

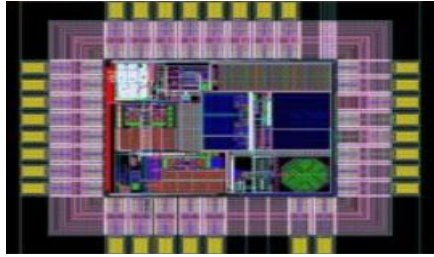
Le Langage VHDL

© E.Dekneuvcl 2019

Quels sont les techniques (modèles) et les moyens (outils,langages) mis à la disposition des descripteurs de hardware?

Modèle de circuit

circuit



Abstraction



ordinateur

(analyse de la fonctionnalité / performances)



Les circuits modernes comportent des millions de transistors (ex: téléphone portable)
Modéliser revient à décrire (sur papier/logiciel) un circuit en faisant abstraction de toute réalisation physique (cablage de composants discrets, programmation de FPGA ou réalisation d'un silicium afin de vérifier son fonctionnement et prédire ses performances et éviter ainsi la réalisation coûteuse de prototypes.

Blocs combinatoires

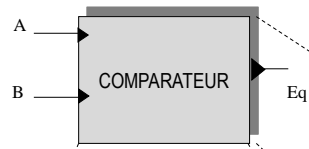


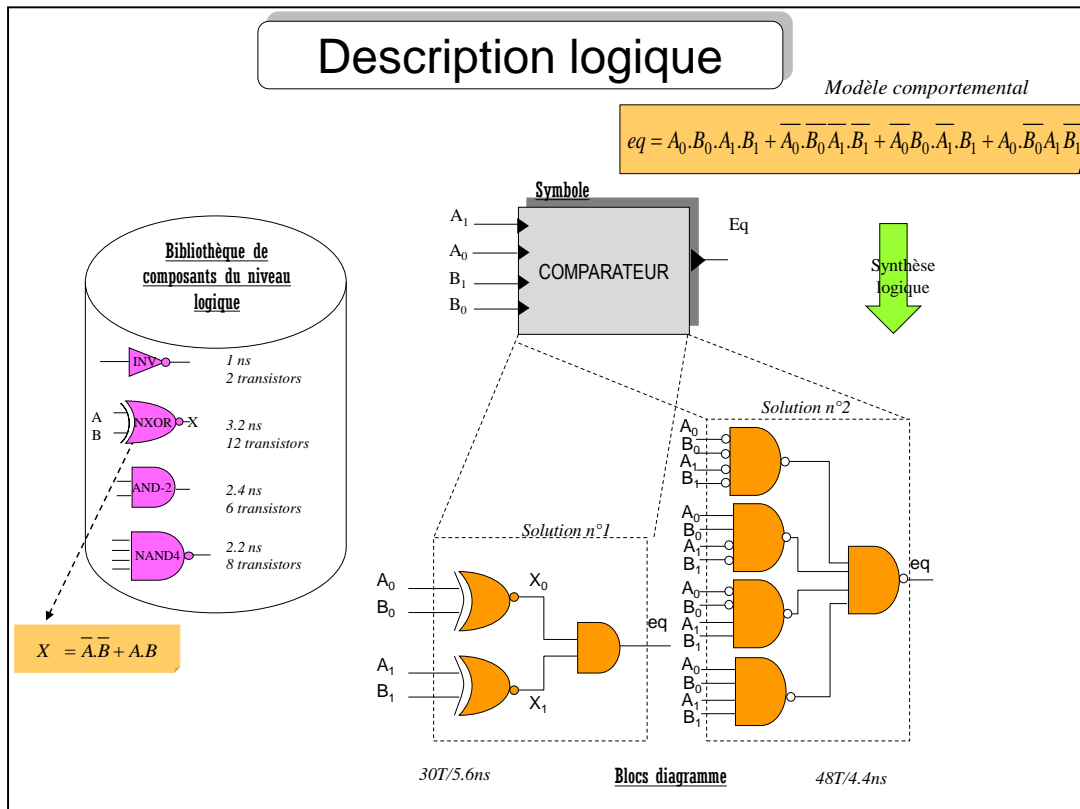
Table de vérité

A ₁	B ₁	A ₀	B ₀	EQ
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
....				...
1	1	1	1	1

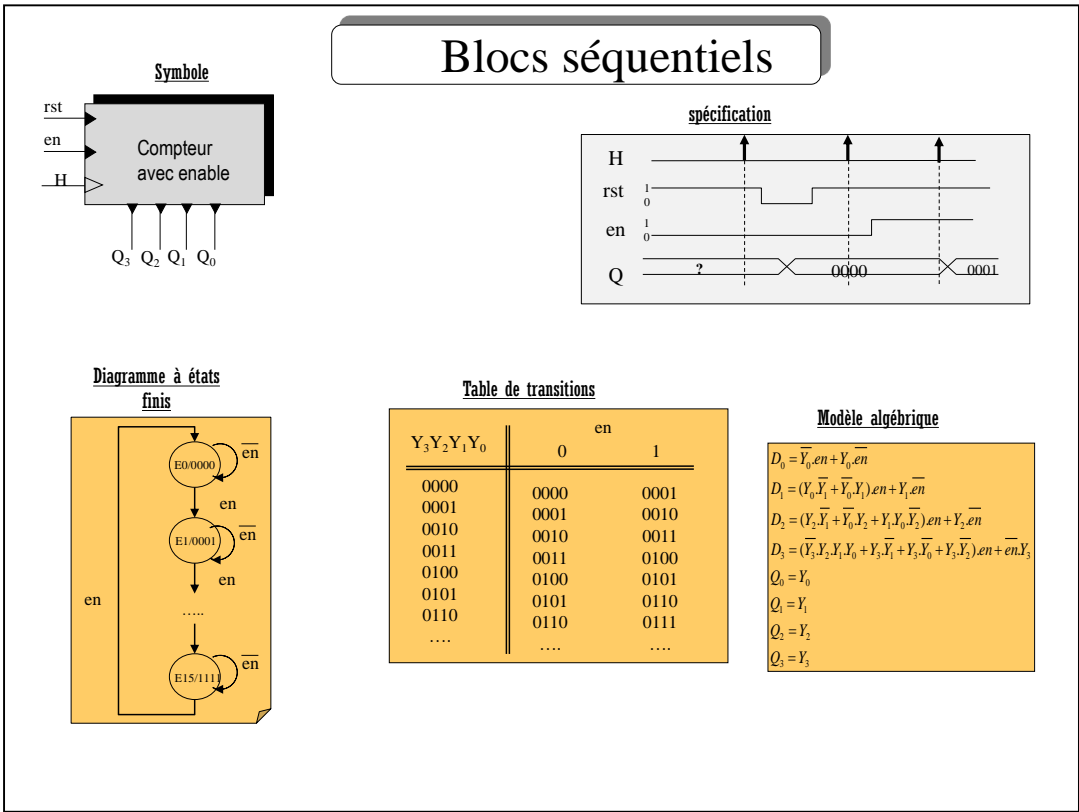
Equations logiques

$$eq = A_0.B_0.A_1.B_1 + \overline{A_0}.B_0.A_1.B_1 + A_0.B_0.A_1.B_1 + A_0.B_0.A_1.B_1$$

On rappelle que la valeur des sorties est déterminée à tout instant par la valeur des entrées. La spécifications de blocs combinatoires tel que le bloc comparateur peut être décrit à partir d'une table de vérité (modèle tabulaire) ou une équation algébrique(modèle mathématique) basée sur les opérateurs ET,OU,NON.



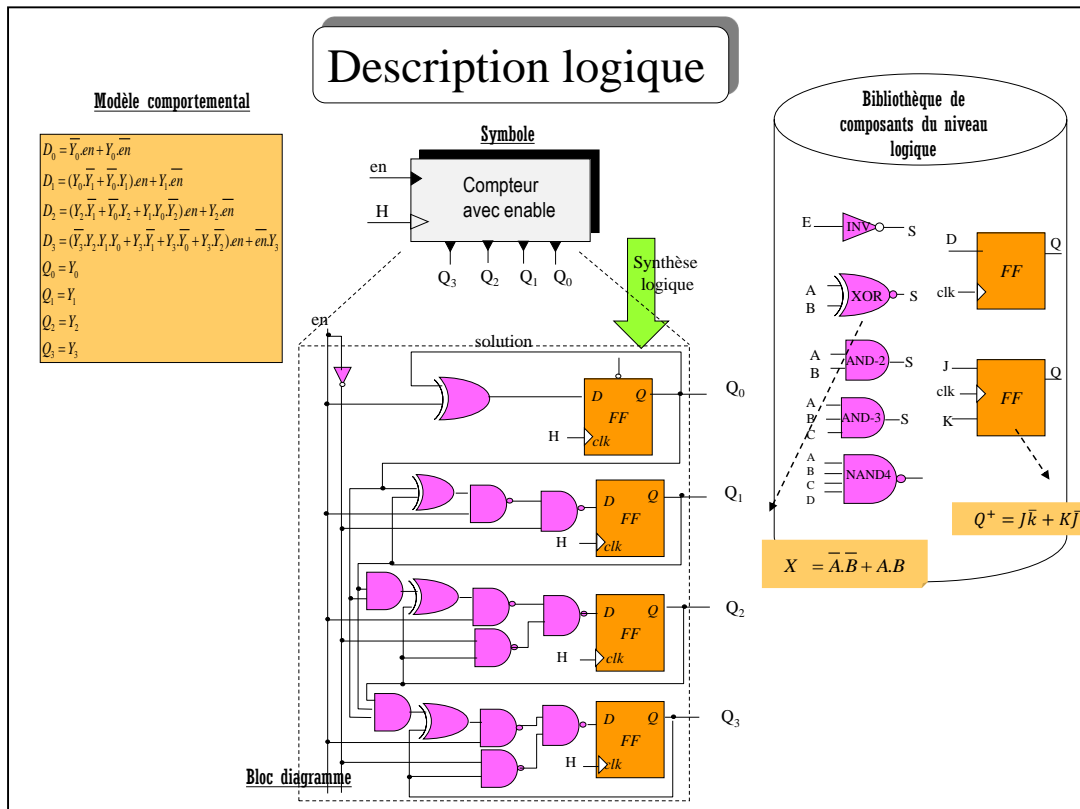
On peut également décrire le composant à partir de sa structure interne comme interconnexion de composants (portes ici), le comportement de chaque composant pouvant à nouveau se décrire par une TV ou une équation. On apporte ici une solution d'implémentation de la description comportementale selon différents critères: timing, surface, consommation, etc. On notera que la solution à base de NXOR est moins chère (30 vs 48 T CMOS) mais moins performante (5.6 vs 4.4 ns) que la solution à base de NAND. Si nécessaire, il est possible de raffiner la description structurelle en dérivant la structure d'une porte NXOR



On rappelle que dans un système séquentiel, la sortie dépend également d'un état interne avec 2 catégories:

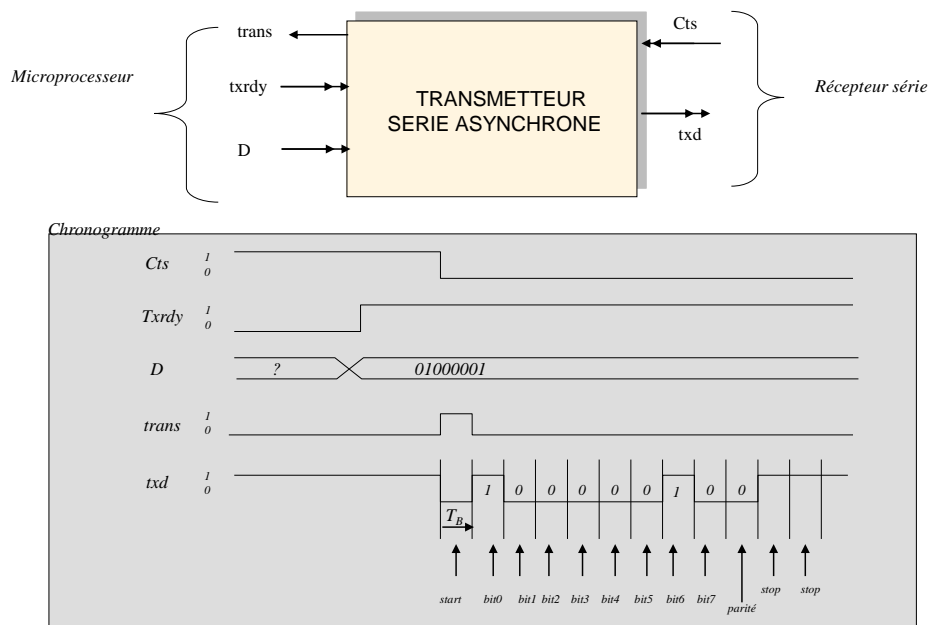
- système séquentiel asynchrone: une modification de l'état peut intervenir à tout instant, sur modification d'une des entrées ou de l'état interne
- système séquentiel synchrone: une modification de l'état ne peut se produire que sur une commande externe d'horloge.

On peut décrire le comportement global du composant par un diagramme à états finis et extraire les équations d'entrées des éléments mémoires et des sorties à partir de la méthode de synthèse tabulaire. Cela suppose une étape de codage des états (ici par les sorties). Ces différents modèles sont un raffinement du chronogramme.



La synthèse logique d'un circuit séquentiel procède de manière similaire sauf que l'extraction des équations d'excitations des bascules nécessite le codage préalable de la machine. La solution utilise des FF comme éléments mémoires (car le système séquentiel est synchrone) en plus de portes. Comme pour les blocs séquentiels, plusieurs mapping sont possible en fonction du codage notamment.

Circuits complexes



Pour des circuits complexes dans lesquelles les transformations de données priment par rapport aux problèmes de contrôle, le modèle des FSM est inadapté.

Spécifications comportementales HDL

Algorithmic State Machine

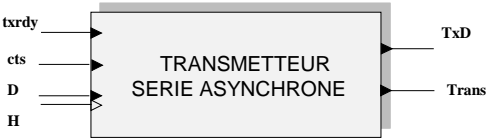
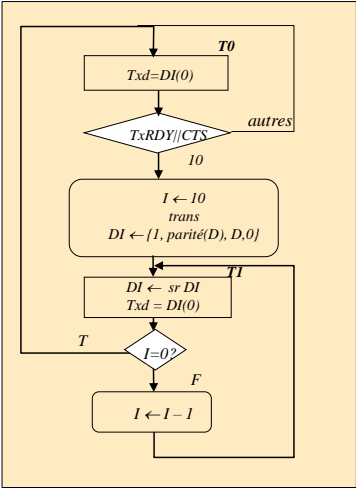
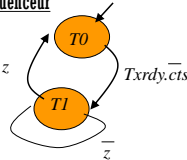


Table d'état étendue

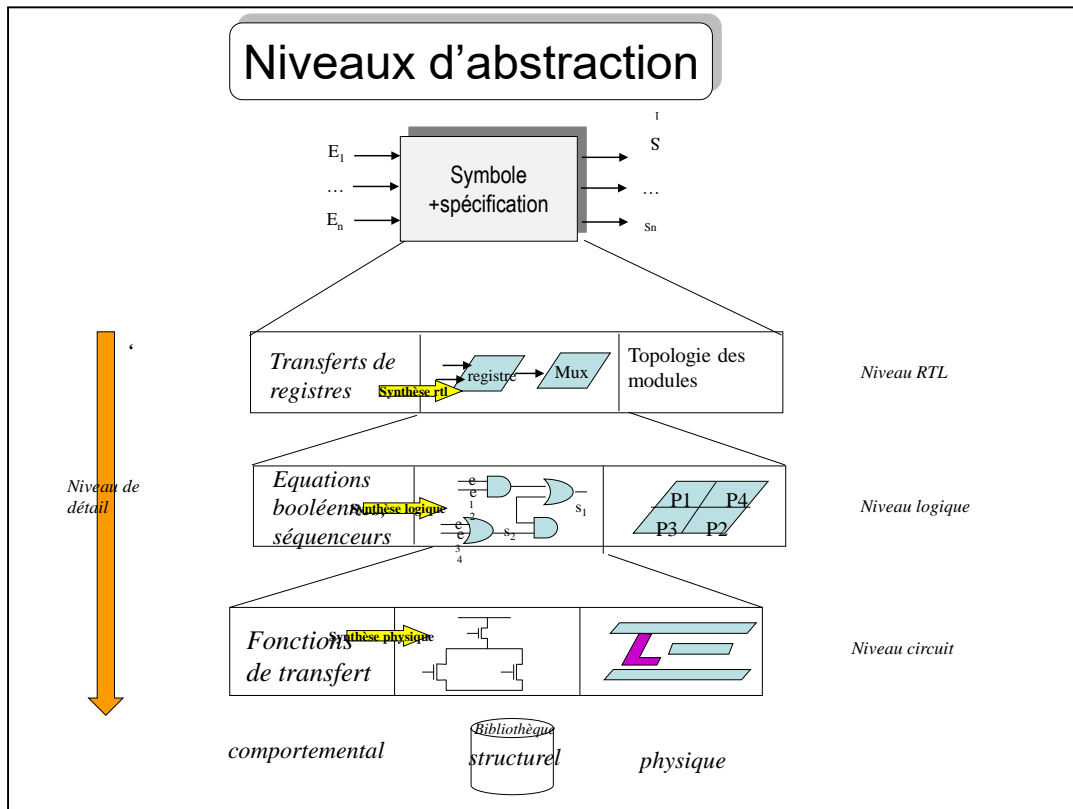
ETAT COURANT	CONDITIONS $1 \dots N$	SORTIE DU CONTROLE		VARIABLES DU DATAPATH		SORTIES DU DATAPATH	ETAT SUIVANT
		Trans		DI	I	TxD	
T0	Txrdy et not cts	T	1	{1,parité(D),D,0}	10	DI(0)	T1
		F	0	DI	I	DI(0)	T0
T1	I=0	T	0	sr DI	I	DI(0)	T0
		F	0	sr DI	I-1	DI(0)	T1

Transfert de registres conditionnels/sequenceur

T0: txrdy.cts/:I <- 10//DI <- {1,parité(D), D,0}
T1: DI <- sr DI//Txdata = DI(0)
T1.Z/:I <- I-1
Z=(I=0)
Txdata=DI(0)



Le composant nécessite la définition de variables telles que I ou DI, variables permettant la spécification d'opérations dites micro-opérations . Un algorithme permet de définir le séquencement de ces micro-opérations Les conditions peuvent être complexes et peuvent faire appel à des opérateurs relationnels quelconques, éventuellement composés par des ET, OU, NON.

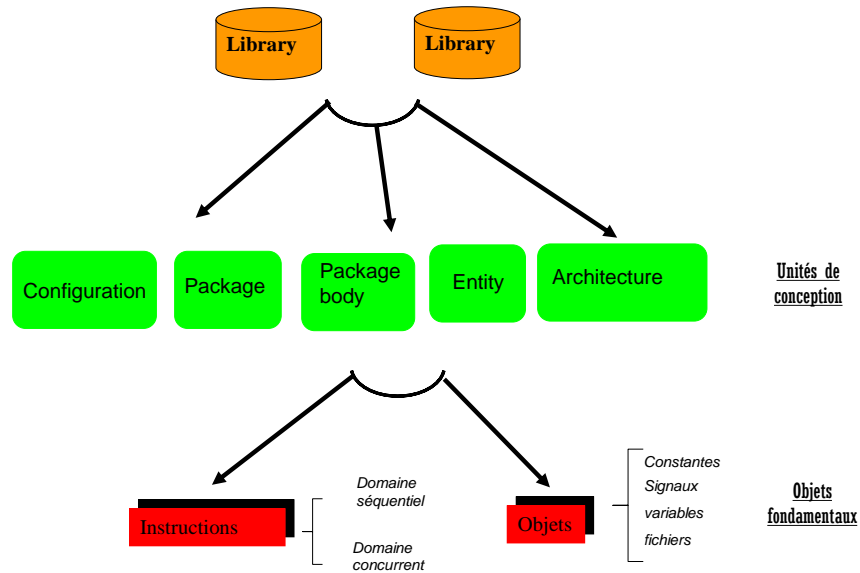


Selon la complexité du circuit, on peut être amené à décrire le circuit à des niveaux d'abstraction plus ou moins détaillé. 2 types de description (domaines) peuvent être établies selon que l'on décrit le composant:

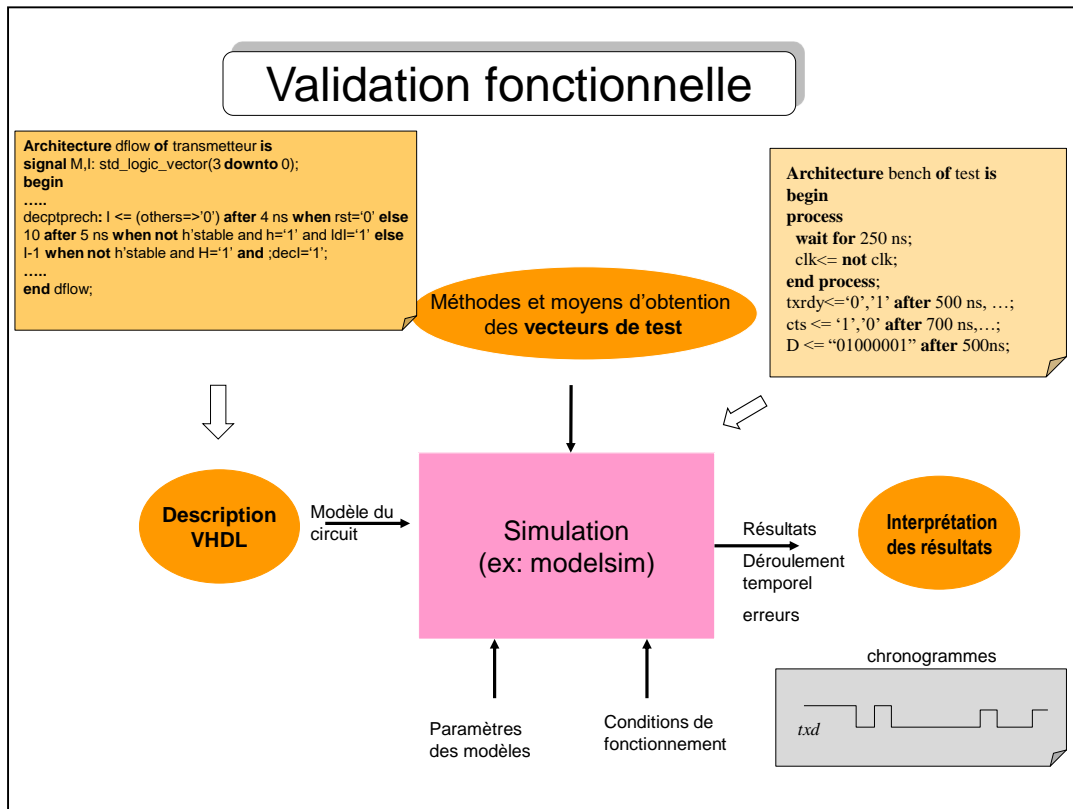
- comportemental: par une description de la relation entrée (état) sorties et en faisant abstraction de toute implémentation
- Structurel: c'est une vue interne qui matérialise une vue externe du circuit à partir d'une bibliothèque de composants
- Physique: c'est la vue qui s'intéresse au placement routage des composants sur le circuit

Le passage du domaine comportemental au domaine structurel (synthèse) s'obtient par MAPPING (allocation) de composants d'une bibliothèque.

VHDL: Vhsic Hardware Description Language



Le VHDL (DoD,81) a pour vocation de décrire un circuit à l'aide d'instructions comme dans un langage classique. L'organisation du langage repose sur une structure arborescente à 3 niveaux et la notion d'entité-unité de conception, les unités de conception pouvant être elles-mêmes des entités encapsulant d'autres unités de conception ou composées d'instructions et objets élémentaires. La bibliothèque stocke le résultat de la compilation des unités de conception qui peut se faire séparément, ce qui facilite la conception modulaire et le travail en équipe.



La validation fonctionnelle est une activité essentielle dans la conception des circuits car elle permet de valider le fonctionnement du circuit (vérifier la description par rapport à la spécification) avant toute implémentation sur une technologie cible. La validation d'une description (structurelle ou comportementale) nécessite la stimulation de la description au moyen d'un jeu de vecteurs (séquence) de tests (stimulus d'entrée) dont la définition peut se faire:

- * manuellement: à l'aide d'un fichier de vecteurs de tests (forme textuelle/testbench) ou à partir d'un éditeur graphique de chronogramme.

- * automatiquement: par des composants ou des procédures modélisant le comportement des entités de l'environnement du circuit (ex: modélisation des cycles de bus d'un processeur).

Les chronogrammes en sortie sont interprétés et peuvent être comparés à ceux issus d'autres niveaux de description du circuit pour vérifier la conformité. La vérification peut être manuelle ou automatique (Smart/intelligent tests).

Organisation et sommaire

- **Eléments fondamentaux**

- A. Modèle entité-architecture
- B. Classes d'éléments
- C. Les types de données

- **Domaines de définition**

- A. Domaine séquentiel
- B. Domaine concurrent
- C. Sous-programmes
- D. Attributs

- **Compléments**

- A. Les fichiers
- B. Configuration
- C. Les packages

- 10 h de cours

- 12 heures de TD

- 5 séances de TP

Modelsim (Mentor Graphics)

- 2 examens de cours

- 1 examen de TP



Jalon
dxi764XUQ