

Source-Level Debugging for FPGA High-Level Synthesis

Nazanin Calagar, Stephen D. Brown, Jason H. Anderson

Presented by: Matthew Schlittler

The Problem

- Lack of debugging techniques
- If a bug is found, must use hardware techniques
- Can be difficult to debug large code bases

Past Work

- Debugging is relatively new research
- Sea Cucumber had multiple limitations
 - Doesn't support Function calls or shared hardware resources
 - Never released

The Solution? - Inspect

- Gives a software view of the problem
- Features:
 - Step through C code
 - Set break points
 - Inspect variables
 - Show relationship between C and RTL
 - Discrepancy detection between simulation and hardware



Fig. 1. Components of Inspect debug framework.

How it works



Debug Database



Fig. 3. Simplified Inspect debug database schema. Figure is auto-generated by MySQL.

High Level C Statements

- Tracks
 - Line Number
 - Column Number
- Helpful when multiple statements in one line.

```
for (int i = 0; i < 100; i++)
{
    std::cout << (i + i - 1) << std::endl;
}</pre>
```

```
HighLevelStatement
```

```
id INT(11)
```

```
line_number INT(11)
```

```
column_number INT(11)
```

IR Instructions

• Tracks

- Function
- Instruction Text
- Position in Function
- High level statement

IRInstr
id INT
position INT(11)
function_id INT(11) (FK)
HighLevelStatement_id INT(11) (FK)
text VARCHAR(2000)

States

- Contains information about Finite State Machine (FSM)
- Tracks
 - Function it belongs to
 - Design name

State
id INT
belongingFunctionId INT(11) (FK)
calledFunctionId INT(11)
design_name VARCHAR(50)

Functions

- Used to help link problem lines of code back to a specific function
- Tracks
 - Function name
 - Start of function

	Function
•	id INT(11)
_	name VARCHAR(100)
	startLineNumber INT(11)

HW signals

• Tracks

- Wires and Registers
- Name
- Bitwidth



Variables

• Tracks

- Name
- Function
- Address Size
- Data Size
- Number of elements

Variable id INT name VARCHAR(200) Function_id INT(11) (FK) tagNum INT addressWidth INT(11) dataWidth INT(11) size INT(11) IRInstr_id INT (FK) type VARCHAR(45)

StateStoreInfo

- Represents the relationship between variable and hardware signal
- Tracks
 - State
 - H/W Signal
 - Memory Offset
 - Port of read/write

StateStoreInfo
id INT
State_id INT (FK)
HWSignal_id INT (FK)
offset INT
port CHAR(1)

Case Study

- Gave example of the debug process
 - Simulation Mode
 - Silicon Mode

```
1 : #include <stdio.h>
2 : /**/
3 : int array[2][2][3] = {{\{1,2,3\}, \{4,5,6\}\}, \ldots
4 : int isPrime (int num)
5:{
6:
      int i, test;
7:
      test = 1;
8 :
      if (num % 2 == 0)
9:
       test = 0;
10:
      else {
11:
        for (i = 2; i < num; i++)
12:
            if (num % i == 0)
              test = 0;
13:
14:
15:
      return test;
16: \}
17: int main (void)
18: {
19:
      int a, b, c, number, result;
20:
      result = 0;
21:
      for (a = 0; a < 2; a++) {
22:
        for (b = 0; b < 2; b++) {
23:
          for (c = 0; c < 3; c++) {
24:
            number = array[a][b][c];
25:
            result += isPrime(number);
26:
          }
27:
        }
28:
29:
      . . .
```

Fig. 4. C code for case study – finding the primes in a 3D array.

😣 🚍 🕘 Inspect - LegUp HLS Source-Level Debugger 1.0



Fig. 5. Screenshot of Inspect during a debugging session.

Simulation Mode

- Uses ModelSim
- Sends commands to ModelSim via TCP

ompile Simulate Add Wave Tools Layout Bookmarks Window Help ē | X 釉 甑 ユ ユ | O - 林 訂 兩 | 夕 ② ※ 與 葱 | 9 | 含 ቀ ⇔ | 翌 100 tr 호 집 집 집 题 题 ⑧ 含 | 1 1 1 0 + 2 | 1 1 + 2 1 1 invit Similate • 1. III II II I Wave - Default : VName 19 Sour 9 + Ab divide by n /b_dvide_by_number/dividend
 /b_dvide_by_number/divisor
 /b_dvide_by_number/quotient 5 4 3 2 • and. 🔕 es (Active) 🚟 🛨 🛃 Ine_42 Now 170000000 fs rsor 1 1 P 4 > 4 Wave × Wave_by_number.vhd × -----Processor se_vr_nemes processor fault1b.divid_by_number(behavioral) umr1c_std(body) fault1b.div_gen_0(div_gen_0_arch) atl_real(body) td_logic_signes(body) ital_time(body) ital_primitives(body) ital_primitives(body) .vpkg(body) .dsp48e1(dsp48e1_v) e.interior.cs.body.pw.wf ucture.interior.cs.body.struct ects.interior.cs.body.tree M DONE n Uom s Iteration: 0 Frocess: /tb_divide_by_number/lime_42 File: ../../../integer_division.srcs/sim_l/new/tb_divide_by_number.vhd s lime_42 at ../../../integer_division.srcs/sim_l/new/tb_divide_by_number.vhd lime 70

ERGA STARTER EDITION 10.5h

Silicon Mode

- Utilizes Altera Signal Tap II
- User must choose signals to watch
- Inspect helps
 - Sliding window-based debug
 - Intelligent signal selection

instance wan	ager: 📉 🔊 🔳 🛄 Acqu	isition in pro	gress			×	JTAG Chain Configurat	ion: JTAG ready	
nstance Rauto_s	Status ignaltap_0 Waiting for th	Enabled	LEs: 1275 1275 cells	Memory: 9728 9728 bits	Small: 0/0 0 blocks	Medium: 2/ 2 blocks	Hardware: DE-SoC [U Device: @2: 5CSE >> SOF Manager:	(BA5 MA5)/5CSTFD5 -	Setup Scan Cl
log: Trig @) 2016/10/27 10:35:48 (0:0:5.8 elap)S@			c	lick to insert tim	ie bar		
Type Alias	Name CounterA:c reset B - CounterA:c count[310] B - CounterA:c resetValue[310] KEY[3]	-3	Ŧ	FCC00000h		X FC FCC00000h	f C00001h X FCC00002	1 1 2h X FCC00003h X F	CC00004
-	∃-SW[90]					3F3h			
ጆ Data	🛲 Setup								
Hierarchy Di	isplay: × [impleCounter ⊦ CounterA:c	Data Log:	gnaltap_0						

Simulation vs Silicon

Benchmark	RTL Sim (s)	Timing Sim (s)	Increase (×)
MIPS	0.9	76	84
MOTION	1	118	118
DFMULL	1.6	248	155
DFADD	0.8	37	46

TABLE I.SIMULATION RUN-TIME COMPARISON.

Other functionality

- Execution management
 - Stepping
 - Statement step
 - Cycle step
 - Breakpoints
- Inspecting Signals
- RTL/Silicon Discrepancy Detection

Testing Inspect – S/W vs RTL

- Took CHStone benchmarks
- Inserted bugs in generated code
- Used Inspect discrepancy's functionality

Results

Benchmark	Bug	# Bugs	1st Bug Cyc	Tot Cyc	Pass/Fail
	Bug #1	7579	503	16919	Fail
	Bug #2	16337	449	17582	Fail
MIPS	Bug #3	5497	1474	17684	Pass
	Bug #1	13264	1785	14030	Fail
	Bug #2	16	27664	30424	Fail
GSM	Bug #3	2817	6362	30642	Fail
	Bug #1	66	62	20212	Fail
	Bug #2	20	19425	19829	Fail
MOTION	Bug #3	88	18953	19816	Fail
	Bug #1	25	138	3294	Fail
	Bug #2	3	479	3318	Fail
DFMUL	Bug #3	21	588	3308	Fail
	Bug #1	51	168	8161	Fail
	Bug #2	735	68	10133	Fail
DFADD	Bug #3	2	400	8257	Pass
	Bug #1	24	1851	6793	Fail
	Bug #2	112	1594	6769	Fail
DFDIV	Bug #3	24	1808	6793	Fail
	GEOMEAN:	138.7	1046.4	10831.9	

TABLE II. AUTOMATED HARDWARE/SOFTWARE DISCREPANCY DETECTION RESULTS.

Testing Inspect – RTL vs Silicon

- Took CHStone benchmarks
- Inserted bugs in generated code
- Introduced bugs by changing FSM

Results

Benchmark	Bug	1st Bug Cyc (# Configs)	Total Cyc (# Configs)	Pass/Fail
MIPS	Bug #1	412 (1)	6552 (4)	Pass
	Bug #2	677 (1)	827 (1)	Fail
	Bug #3	6594 (4)	6640 (4)	Fail
MOTION	Bug #1	8355 (5)	9347 (5)	Fail
	Bug #2	208 (1)	566 (1)	Fail
	Bug #3	8357 (5)	9204 (5)	Fail
DFMUL	Bug #1	486 (1)	1766 (2)	Fail
	Bug #2	208 (1)	∞ (∞)	Fail
	Bug #3	768 (1)	1948 (2)	Pass
DFADD	Bug #1	234 (1)	∞ (∞)	Fail
	Bug #2	208 (1)	4103 (3)	Fail
	Bug #3	983 (1)	4117 (3)	Fail

TABLE III. SILICON DEBUG RESULTS.

Future Work

- Support for watchpoints
- Better correlation between H/W and S/W
- "Wizard-like" functionality

Thoughts on Paper

- Contribution
 - Inspect was released
 - Multiple citations
- Strengths
 - Paper has been cited multiple times
 - Explained the functionality very well
- Weakness
 - Some stuff was not explained, i.e. acronyms

Sources

- <u>https://ieeexplore.ieee.org/document/692</u>
 <u>7496</u> (Paper)
- https://www.llvm.org/
- <u>https://www.mehmetburakaykenar.com/a</u> <u>n-introductory-modelsim-tutorial-for-</u> <u>vivado-xilinx-users/116/</u>
- <u>http://staff.washington.edu/kd1uj/BEE271/</u> <u>Quartus/How%20to%20use%20SignalTap%</u> <u>20II.pdf</u>