# A-QED Verification of Hardware Accelerators

## Abstract

本文提出了一种用于硬件加速器验证的新方法，称为A-QED（Accelerator-Quick Error Detection），**A-QED基于BMC**（Bounded Model Checking）。

- it does not require extensive design specific properties or a full formal design specification like BMC.
- A-QED is effective for both RTL and high-level synthesis (HLS) design flows

A-QED是高效的：

- A-QED detected all bugs detected by conventional verification flow
- A-QED detected bugs that escaped conventional verification flow
- A-QED improved verification productivity dramatically
- A-QED produced short counterexamples for easy debug

## 1.Introduction

### 1.1 硬件加速器验证的困难

- 不像处理器有成熟的指令集（Instruction Set Architecture, ISA），硬件加速器缺乏对其功能和接口的准确描述
- 即使对于同一种功能的加速器，设计方法可能也有很大不同
- 加速器的验证缺乏数十年的验证技术经验积累

### 1.2 Bounded Model Checking

###### 1.3 特性和优点

## 2. Accelerator Model Targeted in This Paper

## 2.1 Accelerator

本文主要集中于以下的硬件加速去：

- **The accelerator is an LCA(Loosely-Coupled Accelerators).** Since an LCA is connected to the SoC interconnect, it can directly access system components such as memories.
- **A handshake protocol is used to communicate between the LCA and the host (e.g., processor core).** This protocol must define when the inputs to/outputs from the accelerator are valid, and also when the accelerator and the host are each ready to receive inputs.
- **The LCA execution is *non-interfering*;** i.e., the result produced by the accelerator for a given input is independent of any other inputs received (earlier or later). LCAs should not be confused with combinational circuits – they are complex sequential circuits

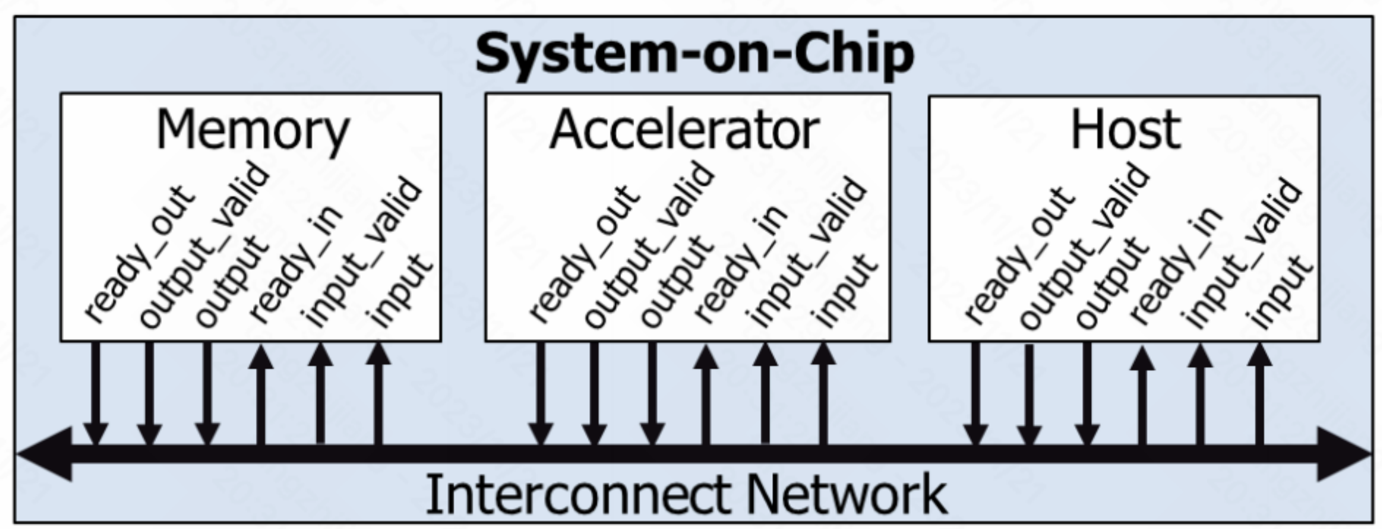

**Fig. 1.** Accelerator model in this paper.

## 2.2 Motivating Example

例子加速器中有四个buffer，其中buffer 4无法接收到时钟信号。

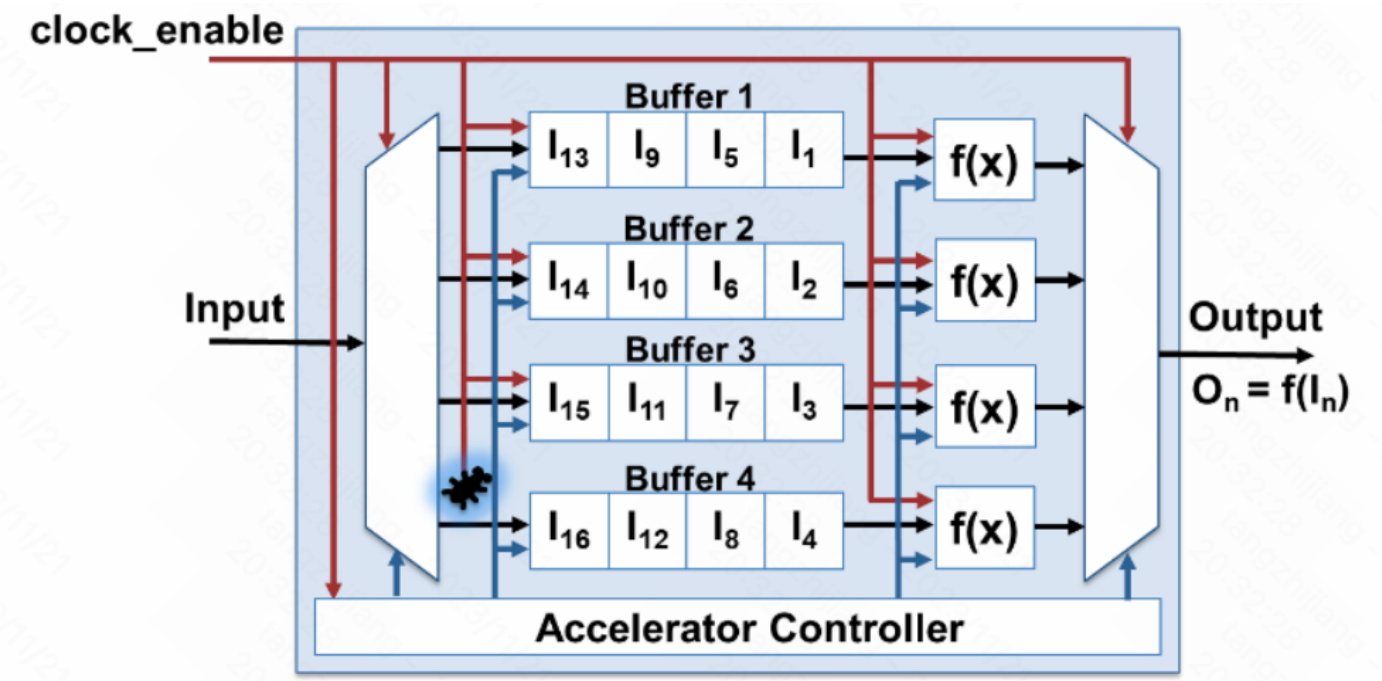


**Fig. 2**. Bug example: *clock_enable* disconnected from Buffer 4.

# 3.Formal Model

## 3.1 Basic Definition

将加速器形式化为一个有限状态转移系统。

**Definition 1:定义有限状态转移系统**$Acc := (S, S_{init}, rdin, A, a_{\perp}, D, O, o_{\perp}, T, F)$，**其中:**

- $S$表示加速器状态的集合，$S_{init} \in S$表示初始状态
- $rdin : S \rightarrow B$表示当前加速器是否处于可以接受输入的状态
- $A$ is a finite set of actions supported by the accelerator. $a_{\perp} \in A$ is a distinguished element of $A$ used to indicate that no operation is being selected or that the provided input is not valid.
- $D$表示输入数据的集合
- $O$表示输出数据的集合，$o_{\perp} \in O$ is a distinguished element of $O$ used to indicate that no output is being produced or that the output produced is not valid;
- $T : S \times A \times D \times B \rightarrow S$ is the state transition function
- $F : S \rightarrow O$ is the output function for action and data inputs

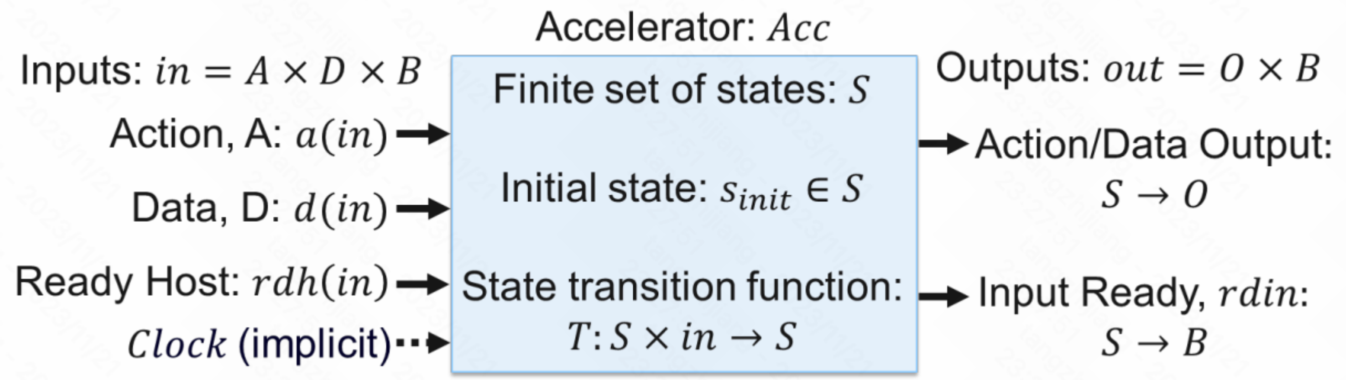


**Fig. 3**. Accelerator transition system model (Def. 1).

加速器的输入$in \in I = A \times D \times B$，包括action、data和一个布尔信号， which is the *host ready* signal, representing whether the host is ready to accept any output being produced in the current state.

记$a(in)$、$d(in)$、$rdh(in)$分别为以上三个维度。有时只需要看action和data维度，记为$ad(in)$。

给定状态$s$和一个输入$in$，下一个状态$s' = T(s, a(in), d(in), rdh(in))$，也记为$s' = T(s, in)$。

At each state $s$, Acc  produces an output $O = F(s)$ and the input ready predicate, $rdin$.

记序列$v =< v_1, \cdots, v_{|v|} >$，使用·将两个序列结合在一起$v = v_1 \cdot v'$，其中$v' =< v_2, \cdots, v_{|v|} >$。

记$C_{in}(s_0, in)$为$in$可用的子序列（captured inputs），对于$\forall in_i \in C_{in}, a(in_i) \neq a_\perp$

## 3.2 Functional Consistency(FC)

函数一致性（functional consistency），希望对于任意相同的输入，加速器都能给出相同的输出。

说加速器$Acc$是函数一致的，当且仅当对于所有的输入$in$、$in'$，如果

- $in^v = C_{in}(S_{init}, in)$，其中$|in^v| = k$
- $o^v = C_{out}(S_{init}, in \cdot in')$，其中$|o^v| \geq k$

则$\forall 1 \leq i < k. ad(in_i^v) = ad(in_k^v) \rightarrow o_i^v = o_k^v$

注意，函数一致的加速器在接收相应的输入之前不应该产生输出

## 3.3 Accelertor Response Bound(RB)

响应边界（Response Bound），要求加速器对于主机想要提供输入或获得输入时，加速器不能永远无响应。

说加速器$Acc$的响应边界为$n$，如果

- 对于任意的输入$in$，如果
  - $|\top(rdin(T(S_{init}, in)))| = k$，则
  - $\exists n. \forall in'. |in'| > n \rightarrow |\top(rdin(T(S_{init}, in \cdot in')))| > k$
- 对于所有的输入$in$，如果
  - $|C_{in}(s_{init}, in)| = k$，则

- $\exists n. \forall in'. |\top(rdh(in'))| > n \rightarrow |C_{out}(s_{init}, in \cdot in')| \geq k$

其中$\top(b)$表示对于布尔序列$b$等于$\top$的子序列。

## 3.4 Total Correctness

定义的FC和RB性质对于加速器是普适的，不依赖于特定的加速器。但这两个性质无法覆盖掉所有的functional bugs。为了完整性，我们现在就一个规范形式化了加速器输出正确性的概念。

### 3.4.1 Pre definition

对于给定的加速器$Acc$，称$Spec : A \times D \rightarrow O$为specification function for all action-data pairs (a,d), $a \neq a_\perp$，defines the expected output $Spec(a, d) \in O$ that the output function $F$ of $Acc$ is expected to produce.

#### functionally correct

称一个加速器是功能正确（functionally correct）的，对于所有的输入$in$、$in'$，如果

- $in^v = C_{in}(s_{init}, in)$，其中$|in^v| = k$并且
- $o^v = C_{out}(s_{init}, in \cdot in')$，其中$|o^v| = k$

则$o^v_k = Spec(ad(in^v_k))$

#### totally correct

An accelerator $Acc$ is *totally correct* with respect to a specification $Spec$ if it is functionally correct with respect to $Spec$ and responsive with a given bound.

#### single-action correct

An accelerator $Acc$ is *single-action correct* with respect to a specification $Spec$ if for every action-data pair $(a, d)$, $a \neq a_\perp$ and input sequence $in$ , if $k$ is the smallest value such that

- $in = (a, d, \perp) \cdot (in_\perp)^k$
- $o^v = C_{out}(s_{init}, in)$ with $|o^v| = 1$,

Then $o^v_1 = Spec(a, d)$

#### strongly connected

An accelerator $Acc$ is strongly connected if for every $in$, there exists $in'$ such that if $s = T(s_{init}, in \cdot in')$ with $|s| = k$, then $s_k = s_{init}$.

### 3.4.2 Proposition 1: totally correct

If a strongly connected accelerator $Acc$ is functionally consistent, responsive with some finite bound, and single-action correct with respect to $Spec$, then it is totally correct with respect to $Spec$.

# 4.A-QED Setup

# 5.Result

# 6.Conclusion