ARM

Formal Verification

Laurent Arditi, Senior Principal Engineer → Thèse vérif physique 12 ans chez ARM
Vincent Abikhattar, Engineer → Promo 2015 PFE en vérif

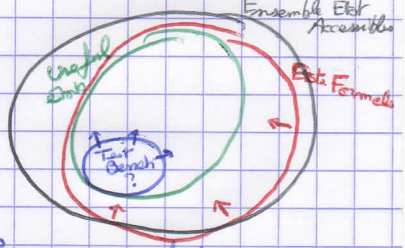
I. ARM

- Acorn + Apple (Smartphone) = ARM
- MARKET
  - Utilisation massive du smartphone
  - ⊕ Maintenant Gadjets
- NEW
  - Serveurs et Iot
- Conception IPs : Architectures
- Argent : Achat Licenses, Royalties
- Relation Etroite avec Fondeurs (Client Fournisseur)
- Utilisation très présente Archi ARM
- CPU eng : Sophia - Cambridge - Austin

II. Design Verification

- Exhaustif = Impossible
- Bon IP : Asserts, Bon par Construction
  - → Code Coverage
  - → Fonctionnal Coverage ⊕ → ✓
    - Coverage Driven Directed / Assertions / Random Unit Test Bench / System C
    - TOP Level Deterministic Test Code — Instructions, Simulation Cycle Accurate
    - FPGA Prototyping — Performances
    - Formal Verif — Property Checking, Assertion, Constraints
- Static Tests, Mathématiques
- Utilisation Model Checking

RTL → Model Checking ← Properties ← Design
Model Checking → Asserts ? → Contre-exemples



Parcours en profondeur VS Parcours en longueur

Propriété vérifiée jusqu'au cycle N ⇐

Bloc to Test ← Assertions Assuming

III. Diving into the Algorithms

- Bit Level ⊕ Propositionnal Logic
- Main Techniques → Theorem Proving (Très lourd et coûteux)
  - → Symbolic Simulation (//)
  - → Model Checking (utilisé partout maintenant)
- ROBDD (Reduced Ordered BDD) ⊕ Opé en tps linéaire ⊕ Tjr vrai ⇒ Etat Final = 1 ; faux ⇒ Etat Final = 0
  - Formule représentée par un BDD
  - circuit
  - → Comment accéder au 0 de l'assertion ⇒ TRACE
- Calculer l'espace des états atteignables (E)
  - * Partir d'un état (début ou fin) et appliquer les equations du circuits ⇒ Exploration — Point fixe atteint ?
  - * Mode Explicite : construction d'une représentation de E : table
  - * Mode Symbolique : structure de donnée adaptée : fonction = BDD !
- Mais CPU Modernes réellement complexes et gros que BDD marchent plus vraiment
- SAT solver : prop binaire satisfiable ?
  - problème NP complet
  - arrive à résoudre avec de + en + de clauses
  - DPLL Algorithm (1962)
  - Déplioge (seq → comb) au faire et à mesure des cycles et assertions jouées.

Passage d'un Pb Séquentiel à un Pb Combinatoire

- BMC (Bounded Model Checking)
  - → Poussée jusqu'à N états

Utilisé aussi → Localisation Erreurs, Learning, Optimisation

IV. Applications

- Assertions "embarquées"
  - * métrique → confiance même si undetermined ?
  - → suffisament d'assertions
- Toutes les branches & statements (Coverage)
  - code mort ? waivers ? existe à ne pas couvrir
- Prove Core Coverage (nb de portes nécessaires pour vérifier cette assertion ?)
- Sequential Equivalence Checking (comparaison entre 2 designs)
  - bug fix ? clock gating ok ?
- Sécurité (chemin possible entre clé qui est sécurisé et pas ?)
- ISA Formal : ARM ARM → XML
  - ↕ comp Design → BUG HUNTING