Hardware Trojans : taxonomie et méthodes de détection

Colloque National du GDR SoC-SiP

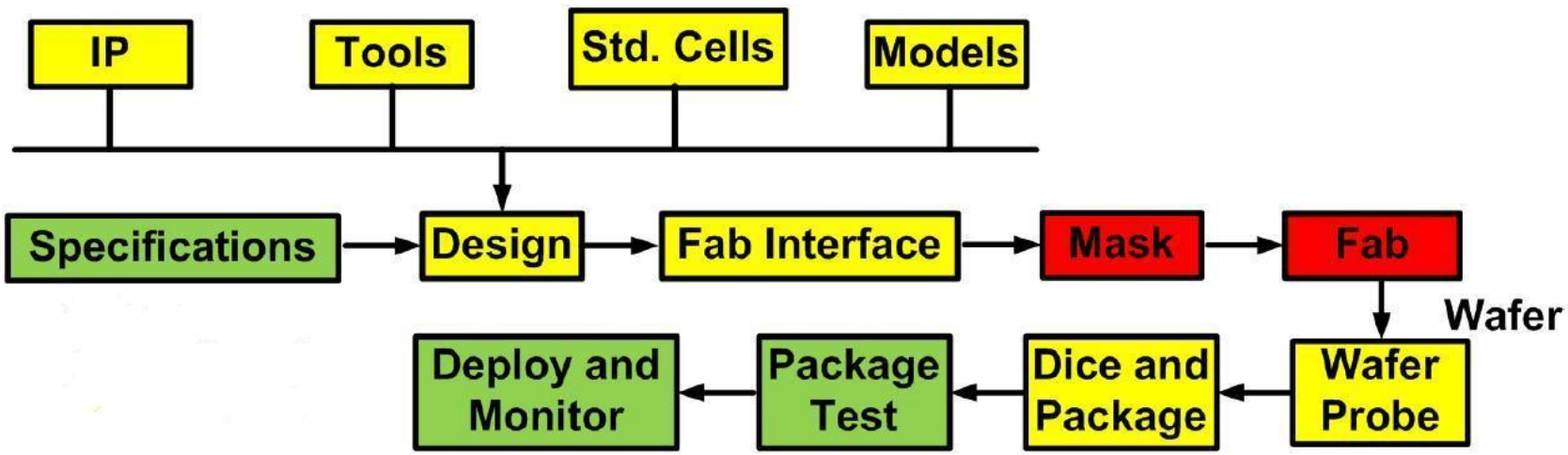
Julien Francq, julien.francq@cassidian.com

13 *juin* 2012

Sommaire

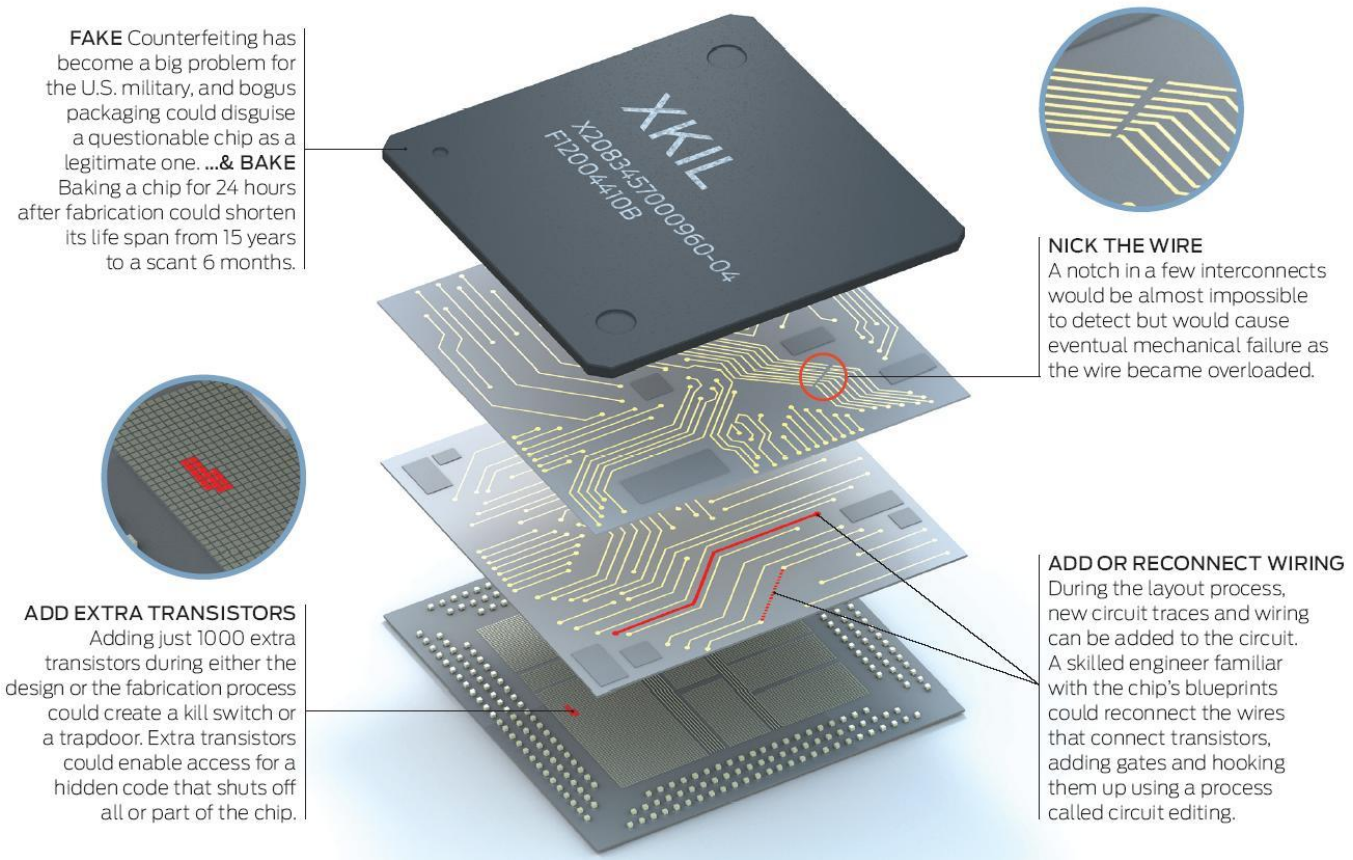
Contexte

- Délocalisation de la fabrication des circuits intégrés
- Difficile d'assurer la confiance



Hardware Trojan (HT)

- Modification frauduleuse d'un circuit intégré à n'importe quelle étape de sa fabrication



HTs : menace théorique ou pratique ?

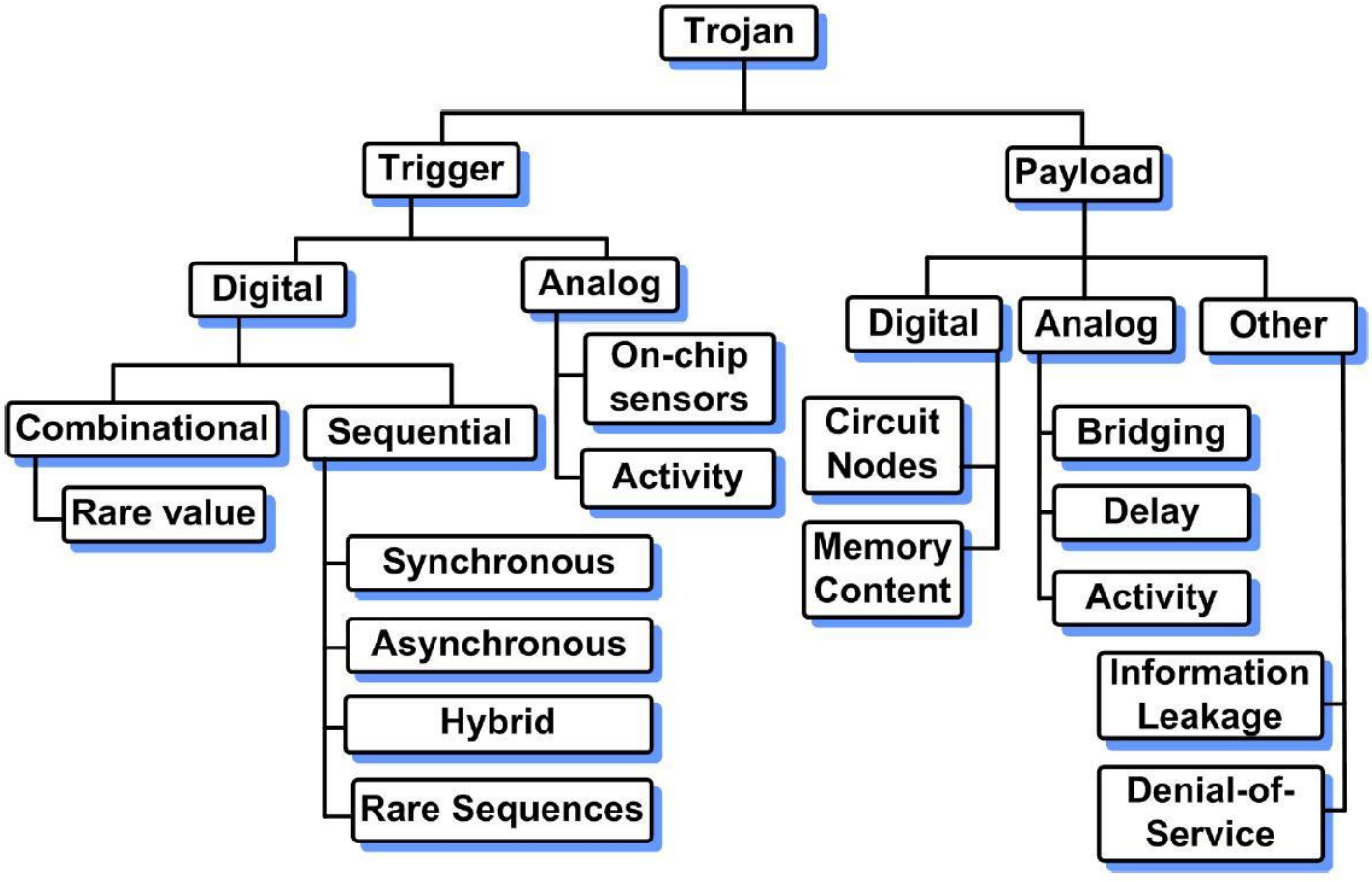
- 2005 : Défense américaine
- 2007 : DARPA "*Trust in IC Program*"
- 2007 : Israël *vs.* Syrie
- 2009 : "*Hot Topic*" de la conférence CHES
- Après 2009 : autres conférences (DATE, HOST, *etc.*)
- [Skorobogatov *et al.* : "*Breakthrough Silicon Scanning Discovers Backdoor in Military Chip*", CHES 2012]
- $\Rightarrow$ HTs : menaces émergentes et réelles

Quantification du risque des HTs

	Surproduction	Clonage logiciel	HTs
Attaquants	Usine	Concurrents	Terroristes
But	Revendre sur le marché gris	Vol de Propriété Intellectuelle	Déni de service, vol de données, sabotage
Impact	Économique	Économique	Risque sur la sécurité, l'économie, les infrastructures (sociétal)
Risque	+++	++	+

- Impact × Risque trop important pour être négligé

Taxonomie des HTs



- Bestiaire riche

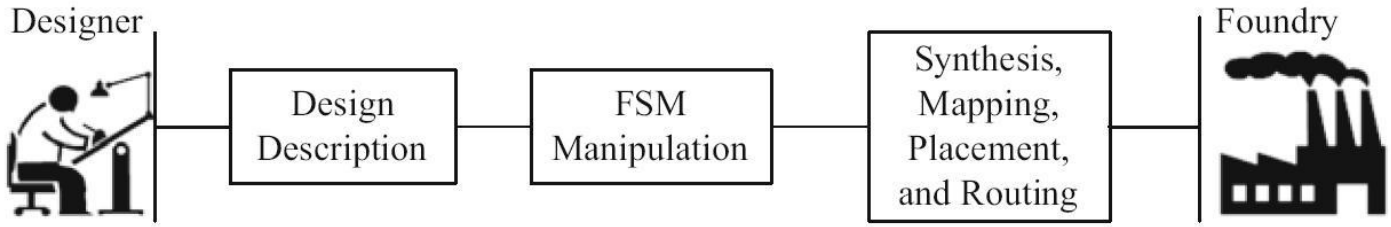
Effets néfastes des HTs

- *Kill switch*
  - Avion de combat
- Dysfonctionner le circuit
  - Satellite fonctionne 6 mois
- Faire fuir un secret
  - Communications chiffrées
- Assister un *malware* en fournissant un *backdoor*
  - Escalade de privilèges, connexions automatiques, vol de *passwords*
- Empêcher le *sleep mode*
  - Autonomie

Bilan

- N'importe quel circuit embarqué peut être infecté
- Impact (pour les fournisseurs et les utilisateurs) des HTs difficile à quantifier, mais peut être important :
  - Économique, financier
  - Réputation
  - Sociétal
- *Software Trojans* / *Hardware Trojans*

Un scénario d'insertion de HTs



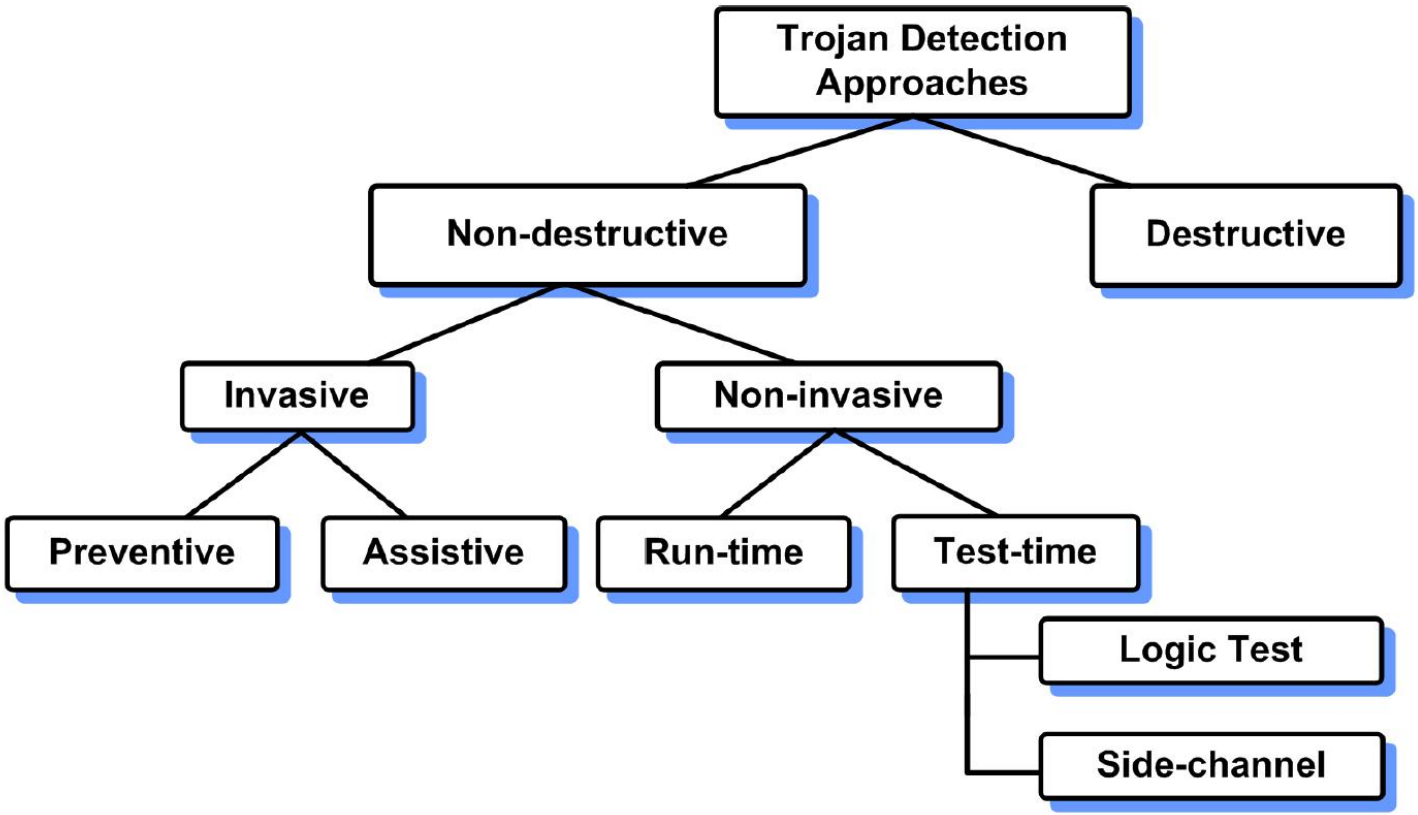
- L'attaquant récupère la description du circuit
- Il considère ses états et transitions comme une machine à états finis (FSM)
- Il y insère des états supplémentaires (HT)
- Il peut ensuite lancer la conception puis la fonderie du circuit infecté

Sommaire

Détecter des HTs ? Pas facile...

1. Les SoCs sont de plus en plus complexes, et détecter une petite modification malicieuse est difficile
2. L'inspection *via reverse-engineering* est coûteuse et difficile
   - N'offre d'ailleurs aucune garantie
3. Par nature, les HTs sont conçus pour être difficilement détectables
   - Notamment par test intégré
4. Par nature, les HTs sont petits
   - Difficile de les détecter par analyse de canaux auxiliaires

Liste des méthodes



- Aucune méthode n'est pleinement satisfaisante aujourd'hui

Rétro-ingénierie

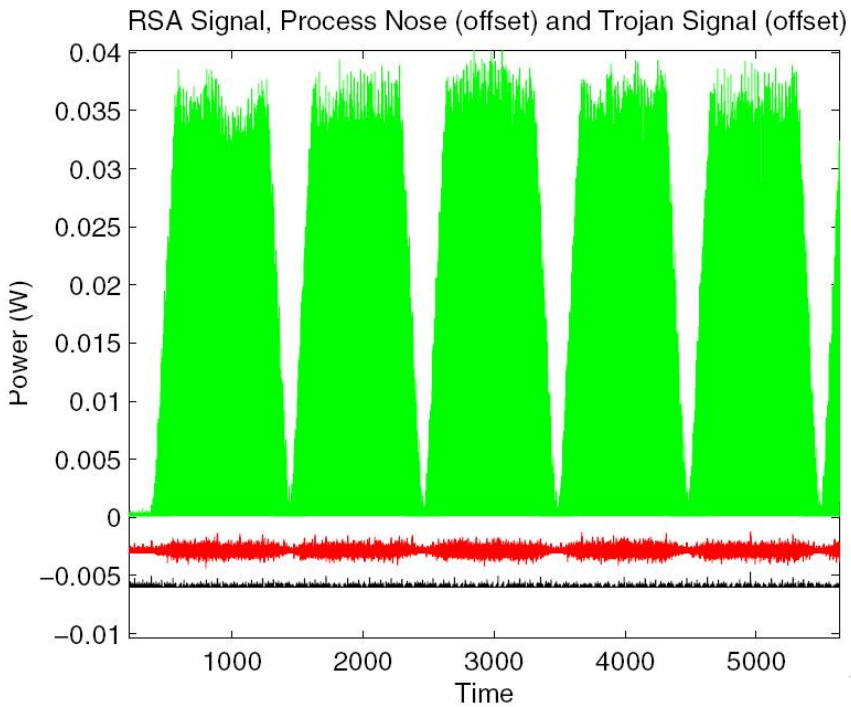
- Circuit testé = circuit jeté
- Coûteux en argent, temps, personnel
- + la densité d'intégration augmente, plus l'investigation est complexe
- On en teste un parmi N
- $\Rightarrow$ Pas adapté

Obfuscation

- Pour insérer un HT, un attaquant a besoin de trouver les points propices d'insertion
  - Évènements rares
- $\Rightarrow$ Obfusquer un circuit rend difficile son appréhension
  - L'insertion des HTs sera donc soit bénigne, soit facilement détectable
- Élégant et à double emploi (Prévention des HTs et protection IP), mais :
  - Le circuit doit être modifié, donc travail supplémentaire pour le concepteur,
  - Le test est plus compliqué et devient plus coûteux.

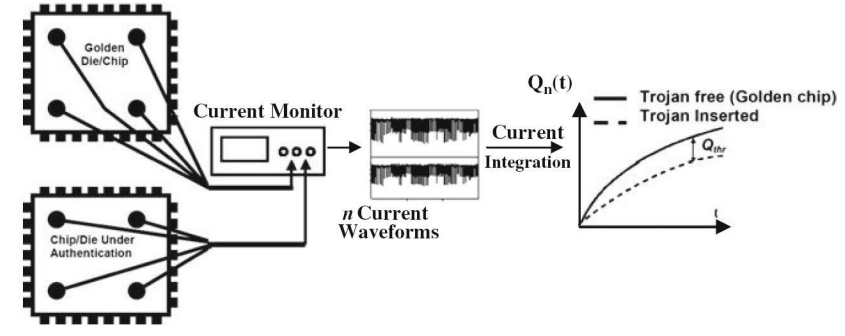
Analyse de canaux auxiliaires globale

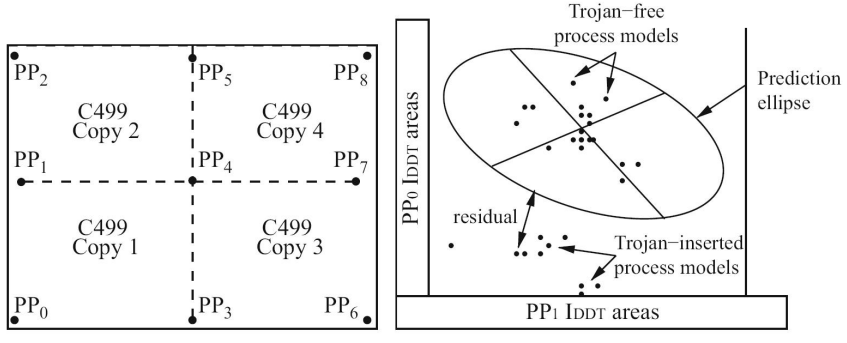
- Canaux auxiliaires : consommation électrique, rayonnement électromagnétique, *etc.*
- Nécessite des circuits certifiés sans HT
  - *Reverse-engineering*
- Puis, acquisition de relevés de référence et comparaison avec les autres circuits



Analyse de canaux auxiliaires locale

- Les mesures locales sont plus efficaces que les mesures globales
- Nécessite encore un circuit certifié sans HT





- Maximiser/Minimiser l'activité dans des régions

Analyse de testabilité

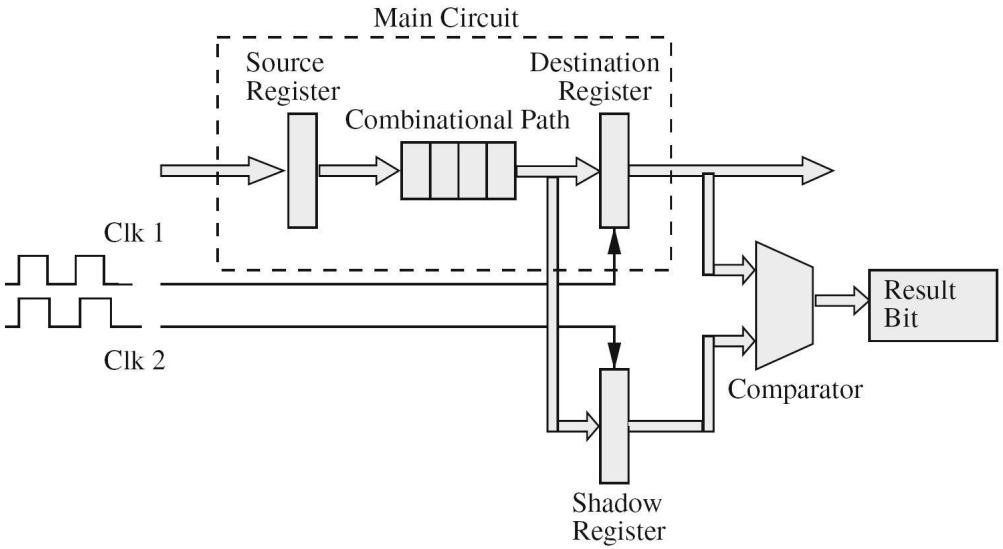
- Approche déterministe difficile
  - Beaucoup de HTs possibles
  - Fonction combinatoire de quelques nœuds du circuit
  - ⇒ Énumération exhaustive impossible
- Approche statistique :
  1. Trouver les événements rares dans le circuit
  2. Établir une liste de HTs insérables
  3. Générer des vecteurs de test et estimer leur couverture
  4. ⇒ Jeux de vecteurs de test de qualité
- 85% de test en moins / aléatoire, mais - efficace avec de nombreux points de *trigger* et long

Verrous technologiques

	Test logique	**SCA**
Avantages	Efficace pour des petits HTs Robuste face au bruit de *process*	Efficace pour des gros HTs Génération des tests facile
Inconvénients	Génération des tests complexe Détection des gros HTs difficile	Vulnérable face au bruit de *process* Détection des petits HTs difficile

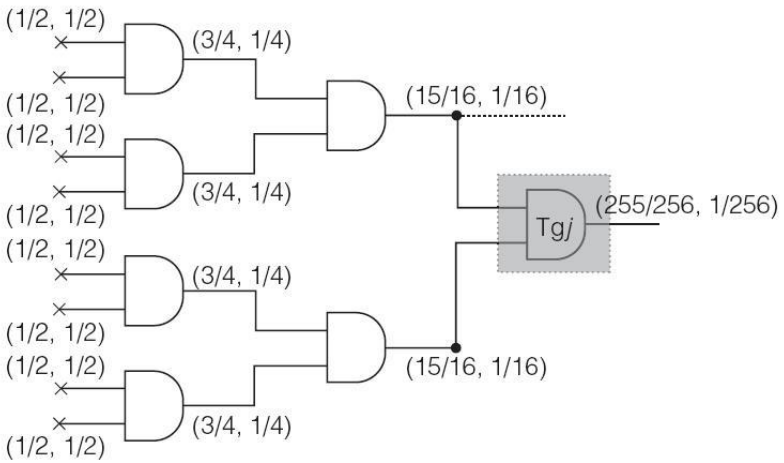
- Test logique et SCA complémentaires

Analyse de délais internes

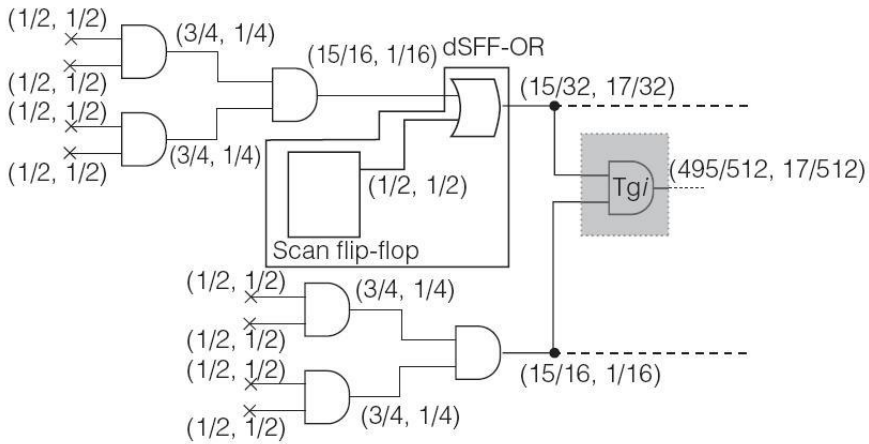


- Mesure des délais entre registres
- Clk1 a la même fréquence que Clk2, mais déphasé
- Millions de chemins ⇒ Gros surcoût

Insertion de bascules



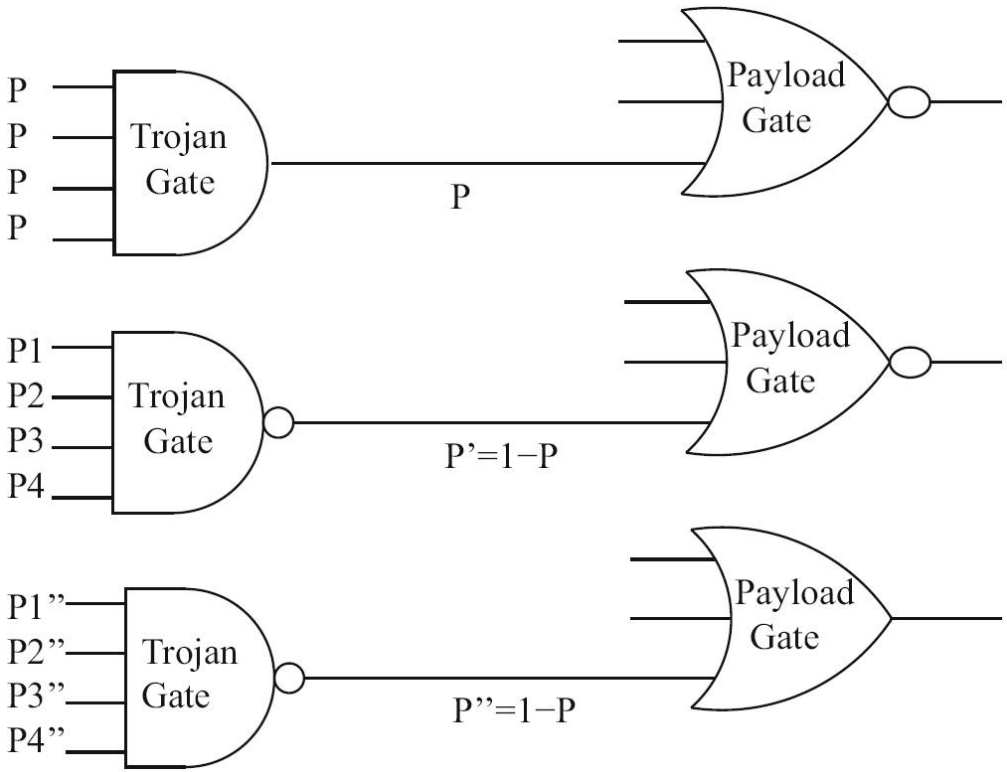
Transition probability at Trojan output = 255/65536
Average clock cycles per transition by GD = 255.6
Average clock cycles per transition by simulation = 250



Transition probability at Trojan output = 8415/262177
Average clock cycles per transition by GD = 30
Average clock cycles per transition by simulation = 33.4

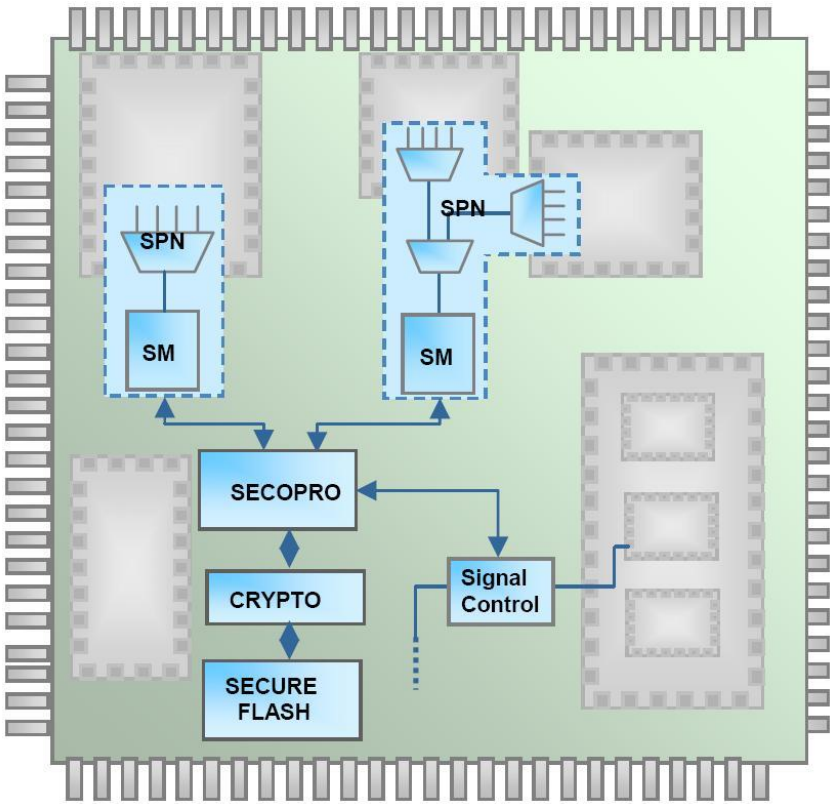
×: Represents scan flip-flop or primary input

VITAMIN



- Peu adapté

Détection en *run-time*



- Désactiver un bloc suspect ou forcer une opération
- SPN : *Signal Probe Network*
- SM : *Security Monitor* ($\sim$ FSM)
- SECOPRO : *Security and Control Processor*
- Configurations chiffrées et stockées dans une Flash sécurisée
- Surcoût ?

Sommaire

Principaux objectifs (1/2)

- Test rapide
  - 1 min. / circuit (au lieu de l'heure)
- Détecter des petits HTs
  - $\leq$ 0,005% de la taille du circuit
- Détecter tout type de HT numérique
  - Combinatoires et séquentiels
- Garantir un fort taux de détection
  - $\leq$ 1% de faux positifs
  - $\leq$ 0,1% de faux négatifs

Principaux objectifs (2/2)

- Méthodes validées sur circuits réels
  - Littérature : simulations
- Obtenir des méthodes résilientes face aux variations de *process* de fabrication des circuits
  - Sur circuits réels
  - Test sur 100 FPGAs (bas coût)
- Méthodes résilientes face à la loi de Moore
  - Test sur des gros et récents FPGAs

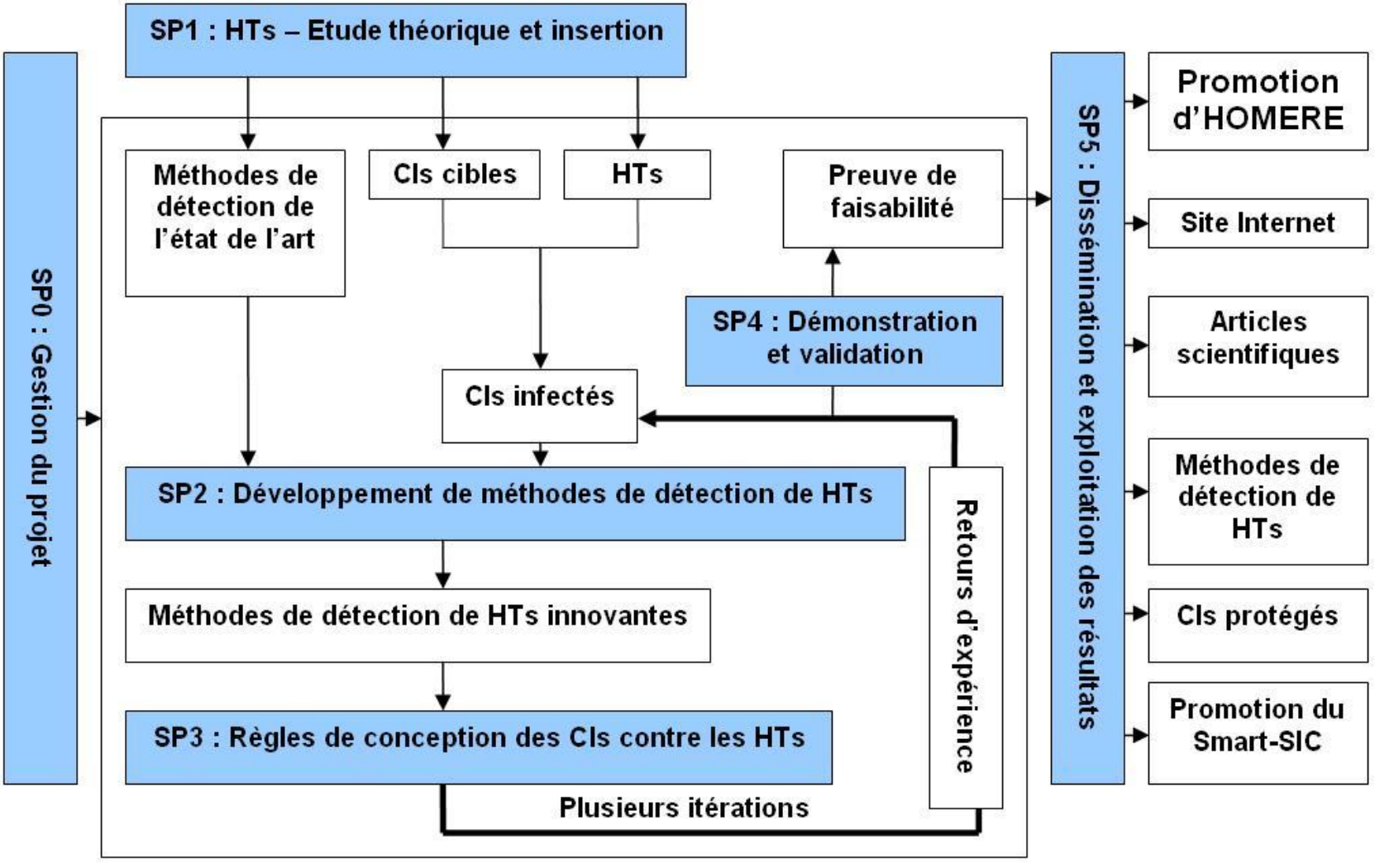
Résultats attendus

- Estimer les difficultés d'un attaquant
- Développer des méthodes de détection de HTs efficaces
- Obtenir de nouvelles versions de circuits facilitant la détection de HTs
- Développer un outil industriel de détection de HTs

Les partenaires

- Grands groupes
  - Cassidian CyberSecurity, Gemalto
- PME
  - Secure-IC
- Académiques
  - ARMINES, CEA-LETI, LIRMM, Télécom ParisTech
- Gouvernemental
  - ANSSI, (DGA)

Organisation du projet



Sommaire

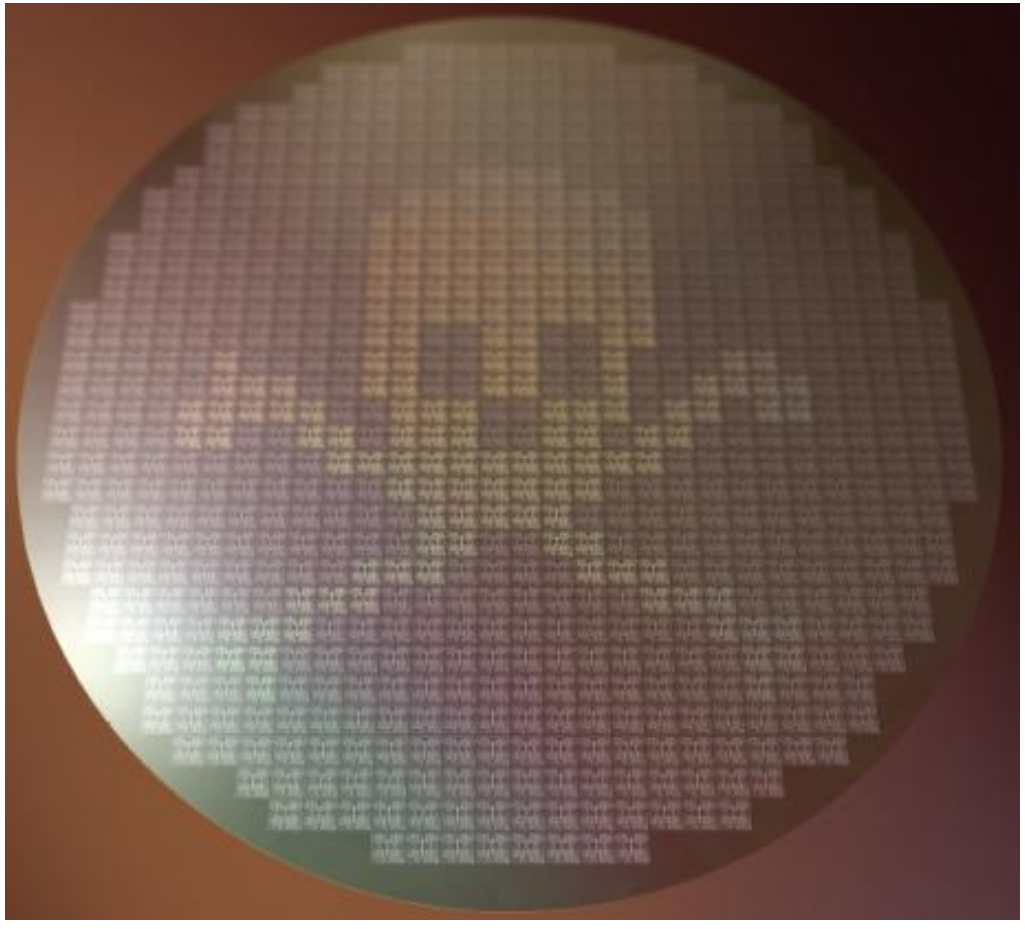
Conclusion

- Les HTs sont des menaces réelles aux conséquences graves
- Détecter / Neutraliser tous les HTs est un vrai challenge
- ⇒ Combiner plusieurs méthodes
  - Analyse de canaux auxiliaires et test intégré
- 3 lignes de défense :
  - Préventives avant fonderie
  - Tests avant déploiement
  - Détection en *run-time*

Quelques réflexions pour HOMERE

- Pour augmenter nos chances de succès, une première série de tests sera effectuée sur des circuits à HTs connus des testeurs
- Pour pouvoir fournir des résultats intègres, une seconde série de tests sera effectuée sur des circuits à HTs inconnus des testeurs
- Augmenter graduellement la difficulté
- Partir de *scenarii* d'attaques réalistes $\rightarrow$ HTs
- On peut être confiant : le *consortium* de HOMERE est solide

Je vous remercie de votre attention. Avez-vous des questions ?



Présentation réalisée à l'aide du *template* Beamer Montpellier