Virtualisation

Emulation de processeurs dans le développement HardWare

Par Arnaud Peeters et Jean Vanneste

Introduction

Emulation: Imiter le comportement d'un système (**la cible**) à partir d'un système différent (**l'hôte**).

- Possibilité d'exploiter un software initialement prévu pour une architecture différente de celle de l'hôte

Reproduction des spécifications du hardware et du comportement digital

- Minimiser l'impact sur le SW lorsqu'on désire changer d'archi HW
- Debugger un design de microcontrôleur avant qu'il ne soit produit

Utilisateurs en conception hardware

Lors du design d'un microcontrôleur:

- Développeurs Hardware:
  - Vérifier les bus et ses données
  - Debugger les erreurs avant le prototype FPGA
- Développeurs Software:
  - Pré-Tester les performances SW sur le système cible

Emulation vs Simulation

- Emulation: Ajoute des fonctions de debugging en imitant le CPU de la cible.
  Généralement en remplaçant le HW de la cible par du SW et un peu de HW
- Simulation: imiter la logique de fonctionnement

In-Circuit Emulation (ICE): le Full ICE (1)

L’hôte émule et remplace le CPU de la cible

Utilisation d'un "pod"

Le plus puissant des outils… aussi le plus cher

- Real-time debugging (pause, run, step-by-step, step-back)
- Trace d’erreur
- Breakpoints avec triggering
- Indépendance du reste du hardware
- Overlay de mémoire

In-Circuit Emulation (ICE): le Full ICE (2)

Inconvénients:

- Flexibilité: émulateur uniquement spécifique à une série d'architecture de cibles
- Prix: Impact sur le choix du CPU lors du design du microcontrôleur
- Dépendance entre l'hôte et le 'pod '
- Encombrant: possibilités de connexion parfois limitées

Background Mode Emulator (BME)

Connexion aux fonctions de debugging via les PINS du processeur

Dépendant d’un certain fonctionnement hardware de la cible

Real-time debugging mais pas à pleine vitesse

Breakpoints PARFOIS disponibles, en fonction du CPU cible

Pas d’overlay de mémoire possible

Pas de trace d’erreur

BME: Joint Test Action Group (JTAG)

Standard pour les BME

Garanti:

- Real-time debugging à vitesse de croisière
- Fonction de trace d’erreurs

Par rapport au Full-ICE: bon marché, connexion plus aisée

Simulation

- Les circuits logiques sont de plus en plus complexes
- La conception et la fabrication du HW sont des processus long et coûteux
- La simulation augmente l'efficacité du flot de conception

2 approches de la simulation

- Simulation : *Type de modélisation d'un système qui peut être mis en oeuvre sur un ordinateur* – A. Morawicc

- Simulation basée sur des langages de description du hardware
- Simulation basée sur des ISS (Instruction-set simulator)

Simulation par langage de description HW

- Permet de décrire le comportement des circuits au niveau des portes logiques
- Très précis et permet de visualiser les détails du comportement du circuit
- Lent

Simulation par langage de description HW

- Utilisation de langages dédiés : Verilog, VHDL...
- Prend en compte les caractéristiques temporelles et fonctionnelles des composants
- Structure en différents niveau d'abstraction
- Deux types d'input :
  - Description du circuit
  - Vecteurs de stimuli

Simulation par langage de description HW

Étapes d'exécution :

1. Compilation du langage de description
2. Élaboration de la hiérarchie du circuit
3. Initialisation de la structure de données (SD)
4. Execution de chaque processus de la SD
5. Execution d'un cycle de simulation

Simulation par langage de description HW

- Compilation : Le code VHDL est traduit en une "représentation cible" de plus bas niveau (code C, asm, executable ou format intermédiaire)
- La représentation cible sert d'input, avec les stimuli d'entrée, au noyau du simulateur qui va simuler le circuit
  - Simulateur interprété : le noyau est le seul executable
  - Simulateur compilé : le noyau, le cicuit et les stimuli d'entrée sont inclus dans un même executable

Exécution de la simulation

- Event-drive simulation
- Time-drive simulation
- Cycle based simulation

Test software sur µ-contrôleur

- On décrit l'architecture du micro-contrôleur et des modules externes via VHDL ou Verilog (parfois fourni par le fabricant)
- On intègre les instructions dans la mémoire du micro-contrôleur

Simulation basée sur des ISS (Instruction-set simulator)

On simule une "machine virtuelle" capable d'interpréter les instructions du micro-contrôleur

- Rapide
- Facile à mettre en oeuvre
- Moins précis
- Ne tiens pas compte des facteurs électroniques
- Pas de simulation avec modules externes

Conclusion

- Utilisés principalement pour le développement
- Chaque technique possède ses forces et ses contraintes; elles peuvent être utilisées à différents moment du développement