Table 1. Details of the Found Bugs. The table shows the details of the 25 bugs found by METPR. The column “Bug description” shows the representation of bugs. The "Bug location" column indicates the location where each bug was detected. The column “Confirm” shows whether the bug has been confirmed by developers; the column “Unknown” indicates whether the bug is new according to ChiPrompt and VTR bug reports. From the table, METPR finds 14 new bugs and 11 known bugs previously found by ChiPrompt. 23 bugs have been confirmed. (Note: "ChiPrompt" is a pseudonym for the actual EDA system used in this study.)

ID	P&R system	Bug location	Bug description	Confirm	Unknown
1	ChiPrompt	Analysis	A path that should have been reported was omitted from the timing report. The bug was caused by a missing function in the component responsible for importing information from the routing module.	✓	✓
2	ChiPrompt	Analysis	The running job was terminated during the timing analysis step after becoming unresponsive for approximately 15 minutes. Fundamental model for loading routing file caused the bug.	✓	✓
3	ChiPrompt	Routing	The process crashed due to encountering duplicate block names during routing a road.	✓	
4	ChiPrompt	Synthesis	Unexpected variables and expressions were discovered in the netlist following synthesis. The abnormal structure cannot be recognized by Preprocessing module.	✓	✓
5	ChiPrompt	Placement	The routability of a netlist was assessed as "false" after placement. However, the architecture has sufficient resources to finish following routing tasks.	✓	
6	ChiPrompt	Routing	Inconsistencies were observed in the routing strings related to LUT costs.	✓	
7	ChiPrompt	Routing	The routing process entered an endless loop to find a path and was terminated after approximately 15 minutes.	✓	✓
8	ChiPrompt	Routing	Some "fixed" values of tree nodes with assigned IDs in the RR graph were incorrectly set to ’false’.	✓	✓
9	ChiPrompt	Routing	The launch tag time exceeded the catch tag time due to a miscalculation.	✓	✓
10	ChiPrompt	Routing	The module responsible for LUT pin swapping incorrectly modified the ’PREP’ parameter in the routing file.	✓	✓
11	ChiPrompt	Routing	Slack calculation failed: either the data arrival time or the data required time was zero, violating the setup and hold time constraints in the timing engine.	✓	✓
12	ChiPrompt	Placement	Not all Design Rule Checking (DRC) rules were met due to a bug in the components that verify and ensure rule compliance.	✓	✓
13	ChiPrompt	Routing	The routing process was not completed as intended. The functions responsible for iterative routing were incomplete.	✓	
14	ChiPrompt	Preprocessing	The variable name "A" unexpectedly conflicted with another variable named "\A[0]".	✓	
15	ChiPrompt	Routing	Incorrect timing computation causing violations of the timing bounds (maxDB > minDB > lowerbound not satisfied).	✓	

ID	P&R system	Bug location	Bug description	Confirm	Unknown
16	ChiPrompt	Placement	A significant performance increase from the single solution to the ARC solution occurred due to an incorrect change from "INPIN" to "INPAD," aimed at reducing high fan-out costs using the timing tree.	✓	✓
17	ChiPrompt	Routing	Routing could not be completed due to perceived limited resources, despite the architecture having sufficient resources that were not recognized by the system.	✓	
18	ChiPrompt	Placement	The register was positioned far from the related I/Os and CLBs because the connection costs of the I/Os were overlooked.	✓	
19	ChiPrompt	Routing	Incorrect timing computation led to violations of timing constraints (min - del.min <= max - del.max not satisfied).	✓	
20	ChiPrompt	Placement	Clock skew execution failed, resulting in improperly placed groups of LUTs and reduced performance.	✓	
21	ChiPrompt	Routing	The routing process failed to complete all pathways and exited prematurely.	✓	✓
22	ChiPrompt	Placement	A LUT with fewer connections was placed far from related blocks, causing significant detours.	✓	✓
23	ChiPrompt	Placement	LUTs were scattered due to an incorrect implementation of clock skew, resulting in a suboptimal solution.	✓	
24	VTR	Placement	Unrelated blocks with the highest "gain" among the candidates were packed into a cluster, leading to excessive wiring.		✓
25	VTR	Placement	Multipliers and related CLBs were placed far from I/Os, resulting in a lack of block aggregation.		✓