Verification Report

System Verilog Verification Environment for Branch Predictor Module

## Kamran Sarfraz, Manahil Malik, Kiran Shoaib

AQL Tech Solutions

# Introduction to the Device-Under-Test (DUT)

The DUT in this plan is a branch predictor module. Branch predictor module has current pc, target pc, branch taken and branch valid as inputs. It also has a clock and active high reset input signals. Data signal for current pc and target pc are 32 bits wide. Branch predictor uses information about taken or not taken branches gathered at run-time to predict the outcome of a branch. It consists of a Branch Transfer Buffer or a Branch History Table, these will update or not depending on branch resolution results. We will be using all possible sequence of instructions in set of three that can go through our branch predictor because the procesor is three staged pipeline.

# Verification Plan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Feature** | **Test Description** | **Ref.** | **Type** | **Result** | **Comments** | **Stimulus** |
| 1 | Branch not taken after branch not taken  after  branch not taken | Give instructions in such a sequence that there are three branches one after other and all are not taken. |  | Transaction |  |  | randomized |
| 2 | Branch taken after branch taken  after  branch  taken | Give instructions in such a sequence that there are three branches one after other and all are taken. |  | Transaction |  |  | randomized |
| 3 | Branch taken  after branch taken  after branch  not taken | Give instructions in such a sequence that first and second is taken and third isn’t taken. |  | Transaction |  |  | randomized |
| 4 | Branch taken  after branch not taken  after branch taken | Give instructions in such a sequence that first is taken, second isn’t taken and third is taken. |  | Transaction |  |  | randomized |
| 5 | Branch taken  after branch not taken  after branch not taken | Give instructions in such a sequence that first is taken. second is’nt taken and third isn’t taken. |  | Transaction |  |  | randomized |
| 6 | Branch not taken  after branch taken  after branch taken | Give instructions in such a sequence that first is not taken. second is taken and third is taken. |  | Transection |  |  | randomized |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | Branch not taken  after branch taken  after branch not taken | Give instructions in such a sequence that first is not taken. second is taken and third is not taken. |  | Transaction |  |  | randomized |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | Branch not taken  after branch not taken  after branch taken | Give instructions in such a sequence that first is not taken. second is not taken and third is taken. |  | Transaction |  |  | randomized |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | Branch after RV32I instruction  After Branch | Give instructions in such a sequence that first is branch second is RV32I instruction and third is branch |  | Transaction |  |  | Directed-randomized |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | Branch  After Branchafter RV32I instruction | Give instructions in such a sequence that first is branch second is Branch and third is RV32I instruction |  | Transaction |  |  | Directed-randomized |
| 11 | RV32I instruction after RV32I instruction  after branch | Give instructions in such a sequence that first is RV32I instruction second is RV32I instruction and third is branch |  | Transaction |  |  | Directed-randomized |
| 12 | RV32I instruction after branch  after RV32I instruction | Give instructions in such a sequence that first is RV32I instruction second is branch and third is RV32I instruction |  | Transaction |  |  | Directed-randomized |
| 13 | Branch after RV32M instruction  After Branch | Give instructions in such a sequence that first is branch second is RV32M instruction and third is branch |  | Transaction |  |  | Directed-randomized |
| 14 | Branch  After Branchafter RV32M instruction | Give instructions in such a sequence that first is branch second is Branch and third is RV32M instruction |  | Transaction |  |  | Directed-randomized |
| 15 | RV32M instruction after RV32M instruction  after branch | Give instructions in such a sequence that first is RV32M instruction second is RV32M instruction and third is branch |  | Transaction |  |  | Directed-randomized |
| 16 | RV32M instruction after branch  after RV32M instruction | Give instructions in such a sequence that first is RV32M instruction second is branch and third is RV32M instruction |  | Transaction |  |  | Directed-randomized |
| 17 | Dependency on last instruction | Give instructions in such a sequence that branch instruction is dependent on last instruction |  | Transaction |  |  | Directed-randomized |
| 18 | Dependency on last two instructions | Give instructions in such a sequence that branch instruction is dependent on last two instruction |  | Transaction |  |  | Directed-randomized |

## Explanation of Different Fields

|  |  |
| --- | --- |
| **No.** | The serial number of the test. |
| **Feature** | The feature which the current test is verifying in full or partially.  The feature is usually on the abstraction level of a user. |
| **Test Description** | A detailed description of the test case being performed.  You can be as verbose as you want. |
| **Ref.** | Reference to the section in the related standard document.  The section number as well as page numbers should be described here. |
| **Type** | Type of the test. Whether the test is an assertion (A) or a transaction (T) type. |
| **Result** | Pass (P) or Fail (F). |
| **Comments** | Any other comments about the test or its results that you want to mention. |
| **Coverage** | Code or Functional |
| **Stimulus** | Mention the kind of stimulus given. i.e directed system verilog |