1156FFFA
C =0
Clock =1
DECODE_VALID =0
DESTOUT[31:0] =00000000
DOUT[31:0] =00000000
DPS_IN1[31:0] =00000000
DRAM_DATAOUT[31:0] =00000000
DRAM_RD_VALID =0
DRAM_WRITE_SRC[31:0] =00000000
```

```
00000000
00000003
00000005                00000000
+ 0000001E   0000001F   00000020
00000000                + 00000020
00000000                1156FFFA
00000005                  00000000
0000+ 00000006            00000000
00000000
00000000
00000000
```

Results

- **Four times faster** to find a bug
- Easy to **follow data** through complex designs
- Helps to **understand large codebases**
- Frees engineers' time to **shift left**:
    - **Bring-up more** application layers before tape-out
    - Explore alternative **PPA optimizations**
    - Improve **coverage closure**
    - Improve **stability** (by not ignoring the challenging bugs)
    - **Reduce time-to-market**

2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

Questions?

info@undo.io

accellera
SYSTEMS INITIATIVE