2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Bus Trace System: Automating Bus Traffic Debugging in IP-XACT Based SoC

Wonyeong So, Yonghyun Yang, Sun-il Roe, Moonki Jang, Youngsik Kim, Seonil Brian Choi

SAMSUNG

accellera
SYSTEMS INITIATIVE

Contents

- Introduction
  - Debugging approaches for bus traffics
- Proposed methodology
  - SoC bus navigation
  - Automatically finding debug scope during simulation
- Actual usage example
- Expected application result
- Conclusion
- Questions

Debugging approaches for bus traffics

- Waveform viewing tool provided by EDA vendor
- Relying on experiential knowledge
- Increase debug time corresponding to the complexity of bus architecture

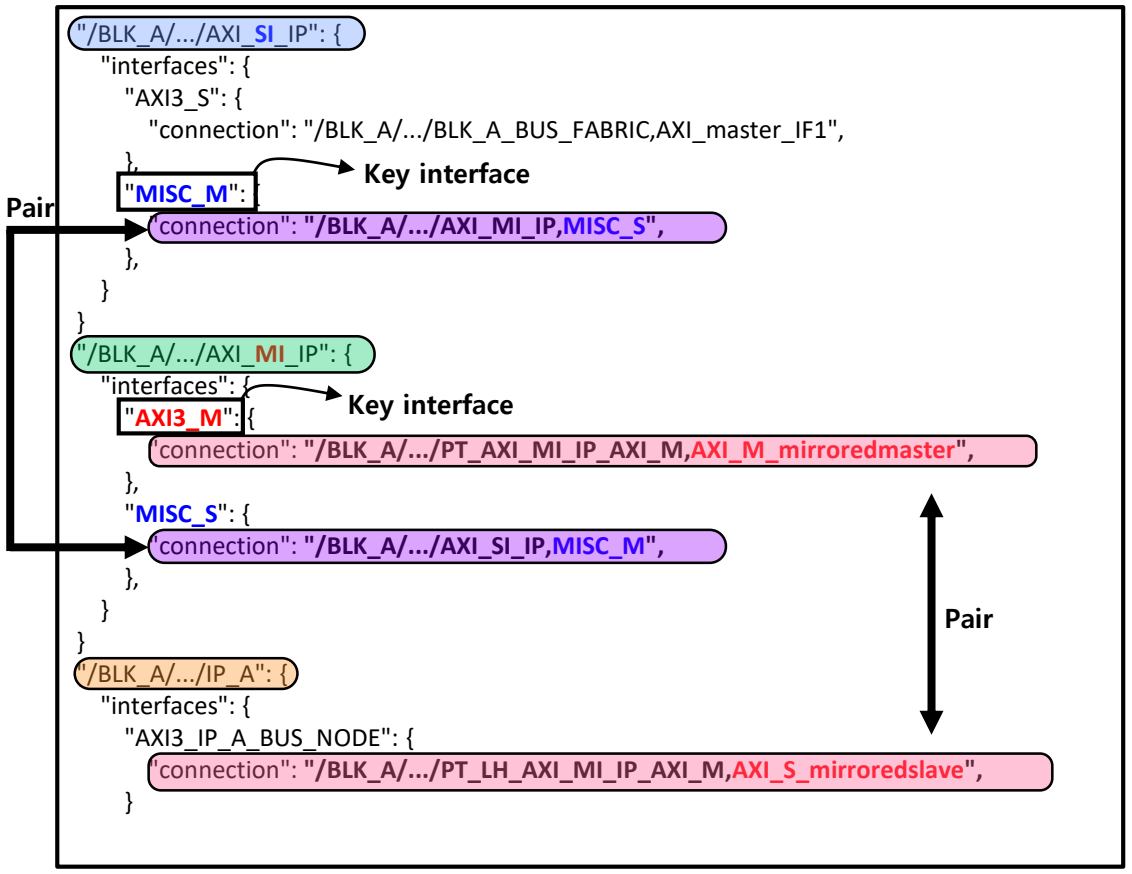
Proposed methodology - SoC bus navigation(1)

- Convert IP-XACT to JSON
  - Utilizing Magillem tool
- Generation bus branch information
  - Bus routing spec file
- Convert UVM register definition system Verilog(SV) file to JSON

Proposed methodology - SoC bus navigation(2)

- Generation raw bus navigation

Converted JSON from IP-XACT



Raw bus navigation file

```
"/BLK_A/.../AXI_SI_IP": [
  "/BLK_A/.../AXI_MI_IP",
],
"/BLK_/.../AXI_MI_IP": [
  "/BLK_A/.../IP_A"
],
"/BLK_A/.../IP_A": [
  "/BLK_A/.../BUS_FABRIC"
],
"/BLK_A/.../BUS_FABRIC": [
  "/BLK_A/.../IP_0",
  "/BLK_A/.../IP_1",
  "/BLK_A/.../IP_2",
  "/BLK_A/.../IP_3",
],
...
```

Proposed methodology – SoC bus navigation(3)

- Generation final bus navigation

Raw bus navigation file

```
"BLK_CPU/.../CPU" : "BLK_CPU/.../MMU", "BLK_CPU/.../MMU" :
"/BLK_CPU/.../IP_CPU",
...
"/BLK_CPU/.../AXI_SI_IP" : "/BLK_CPU/.../AXI_MI_IP",
"/BLK_CPU/.../AXI_MI_IP" : "/BLK_A/.../AXI_SI_IP",
...
"/BLK_A/.../AXI_SI_IP" : "/BLK_A/.../AXI_MI_IP"
"/BLK_A/.../AXI_MI_IP": "/BLK_A/.../BUS_FABRIC",
...
"/BLK_A/.../BUS_FABRIC": [
  "/BLK_B/.../AXI_SI_IP",
  "/BLK_C/.../AXI_SI_IP",
  "/BLK_D/.../AXI_SI_IP",
  "/BLK_E/.../AXI_SI_IP",
  "/BLK_F/.../AXI_SI_IP",
  "/BLK_G/.../AXI_SI_IP",
  "/BLK_H/.../AXI_SI_IP",
  "/BLK_I/.../AXI_SI_IP"
],
"BLK_B/.../AXI_SI_IP" : "BLK_B/.../AXI_MI_IP"
...
"BLK_C/.../AXI_SI_IP" : "BLK_C/.../AXI_MI_IP"
...
```

Starting point

Bus routing instance

Counter instance candidates

Bus routing information

```
"/BLK_A.../BUS_FABRIC"
 "addr": {
    "AXI_master_IF0": [
      "0x0900_0000~0x0940_FFFF",
      "0x2000_0000~0x210F_FFFF",
      "0x21B0_0000~0x21FF_FFFF"
    ],
    "AXI_master_IF1": "0x2120_0000~0x212F_FFFF",
    "AXI_master_IF2": "0x2130_0000~0x214F_FFFF",
    "AXI_master_IF3": "0x2150_0000~0x215F_FFFF",
    "AXI_master_IF4": "0x2160_0000~0x216F_FFFF",
    "AXI_master_IF5": "0x2170_0000~0x217F_FFFF",
    "AXI_master_IF6": "0x2180_0000~0x218F_FFFF",
    "AXI_master_IF7": "0x2190_0000~0x219F_FFFF",
}
```

Counter interface

Matched address range

Bus routing connection information

```
"/BLK_A/.../BUS_FABRIC"
"branch_info": {
   "AXI_master_IF0": "/BLK_B/.../AXI_SI_IP",
   "AXI_master_IF1": "/BLK_C/.../AXI_SI_IP",
   "AXI_master_IF2": "/BLK_D/.../AXI_SI_IP",
   "AXI_master_IF3": "/BLK_E/.../AXI_SI_IP",
   "AXI_master_IF4": "/BLK_F/.../AXI_SI_IP",
   "AXI_master_IF5": "/BLK_G/.../AXI_SI_IP",
   "AXI_master_IF6": "/BLK_H/.../AXI_SI_IP",
   "AXI_master_IF7": "/BLK_I/.../AXI_SI_IP",
```

Counter instance

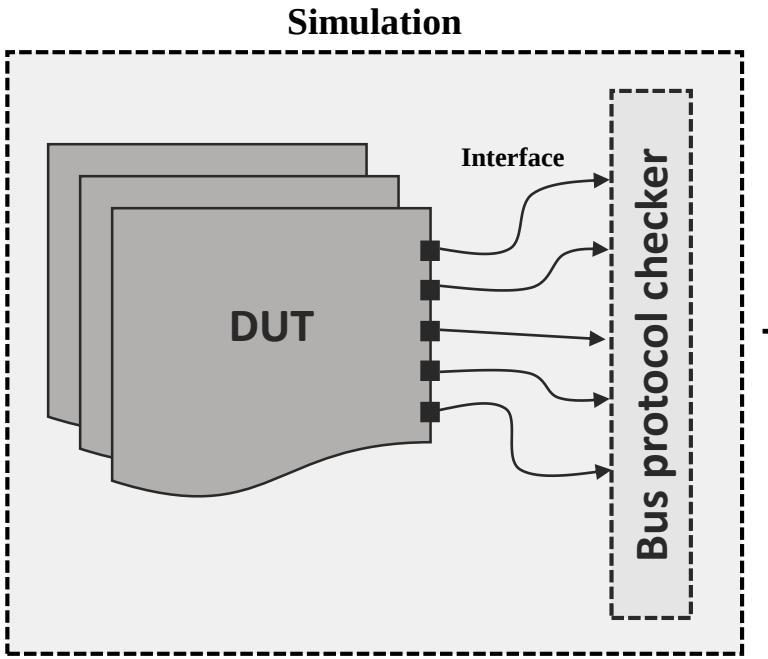
Final bus navigation

```
...
"IP_X": {
   "BASE_ADDR": "0x200F_0000",
   "PATH": [
      "/BLK_CPU/.../CPU",
      "/BLK_CPU/.../MMU",
      "/BLK_CPU/.../GLUE_LOGIC",
      ...
      "/BLK_CPU/.../AXI_SI_IP",
      "/BLK_CPU/.../AXI_MI_IP",
      ...
      "/BLK_A/.../AXI_SI_IP",
      "/BLK_A/.../AXI_MI_IP",
      "/BLK_A/.../BUS_FABRIC",
      "/BLK_B/.../AXI_SI_IP",
      "/BLK_B/.../AXI_MI_IP",
       ...
      "/BLK_X/.../AXI_SI_IP",
      "/BLK_X/.../AXI_MI_IP",
      "/BLK_X/.../IP_X",
   ]
},
...
```

Connection ordering

Proposed methodology - Automatically finding debug scope during simulation (1)

- Connect SV interface at Design Under Test(DUT)

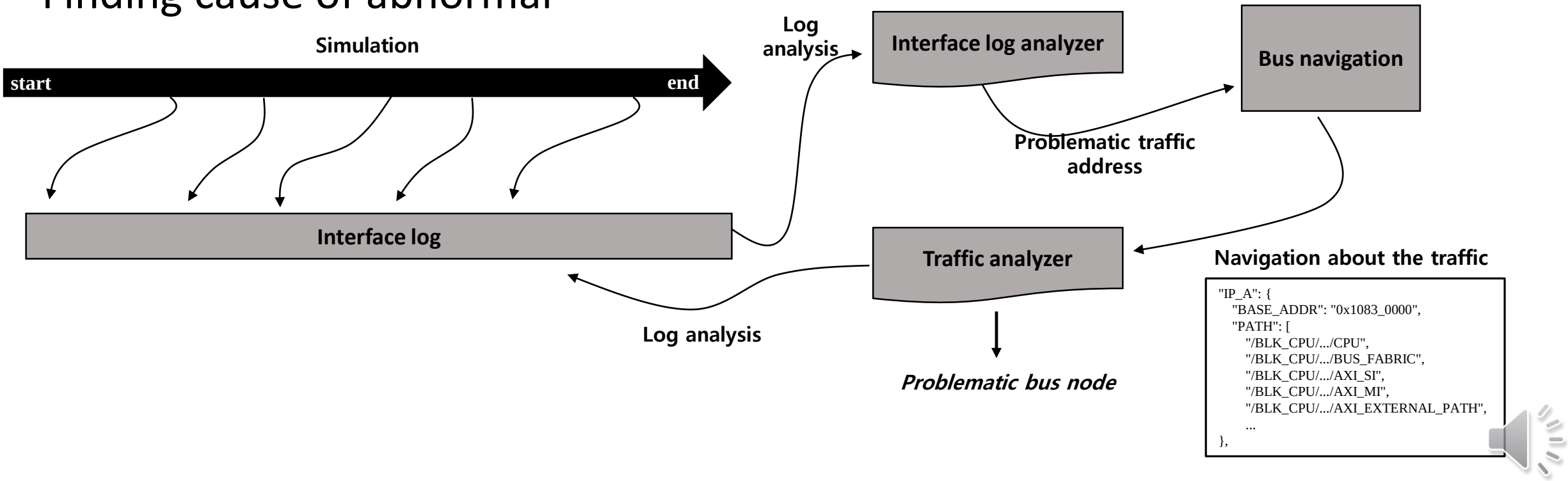


Interface log

{"time":"15572466","intf_name":"BLK_A/.../AXI_BUS_NODE_A/AXI_master_IF1","araddr":"0x000012000","arid":"0x001","arsize":"0x2","arlen":"0xf","arburst":"0x2","arcache":"0x3","arlock":"0x0","arprot":"0x1","aruser":"0x00002000003"}
{"time":"15572466","intf_name":"BLK_A/.../AXI_BUS_NODE_A/AXI_slave_IF1","araddr":"0x000012000","arid":"0x0","arsize":"0x2","arlen":"0xf","arburst":"0x2","arcache":"0x3","arlock":"0x0","arprot":"0x1","aruser":"0x00002000003"}
{"time":"15572477","intf_name":"BLK_A/.../IP_A/AXI00_S","rdata":"0x0b000820","rid":"0x003","rresp":"0x0","ruser":"0x0","rlast":"0x0","rdata_width":"0x00000020"}
{"time":"15572477","intf_name":"BLK_A/.../AXI_BUS_NODE_B/AXI3_M_LH","rdata":"0x0b000820","rid":"0x0","rresp":"0x0","ruser":"0x0","rlast":"0x0","rdata_width":"0x00000020"}
{"time":"15572477","intf_name":"BLK_A/.../AXI_BUS_NODE_C/AXI_master_IF0","rdata":"0x0b000820","rid":"0x003","rresp":"0x0","ruser":"0x0","rlast":"0x0","rdata_width":"0x00000020"}
{"time":"15572477","intf_name":"BLK_A/.../AXI_BUS_NODE_C/AXI_slave_IF1","rdata":"0x0b000820","rid":"0x001","rresp":"0x0","ruser":"0x0","rlast":"0x0","rdata_width":"0x00000020"}
{"time":"15572477","intf_name":"BLK_A/.../IP_B/AXI_master_IF1","rdata":"0x0b000820","rid":"0x001","rresp":"0x0","ruser":"0x0","rlast":"0x0","rdata_width":"0x00000020"}
{"time":"15572477","intf_name":"BLK_A/.../IP_B/AXI_slave_IF1","rdata":"0x0b000820","rid":"0x0","rresp":"0x0","ruser":"0x0","rlast":"0x0","rdata_width":"0x00000020"}

Proposed Methodology - Automatically finding debug scope during simulation (2)

- Simulation with Python script
- Finding cause of abnormal



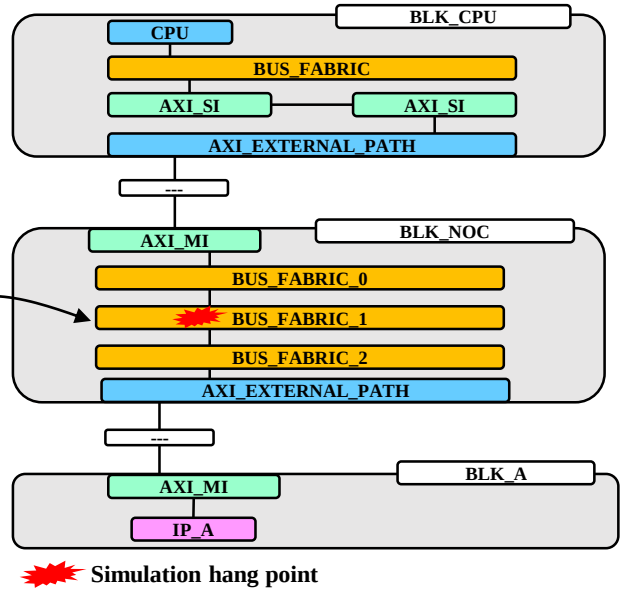
Proposed Methodology - Automatically finding debug scope during simulation (3)

- Show debugging result to user with problematic bus node

Terminal

Simulation start…
…
BTS::TIMER:: Reach the threshold
BTS::Trigger BTS…
……
BTS:: BTS has been triggered automatically by abnormal state.
…
BTS:: Analyzing the interface log …
BTS:: BLK_CPU didn`t get write response about 0x1083_0808
BTS:: Problematic instance is “BUS_FABRIC_1”
BTS:: Please debug “BUS_FABRIC_1” first about 0x1083_0808 traffic
BTS:: Result – Bus hang. Simulation end
Problematic bus node is “*BUS_FABRIC_1*”

DUT



Actual usage example

- Master traversed 11 instances to access the slave, resulting in a bus hang

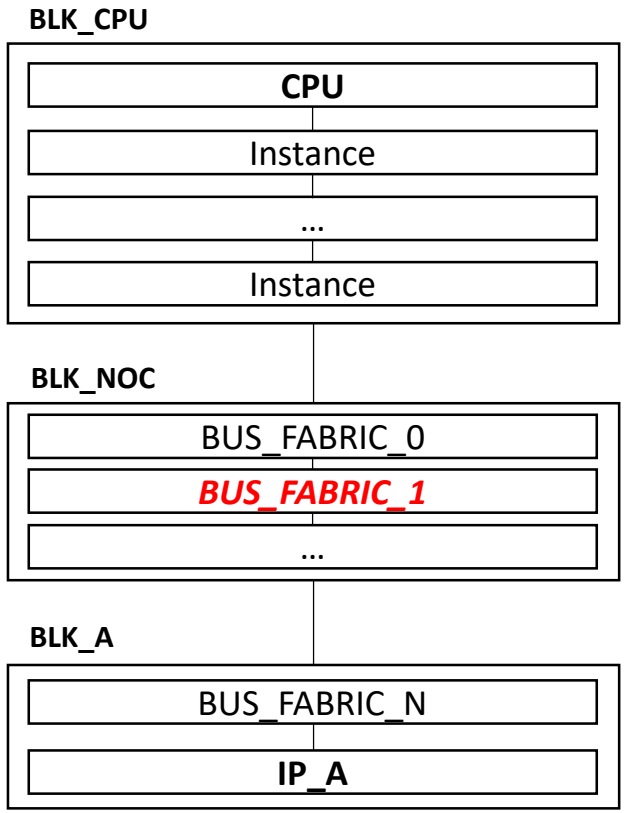
```
Read burst response timeout

A Read burst has reached a response timeout and was discarded from top.cdn_ps_axi_if_peri_active Monitor.
The last read transfer for this transaction had not been received from the slave after 30000 clock cycles from the start of the address phase.
Timeout burst:   kind: INCR address: address: 0x00010830008 id: 41 len: 1 size: WORD access: NORMAL
Title:  Read burst response timeout
Expected value:
Actual value:
Debug action:  The read response timeout is currently set to 30000 clock cycles.
If this is too short, please increase this value.
Relevant data item:   burst-@10331
AMBA AXI Protocol Spec version:  1.0
Section(s):  NA

BTS triggering
==========
Execution BTS for read burst timeout
BTS: : Execution Read Timeout Debugger
Problematic Bus Node :   top.BLK_NOC.BUS_FABRIC_1
==========
```

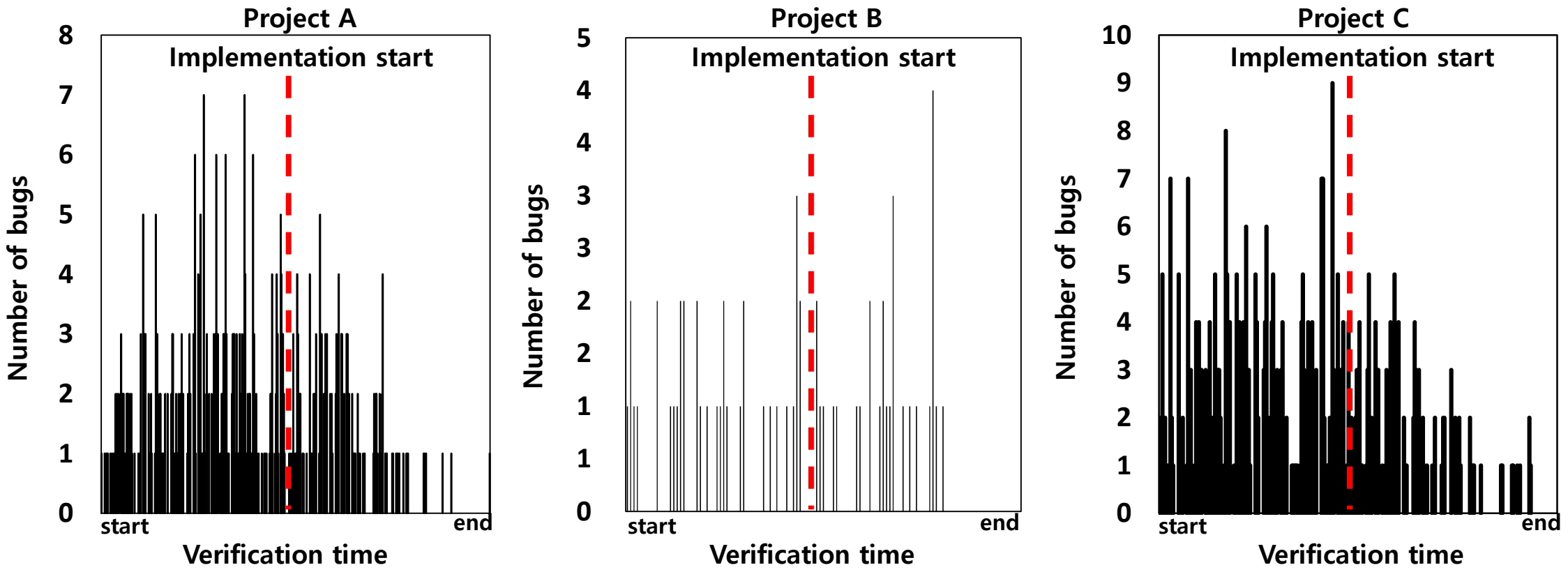
Navigation for IP_A

```
"IP_A": {
  "BASE_ADDR": "0x1083_0000",
  "PATH": [
    "/BLK_CPU/.../CPU",
    "/BLK_CPU/.../BUS_FABRIC",
    "/BLK_CPU/.../AXI_SI",
    ...
    "/BLK_NOC/.../BUS_FABRIC_0",
    "/BLK_NOC/.../BUS_FABRIC_1",
    "/BLK_NOC/.../BUS_FABRIC_2",
    ...
    "/BLK_A/.../IP_0",
  ]
},
```



Expected application result

- Around 10-12% of all errors can be identified with BTS
- Bugs mainly occurred in the early stages of the project



Conclusion

- Fast debug methodology in the increasingly complex SoC architecture trend
- High debugging proficiency independent of experience
- Achieving quick stabilization of bus integration
- Limitations in relying on IP-XACT

Questions