DESSIN
Développement de SystèmeS Intégrés Numériques

http://www.comelec.enst.fr/dessin/

Première partie

Lirida Naviner
Jean-Luc Danger
Yves Mathieu
Jean Provost

Avril 2004

Département Communications et Electronique
Ecole Nationale Supérieure des Télécommunications

Introduction

Présentation. L'industrie du semi-conducteur, économie, domaines applicatifs, technologies.

Plan

- La brique DESSIN
- L'industrie du semi-conducteur
- Perspectives d'évolution
- Les SoC (System on Chip)
- Les métiers

Plan

- **La brique DESSIN**
- **L'industrie du composant**
- **Perspectives d'évolution**
- **Les SoC (System on Chip)**
- **Les métiers**

DESSIN, objectifs

- **Vous présenter la filière des systèmes intégrés numériques**
- **Vous former aux techniques et méthodes de conception**
- **Vous faire découvrir les outils, leurs possibilités, leurs limites, leur coût, leurs évolutions possibles**
- **Vous permettre**
  - ***De vous intégrer à une équipe de concepteurs***
  - ***De dialoguer avec "les gars du hard"***
  - ***D'embarquer pour la planète CAO***

DESSIN, moyens

- **60 TH (90 heures)**
- **12 postes de travail (F301)**
- **Enseignants**
  - ***Lirida Naviner, Jean-Luc Danger, Yves Mathieu, Renaud Pacalet, Alexis Polti (ENST)***
  - ***Frédéric Pétrot (LIP6)***
- **Conférenciers**
  - ***Yves Mathys (Motorola Genève)***
  - ***Stéphane Mancini (ENSERG Grenoble)***

DESSIN, moyens

- **Des étudiants**
  - ***Brique AMEN***
  - ***Stage***
  - ***Pas stage***
- **Un bouquin : «** Application-Specific Integrated Circuits **», Michael John Sebastian Smith, Addison Wesley**
- **Beaucoup de bonne volonté**

DESSIN, organisation

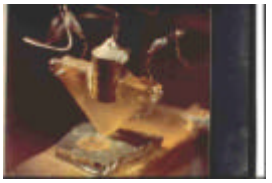
- **De la théorie (très pratique)**
  - ***Module ASOM***
  - ***25 TH de leçon, devoir sur table***
- **De la pratique (très théorique)**
  - ***Module CANEX***
  - ***7 TH de cours/TD, 20 TH de TP***
- **Des conférences**
  - ***Yves Mathys (Motorola, 3 TH)***
  - ***Stéphane Mancini (ENSERG, 6 TH)***

Plan

- **La brique DESSIN**
- **L'industrie du semi-conducteur**
- **Perspectives d'évolution**
- **Les SoC (System on Chip)**

Un peu d'histoire

- **Fin du XIX^{ème} siècle**
  - ***Démonstration du potentiel des technologies électroniques pour les transmissions sans fil***
- **Seconde guerre mondiale**
  - ***Application au calcul (décryptage)***
  - ***Projet Manhattan***
  - ***Guidage des bombardiers***
  - ***ENIAC (ENorme Instrument A Calculer)***
- **1947-1954**
  - ***Invention puis commercialisation des transistors à semi-conducteurs***
  - ***Les « transistors » remplacent les postes radio à tubes***
- **Années 60 et 70**
  - ***Premiers circuits intégrés, LSI***



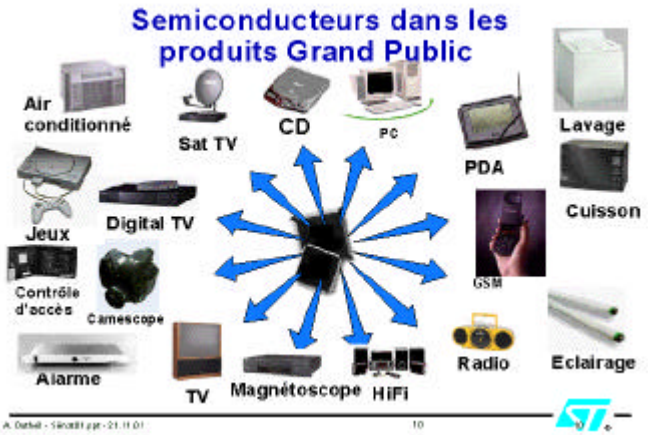
Un peu d'histoire

- **1960-2002**
  - ***Réduction des géométries ($2^{40/3}$ » 10000)***
  - ***Evolution anticipée (Feynman, Moore)***
    - **Lisibilité des progrès technologiques**
    - **Anticipation des nouveaux marchés**
    - **Coûts (croissants) de R&D**
  - ***Industrie au cœur de la croissance économique des 50 dernières années***
- **Aujourd'hui**
  - ***Circuits mixtes (télécommunications)***
  - ***Micro-systèmes (capteurs, actionneurs intégrés)***
  - ***Nano-technologies (horizon 5 à 10 ans)***
  - ***130 Þ 90 nm***
    - **Coûts de R&D de plus en plus élevés**
    - **La demande (ordinateurs, téléphones mobiles, etc.) stagne**
    - **Peu de produits vraiment nouveaux (VHS Þ DVD, caméscopes Þ numériques, téléphones fixes => mobiles)**
  - ***Crise financière et économique des TIC***

L'importance des « puces »

- **Invisibles**
  - ***Masquées par les services qu'elles fondent***
- **Stratégiques**
  - ***Militaire, énergie, espace, santé, progrès scientifiques***
- **Impact considérable**
  - ***Sur les comportements sociaux***
  - ***Sur la vie économique***
- **Responsables de l'essentiel de l'évolution des produits et services**

L'importance des « puces »



Exemple : l'automobile

Pourcentage de l'électronique dans le coût direct

- *2000 : 20%*
- *2010 : 40%*

Sécurité	Confort, aides à la conduite	Performances, environnement	Productivité du temps de transport
Freinage urgence ABS EVC ASC Eclairage Xenon Vision nocturne Airbag Prétensionneur Ceintures Anticollision Anti-effraction/vol Commandes essuie-glace Démarrage carte à puce Pression pneus Détecteur de pluie	Climatisation Direction électrique Vitres électriques Navigation Affichage multimédia Calculateur de maintenance Localiseur	Injection directe essence Injecteur pompe diesel Moteur hybride Turbo géométrie variable Contrôle numérique moteur Multiplexage Suspension active Alterno-démarreur Transmission continue	Bureau embarqué Salon embarqué

Economie du semi-conducteur

- **Un poids croissant dans l'économie**
- **Rapport de coût exponentiellement décroissant depuis 40 ans**
  - *1 Mbit de mémoire en 1973 : 75000 €*
  - *1 Mbit de mémoire aujourd'hui : 0,05€*
- **Extension prévisible**
  - *Marchés émergents (Asie)*
  - *Technologie pervasive*
    - 1/20éme de la valeur d'un téléviseur en 1960, 1/6éme aujourd'hui
    - 15% de la valeur d'un PC en 1990, 40% aujourd'hui

Croissance depuis 1970

PNB Mondial	Electronique	Semi-conducteurs
2,7 %	10 %	66 %

Chiffre d'affaire mondial

1965	1985	1995	2000	2005
1,5 G$	25 G$	144 G$	200 G$	300 G$

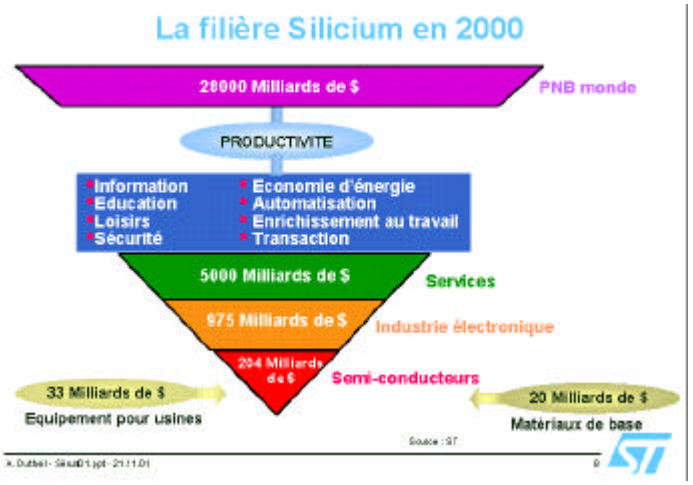
Effet de levier

Achats SC	CA ventes mobiles	CA opérateurs
30 G$	200 G$	500 G$

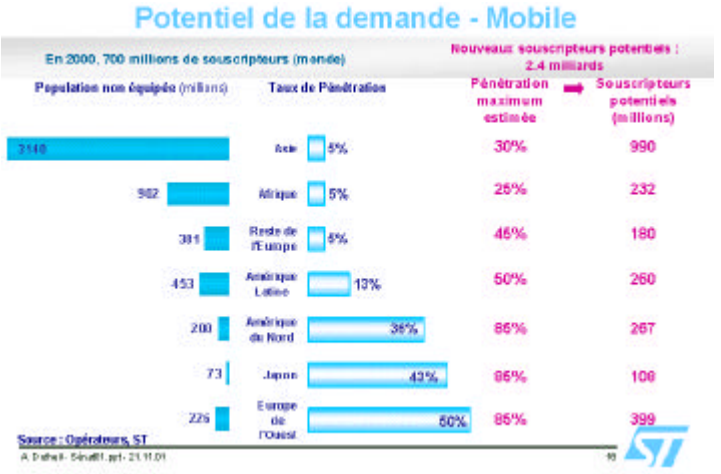
Consommation annuelle (transistors par habitant)

2002	2005	2010
50 M	200 M	1 G

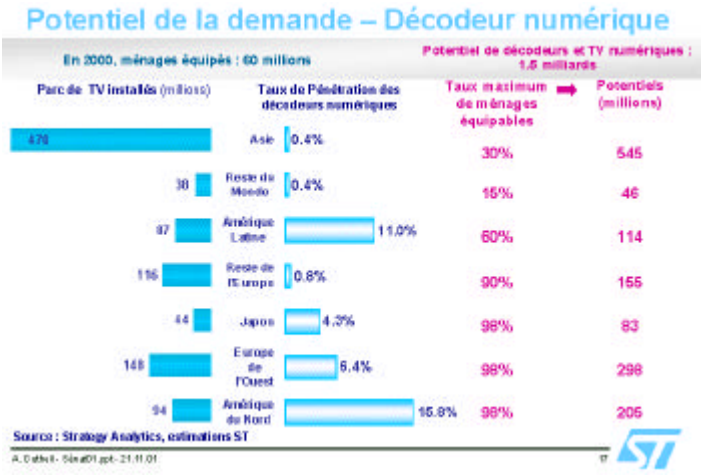
L'effet de levier



Perspectives d'évolution



Perspectives d'évolution



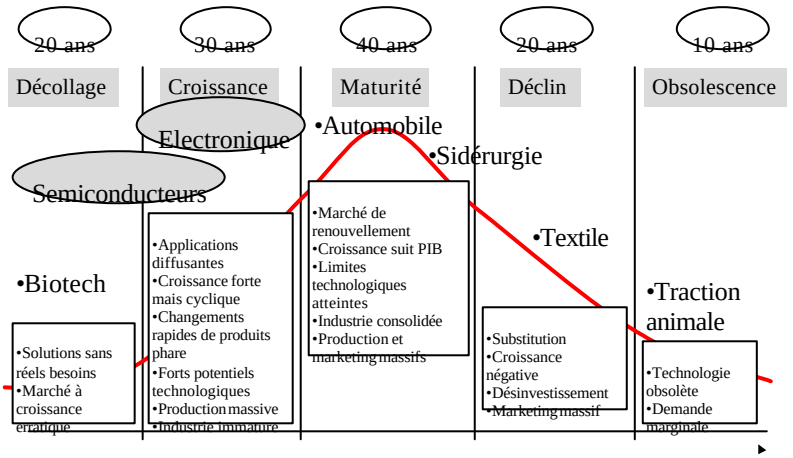
Les points « durs »

- **Croissance annuelle moyenne : 15% depuis 30 ans**
- **Accélération actuelle : une génération tous les 2-3 ans au lieu de 3-4 ans**
- **Coûts de R&D exponentiels : 15% du chiffre d'affaire**
- **Coûts des masques : 500 à 1000 k-euros**
- **Coût des unités de production :**
  - ***1,5 G-euros en 2000 pour 200 mm***
  - ***2,5 G-euros en 2003 pour 300 mm***
  - ***6 G-euros en 2010 pour 450 mm (4 centrales nucléaires)***

Les points « durs »

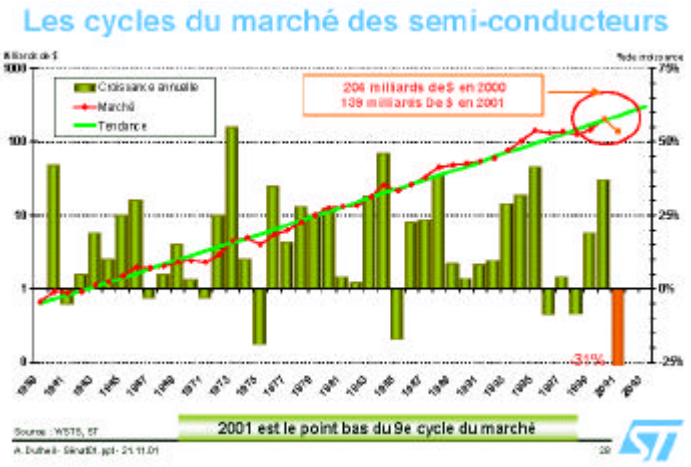
- **CAO**
  - ***Les progrès des outils sont très en deçà des progrès technologiques***
  - ***La productivité des concepteurs n'augmente pas assez rapidement***
  - ***Les erreurs de conception sont plus nombreuses***
  - ***On réutilise des éléments anciens***
- **Formation**
  - ***Spécialisation due à la complexité***
  - ***Déficit d'intégrateurs interdisciplinaires***

Le marché des semi-conducteurs



Source : STMicroelectronics 1998

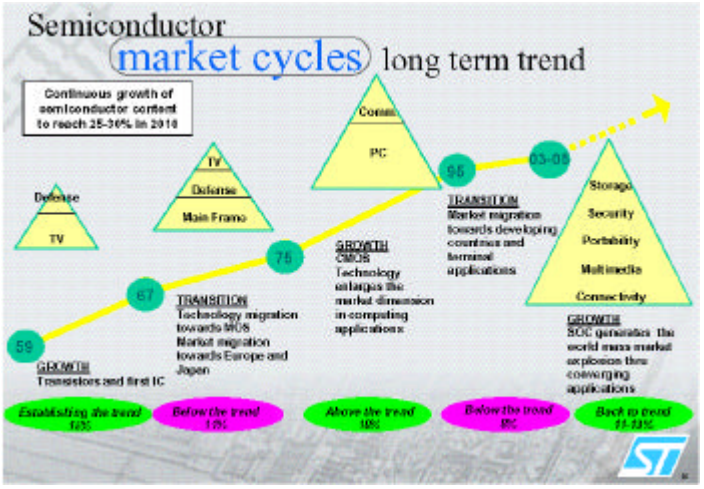
Les cycles du marché des semi-conducteurs



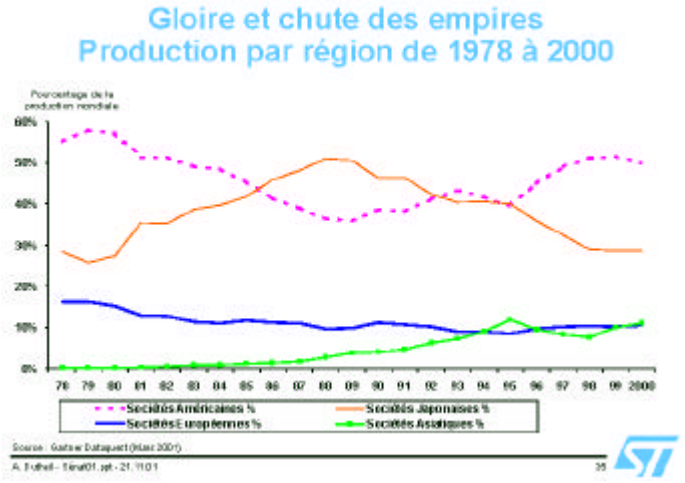
Les cycles du marché des semi-conducteurs

- **Les variations sont corrélées :**
  - ***Avec les tendances lourdes de l 'économie mondiale (Choc pétrolier, stagnations, reprises)***
  - ***Avec l'arrivée de nouveaux produits***
    - **Audio-vidéo (années 60)**
    - **PC**
  - ***Avec l'arrivée de nouveaux usages***
    - **Internet**
    - **Téléphonie mobile**
  - ***Avec la saturation de ces marchés***
- **La crise actuelle est une combinaison**
  - ***De la saturation de certains marchés des télécommunications***
  - ***De la situation de l'économie mondiale***

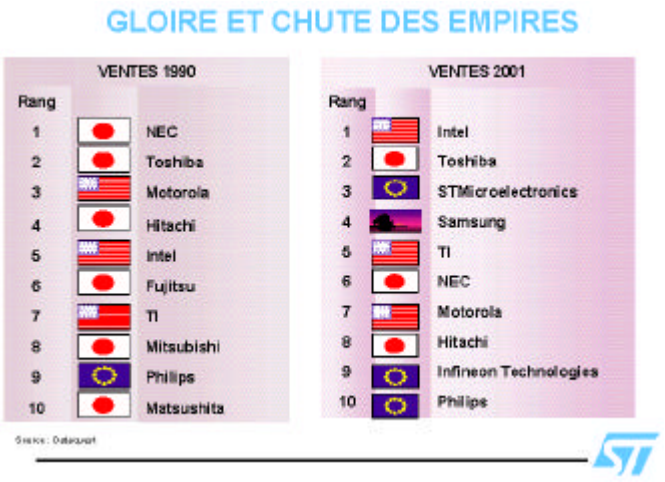
Les cycles du marché des semi-conducteurs



Répartition mondiale



Répartition mondiale



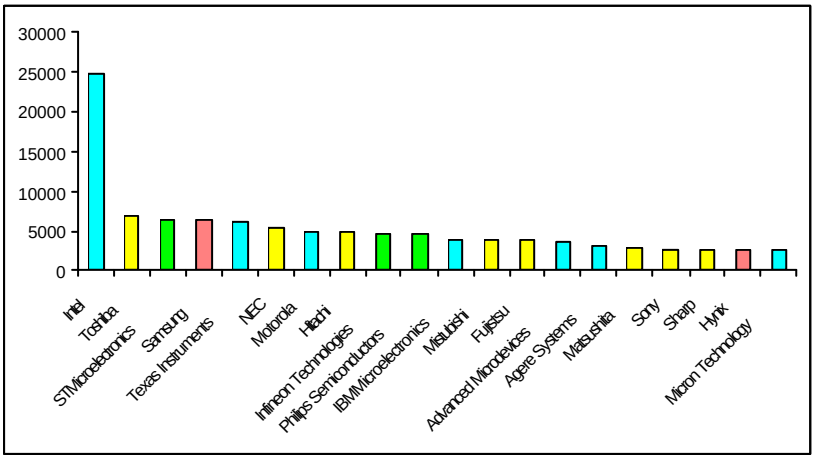
Les grands du semi-conducteur

Classement et revenus des 20 principales entreprises de semi-conducteurs

Origine	Classement	Entreprise	Ventes 2001	Parts de marché 2001 (%)
US	1	Intel	24927	16,10
JP	2	Toshiba	6783	4,38
EU	3	STMicroelectronics	6360	4,11
AS	4	Samsung	6303	4,07
US	5	Texas Instruments	6060	3,91
JP	6	NEC	5389	3,48
US	7	Motorola	4828	3,12
JP	8	Hitachi	4724	3,05
EU	9	Infineon Technologies	4512	2,91
EU	10	Philips Semiconductors	4402	2,84
US	11	IBMMicroelectronics	3892	2,51
JP	12	Mistubishi	3876	2,50
JP	13	Fujistsu	3786	2,45
US	14	Advanced Microdevices	3701	2,39
US	15	Agere Systems	2975	1,92
JP	16	Matsushita	2804	1,81
JP	17	Sony	2570	1,66
JP	18	Sharp	2519	1,63
AS	19	Hynix	2426	1,57
US	20	Micron Technology	2410	1,56

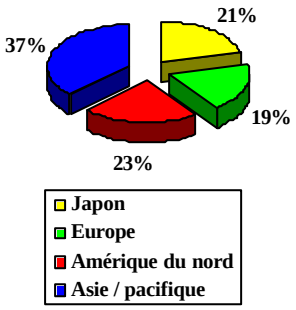
Source : Gartner Dataquest 2002

Les grands du semi-conducteur

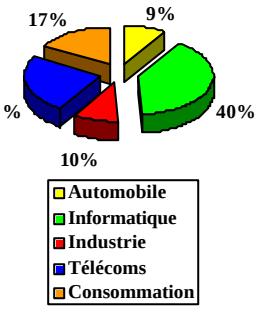


La demande mondiale 2001

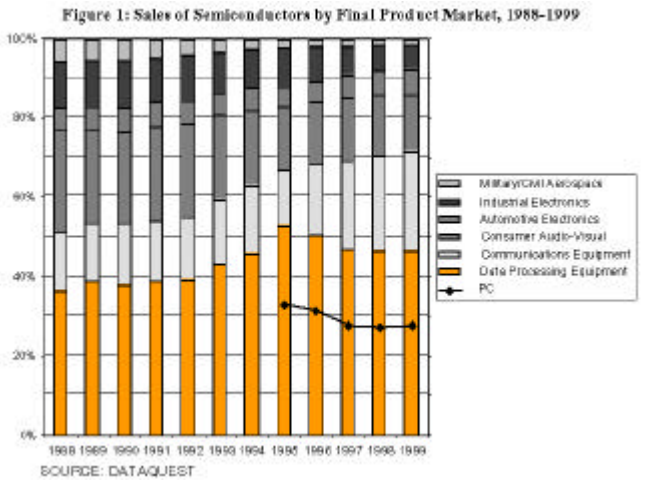
Par région



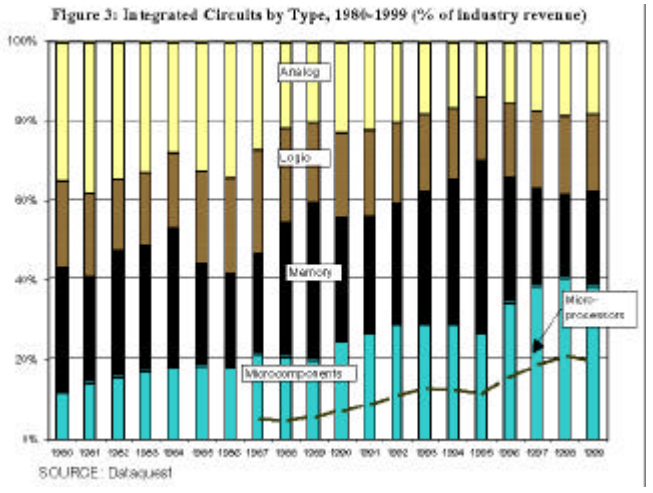
Par application



Ventes de semi-conducteurs par segments de marché, 1988-1999



Circuits intégrés par type, 1980-1999 (en % du revenu total)



Exemple de marché : réseau

	BUSINESS NETWORKING	NETWORK INFRASTRUCTURE	CONSUMER NETWORKING	ALL ELECTRONICS
REPRESENTATIVE PRODUCTS	PC, server, network interface cards, 3G cell phone, Workstation	router, switch, cellular basestation	PC, 3G cell phone, set-top box, Internet appliance	Networked products plus non-networked goods such as faxes, TV sets, automotive
SHARE OF CHIP MARKET REVENUE IN 1999	30%	3%	8%	100%
FORECAST CAGR* TO 2004	11%	9%	17%	14%

*CAGR: compound annual growth rate
SOURCE: calculated from Dataquest reports issued in Spring 2000

Exemple de marché : réseau

System Type	1999 IC Revenue ($ Million)	ICs As % Of System Wholesale Price	As % Of All IC Revenues	IC Revenue CAGR* forecast to 2003
PCs	49 079,00	38,00	28,36	15,00
Entry-Level Servers	3 348,00	41,00	1,93	17,00
Workstations	3 275,00	37,00	1,89	6,00
Midrange Computers	3 028,00	18,00	1,75	11,00
Central Office Telecom Equipment	2 105,00	7,00	1,22	6,00
LAN Switch	1 518,00	15,00	0,88	10,00
Digital WAN	1 175,00	13,00	0,68	11,00
Analog Modems	1 125,00	24,00	0,65	-9,00
Mainframe, Supercomputers	1 076,00	7,00	0,62	-4,00
Routers	852,00	12,00	0,49	11,00
LAN Cards	850,00	20,00	0,49	11,00
Fiber Optic Transmission Equipment	679,00	10,00	0,39	10,00
Remote Access Equipment	514,00	19,00	0,30	23,00
LAN Hub	465,00	24,00	0,27	-10,00
Handheld Computers	447,00	25,00	0,26	36,00
ISDN Terminal Adapters	307,00	26,00	0,18	1,00
Broadband Local Loop Systems	203,00	17,00	0,12	52,00
DSL Premises Equipment	134,00	26,00	0,08	38,00
Cable Modem Premises Equipment	92,00	21,00	0,05	11,00
Internet Audio Players	56,00	33,00	0,03	57,00
Internet-Enabled Set-Top Boxes	45,00	41,00	0,03	76,00
Internet-Enabled Wireless Infrastructure	24,00	9,00	0,01	195,00
ALL NETWORKED SYSTEMS	70 398,00	27,00	40,68	15,00
ALL SYSTEMS	173 027,00	17,00	100,00	17,00

*CAGR: compound annual growth rate
SOURCE: calculated from Dataquest reports issued in Spring 2000

Plan

- La brique DESSIN
- L'industrie du composant
- Perspectives d'évolution
- Les SoC (System on Chip)
- Les métiers

Perspectives

- Les prédictions sont difficiles
- Les nouvelles technologies arrivent tous les 2 ans (au lieu de 3 d'après la loi de Moore)
- La fréquence des microprocesseurs double tous les 4 ans
- La capacité de mémorisation est multipliée par 10 tous les 5 ans
- Un circuit actuel contient des centaines de millions de transistors
- Ses interconnexions ont une complexité extraordinaire

Perspectives

- **Le cuivre va remplacer l'aluminium**
- **D'autres matériaux, utilisés depuis plus de 30 ans, vont changer dans les prochaines années**
- **Depuis ses débuts l'industrie du SC est tirée par un segment de marché fort**
  - ***1947-1970 : applications militaires***
  - ***1970-1980 : gros ordinateurs***
  - ***1980-1985 : jeux vidéo et magnétoscopes***
  - ***1985-1999 : PC, télévision numérique***
  - ***1999-2001 : télécommunications (sans fil, large bande, optique)***
  - ***Et après ? Peut-être les données (modem câble, DSL, réseaux optiques, boucle locale, etc)***

Les secteurs à surveiller

- **Microsystèmes**
- **Têtes de lecture de disques durs**
- **Billettique sans contact**
- **Nouveaux matériaux pour les batteries pour l'électronique portable (lithium-ion polymère, piles à combustible)**
- **Technologies émergentes d'écrans plats (plasma, micro-pointes, électroluminescence)**
- **Nouveaux substrats (silicium sur isolant, carbure de silicium)**

L'évolution des CI depuis 1970

Grandeurs	1970	2000
Inflation en Francs	1	5,5
Prix de 10^6 transistors	10 000 $	1 $
Marché (y compris discrets)	2 G$	226 G$
Taille d'un MPU	12 mm^2	200 mm^2
Prix d'un MPU	125 $	200 $
Taille de gravure	0,01 mm	0,15 µm
Nombre de couches	1	8
Transistors/puce	2 300	64 000 000
Fréquence	200 kHz	1 GHz
Coût d'une usine	10 M$	1 500 M$
Diamètre de wafer	1 pouce	12 pouces
Nombre d'opérations de fabrication	20	800

Vers les nanotechnologies

Perspectives d'évolution des circuits intégrés numériques

Année	2001	2002	2003	2004	2005	2006	2007	2010	2013	2016
Wafer (mm)	300	300	300	300	300	300	300	300	450	450
M Tr. / cm²	39	48,6	61,2	77,2	97,2	122,5	154,3	309	617	1235
Nombre max métaux	10	10	12	13	14	14	14	14	15	15
Taille max (mm²)	280	280	280	280	280	280	280	280	280	280
Alimentation (V)	1,1	1	1	1	0,9	0,9	0,7	0,6	0,5	0,4
Fréquence (MHz)	1684	2317	3088	3990	5173	5631	6739	11511	19348	28751
Consommation max (W)	61	75	81	85	92	98	104	120	138	158
Consommation max portable (W)	2,4	2,6	2,8	3,2	3,2	3,5	3,5	3	3	3
Gravure (nm)	90	75	65	53	45	40	35	25	18	13
Capacité DRAM (Gbits)	0,54	0,54	1,07	1,07	2,15	2,15	4,29	8,59	34,36	68,72
Taille max DRAM (mm²)	127	100	118	93	147	116	183	181	239	238
1/2 pitch DRAM (nm)	130	115	100	90	80	70	65	45	32	22

Source : SIA 2002

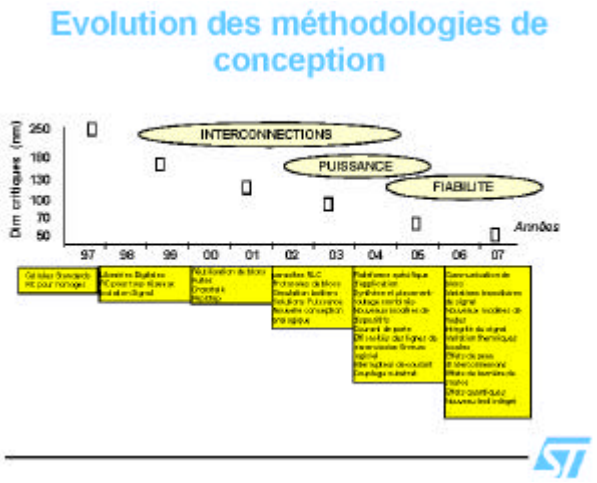
Perspectives d'évolution des circuits intégrés numériques

Interconnexions : Cu Low K et...

Generation Technologique	1999 180nm	2001 130nm	2003 100nm	2005 70nm	2008 50nm	2011 35nm
Nombre de niveaux métal—DRAM	3	3–4	4	4	4	4
Nombre de niveaux métal - logique	6–7	6–7	10-11	12	13-14	14
Constante diélectrique matériau inter métal (k)	3.6	2.7	2	1.9	<1.9	1.6
Constante diélectrique effective (k)	3.6-4.0	2.8-3.6	2.2-2.9	1.8	<1.8	<1.5
Epaisseur de barrière (nm)	20	13	12	0	0	0
Résistivité Effective des conducteur (µW.cm)	2.2	2.2	2.2	2.2	<1.8	<1.8
Densité de courant - (A/cm²)	5.8E6	9.0E6	1.4E6	1.8E6	3.7E6	4.6E6
Fiabilité — (FIT/via/meter) x 10^2	0.46	0.27	0.16	0.1	0.06	0.03

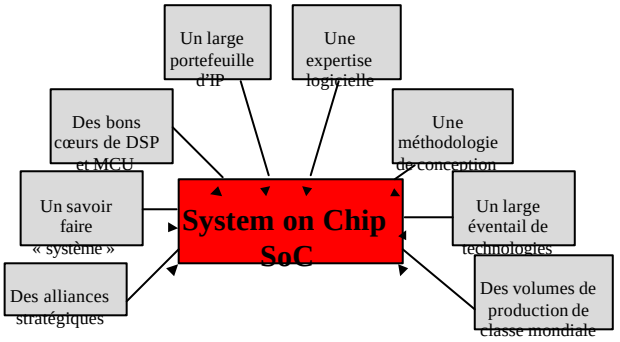
Source: ITRS Roadmap

Les perspectives de la CAO



Perspectives

- **Les évolutions technologiques ont permis l'intégration des composants de base, des fonctions, des sous-systèmes, …**
- **Les industriels qui veulent accéder au niveau système ont besoin de :**



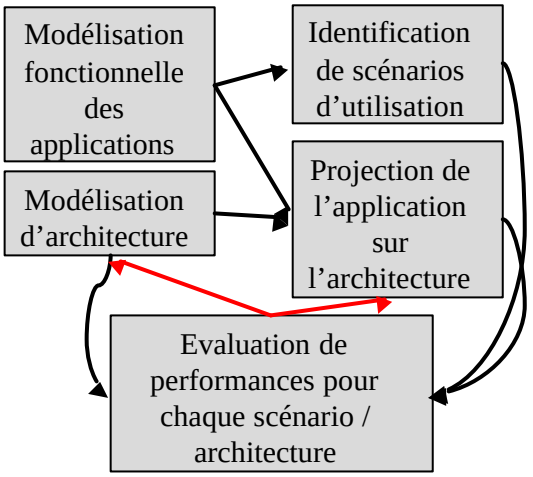
Plan

- La brique DESSIN
- L'industrie du composant
- Perspectives d'évolution
- Les SoC (System on Chip)
- Les métiers

Qu'est-ce qu'un SOC ?

- "System" : une collection d'éléments en interaction.
- "On Chip" : sur un seul circuit.
- Mais tous les circuits intégrés ne sont pas des SOC :
  - *Un SOC implémente une fonction "complète"*
    - Décodeur MPEG2 vidéo + audio + système + transport + graphique + interface utilisateur
    - Terminal GSM : tout sauf la RF
  - *Les éléments constitutifs d'un SOC sont "complexes", réutilisables et de nature variée :*
    - Fonctions analogiques (convertisseurs A/N, filtres, etc.)
    - Fonctions numériques câblées (décodeur de Viterbi)
    - Fonctions logicielles (sur micro-contrôleur, DSP, RISC)
    - Composants de base (mémoires)

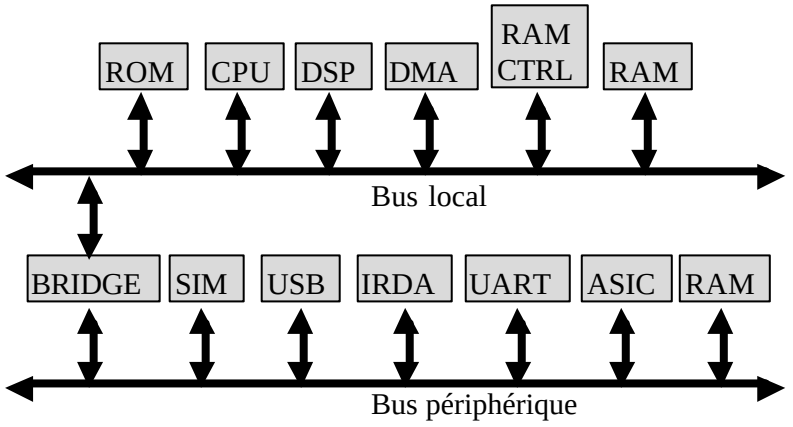
SoC : exploration d'architecture



Exemple d'application

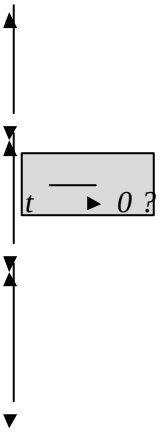
- **Téléphonie mobile de troisième génération**
- **Traitement en bande de base et couches applicatives**
- **Télécommunications (multi-standard)**
- **Applications**
  - ***Voix***
  - ***Vidéo***
  - ***Graphique 2D / 3D (animation faciale)***
  - ***Données (WEB, MP3, MPEG4, etc.)***
  - ***Agenda, carnet d'adresses (fonctions PDA)***
  - ***Jeux, …***

Exemple d'architecture



Exemple de scénario

- **Situation 1 :**
  - ***W-CDMA***
  - ***Lecture vidéo MPEG4***
- **Situation 2 :**
  - ***W-CDMA***
  - ***Visiophonie***
- **Situation 3 :**
  - ***Perte de couverture W-CDMA***
  - ***Passage en GSM***
  - ***Animation labiale***



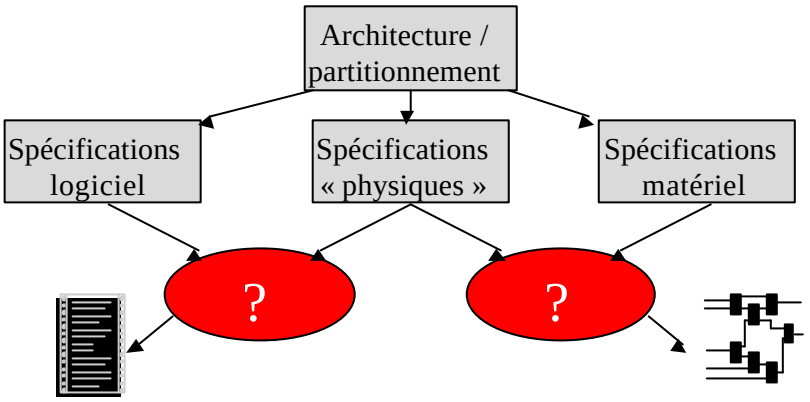
Exploration d'architecture

- **Partitionnement logiciel / matériel,**
- **Choix des cœurs de micro-processeur, des puissances de calcul et des taux de parallélisme,**
- **Spécification des modules logiciels et matériels,**
- **Choix des architectures d'interconnexion,**
- **Dimensionnement des ressources partagées (mémoires, bus), des caches,**
- **Détection des inter blocages,**
- **Vérification des contraintes de temps réel, …**

Exploration d'architecture

- **Simulation**
  - ***Sur des modèles précis au cycle près***
    - **100 % fonctionnelle**
    - **Temps de simulation importants, couverture réduite**
    - **Développements préalables longs et coûteux**
  - ***Sur des modèles approchés***
    - **Non fonctionnelle**
    - **Temps de simulation réduits, couverture augmentée**
    - **Délais de mise en œuvre raccourcis**

SoC : raffinement



SoC : réutilisation

- **Utilisation des composants virtuels (IP)**
  - ***Environnements variables***
  - ***Fortement génériques***
  - ***Origines variées***
  - ***Impacts sur la stratégie de vérification***
- **Conception d'IP**
- **Maintenance d'IP**

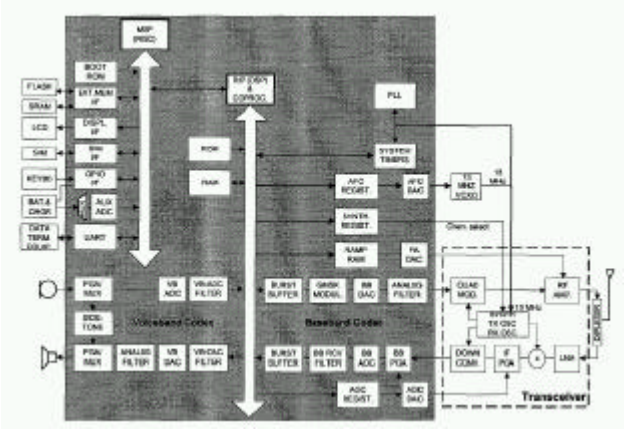
SoC : les contraintes

- **Augmentation de la complexité**
- **Augmentation des coûts fixes**
- **Réduction des temps de développement**
- **Performances des outils insuffisantes**

Il faut concevoir très vite des systèmes complexes, sans erreurs et avec des outils insuffisants

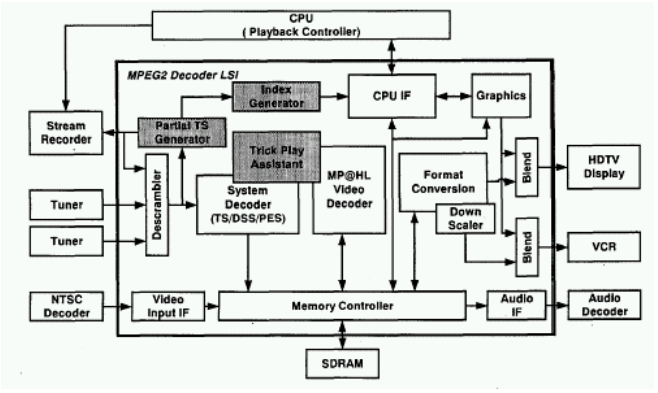
Exemple de SoC

- **GSM**



Exemple de SoC

- Multimédia



Les SoC, qu'est-ce qui change ?

- Connaissance système indispensable
- Processeurs embarqués (« Co-design »)
- Mémoires embarquées (les bandes passantes augmentent)
- Réutilisation, approche « composants virtuels » (IP)
- Standards de bus (VCI, Sonics, SRS, AMBA, …)
- Circuits mixtes
- Basse consommation
- Testabilité
- Complexité

Plan

- La brique DESSIN
- L'industrie du composant
- Perspectives d'évolution
- Les SoC (System on Chip)
- Les métiers

Les métiers

- Différents « matériaux »
  - *Numérique*
  - *Analogique*
  - *Logiciel*
  - *Matériel*
  - *Equipements*
  - *CAO*
  - *Argent*
  - *Hommes, …*
- Différentes structures
  - *Fondeurs*
  - *Sansfabs*
  - *Sanschips*
  - *Equipementiers, …*

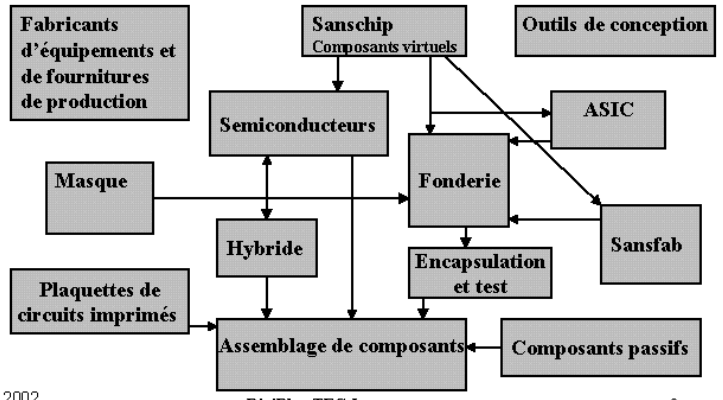
Différents contextes
- Système
- Composant, …

Différents aspects
- Conception
- Validation
- Fabrication
- Test, …

Différents niveaux
- Application
- Fonction
- Logique
- Physique, …

Les métiers

Structure de la grappe microélectronique:



Juillet 2002

DigiPlan TEC Inc.

9

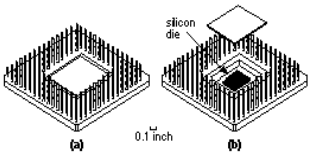
Structure du flot de conception ASIC

Les grandes classes de flots, évaluation de complexité, de risques, de coûts.

Plan

- Les ASIC, types, économie
- Le flot de conception ASIC
  - *Spécification, modélisation*
  - *Le modèle synthétisable*
  - *Synthèse, optimisation*
  - *Test*
  - *Placement / routage*
- Les méthodes de validation
  - *Simulation*
  - *Preuve*
- Les outils
- Conclusion

Les circuits ASIC

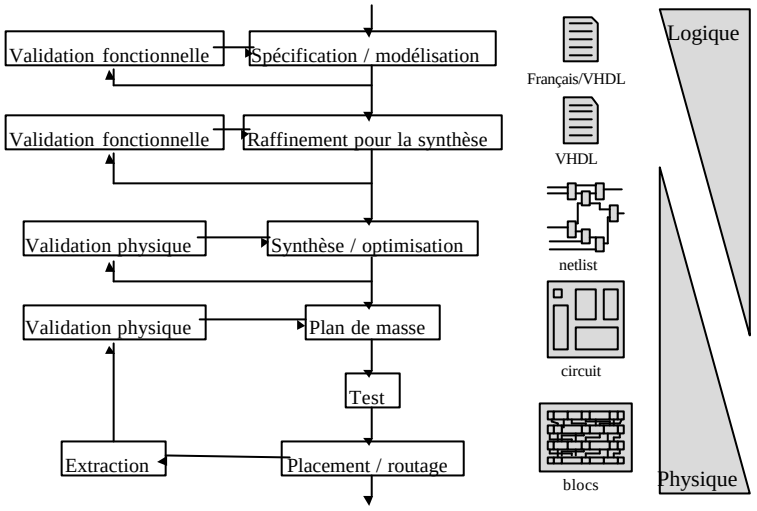


- Les boîtiers sont généralement en céramique ou en plastique et du type PGA ou BGA
- La surface varie de quelques mm2 à quelques cm2 ; on la mesure aussi en transistors ou en portes NAND2
- On caractérise le procédé par la longueur minimale de canal (0.18 m) ou le « lambda » (0.09 m)
- 1970 : bipolaire, une dizaine de portes
- 2000 : CMOS, plusieurs millions de portes

Les types de circuits ASIC

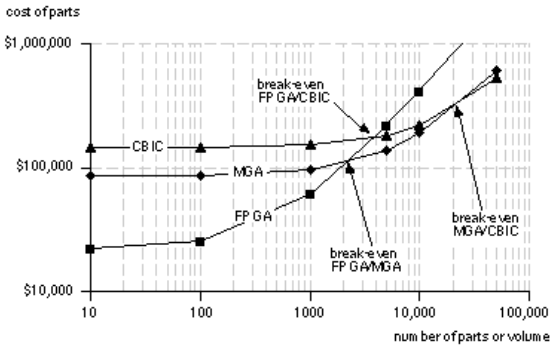
- Full custom
- Standard cells
- Gate array
  - *Avec canaux de routage*
  - *Sans canaux de routage*
  - *Structurés*
- Circuits logiques programmables
  - *EPLD*
  - *FPGA*
  - *Structurés*

Le flot de conception ASIC



L'économie des ASIC

Type	Coût fixe	Coût par unité
FPGA	21800	39
Gate Array	86000	10
Standard Cell	146000	8



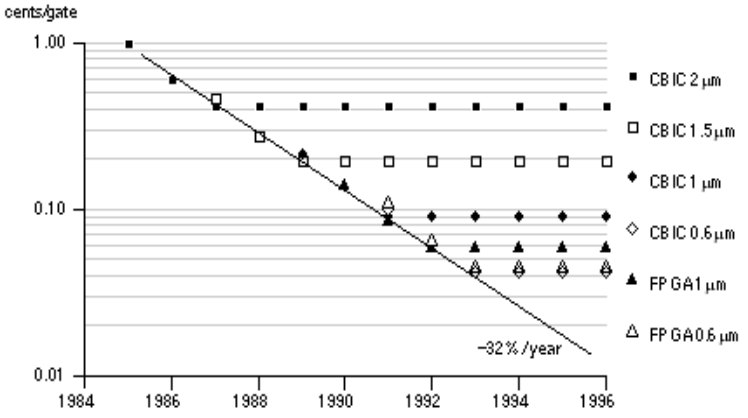
L'économie des ASIC, coût fixe

	FPGA		MGA		CBIC	
Training:	$800		$2,000		$2,000	
Days		2		5		5
Cost/day		$400		$400		$400
Hardware	$10,000		$10,000		$10,000	
Software	$1,000		$20,000		$40,000	
Design:	$8,000		$20,000		$20,000	
Size (gates)		10,000		10,000		10,000
Gates/day		500		200		200
Days		20		50		50
Cost/day		$400		$400		$400
Design for test:			$2,000		$2,000	
Days				5		5
Cost/day				$400		$400
NRE:			$30,000		$70,000	
Masks				$10,000		$50,000
Simulation				$10,000		$10,000
Test program				$10,000		$10,000
Second source:	$2,000		$2,000		$2,000	
Days		5		5		5
Cost/day		$400		$400		$400
Total fixed costs	$21,800		$86,000		$146,000	

L'économie des ASIC, coût unitaire

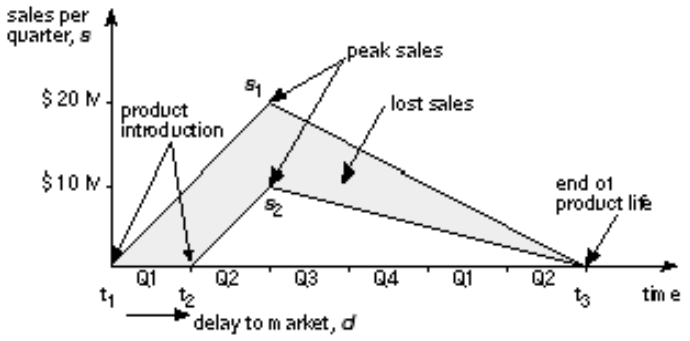
	FPGA	MGA	CBIC	Units
Wafer size	6	6	6	inches
Wafer cost	1,400	1,300	1,500	$
Design	10,000	10,000	10,000	gates
Density	10,000	20,000	25,000	gates/sq.cm
Utilization	60	85	100	%
Die size	1.67	0.59	0.40	sq.cm
Die/wafer	88	248	365	
Defect density	1.10	0.90	1.00	defects/sq.cm
Yield	65	72	80	%
Die cost	25	7	5	$
Profit margin	60	45	50	%
Price/gate	0.39	0.10	0.08	cents
Part cost	$39	$10	$8	

L'économie des ASIC, évolution



L'économie des ASIC, modèle de profit

- **s1 = $60,000,000**
- **s2 = $25,000,000**



Les bibliothèques de cellules standard

- **Un élément clé du flot**
- **Un modèle de cellule comporte**
  - ***Un dessin au micron***
  - ***Un modèle fonctionnel***
  - ***Un modèle VHDL/Verilog***
  - ***Un modèle de temps de propagation***
  - ***Une stratégie de test***
  - ***Un schéma électrique***
  - ***Une icône schématique***

$Z = \overline{I}$

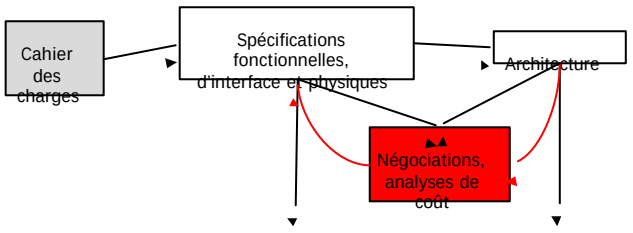
```
module IN01D1(I, Z);
  input I;
  output Z;

  assign Z = ! I;
endmodule
```

$tp_{lh} = tp0_{lh} + D tp_{lh} * Cl$

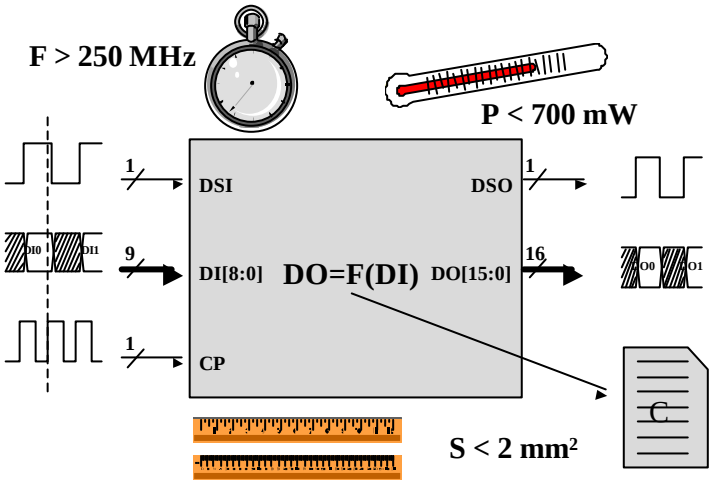
- **Toutes ces représentations ne sont pas fournies**
- **Le choix d'un fondeur (technologie et bibliothèque de cellules) est déterminant**

Le flot de conception ASIC

- **Spécification, architecture**
  - ***Fonctionnelle***
  - ***Interface***
  - ***Physique***
  - ***Validées par analyse de coûts***
    - **Négociations client**
    - **Négociations fournisseurs (CAO, fonderie, services internes)**

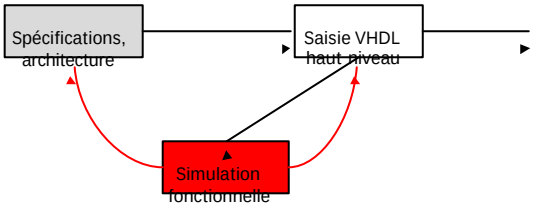


Spécification : fonctionnalité, interface, physique



Le flot de conception ASIC

- **Modélisation haut niveau**
  - ***Produit un modèle de référence précis au bit près et au cycle près***
  - ***Maintenu à jour tout au long du flot***
  - ***Validé par simulations***

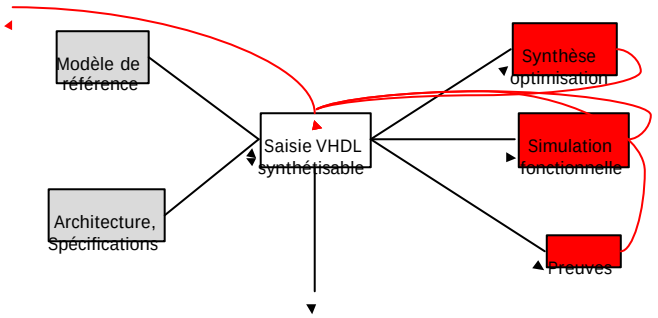


Le modèle de référence

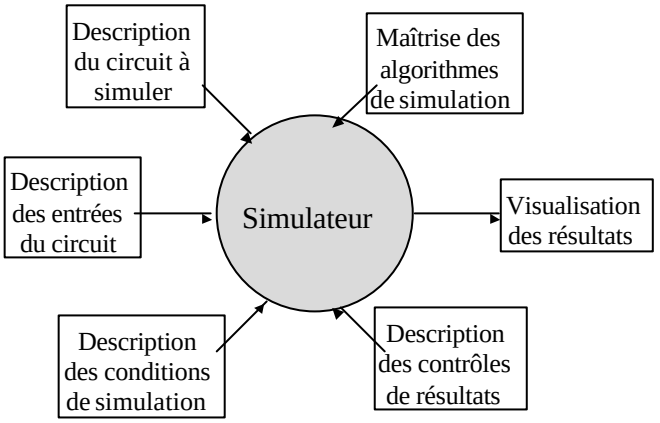
- **Référence tout au long du projet**
- **Précis au bit près et au cycle près**
- **Remis à jour lors des évolutions**
- **Rapide en simulation**
- **Proche du matériel**
  - ***Hiérarchie compatible***
  - ***Flots de données compatibles***
- **Loin du matériel**
  - ***Bonne couverture des erreurs***
- **Validé par simulations fonctionnelles**
  - ***Comparaisons avec un modèle algorithmique***

Le flot de conception ASIC

- **Le raffinement pour la synthèse**
  - ***Produit un modèle synthétisable***
  - ***Validé par synthèse, simulation et preuve***



Le simulateur



La simulation

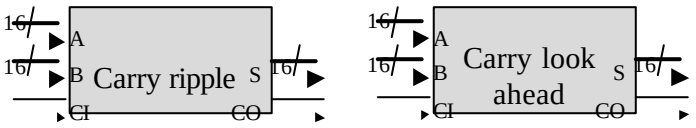
- **Induit des développements supplémentaires**
  - ***Modèle de référence***
  - ***Environnement de simulation***
- **Une simulation n'est jamais exhaustive**
  - ***On peut prouver un dysfonctionnement***
  - ***On ne peut pas prouver le bon fonctionnement***
- **On peut constater des comportements différents avant et après synthèse**
- **Souvent coûteuse en temps de calcul**

La preuve

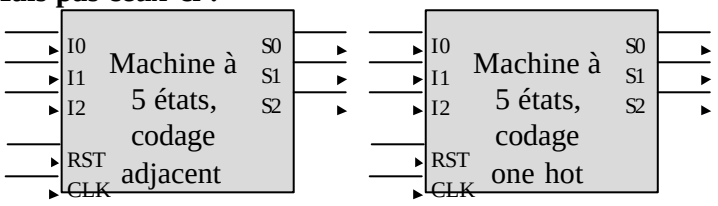
- **Ne s'applique (pour l'instant) qu'aux modèles synthétisables**
  - ***Trois classes d'outils :***
    - **Equivalence structurelle (equivalence checking)**
      - ***mêmes registres aux mêmes endroits***
      - ***rapide***
      - ***gros circuits (millions de portes)***
    - **Equivalence séquentielle**
      - ***circuits fonctionnellement équivalents***
      - ***registres différents***
      - ***lent et gourmand en mémoire***
      - ***petits circuits (centaines de registres)***
    - **Preuve de modèle (model checking)**
      - ***comparaison entre circuit et propriétés logico-temporelles***
      - ***lent et gourmand en mémoire***
      - ***petits circuits (centaines de registres)***

La preuve

- **Ces deux circuits sont structurellement équivalents :**

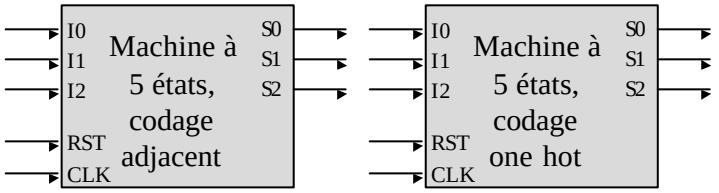


- **Mais pas ceux-ci :**



La preuve

- **Ces deux circuits sont séquentiellement équivalents :**

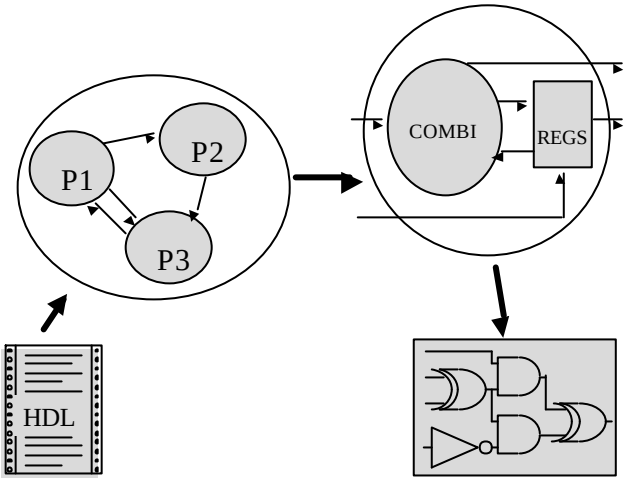


- **Exemples de propriétés logico-temporelles :**
  - *les deux feux ne sont jamais verts simultanément (sécurité)*
    `AG(!(f1_vert * f2_vert))`
  - *si un client demande le bus il finit par l'obtenir (vivacité)*
    `AG(c1_request -> AF(c1_acknowledge))`
  - *un client maintient sa requête jusqu'à obtention du bus (vivacité)*
    `AG(c1_request -> AX(c1_request + c1_acknowledge))`

Le modèle synthétisable

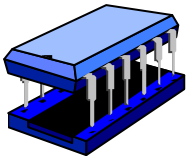
- **La synthèse est purement fonctionnelle**
- **Le synthétiseur n'accepte que du VHDL**
- **Le synthétiseur n'accepte pas tout VHDL**
- **Le style d'écriture est important**
- **Les options de synthèse ne sont pas toujours du VHDL**
  - ***Commentaires interprétés***
  - ***Directives externes***
- **Des paquetages sont fournis**
- **Les performances en simulation sont critiques**
- **Validation conjointe fonction / synthétisabilité**
- **Les validations fonctionnelles peuvent utiliser la preuve (Model checking, Equivalence checking)**

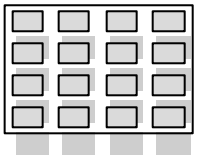
La synthèse

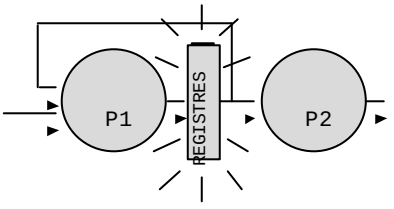


La synthèse

- **Le VHDL synthétisable :**
  - ***Purement matériel***
    - **Pas de pointeurs**
    - **Pas de fichiers**
    - **Des types réalistes (pas de réels)**
    - **Pas de clause `after`**
  - ***Ressources identifiées, allouées, ordonnancées***
  - ***Complexité réduite (forte granularité)***







La synthèse

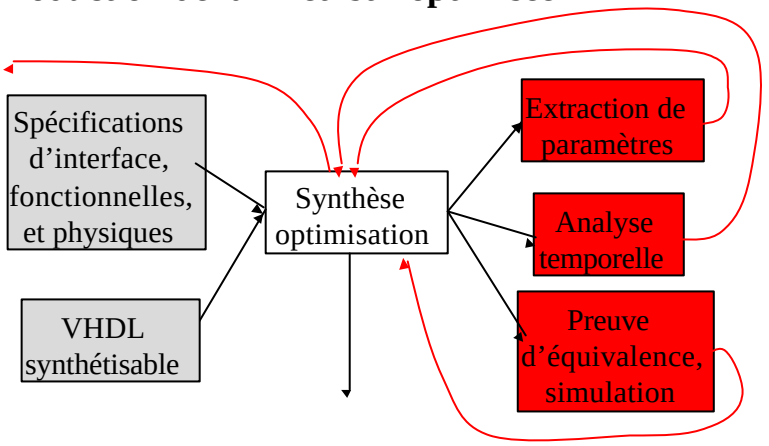
- **Les directives de synthèse**
  - ***soit commentaires interprétés***
  - ***soit attributs VHDL***
  - ***soit commandes du synthétiseur***
  - ***usages multiples***
    - `synthesis on/off`
    - `translate on/off`
    - `set` et `reset`
    - **architectures arithmétiques**
    - **codage de types énumérés**
    - **logique multiplexée ou câblée (pour les `case`)**
    - **codage et optimisations de machines à états**
    - **interprétation de fonctions de résolution**
    - **...**

La synthèse : validations

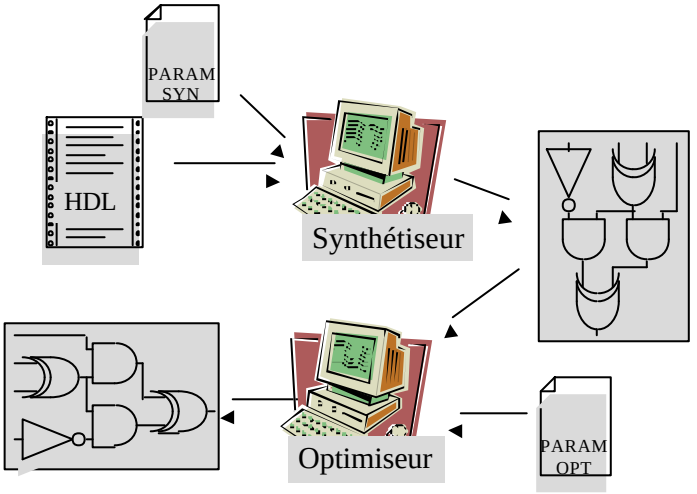
- **Etudier attentivement les messages :**
  - ***Erreurs de syntaxe***
  - ***Alertes aux "latches"***
  - ***Alertes aux listes de sensibilité incomplètes***
- **Compter les registres**
- **Estimer la complexité (additionneurs)**
- **Tenir compte du temps CPU de synthèse**

Le flot de conception ASIC

- **Production de la « netlist » optimisée**



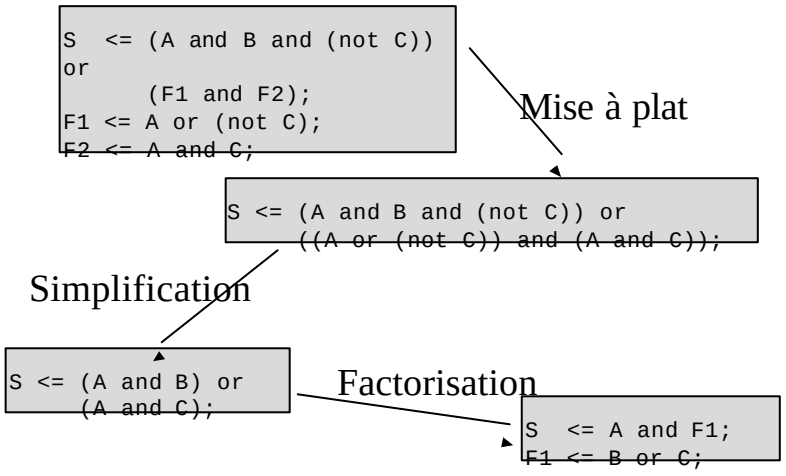
Synthèse - optimisation



Synthèse - optimisation

- **Les étapes**
  - ***analyse du code source***
    - **extraction des parties combinatoires**
    - **extraction des unités de mémorisation**
  - ***construction d'une représentation interne***
  - ***minimisation logique (Karnaugh)***
  - ***optimisation logique***
    - **factorisations**
    - **substitutions**
    - **éliminations**
    - **...**
  - ***construction d'une correspondance technologique indépendante***
  - ***optimisation de la correspondance technologique***

Synthèse - optimisation

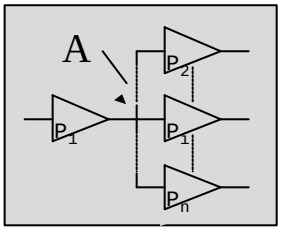


L'optimisation

- **Prise en compte des spécifications physique**
  - ***Surface***
  - ***Vitesse***
  - ***Environnement***
- **Long et coûteux en temps de calcul**
  - ***Pas d'optimisations inutiles***
  - ***Granularité fine***
- **Reproductibilité indispensable**
- **Prise en compte nécessaire des autres étapes du flot**
  - ***Test***
  - ***Plan de masse***
  - ***Placement / routage***

Les capacités prédites

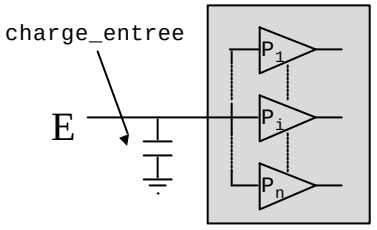
- **L'optimiseur utilise un modèle prédictif de capacités de routage à deux paramètres :**
  - ***defCap : capacité par défaut d'un nœud du réseau de portes***
  - ***incCap : capacité incrémentale par connecteur sur le nœud***
- **Deux modes :**
  - ***inclusion***
  - ***global***




```
cap(A) =
  capSortie(P1) +
  defCap +
  (n + 1) * incCap +
  capEntrée(P2) +
  ... +
  capEntrée(Pn)
```

Les capacités d'entrée

- **L 'optimiseur calcule les capacités des nœuds d'entrées à partir :**
  - ***du modèle prédictif de capacités de routage***
  - ***de la spécification des capacités externes des entrées***

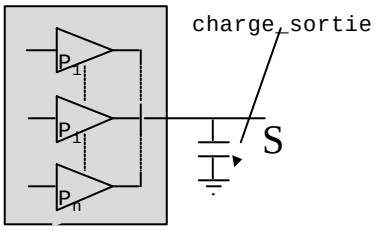



```
cap(E) =
  charge_entree +
  defCap +
  (n + 1) * incCap +
  capEntrée(P1) +
  ... +
  capEntrée(Pn)
```


Les capacités de sortie

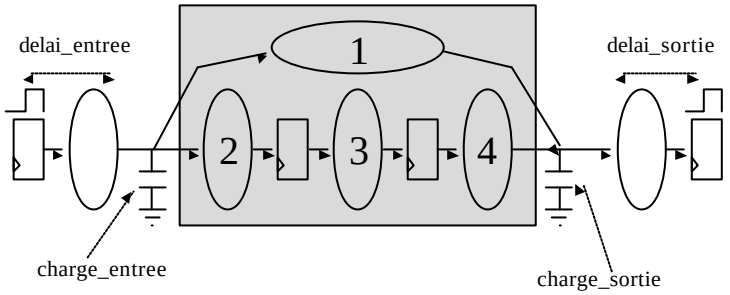
- **L 'optimiseur calcule les capacités des nœuds de sortie à partir :**
  - ***du modèle prédictif de capacités de routage***
  - ***de la spécification des capacités externes des sorties***

```
cap(S) =
  charge_sortie +
  defCap +
  (n + 1) * incCap +
  capSortie(P1) +
  ... +
capSortie(Pn)
```



Les contraintes temporelles

- **On spécifie :**
  - ***les horloges et leurs caractéristiques***
  - ***les chemins multi-cycles***
  - ***les dates d 'établissement souhaitées pour les sorties***
  - ***les délais externes***



Autres options

- **Pour chaque entrée l 'optimiseur doit connaitre les caractéristiques de la porte pilote :**
  - ***temps nécessaire à la porte pilotant l'entrée pour charger une capacité donnée***
- **D 'autres paramètres définissent :**
  - ***le temps de transition maximal toléré sur tout nœud du réseau de portes (entrées et sorties comprises)***
  - ***la capacité maximale autorisée en charge utile de chaque porte***
  - ***le nombre maximal de portes attaquées par un signal***
- **On peut modifier la hiérarchie**
- **On peut accorder plus ou moins de puissance de calcul aux différentes étapes**
- **...**

L'optimisation : validations

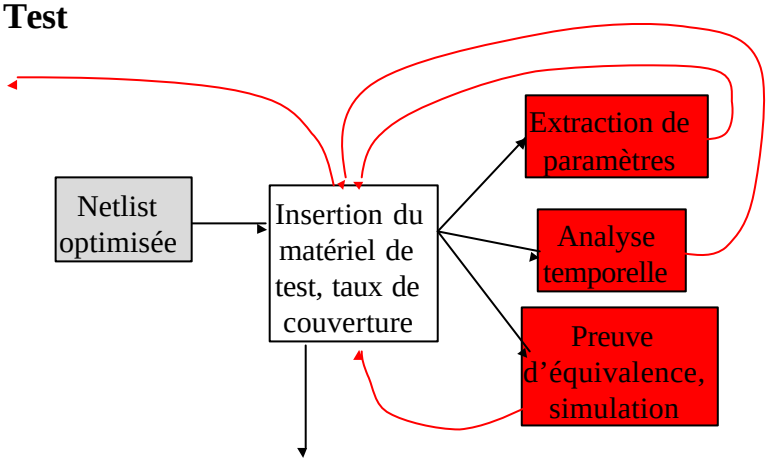
- **Etudier attentivement les messages :**
  - ***Syntaxe de la netlist***
  - ***Nombre de registres***
  - ***Complexité (additionneurs)***
  - ***Contraintes temporelles***
- **Tenir compte du temps CPU**
- **Analyse syntaxique**
- **Analyse temporelle**
- **Preuve d'équivalence**
- **Simulation logico-temporelle**
  - ***recherche des défauts d'initialisation***
  - ***recherche des violations de temps de setup et de hold***
  - ***accélérateur matériel ?***

L'analyse temporelle

- **Le calculateur de délais est le coeur de l'optimiseur**
- **Il utilise des modèles de temps de propagation variés**
  - ***Prop ramp delay***
  - ***Input slope***
  - ***Wave tabular***
- **Ses performances en vitesse de calcul sont critiques**
- **On l'utilise seul**
  - ***pour évaluer un assemblage non globalement optimisé***
  - ***pour évaluer l'impact de modifications***
    - **conditions d'utilisation**
    - **test**
    - **placement / routage**
- **Les options sont proches de celles de l'optimisation**

Le flot de conception ASIC

- **Test**



Le test

- **Il s 'agit d 'un test de fabrication et non d 'une validation fonctionnelle**
- **C 'est le concepteur qui fournit les séquences de test**
- **C 'est en général le fondeur qui teste**
  - ***sur le wafer***
  - ***après empaquetage***
- **On teste aussi les cartes**
- **Le test de fabrication est d 'une très grande importance économique**
- **Exercice : quelle est la probabilité de défaut d 'une carte**
  - ***comportant 10 ASIC de 1,5 cm²***
  - ***fabriqués avec une probabilité de défaut par cm² de 0,1***
  - ***et un taux de couverture de test de 97%***

Le test

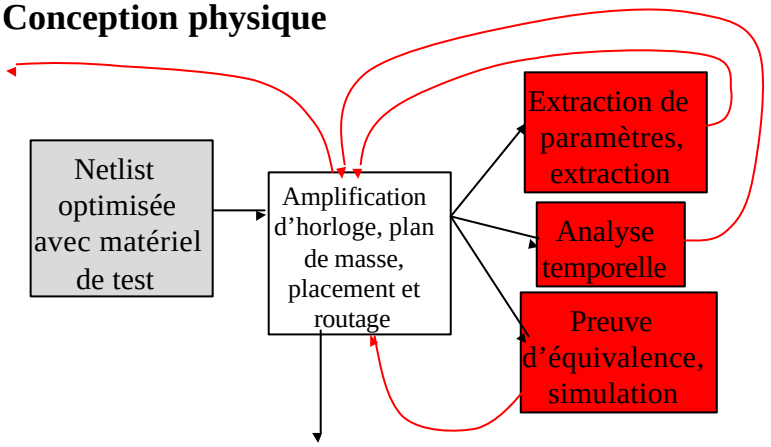
- **Le modèle de faute est souvent :**
  - ***une faute est un collage à 0 ou à 1 d 'une équipotentielle***
  - ***il n 'y a qu'une faute par circuit***
- **La simulation de fautes consiste à simuler le circuit en supposant qu 'il comporte l 'une des nombreuses fautes possibles ; la faute est dite « couverte » par la séquence de test si les sorties du circuit fautif ne se comportent pas comme celles d 'un circuit sans faute**
- **Un taux de couverture de 100% correspond à une séquence de test qui met en évidence toutes les fautes possibles**
- **Le taux de couverture dépend très fortement de la puissance de commandabilité et d 'observabilité**

Le test

- **Stratégies multiples :**
  - ***Principe du " Scan path "***
  - ***BIST***
  - ***Autotest***
  - **...**
- **Le test modifie les caractéristiques temporelles**
- **Augmente la surface**
- **Le taux de couverture est critique**
- **Il existe des interactions particulières avec le placement**
  - ***Connexion des chaînes de scan après placement***
- **Le test de cartes possède ses stratégies propres (Boundary Scan, norme JTAG) qui influent sur les circuits**

Le flot de conception ASIC

- **Conception physique**

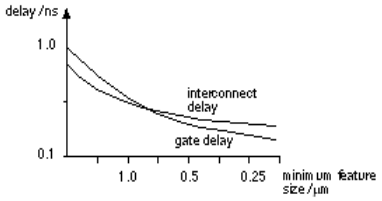


Placement / routage

- **Long et coûteux en temps de calcul**
- **Modifie les caractéristiques temporelles**
- **Dimensionnement des alimentations**
- **Impacts sur l'architecture**
  - ***Partitionnement***
  - ***Segmentation des bus***
- **Impacts sur la synthèse / optimisation**
  - ***Temps***
  - ***Surface***
- **Outils complexes**
- **De plus en plus souvent réalisé par le fondeur**

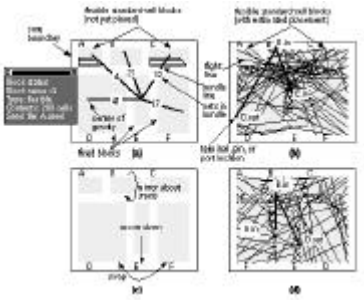
Placement / routage

- **Les buts de la réalisation d 'un plan de masse :**
  - ***définir l 'arrangement topologique des blocs sur le circuit***
  - ***décider de l 'emplacement des plots d 'entrée / sortie***
  - ***décider de l 'emplacement et du nombre des plots d 'alimentation***
  - ***décider de la stratégie de distribution de l 'alimentation***
  - ***décider du type et de l 'organisation de la distribution d'horloges***
  - ***identifier les délais d 'interconnexion critiques:***



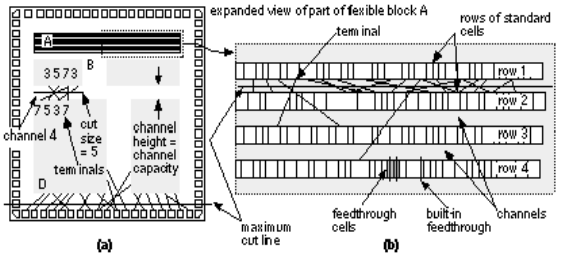
Placement / routage

- **Les outils de « floor - planning »**
  - ***aident au dimensionnement des blocs « mous »***
  - ***permettent d 'identifier les congestions ...***
  - ***... et de les résoudre***
  - ***aident à l 'estimation de qualité des stratégies d 'alimentation ...***
  - ***... et de distribution des horloges***



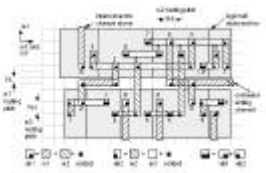
Placement / routage

- **Le placement**
  - ***consiste à organiser les cellules standard en rangées***
  - ***interagit avec la stratégie de test dite du « scan path »***
  - ***doit conduire à une congestion minimale***
  - ***utilise des algorithmes classiques d 'optimisation (recuit simulé)***



Placement / routage

- **Le routage se fait à deux niveaux :**
  - ***routage cellules dans les zones de cellules standard***
  - ***routage bloc, entre les blocs « durs »***
  - ***le routage bloc est en général plus délicat***
    - **les canaux de routage ont des formes plus variées**
    - **les alimentations posent problème**
  - ***doit souvent être piloté manuellement pour les noeuds critiques***
  - ***fournit une bonne estimation des capacités d 'interconnexion***



Placement / routage

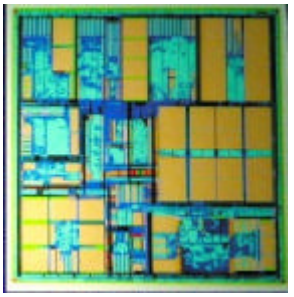
- **Les dernières étapes :**
  - ***extraction des capacités d 'interconnexion***
  - ***analyse temporelle rétro - annotée***
  - ***simulation logico - temporelle rétro - annotée***
    - **consommation**
    - **setup, hold**
  - ***Design Rules Check (DRC)***
  - ***Electrical Rules Check (ERC)***
  - ***Bonding Diagram***
  - ***Tape - out***
  - ***Champagne ?***

Conclusion

- **La conception logique est critique**
- **Forte valeur ajoutée**
- **Détermine presque exclusivement la qualité du produit**
- **Les validations sont très coûteuses**
- **Les choix matériels sont en grande partie dictés par des contraintes non matérielles**
  - ***Partitionnement***
  - ***Techniques de validation***
  - ***Performances des outils***
  - ***Maîtrise de la complexité***
  - ***Réutilisabilité***
  - **...**

Matériel et logiciel

Co-design, réutilisabilité et interfaces
pour les SoCs



Le pourquoi d'un SoC

- **Evolution de la complexité des systèmes**
  - ***100 millions de transistors/puce en 2003***
  - ***Intégration X2 tous les 2 ans (Loi de Moore)***
- **Optimisation des coûts de fonctions complexes**
  - ***Intégration traitement logiciel + Matériel***
  - ***Intégration traitement numérique + analogique***
- **Adaptation rapide au marché**
  - ***Développer + vite pour être le premier sur le marché***
  - ***Offrir une base performante et facilement adaptable***
    - **Utilisation de briques matérielles pour assurer les performances**
    - **Utilisation de processeurs et de briques génériques pour assurer la flexibilité**

Problèmes de conception d'un SoC

- **Les ressources de développement (outils/méthodes/hommes) n'évoluent pas aussi vite que la technologie**
- **Les coûts de développement + fabrication ont une croissance exponentielle (environ 500Keuros pour un circuit de 100mm^2)**
  - ***nécessité de faire un circuit fonctionnel du premier coup***
- **L'augmentation de complexité va à l'encontre de la diminution du temps de développement**
- **SOLUTIONS : outils et méthodes basés sur**
  - ***Des développement conjoints à haut niveau (co-design)***
  - ***La réutilisation des blocs préconçus***

PLAN

1. **Architectures d'un « SoC »**
2. **Notion de « co-design »**
3. **Les « IPs »**
4. **Interfaçage entre blocs internes**

Architecture

Une architecture n'est pas seulement une finalité structurelle et fonctionnelle, Da façon à garantir le respect des contraintes fonctionnelles, économiques et physiques, elle nécessite préalablement des études :

- d'adaptation à l'algorithme
- de partitionnement
- de méthodologie de conception

Selon la société ARM computer :

Un système : " Something so complex that it doesn't work first time"

Une Architecture : "The sum of the *Partitioning* and *Implementation* decisions taken in the design phase ...

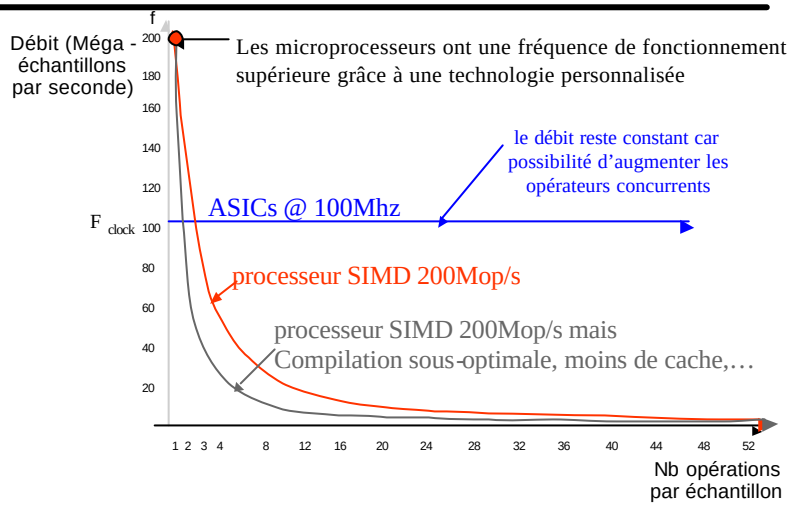
(including Conventions, Reuse, Economics, History & Experience)

... which together ... Implement a *Serviceable Solution* "

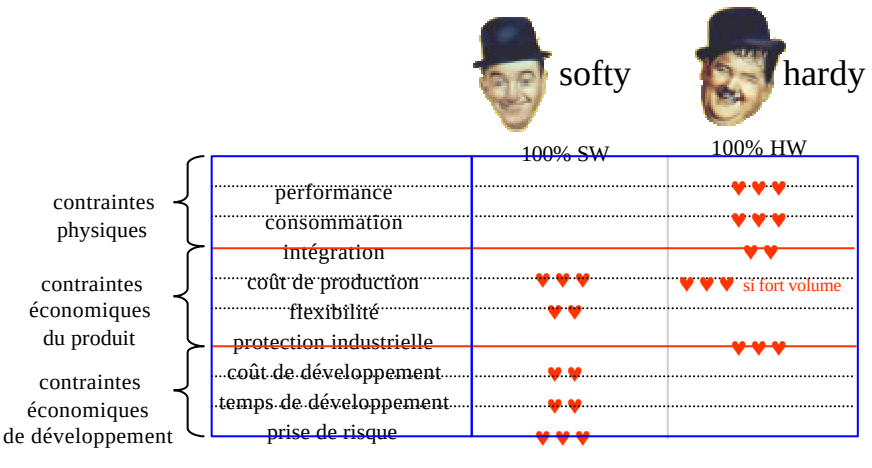
Architectures

- logicielles (SW) : **processeur + Logiciel**
  - ***flexibilité***
  - ***faible temps de conception***
- matérielles (HW) : **ASIC**
  - ***performances***
  - ***consommation***
  - ***protection industrielle***
- Mixtes
  - ***Tire profit des 2 approches => cas d'un SoC***

Performances



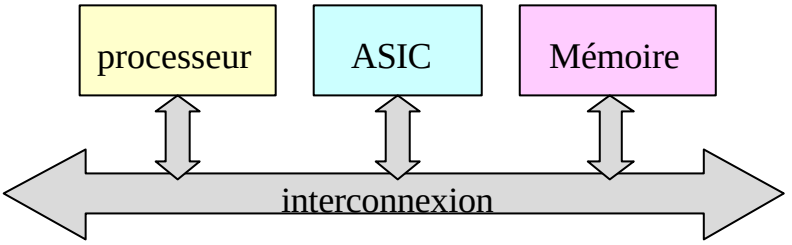
Complémentarité logiciel/matériel



*Fabless

Déclinaisons architecturales dans les SoC(1)

- **processeur(s) + ASIC(s)**
  - ***L'ASIC est un coprocesseur "accélérateur"***



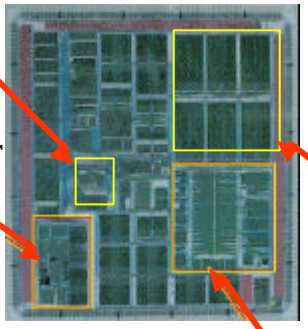
Déclinaisons architecturales dans les SoC (2)

- ASIP "Application Specific Integrated Processor"
  - le processeur est conçu spécialement pour le système. Le développement d'un compilateur est nécessaire.

	ASIC	ASIP	proc+ASIC	processeur
Performances	+++	++	+	
Consommation	+++	+	+	
Flexibilité	-	++	++	++
Temps de conception		--	-	+++

Exemple de système

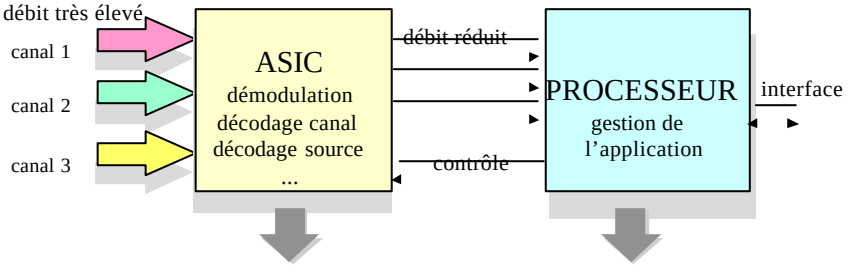
circuit bande de base GSM



- 400K portes
- 100Koctets mémoire
- 81mm2 on .5µ (1997)

Exemple de répartition des tâches

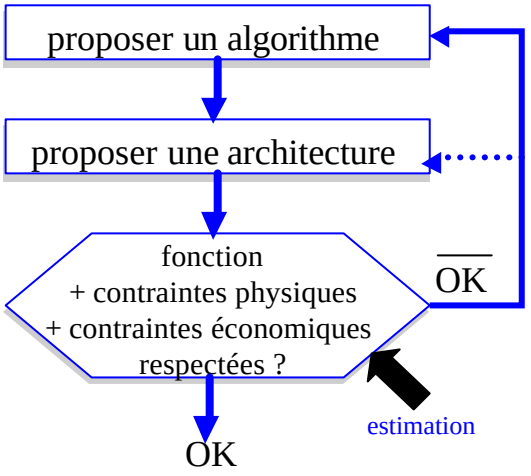
système d'acquisition



couches de communication physiques
avec des opérations rapides et répétitives

contrôle centralisé, gestion mémoire de masse, interface homme machine, gestion des couches hautes de communication

Faisabilité architecturale



Adéquation
Algorithme
Architecture

contraintes physiques :

- puissance de calcul
- capacité mémoire
- débits mémoire
- nombre d'E/S
- débits E/S
- consommation
- fréquence d'horloge
- taille totale
- ...

Quelle architecture optimale ?

Difficile d'y répondre

- **D'abord connaître parfaitement les besoins :**
  - ***la fonction mais aussi les contraintes économiques et physiques***
  - ***Un retour sur l'algorithme peut être nécessaire***
- **Les « meilleures » solutions sont souvent mistes HW/SW**
  - ***Un bon « partitionnement » tire profit de la complémentarité***
    - **SW : coûts de développement réduits, flexible**
    - **HW : hautes performances, consommation réduite**
- **Penser une architecture complexe c'est aussi penser aux méthodes et outils de conception et validation**
  - ***co-conception (co-design) si choix (processeur + ASIC) ou ASIP***
  - ***réutilisation des blocs déjà conçus « IPs »***

PLAN

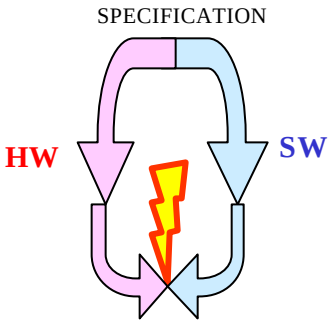
1. Architectures d'un « SoC »
2. Notion de « co-design »
3. Les « IPs »
4. Interfaçage entre blocs internes

Qu'est ce que le co-design ?

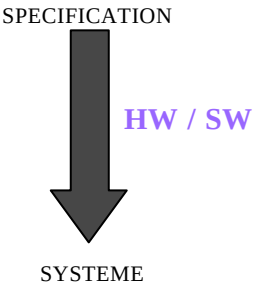
- **Conception (spécifier+valider+synthétiser) concurrente (matérielle + logicielle) d'un système électronique complexe**
- **étapes clés :**
  - ***Partitionnement :* Décider quelles tâches sont effectuées en SW et quelles autres sont effectuées en HW et comment se fait l'interface entre HW et SW**
  - ***Modélisation :* Coder la spécification de façon à représenter du mieux possible la fonctionnalité du système et ses contraintes.**
  - ***Co-Validation :* valider simultanément les parties matérielles et logicielles. Valider la fonction, la cohérence temporelle des tâches, analyser le respect des contraintes et préparer la synthèse**
  - ***Co-synthèse* : Effectuer la compilation (SW) et la synthèse logique (HW).**

Conception concurrente HW/SW

flot de conception classique



Flot de conception “codesign”



Il n’y a plus qu’une équipe HW/SW

Interêts du « codesign »(1)

- **Conception rapide de système intégré (SoC)**
  - ***Réduction du temps de mise à disposition sur le marché ou “time-to-market”***
- **Réduire la difficulté à valider et déboguer les systèmes complexes**
  - ***réduction des erreurs au plus haut niveau***
  - ***vérifier les contraintes d’inter-opérabilité et de temps réel***
- **Converger vers une solution optimale au sens des contraintes physiques et économiques (partitionnement)**

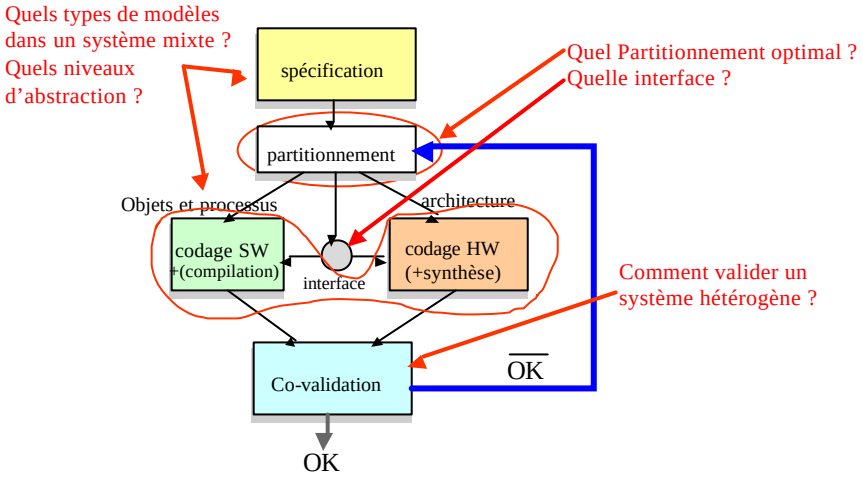
Interêts du « codesign »(2)

- **Tirer profit des bloc préalablement conçus (IPs)**
  - **cœur de processeurs,DSP, microcontrôleurs**
  - **périphériques, coprocesseurs**
  - **cœurs reconfigurables**
- **Conception d' ASIP**
  - ***Faire les compromis entre la complexité du logiciel et la complexité du processeur (attention à la complexité du compilateur)***
- **Reconfiguration dynamique des systèmes**
  - ***exemple : estimation de mouvement avec un algorithme génétique***

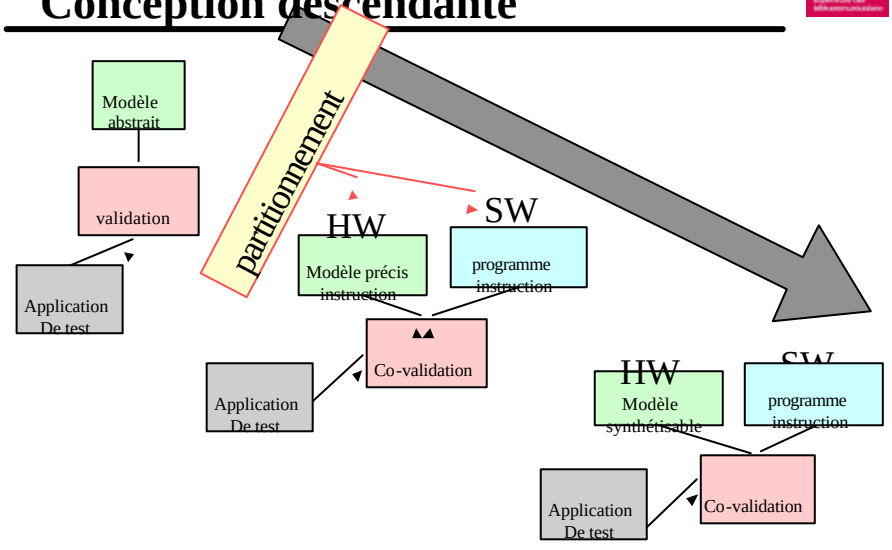
Tâches du co-design

- **Modélisation du système complet**
  - ***Etude de la fonctionnalité et des tâches concurrentes indépendamment du partitionnement HW/SW***
- **Partitionnement HW/SW**
  - ***hypothèses architecturales***
  - ***choix du processeur***
  - ***choix de l'interface HW/SW***
  - ***estimation des contraintes physiques et économiques***
- **Modélisation HW/SW, Synthèse, Validation**
  - ***Analyser la fonctionnalité et le respect des contraintes pour des granularités temporelles de plus en plus fines (ie instruction=>cycle=>ns)***

flot de conception du co-design

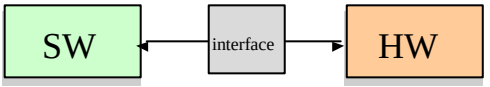


Conception descendante



partitionnement

- **A partir d'une fonction, trouver une architecture HW/SW**



- **Faire en sorte que cette architecture soit optimale au sens des contraintes physique et économiques : coût, performances, consommation, flexibilité…**
- **Problème NP complet**
  - ***Automatisation basée sur des méthodes heuristiques plutôt qu'exactes, trop coûteuses en calcul.***

Choix du processeur

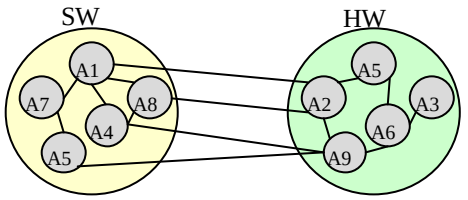
- **Processeurs standards :**
  - ***Interface homme - machine ou réseau, couches applicatives hautes***
  - ***Simplicité de mise en œuvre, flexibilité***
  - ***Performances limitées***
- **DSP :**
  - ***Traitement du signal, opérations de calcul répétitives***

Note : Les performances du processeur dépendant aussi
- Du compilateur
- De la mémoire cache (vitesse, taille, algorithme)
- De la bibliothèque de fonctions logicielles (écrites en assembleur)
- De la qualité de ses périphériques (communications, contrôle d'interruption, DMA, contrôleur mémoire externe,…)

Algorithmes de partitionnement(1)

- **Regrouper les objets en 2 partitions.**

Fonction={SW,HW}



- **Passer les objets d'une partition à une autre en minimisant une fonction de coût (ie performances, coût économique, complexité de l'interface,…)**

Algorithmes de partitionnement(2)

- **Approche orientée SW :** au début (SW=Fonction, HW=0), les objets migrent vers le HW jusqu'à ce que la contrainte de performances soit atteinte et que la fonction de coût soit minimum

- **Approche orientée HW :** au début (HW=fonction,SW=0), les objets migrent vers le SW tant que la contrainte de performances reste atteinte et que la fonction de coût est minimum

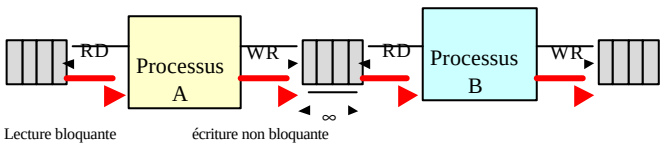
- Le meilleur partitionnement est en général obtenu manuellement en prenant en compte l'expérience et la culture de l'équipe HW/SW

Modélisation du système

- **Le modèle idéal doit être :**
  - ***Formel (manipulable automatiquement)***
  - ***Non contradictoire***
  - ***Complet (avec toutes les propriétés)***
  - ***Facile à lire et à modifier***
- **La modélisation s'appuie sur**
  - ***Un « modèle » de calcul pour gérer les opérations concurrentes***
  - ***Un langage informatique***
  - ***Un niveau d'abstraction*** *(ie synthétisable/non-synthétisable)*

modèle de calcul (1)

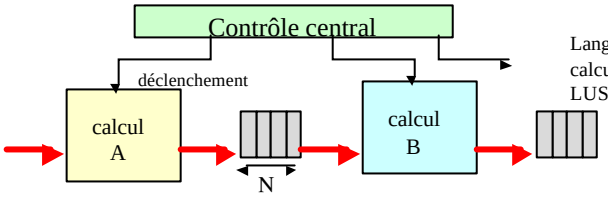
- **Les règles de calcul dépendent du type d'application**
  - ***Temps continu (système analogique)***
  - ***Temps discret. Modèles principaux :***
    - **Réseau de processus**
      - ***Les éléments effectuent des calculs indépendamment des autres (processus actifs). Les données transitent par des FIFOS infinies***



- Pas de vision sur la cohérence temporelle (pas de communication entre processus)

modèle de calcul (2)

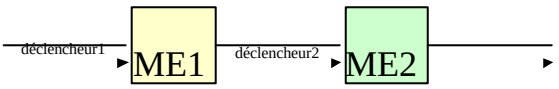
- **« Data flow » synchrone (un calcul par cycle)**
  - *Chaque élément traite un **paquet de N données** d'une façon **synchrone**.*
  - *Les éléments sont déclenchés suivant une condition particulière (état des FIFOs, temps de calcul des autres éléments,…). Une fois déclenché, un élément opère un calcul de N données.*
  - *Permet l'estimation du temps et de la mémorisation*
  - *Adapté aux traitements, pas au contrôle*



Langage approprié à ce type de calcul :
LUSTRE

modèle de calcul (3)

- ***« control flow » synchrone (un calcul par cycle)***
  - *Les éléments sont des **machines à états synchrones** déclenchées par la sortie d'un autre élément (les éléments sont dits « réactifs »). Leur structure de machine à état génère une latence de calcul.*
  - *Modèle bien adapté à l'analyse des communications et dépendances de données entre des sous-systèmes réactifs entre eux*
  - *Modèle permettant la vérification formelle*



modèle de calcul (4)

- ***Par événements***
  - *Chaque élément est un processus déclenché suivant un évènement=(signal, valeur,temps) et renvoie un autre évènement. Un gestionnaire d'évènements gère l'ensemble grâce à un échéancier associé à chaque signal*
  - *calcul adapté pour la simulation des modèles écrits en langage de description de matériel*
  - *modèle demandant beaucoup de puissance de calcul pour la simulation système*

Niveau d'abstraction synthétisable

- **Nécessité de faire un effort de codage de façon à permettre la synthèse :**
  - ***Architecture matérielle***
    - **Sans types et objets « abstraits »:fichiers, temps absolu, pointeurs,…**
    - **Avec les unités de mémorisation identifiées (niveau RTL)**
  - ***Code logiciel***
    - **Utilisation d'un langage évolué pour la compilation ou d'un langage machine**
- **Synthèse directe au niveau système**
  - ***Synthèse d'architecture VHDL/Verilog***
    - **pas encore au point**

Langages de modélisation

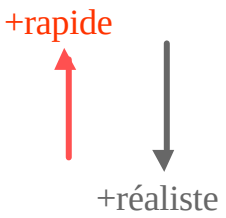
- **Modèles système**
  - ***C/C++***
  - ***VHDL/VeriLog***
  - ***Langage émergent :* SystemC *: langage mixte***
  - ***Langages informatiques divers***
    - **JAVA, ESTEREL, ADA …**
- **Modèles « synthétisables » ou compilables**
  - ***HW : VHDL/Verilog*** *(pas encore systemC)*
  - ***SW : C/C++, assembleur***
- **Modèles synthétisés ou compilés**
  - ***HW : VHDL/Verilog, EDIF (GDSII si placé et routé)***
  - ***SW : executable en langage machine***

Co-validation

- **Validation fonctionnelle par preuve formelle.**
  - ***Définition de propriétés logico-temporelles => pas d'environnement de test.***
  - ***Perfectible pour un modèle abstrait car nécessite une description de type machine à états.***
  - ***Complémentaire à la simulation pour des cas longs à simuler***
- **Co-simulation**
  - ***Simulation concurrente du C et du VHDL/Verilog***
  - ***Temps de simulation parfois très longs pour les modèles synthétisables***

Exécution d'une co-simulation

- **Sur station de travail**
  - ***Le code logiciel est exécuté sur un modèle de simulation du processeur***
  - ***Le code matériel est simulé en s'interfaçant avec le processeur (pas de testbenchs)***
  - ***Différents modèles de processeur***
    - **« Bus cycle accurate »**
    - **« Instruction Set Accurate »**
    - **« Cycle accurate »**
    - **« Time accurate »**



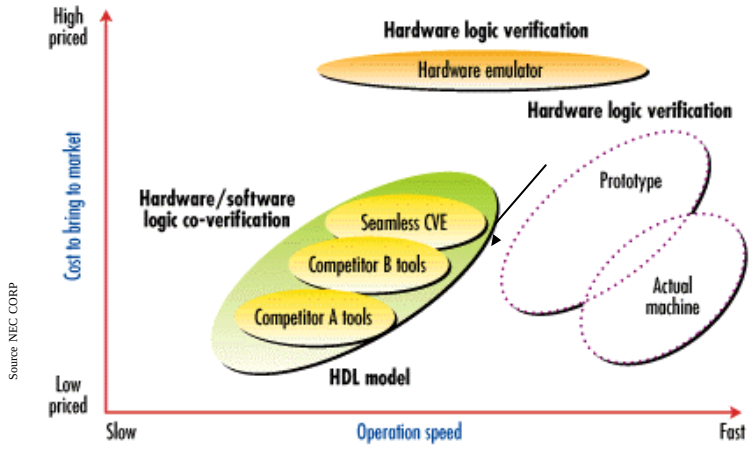
Co-vérification par prototypage

- **Utilisation d'une cartes munies de processeurs et FPGAs**
  - ***Le logiciel tourne sur le processeur cible avec l'utilisation éventuelle d'un débogueur intégré « ICE ».***
  - ***Le code matériel est synthétisé dans une technologie FPGA.***
  - ***Les + :***
    - La vérification nettement plus rapide que la co-simulation
    - le logiciel peut tourner en temps réel ou presque
  - ***Les limites :***
    - L'accès aux nœuds internes est restreint
    - Nécessité d'un code matériel synthétisable
    - La taille du FPGA, des ressources mémoires et des E/S sont limitées
    - L'environnement de validation est plus long à mettre en oeuvre

Co-vérification par émulation

- **Utilsation d'un accélérateur matériel**
  - ***Le code logiciel est exécuté soit sur un vrai processeur ou sur un processeur émulé dans l'accélérateur***
  - ***Le code matériel est synthétisé dans une technologie de matrices FPGA spécialisées où tous les nœuds sont accessibles***
  - ***les + :***
    - L'accès à tous les nœuds
    - L'accélération de la co-simulation
  - ***Les limites :***
    - Le coût
    - Nécessité d'un code synthétisable
    - L'éxécution non temps réel

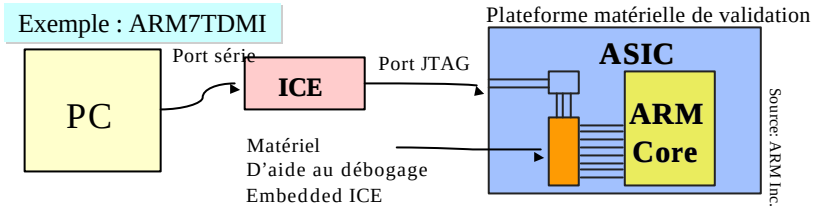
Comparaison des approches de co-vérification



Source NEC CORP

Débogueur intégré au processeur

- **Certains coeurs de processeur disposent d'un circuit interne dédié au débogage (Embedded ICE** In Circuit Emulator**)**
  - Module de Mise de points d'arrêt sur les addresses et les valeurs des données
  - Module de mémorisation des instructions exécutées (TRACE)
  - Possibilité de contrôler de l'extérieur (arrêt démarrage, points d'arrêts, téléchargement d'executables, lecture registres et mémoire…)
  - Nécessité d'un port de communication externe : JTAG,BDM,…
  - Ce débogueur peut être supprimé pour la production



résumé co-design

- **Modélisation au niveau système recommandée**
  - ***Rapidité de simulation***
  - ***Découvertes des erreurs système***
- **Pas d'outil de partitionnement HW/SW automatique**
- **Méthodes de co-validation par simulation lentes et perfectibles.**
  - Amélioration par
    - *utilisation de modèles de processeurs plus haut niveau*
    - *validation par preuve formelle*
    - *utilisation de prototypes ou d'accélérateurs matériels (plus coûteux)*
- **perspectives:**
  - *Utilisation d'outils et méthodes homogènes, fiables et inter-opérables (Méthodes à base de SYSTEMC et VHDL ?)*
  - *Accélérer la validation avec des outils de vérification formelle à différents niveaux d'abstraction*

Pointeurs intéressants

Modèlisation haut niveau

Ptolemy II **: pour la modélisation des systèmes hétérogènes avec plusieurs mode de calculs**
http://ptolemy.eecs.berkeley.edu/
Polis **: basé sur ptolemy et le langage ESTEREL**
http://www-cad.eecs.berkeley.edu/Respep/Research/hsc/abstract.html
Esterel : **langage orienté machines à états réactives pour la modélisation système**
SystemC : http://www.systemc.org/

Co-simulation

http://www-sop.inria.fr/meije/esterel/esterel-eng.html
N2C **: co-simulateur multi_niveaux de la société Coware**
http://www.coware.com/cowareN2C.html
Seamless : **co-simulateur de la société Mentor graphics**
http://www.mentorg.com/seamless/products.html
VCC : **conception au niveau système de la société Cadence**
http://www.cadence.com/products/vcc.html

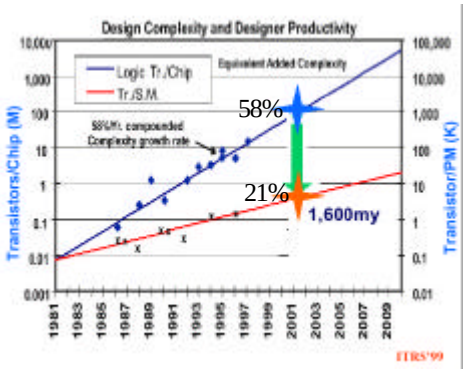
Accélération matérielle

http://www.cadence.com/products/palladium_new.html
http://www.mentor.com/emulation/

PLAN

1. **Architectures d'un « SoC »**
2. **Notion de « co-design »**
3. **Les « IPs »**
4. **Interfaçage entre blocs internes**

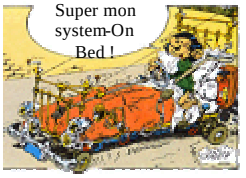
Réutilisation des blocs déjà conçus



La taille des circuits augmente plus vite que la productivité par ingénieur

Il devient de moins en moins possible de concevoir un SoC entièrement. Des parties déjà conçues doivent être réutilisées

« design reuse »



Exemples de complexité

Toshiba « Emotion Engine »
Processeur graphique de la Playstation 2
13,5 Mtransistors
225mm2 en 0.25µ
200 ing. X 4 ans = 800 hommes.ans

Intel « Itanium »
Processeur 64 bits EPIC
25,4 Mtransistors
0.18µ
4500 hommes.ans

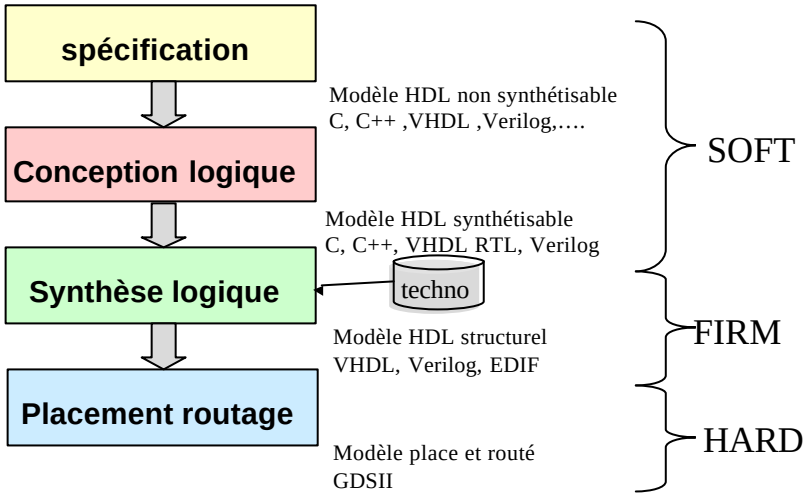
« design reuse »

- **Développer des « composants virtuels » réutilisables**
- **Problèmes :**
  - ***Une compagnie seule ne peut pas toujours concevoir tous les composants dont elle a besoin***
    - **Commerce de composants virtuels**
    - **Nécessité d'instaurer une protection juridique => « intellectual Property »**
    - **Nécessité d'utiliser un même formalisme pour la modélisation et l'utilisation pour l'adaptation rapide au système**

Niveaux des « IPs» matérielles

- **« Soft» IP**
  - ***Modèle de description de matériel. Peut être non synthétisable pour la validation purement fonctionnelle.***
  - ***Flexible, paramétrable***
  - ***Indépendant de la technologie***
  - ***Difficile à protéger***
- **« Firm» IP**
  - ***Liste d'équipotentielles post synthèse logique***
  - ***Dépend de la technologie***
  - ***Assez difficile à protéger***
- **« Hard» IP**
  - ***Liste d'équipotentielles placée et routée***
  - ***Contient les temps de propagation***
  - ***Facile à protéger (boîte noire avec uniquement les E/S)***

Usage et format des IPs



La validation des IPs

- **Un environnement de validation est nécessaire**
  - ***Pour la validation fonctionnelle du bloc au sein du système***
    - **Modèle de description pour la simulation (si possible haut niveau)**
- **Un environnement de test de production est nécessaire**
  - ***Pour détecter les défauts du circuit (cas des Ips hard)***
    - **Définir une Stratégie de test pour une bonne couverture de faute : Test fonctionnel , scan test, BIST,…**

Les fournisseurs d'IPs

- **Sociétés d'études et de conception**
  - ***Sociétés sans fonderie (« fab ») dont le profit vient des droits sur les licences***
    - **DSP Group (IP pour les télécoms), ARM (processeurs RISCs),**
  - ***offre incluant des IPs SOFT, de simulation et synthèse.***
- **Sociétés de semi-conducteurs**
  - ***Peuvent fournir des Ips HARD en plus des Ips SOFT***
  - ***TI, Motorola, Lucent, Altera,Xilinx,LSI Logic, STM***
- **Equipementiers**
  - ***ALCATEL, Adaptec***
- **Fournisseurs d'outils de CAO**
  - ***Fournissent des Ips SOFT uniquement***
    - **Mentor Graphics, Cadence, Synopsys,…**

Les types d'IPs

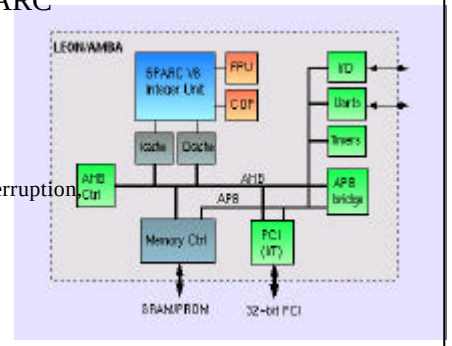
- **Processeurs:**
  - ***LSI logic CW4001/4010/4100, ARM 7TDMI, ARM 810,NEC 85x, Motorola 680x0, IBM PPC,…***
- **DSPs:**
  - ***TI TMS320C54X, Pine, Oak,…***
- **Composants de traitement spécialisés**
  - ***Cryptographie, traitement d'images, multimédia : JPEG codec, MPEG decoder***
- **Contrôleurs mémoire et bus : SDRAM, USB, PCI, UART,AMBA**
- **Réseaux : ATM, Ethernet**

Pointeurs : IP commerciales www.design-reuse.com
IP « libres » www.opencore.org

Cœur de processeur libre : LEON

• Processeur 32 bits compatible SPARC conçu à l'ESA

•5 étages de pipeline
• cache instructions et cache données
• MAC et division câblées
• Contrôleur mémoire SRAM ou ROM
• Périphériques : uarts, timers, contrôleur d'interruption, port E/S 16 bits
• bus AMBA-2.0 AHB and APB
•Interfaces co-processeur
• Mode veille



http://www.gaisler.com/leonmain.html

Cœur sous license « Lesser GPL » (source pas nécessaire)
testbench VHDL sous licence « GPL »

.

LEON dans les SoC

Société	techno	caches	fmax	FPU	taille core
UofToronto	TSMC 0.18µ	4KB	95MHz	non	1,1mm^2
iROC :	? 0.25µ	?	100MHz non	?	
ATMEL :	0.35µ durcie	4KB	50 MHz oui	30mm^2	
XILINX :	Virtex XCV800-6	?	35MHz	non	4800 cells
ALTERA :	APEX20K200E-1	?	37 MHz	non	6700 cells

Pour développer un SoC à base de LEON :
http://www.leox.org

IPs matérielles : LEON, AMBA
IPs logicielles : Linux sans MMU

PLAN

1. Architectures d'un « SoC »
2. Notion de « co-design »
3. Les « IPs »
4. Interfaçage entre blocs internes

solutions

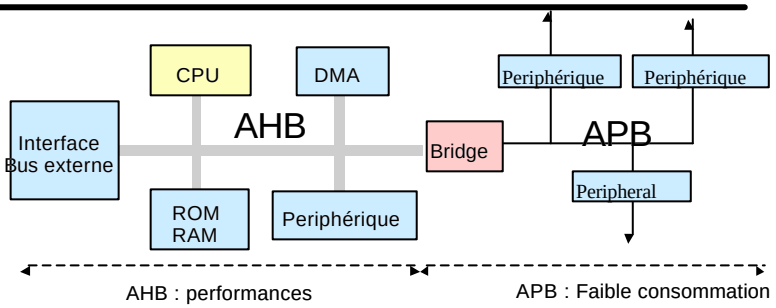
Les bus

- Logique 3 états peu recommandée
- Nécessite un arbitre si plusieurs maîtres potentiels
- 1 seul accès à la fois
- Beaucoup de standards existants
  - *PCI, Pibus, AMBA (pas de 3 états mais des MUX)*

Les liaisons point à point

- Plusieurs accès simultanés
- Adaptés au architectures en pipeline
- Pas de flexibilité

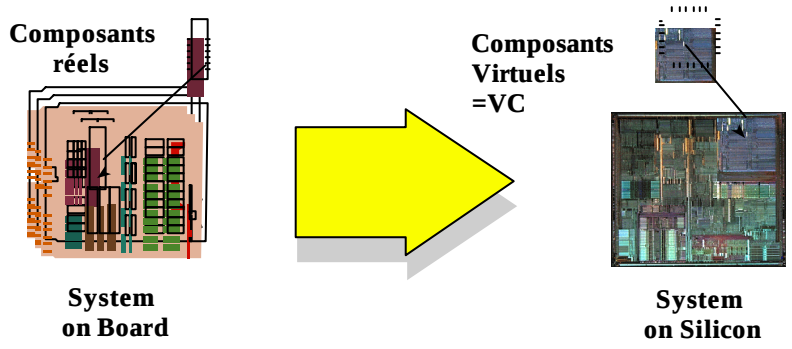
Le bus AMBA



- Synchrone Adresse/donnée séparées.
- Multi-maître – nécessite un arbitre.
- Chemin de données 32- 64- 128- 256 bits
- Bus d'addresse 32 bits
- Pipeline
- Accès en rafales (4- 8- 16).
- Pas de 3 états => utilisation de multiplexeur

- Synchrone Adresse/donnée séparées.
- Un seul maitre : le « bridge ».
- Chemin de données 32- 64- 128- 256-bit
- Bus d'addresse 32 bits
- Pas de pipeline.
- Transferts en 2-cycles. Pas de rafales.
- Statique – consommation nulle au repos

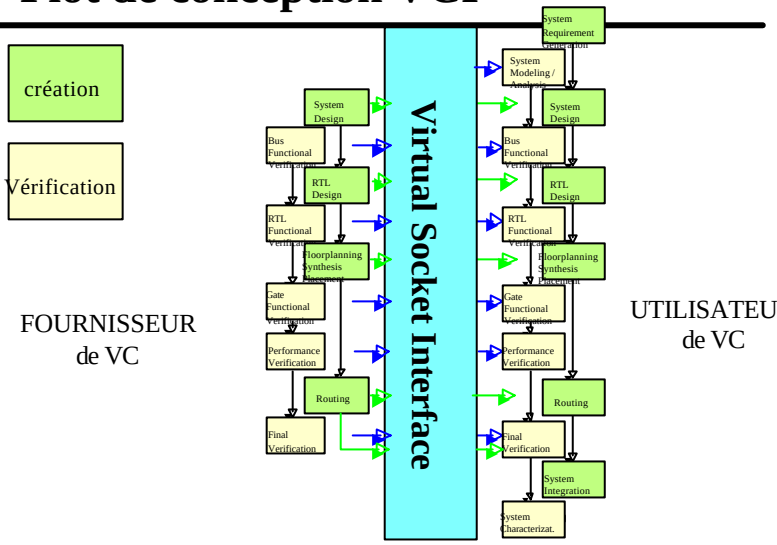
L'initiative industrielle : VSIA



Consortium d'industriels pour spécifier les interfaces entre le fournisseur de VC et l'utilisateur de VC

http://www.vsia.com

Flot de conception VCI



Actions VSIA

9 actions

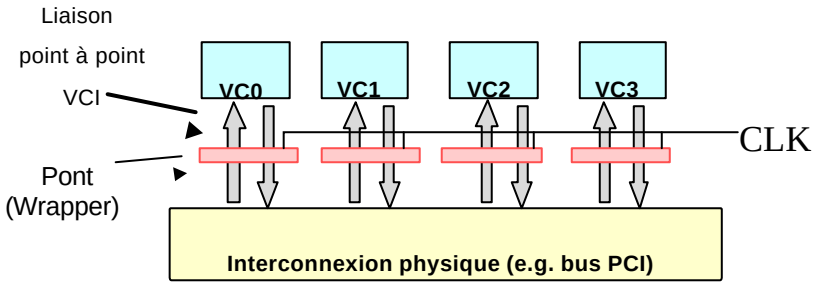
- **Bus internes (OCB : On Chip Bus)**
- **Conception au niveau système**
- **Composants analogiques et mixtes**
- **Protection des IPs**
- **Validation fonctionnelle**
- **Validation post synthèse**
- **Test pour la production**
- **Transfert des composants virtuels entre compagnies**
- **Qualité**

Objectifs du groupe de travail OCB

- **Interface générique entre IPs**
- **Ne pas réinventer un nouveau bus**
  - ***Il existe beaucoup de solutions performantes***
    - **PCI, PCI-X**
    - **AMBA**
    - **PI-BUS**
    - **ST-BUS**
    - **Silcon Backplane,…**

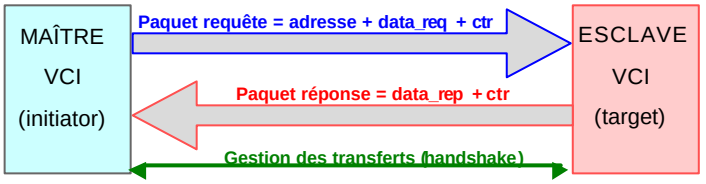
Concepts principaux (1)

Liaison point à point synchrone



Concepts principaux(2)

Communication MAÎTRE ⇔ ESCLAVE



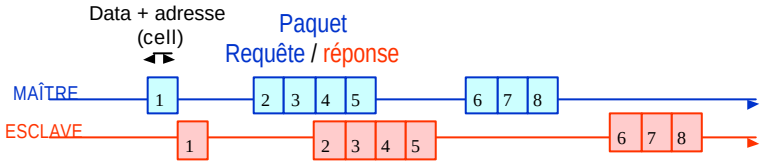
- **Espace addressable partagé**
- **Pas de bus bidirectionnel (pas de 3 états)**
- **Bus MAÎTRE = bus adresse + bus data_requête,**
- **Bus ESCLAVE = bus data_réponse**
- **Largeur des bus données adaptables n*8 bits**

Concepts principaux (3)

- **Signalisation de l'atomicité d'un groupe de données. Définitions :**
  - **Groupe de données =PAQUET**
  - **Transfert (requête+réponse) = TRANSACTION**
- **La transaction d'un paquet doit s'effectuer sans coupure et si possible en rafale**
- **Signal EOP pour signaler la fin d'un paquet (mais pas forcément la fin de la transaction)**
- **La paquets peuvent être chaînés**
- **L'esclave peut signaler une erreur de transfert et sa nature.**

Concepts principaux(4)

Temps de transmission du MAÎTRE indépendant de l'ESCLAVE
« SPLIT PROTOCOL »

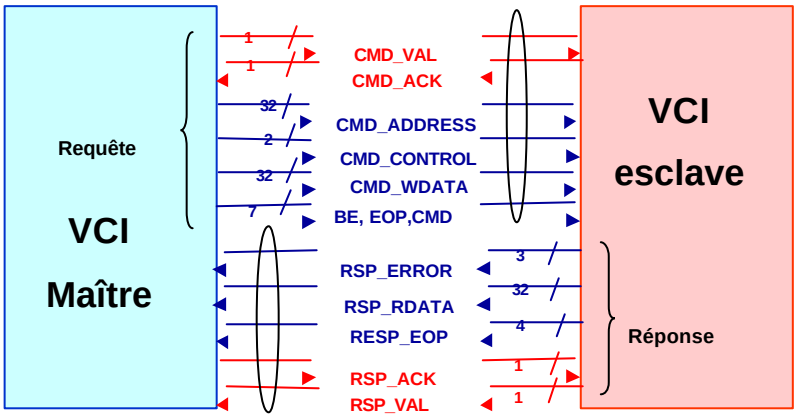


2 gestions de la transaction (HANDSHAKE) :

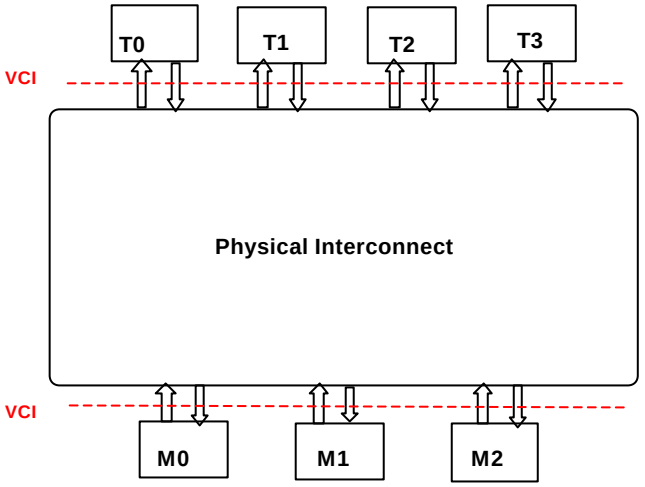
1 Handshake pour les transferts Maître=>Esclave

1 Handshake pour les transferts Esclave=>Maître

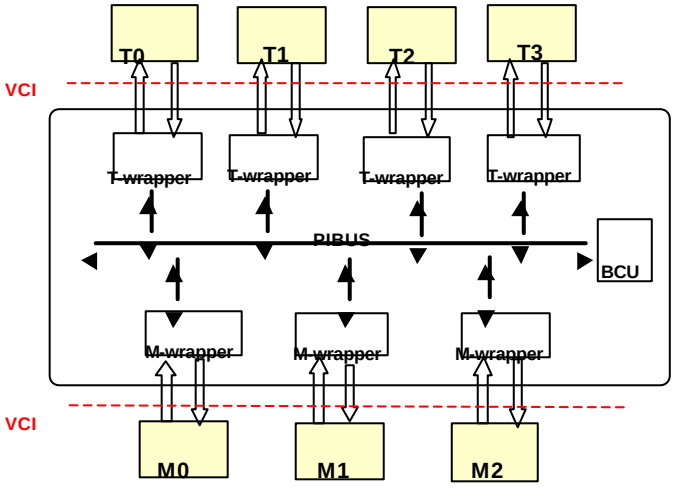
Exemple d'Interface physique



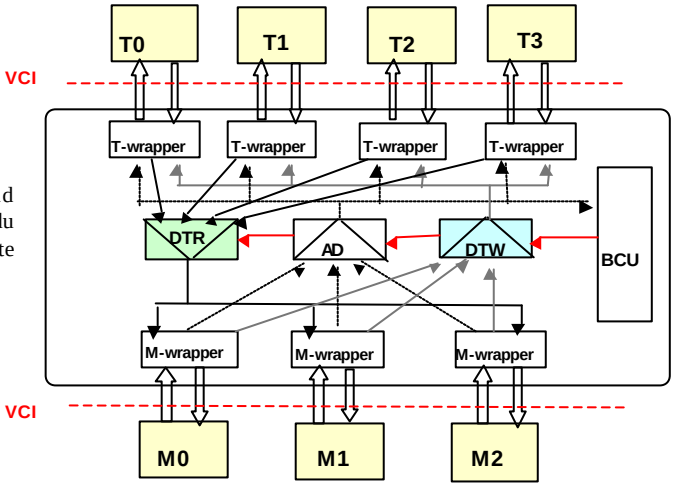
Utilisation du VCI



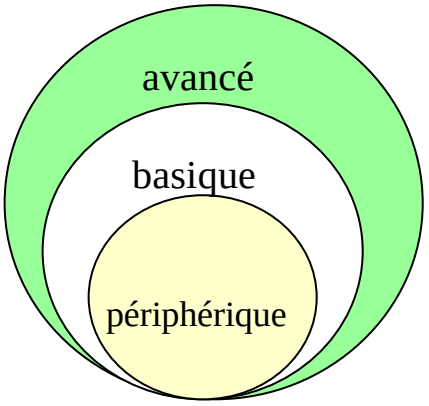
Exemple VCI/PI-bus



Exemple VCI / AMBA AHB



3 niveaux

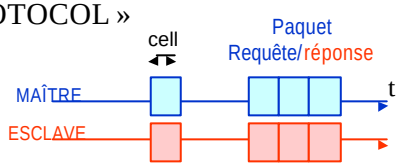


Les niveaux sont Inter-opérables

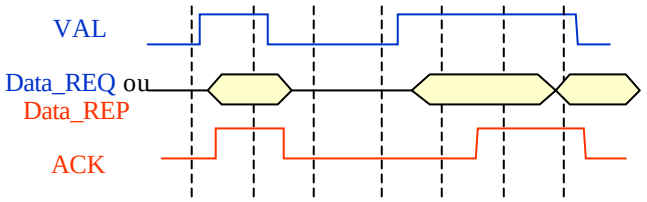
les IPs de niveau supérieur doivent être contraints pour dialoguer avec les IPS de niveau inférieur

Protocole « périphérique »

Chronogrammes des transmissions MAÎTRE ⇔ ESCLAVE imbriqués .Pas de « SPLIT PROTOCOL »



1 seul « Handshake »



Protocole « basique »

- **« SPLIT PROTOCOL»**
- **Une requête implique nécessairement un réponse**
- **Possibilité de chaîner les paquets**
- **4 opérations possibles :**
  - **lecture**
  - **écriture**
  - **Pas d'opération**
  - **Lecture verrouillée (l'esclave attend que la donnée soit modifiée pour répondre)**
- **Attributs sur la gestion des adresses dans les paquets**
  - **Adresses adjacentes**
  - **Adresses cycliques**
  - **Adresse constante**
  - **Adresses aléatoires (pas d'attributs)**
  - **Longueur du paquet (en plus de EOP)**

Protocole « avancé »

- **Idem protocole « basique » +**
  - ***possibilité d'avoir les paquets retour dans le désordre***
  - ***Taille des paquets requête/réponse pouvant être différents .***
    Exemple :
    - **paquet requête = 1 cellule (adresse+ attributs+lecture)**
    - **Paquets réponse = nombre de cellules demandées**
  - ***Optimisation des latences (« arbitration Hiding ») entre la transmission de 2 PAQUETS***
  - ***Identifiants***
    - **Nécessité d'un identifiant par PAQUET pour gérér l'ordre des PAQUETS**
    - **Identifiant fixe pour chaque MAÎTRE**
    - **Identifiant par « processus » ou THREAD. Permet de créer des MAÎTRES virtuels**

Conclusions

- **Interface simple et souple**
- **Permet la conception d'IPs indépendamment des BUS**
- **La norme VCI n'inclut pas :**
  - ***Isochronisme***
  - ***Traitement des interruptions***
  - ***Gestion des caches (SNOOPING)***

Interface avec le monde extérieur

Lirida NAVINER/Jean-Luc DANGER

Plan

- **Les interfaces**
  - ***Les problèmes de l'assemblage***
  - ***Protocoles de communication***
- **L'assemblage**
  - ***Boîtiers : Contraintes et types***
  - ***PCB et MCM***
- **Les mémoires**
  - ***Architectures et Utilisation***

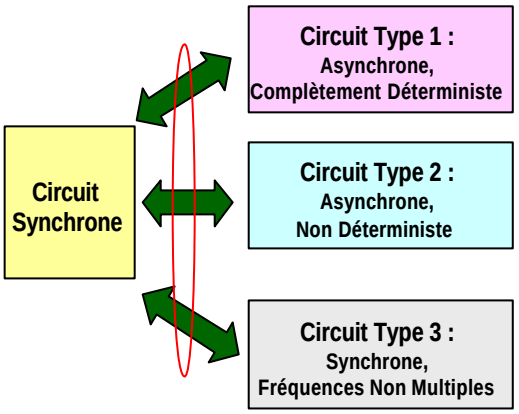
Le Problème de l'Assemblage

- **Un système**
  - ***Plusieurs composants***
    - **Mémoires, processeurs, bus, etc.**
  - ***Evolutions technologiques différentes***
    - **Caractéristiques différentes (tensions, vitesse, etc.)**
  - ***Modes de fonctionnement différents (synchrone, asynchrone)***
- **Bon séquencement des opérations**
  - ***Adaptation des paramètres électriques, cycles d'attente, circuits de synchronisation, protocoles de requête/acquittement, arbitres, adaptation d'horloge, etc.***

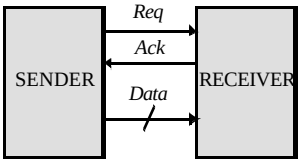
Séquencement d'Opérations

- **Système synchrone**
  - ***L'horloge agit comme un signal d'achèvement***
    - **Elle garantit le séquencement correct des événements**
- **Système asynchrone**
  - ***L'achèvement est assuré par analyse temporelle***
    - **Le séquencement est implicite dans la logique**
  - ***Système auto-séquencé***
    - **Achèvement garanti par un signal d'achèvement (autre que l'horloge)**
      - ***Séquencement garanti par protocole d'acquittement***
  - ***Fréquences d'horloge non multiples***

Le Problème de la Synchronisation

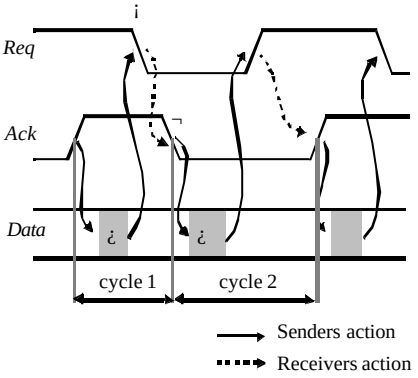


Protocole d'Acquittement



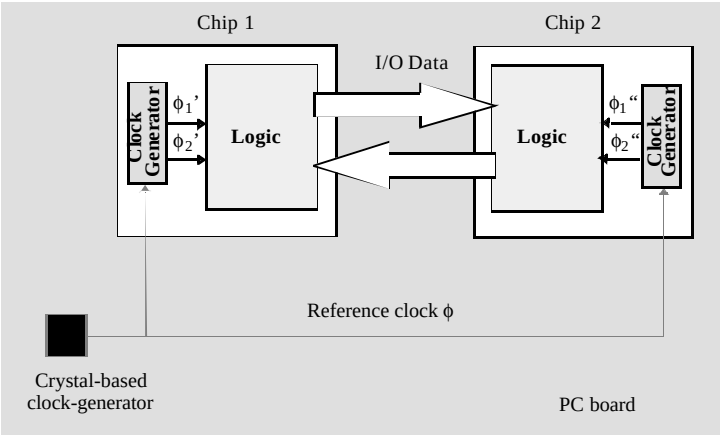
(a) Sender-receiver configuration

(b) Timing diagram



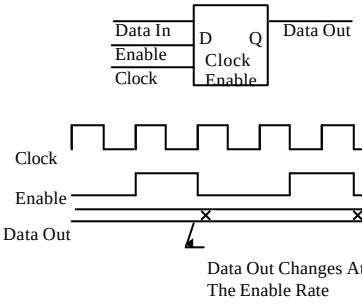
Two-Phase Handshake

Le Problème des Horloges



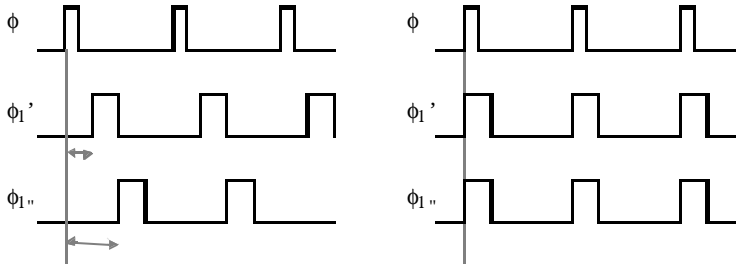
Le Problème des Horloges

- **Bases de temps différentes de l'horloge système**
  - ***Problème pour stocker, traiter, transférer des données***
    - **Générer des signaux de validation et utiliser les contrôles de validation plutôt que'utiliser plusieurs horloges différentes**



Si Horloges Différentes …

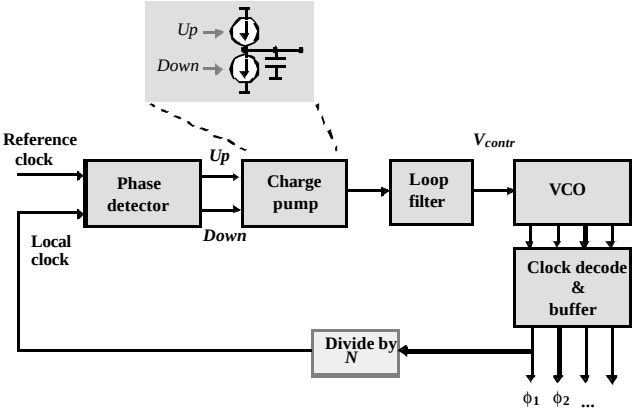
Veiller au problème de skew !



(a) Skew of local clock signals with respect of reference clock.

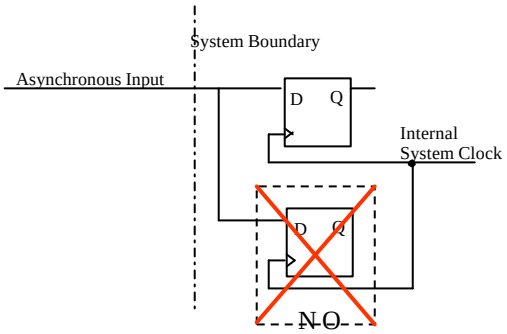
(b) Local clock signals as produced by PLL based clock generator.

Générateur d'Horloge Basé sur PLL



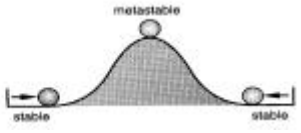
Synchronisation d'Entrées Asynchrones

Synchronisation en un seul endroit !



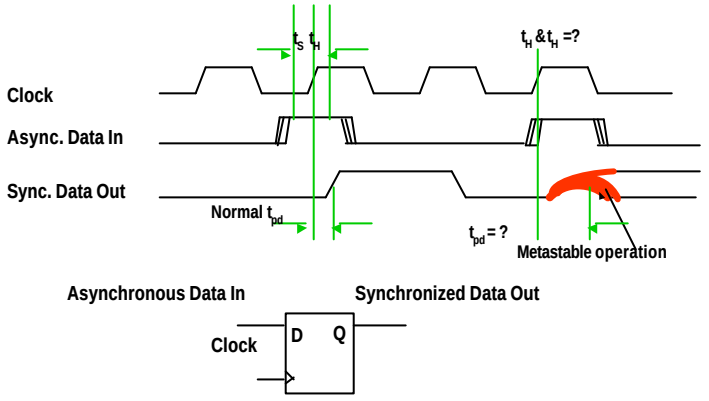
Méta-stabilité et Synchronisation

- **Les données sont échantillonnées sur front montant de l'horloge**
  - ***Si les niveaux de tension ou les T_{su}/T_h ne sont pas respectés, la sortie de la bascule se trouve dans un état méta-stable***



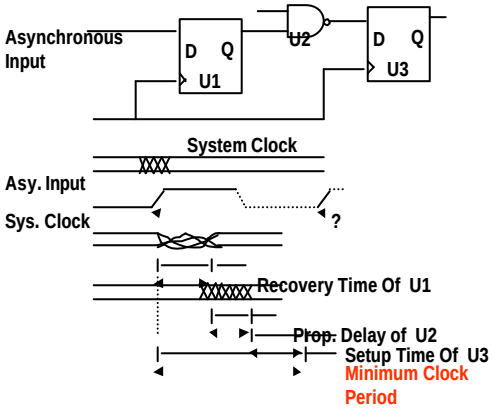
- **Méta-stabilité**
  - ***L'une des sources les plus fréquentes de problèmes de synchronisation***

Méta-stabilité



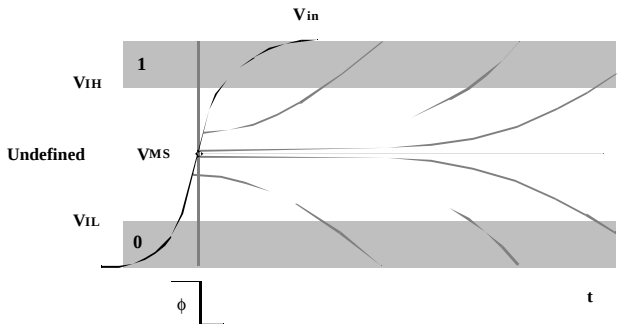
Temps de résolution : temps nécessaire pour avoir une valeur logique valide

Méta-stabilité



Le temps de résolution doit être ajouté pour la détermination de la fréquence maximale !

Méta-stabilité



$$v(t) = V_{MS} + (v(0) - V_{MS})e^{t/\tau}$$

Méta-stabilité

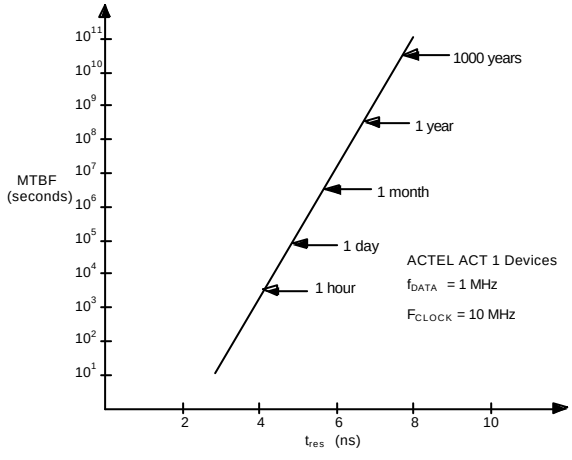
- **Attendre "suffisamment" pour que la bascule sorte de l'état méta-stable**
  - ***Il n'y a pas de borne supérieure ...***
  - ***Les mesures sont statistiques (MTBF, MTBU)***
- **Forcer la bascule à une valeur logique valide**
  - ***Améliorer le système de synchronisation***
    - **Mise en cascade de plusieurs bascules**
    - **Circuits de synchronisation plus rapides**

Medium Time Between Failures

$$MTBF = e^{\left(\frac{K_2 t_r}{K_1 F_{CLK} F_{DATA}} \right)}$$

- t_r *est le temps de résolution*
- K_1 *et* K_2 *sont des constantes, caractéristiques des bascules*
- F_{CLK} *et* F_{DATA} *sont les fréquences de l'horloge de synchronisation et des données asynchrones, respectivement*

MTBF et Temps de Résolution



Erreur de Convergence

- **Probabilité de convergence vers une valeur erronée**

$$p = T_0 e^{-\frac{t_r}{t_c}}$$

  - **t_r est le temps de résolution**
  - **T_0 et t_c sont des constantes, caractéristiques de la bascule**

- **Mean time between upsets (MTBU)**

$$MTBU = \frac{e^{-\frac{t_r}{t_c}}}{T_0 f_{clock} f_{data}}$$

Exemple

- **t_r = 5 ns, t_c =0.1 ns, T_0 = 0.1s:**

$$p = 0.1e^{\frac{-5\times10^{-9}}{-0.1\times10^{-9}}} = 2\times10^{-23}$$

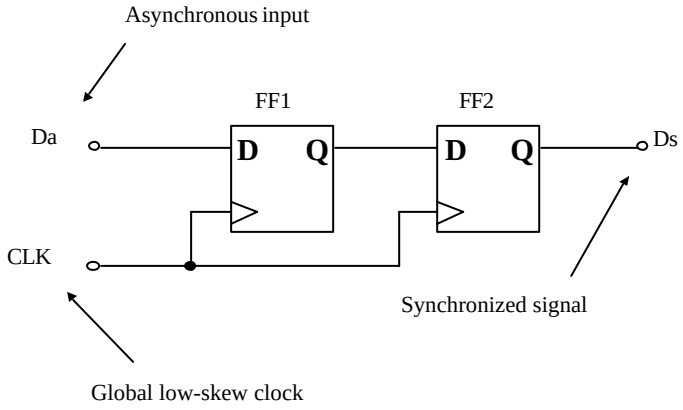
- **f_{clock} = 100 MHz et f_{data} = 1 MHz:**

$$MTBU = \frac{e^{\frac{-5\times10^{-9}}{-0.1\times10^{-9}}}}{(100\times10^{6})(1\times10^{6})(0.1)} = 5.2\times10^{8}\ \text{sec} \approx 16\ years$$

Pour un bus de 64 bits, le MTBU sera de trois mois

Circuit de Synchronisation

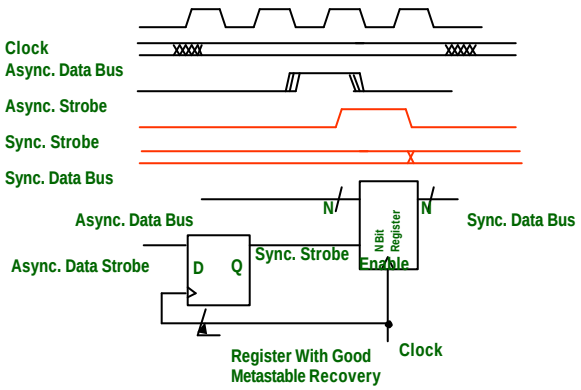
Double bufferisation



A5

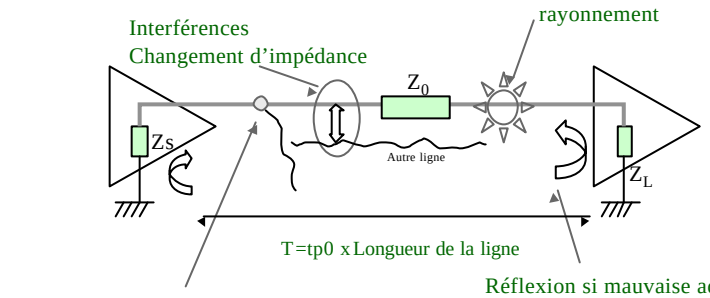
Synchronisation de Bus Asynchrone

Synchronisation des signaux de contrôle



La connexion entre circuits et cartes

- **Le comportement d'une ligne : domaine des hyperfréquences**



Connexions => Changement d'impédance

Réflexion si mauvaise adaptation d'impédance

$$s = \frac{Z_0 - Z_L}{Z_0 + Z_L}$$

La connexion entre circuits et cartes

- **Les contraintes**
  - ***Haut débits, faible rayonnement électromagnétique, faible consommation, immunité au bruit, sortance élevée pour les bus et fonds de panier***
- **Les solutions**
  - ***Sur ligne simple***
    - **1 seul fil référencé par rapport à la masse**
  - ***Différentielle***
    - **2 fils portant des grandeurs complémentaires**

Connexion par fil simple : « single ended » transmission

Inconvénients

- ***Faible immunité au bruit***
  - **Nouvelles technologies CMOS à faible Vdd (=> excursion (swing) très faible :0,86v si Vdd=1,8v)**
  - **Beaucoup de Bruit d'alimentation en électronique numérique (commutations synchrones)**

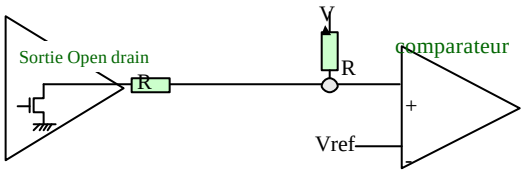
Sortie classique : 3 états (TTL, LVTTL,CMOS)

- ***Performances sous-optimales pour les bus (sortance élevée) et les connexions DRAM (débits élévé)***

Connexion des bus (fonds de panier)

Sorties open drain

- ***Standard BTL*** ***(Backplane Transfer Logic)***
- ***Standard GTL*** ***(Gunning Transfer Logic)***

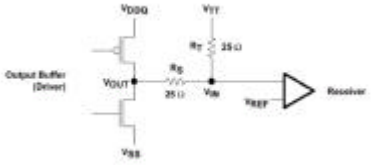


Pas de 3 états à gérer
Adaptation de tensions facile
Adaptation d'impédances
Tension de référence Vref très précise

	V	R	Vref
BTL	2,1v	29Ω	1,55v
GTL	1,2v	50Ω	0,8v

Connexion des DRAM

- **SDRAM (pour bus PC133MHz)**
  - ***Sortie 3 états => difficile d'augmenter les débits***
- **DDR SDRAM (Double Data Rate en utilisant un horloge à fréquence double)**
  - ***Sorties SSTL_2 (Series Stub Terminated Logic)***

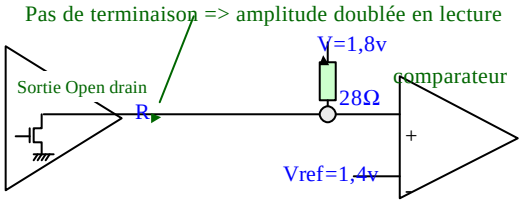


PARAMETER	V_{CC} = 3.3 V	V_{CC} = 2.5 V
V_{REF}	1.5	1.25
V_{IH}(min)	V_{REF} + 0.2	V_{REF} + 0.18
V_{IL}(max)	V_{REF} – 0.2	V_{REF} – 0.18

- ***Mécanisme de synchronisation par PLL pour multiplier l'horloge par 2 et réduire le skew (retard d'horloge interne à la SDRAM) et ainsi le temps d'accès à la donnée => fclk=266MHz=> fclock virtuelle=533MHz)***

Connexion des DRAM (2)

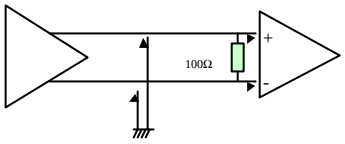
- **RAMBUS**
  - ***bus série 16 bits @400MHz sur 2 fronts***
  - ***PLL à chaque circuit pour doubler la fréquence (800M_mots de 16 bits/s) et réduire le temps d'accès***



Un fil de masse entoure chaque signal

Connexion différentielle

- **2 fils par bit**
  - ***L'information sur chaque fil est complémentaire***
    - **Permet de tirer parti de la réjection en mode commun des amplificateurs d'entrée**
    - **Excursion (swing) réduite**
    - **Consommation et Rayonnement réduits**
    - **Bonne immunité aux bruits**



Parameter	LVDS	Optics	GTL	TTL
Output voltage swing	±350mV	n/a	1.2V	2.4V
Receiver threshold	±100mV	n/a	160mV	1.2V
Speed (Mbps)	>400	>1000	<200	<100
Dynamic power	Low	Low	High	High
Noise	Low	Low	Med	High
Cost	Low	High	Low	Low

Source : NS

Plan

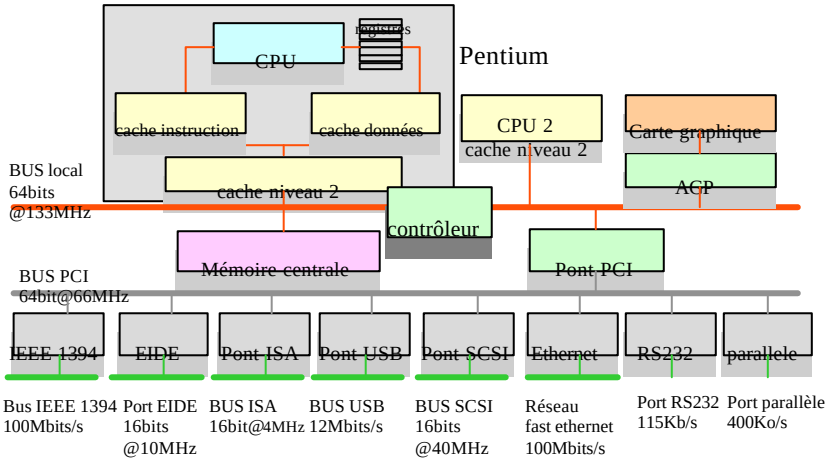
- **Les interfaces**
  - ***Les problèmes de l'assemblage***
  - ***Protocoles de communication***
- **L'assemblage**
  - ***Boîtiers : Contraintes et types***
  - ***PCB et MCM***
- **Les mémoires**
  - ***Architectures et Utilisation***

Interfaces entre circuits : protocoles de communication

Connexions courtes, hauts débits :

- ***Bus rapide (>100 Mo/s)***
  - **Gestion du bus et arbitrage nécessaire si plusieurs maîtres**
  - **Mode rafale**
  - **Bus donnée >=32 bits**
  - **Interruptions**
  - **Protocole asynchrone pour gérer le non-déterminisme des temps de réponse**
  - **Les + : « PLUG AND PLAY », isochronisme**
- ***Point à point dans certaines applications (circuit graphique)***

Exemple : Communications dans un PC



Exemple de bus : PCI

- **Fréquence de fonctionnement**
  - *33 / 66 MHz*
- **Bus d'adresse/données**
  - *32 / 64 bits*
- **Débit**
  - *132 / 528 Mo/s*
- **Trois modes d'accès**
  - *Mode maître, esclave ou maître/esclave*
  - *I/O, Mémoire, Configuration*
- **Brochage minimal**
  - *47 broches pour un esclave, 49 pour un maître*

Principaux Signaux

Broches communes à toutes les cartes

- ~~AD[31..0]~~ : Bus d'adresse/données multiplexées
- ~~CBE[3..0]~~ : Commandes/Byte enable
- ~~FRAME~~ : Indicateur de transaction
- ~~IRDY~~ : Indique que le maître est prêt
- ~~TRDY~~ : Indique que l'esclave est prêt
- DEVSEL : Indique qu'un esclave a reconnu un accès

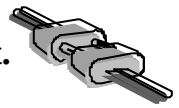
Pour le « Handshake » du protocole asynchrone

Broches spécifiques

- ~~IDSEL~~ : Sélecteur de carte pour configuration
- ~~REQ~~ : Indique qu'un maître demande le bus
- GNT : Indique à un maître qu'il a le bus

Plug 'n' play...

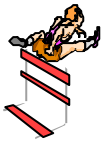
- Chaque carte possède son espace de configuration.
- Le système se configure automatiquement.

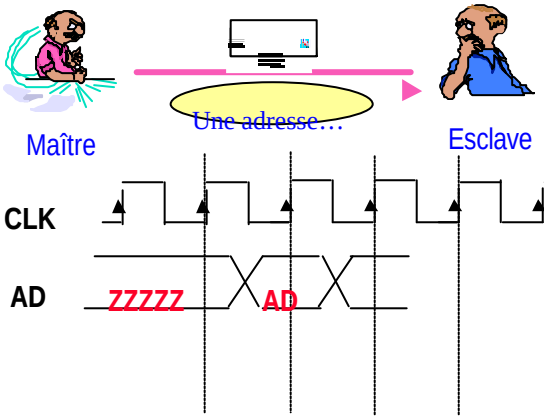


Utilisation simple...

Rapide...

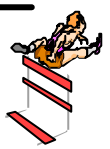
- Gagner du temps sur les transactions : burst

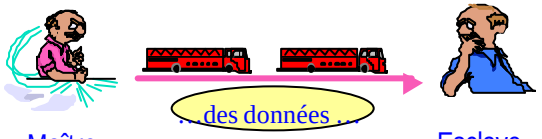


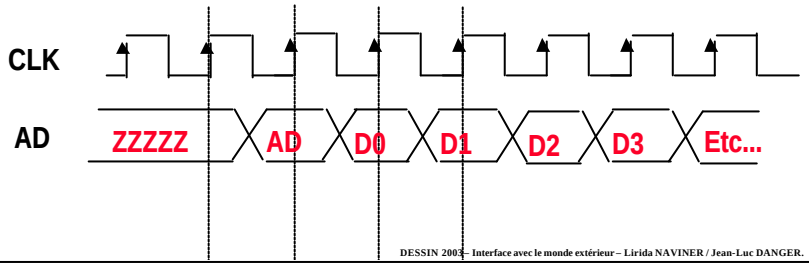


Rapide …

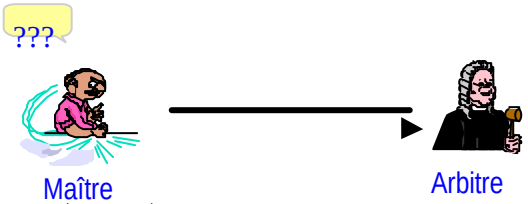
- Gagner du temps sur les transactions : burst

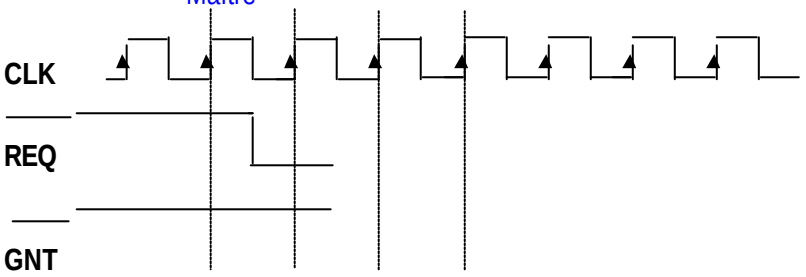




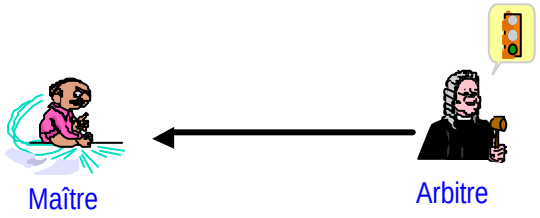


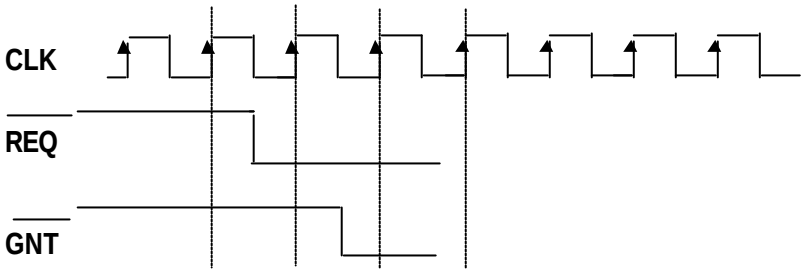
Transaction PCI



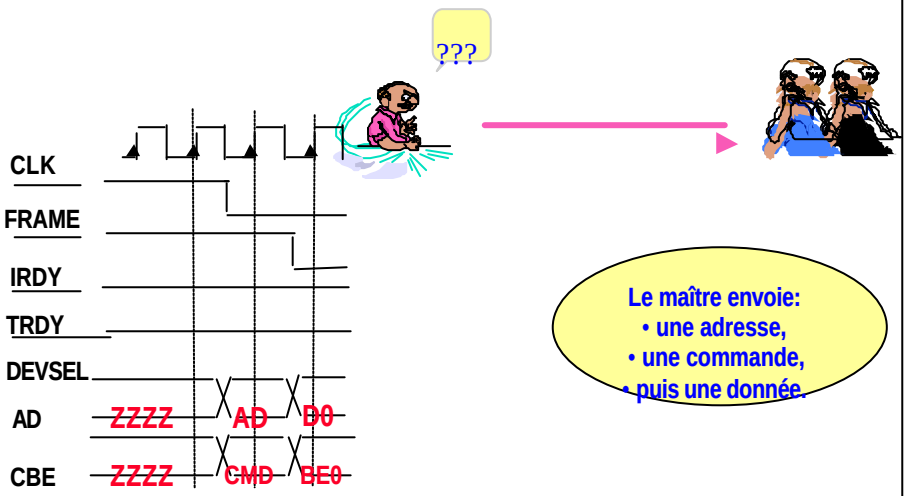


Transaction PCI



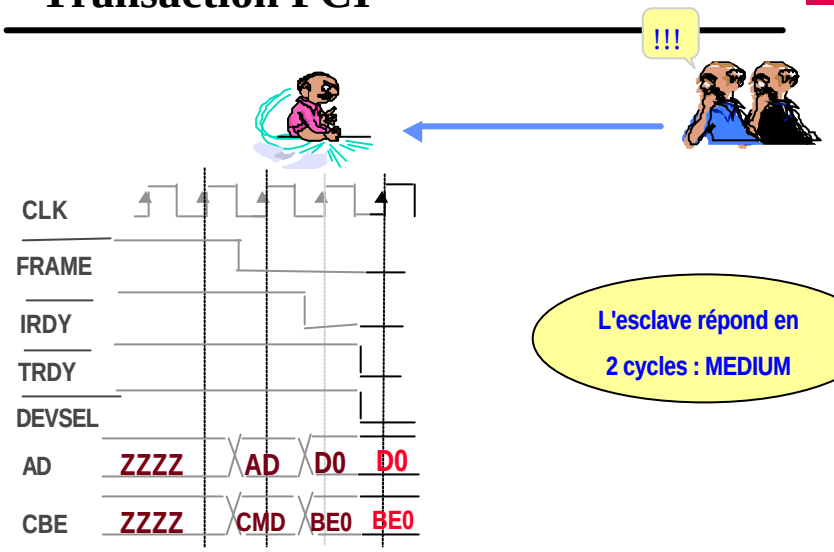


Transaction PCI



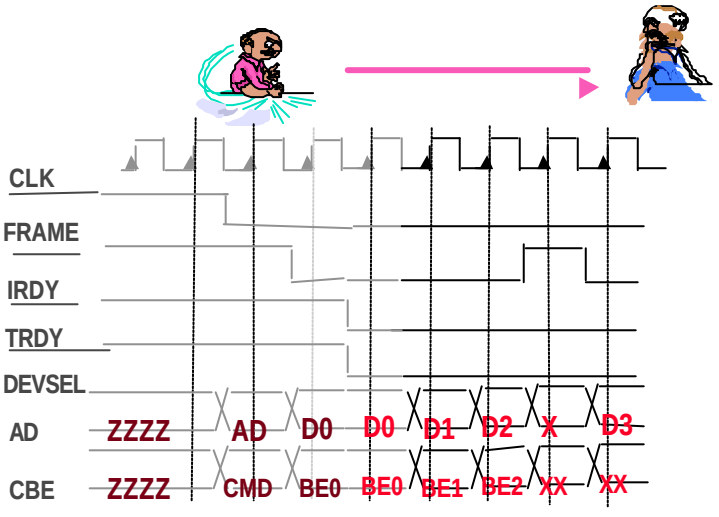
Le maître envoie:
- une adresse,
- une commande,
- puis une donnée.

Transaction PCI

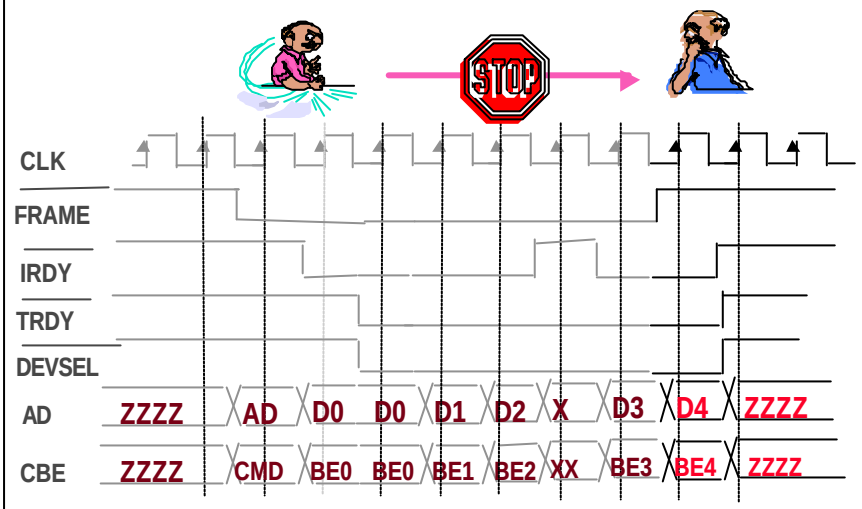


L'esclave répond en 2 cycles : MEDIUM

Transaction PCI



Transaction PCI



Plan

- **Les interfaces**
  - *Les problèmes de l'assemblage*
  - *Protocoles de communication*
- **L'assemblage**
  - *Boîtiers : Contraintes et types*
  - *PCB et MCM*
- **Les mémoires**
  - *Architectures et Utilisation*

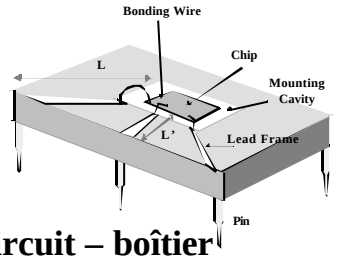
Boîtiers : Contraintes

- **Electriques**
  - ***Pas ou peu de parasites***
- **Mécaniques**
  - ***Robustesse***
- **Thermiques**
  - ***Elimination efficace de la chaleur***
- **Economiques**
  - ***Bas prix***
- **Nombre de Broches**
  - ***Suffisant pour le circuit***

Boîtiers - Contraintes

- **Nombre de broches**
  - $N = K.G^b$
    - **N est le nombre d'E/S**
    - **K est le nombre d'entrées/porte**
    - **G est le nombre de portes**
    - b **est une constante = 0.6**
  - ***Réduction du pitch. : 2,54mm => 0,5 mmm***
  - ***Broches en disposition matricielle plutôt que périphérique***
- **Dissipation thermique**
  - ***Boîtier céramique plus intéressant que boîtier plastique***
  - ***Mécanisme de rafraîchissement sur le boîtier***

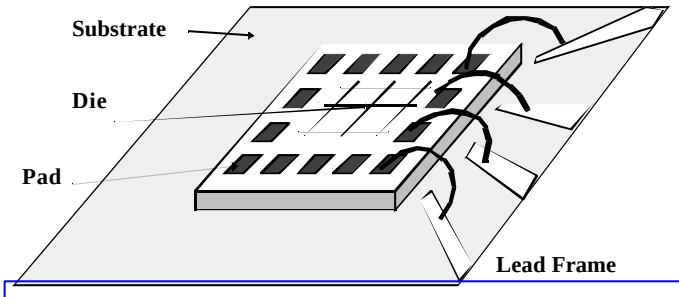
Boîtiers : 2 Niveaux d'Interconnexion



- **Connexion circuit – boîtier**
  - ***wire bonding, TAB, flip-chip mounting***
- **Connexion boîtier – carte**

Connexion Circuit - Boîtier

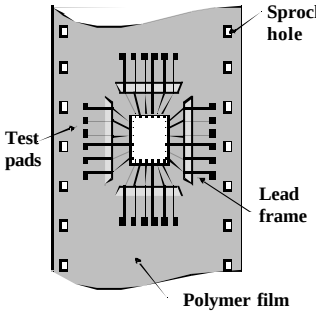
Wire Bonding



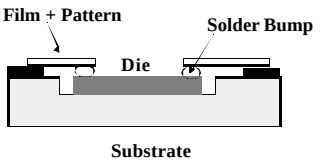
Les fils sont connectés les uns après les autres
Grand nombre de broches => emplacement difficile
Propriétés électriques dégradées (haute inductance, inductance mutuelle avec les broches voisines)
Evaluation difficile des éléments parasites

Tape-Automated Bonding (TAB)

Connexion du circuit à un metal imprimé sur un film polymère.
Connexion entre broches du circuit et film polymère par soudure
Connexion au corps du boîtier avec des connecteurs de pression



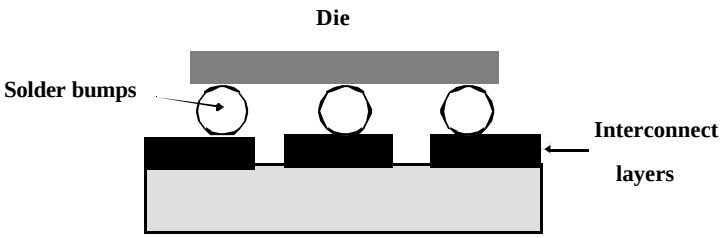
(a) Polymer Tape with imprinted wiring pattern.



(b) Die attachment using solder bumps.

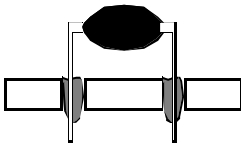
Toutes les connexions sont faites au même temps
Principe de l'impression => réduction du pitch plus facile
Diminution des fils => amélioration des propriétés électriques

Flip-Chip Bonding

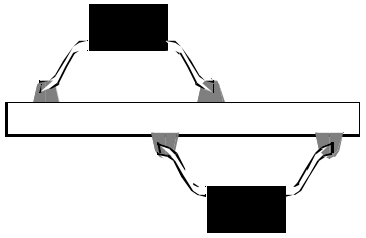


Connexion directe du circuit au substrat par soudure
Propriétés électriques supérieures
Les broches peuvent être placées n'importe où dans sa surface du chip
(et pas forcément sur la périphérie)
Meilleure gestion des problèmes de la distribution des horloges et de l'alimentation
(Cu et Au meilleurs que Al)

Connexion Boîtier-Carte



(a) Through-Hole Mounting

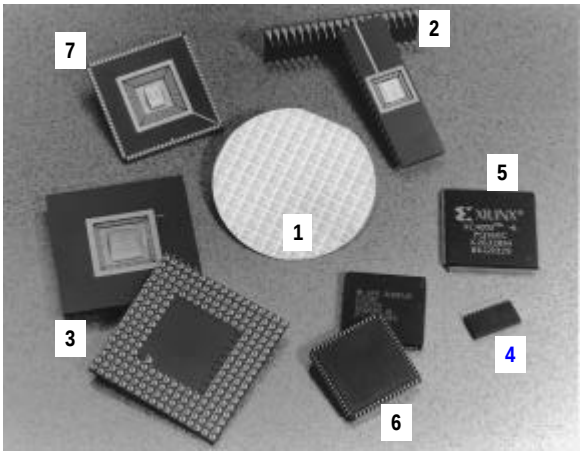


(b) Surface Mount

Montage « standard »
Adapté aux boîtiers DIP
Pb : diminution de la densité avec le nombre de broches
Pitch minimal : 2.54 mm

Montage « en surface »
Circuit connecté directement à la surface de la carte par soudure
Densité augmentée : les trous sont éliminés, le pitch est réduit, les circuits peuvent être placés des deux côtés de la carte

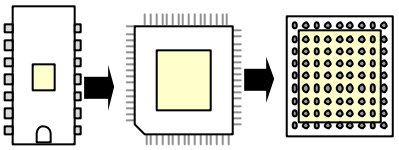
Les Boîtiers : Types



1 – Bare die
2 – DIP
3 – PGA
4 – Small-outline IC
5 – Quad flat pack
6 – PLCC
~~7 – Leadless carrier~~

4, 6 et 7 : montage ~~« en surface »~~

Evolution des Boîtiers

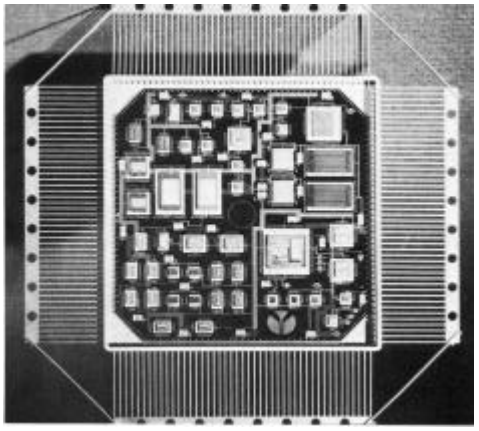


Boîtier	Pitch	Broches
DIP	2,54 mm	64
PGA	2,54 mm	> 300
Small-outline IC	1,27 mm	28
Quad flat pack	0,5 mm	240
PLCC	1,27 mm	124
Leadless carrier	0,75 mm	124
BGA	1 mm	>500

Connexion Circuit – Carte

- **Adaptation des techniques précédentes pour connexion sur un substrat.**
  - ***Les matériaux : époxy (comme les cartes) et silicium (adaptation « naturelle » des circuits)***
- **Avantages :**
  - ***Augmentation de la densité d'assemblage et des performances***
  - ***Réduction de la consommation (besoin moindre en amplificateurs)***
- **Inconvénients :**
  - ***Coût (justifié si besoins en performances et densité)***

MCM : Multi-Chip Modules



53 circuits et 40 composants discrets

Les Limites du PCB

- **Complexité d'intégration**
  - ***Dépendante du nombre de composants***
    - **Intérêt des SOC (system on a chip) et MCM**
- **Propagation de signaux HF sur les lignes**
  - ***Les lignes doivent être optimisées pour éviter interférences, réflexions électriques, rayonnement électromagnétique***
  - ***Standard d'interface spécifiques : différentiel (LVDS) , avec Vref (SSTL pour SDRAM), (GTL pour processor backplane)***
- **Testabilité**
  - ***Boundary scan test avec JTAG, points de test multiples, tests optiques***

Réalisation de PCB : Un métier à part entière !

Plan

- Les interfaces
  - *Les problèmes de l'assemblage*
  - *Protocoles de communication*
- Les boîtiers
  - *Contraintes et types*
  - *Assemblage PCB et MCM*
- Les mémoires
  - *Architectures et Utilisation*

Mémoires - Introduction

- Principe de base
  - *Un « plan » de cellules où sont stockées des informations*
  - *Un mécanisme d'accès aux informations dans les cellules*
- Contexte
  - *Produits haute performance :*
    - Demande pour de bandes passantes de plus en plus grandes
    - Besoin de cellules et de mécanismes d'accès performants
    - Développement de nouvelles technologies

Mémoires - Classification

- **Fonctionnalité**
  - ***Lecture/Lecture-écriture***
- **Mécanisme de mémorisation**
  - ***Statique/Dynamique***
- **Volatilité**
  - ***Volatile/Non Volatile***
- **Type d'accès**
  - ***Aléatoire/Non aléatoire***
- **Nombre de ports entrée/sortie**
  - ***Unique/Multiple***

Classification - Tableau Récapitulatif

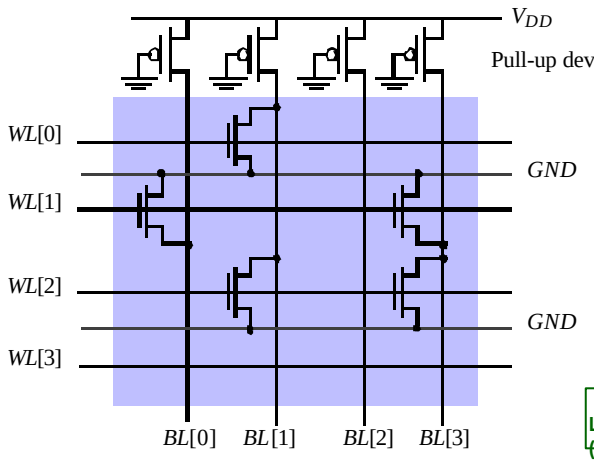
Volatiles — Non Volatiles

RWM		NVRWM	ROM
Random Access	**Non-Random Access**	EPROM E^2PROM FLASH	Mask-Programmed Programmable (PROM)
SRAM DRAM	FIFO LIFO Shift Register CAM		

Architecture – Le Cœur

- **ROM**
  - ***Structure générale***
    - **Matrice de transistors placés sur une grille word-line/bit-line**
      - ***Programmation d'une valeur par validation/inhibition sélective (faite par altérations au niveau masque)***
  - ***Eléments de base***
    - **Structures NOR et NAND**
    - **Structure pré-chargée**

MOS NOR ROM



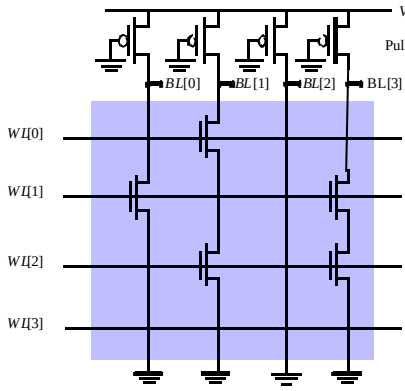
Programmation par masque : présence de transistor = « 0 » stocké

Condition normale : BL « high » (grâce au pull up) et WL « low » Lecture : WL « high »

Une grande partie de la surface est dédiée au contact bit-line et à la connexion à la masse

Les transistors P doivent être petits (résistivité)

MOS NAND ROM



Programmation par masque : présence de transistor = « 1 » stocké

Condition normale : BL « high » (grâce au pull up) et WL « high »
Lecture : WL « low »

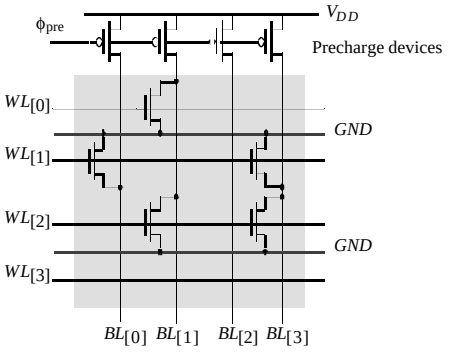
Avantage : La cellule de base consiste en un seul transistor et il n'y a aucun contact avec les sources d'alimentation.
Inconvénient : Consommation statique

All word lines high by default with exception of selected row

NOR ROM – Structure Pré-chargée

Opération en deux temps :
1 - Pré-charge de BL à travers P, puis coupure.
2 - Après, fonctionnement équivalent à NOR.

Avantages : Consommation moindre et pas de conflit (pas de résistance pull up)

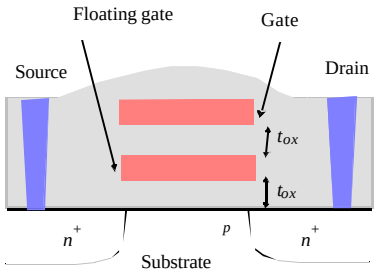


PMOS precharge device can be made as large as necessary, but clock driver becomes harder to design.

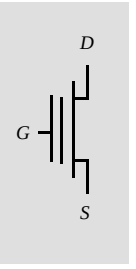
Architecture – Le Cœur

- **NVRWM (Mémoires R/W non volatiles)**
  - ***Mêmes structures que ROM, mais***
    - **Les transistors permettent un changement électrique de leur seuil**
    - **La programmation est faite électriquement**
  - ***Effacement des anciennes valeurs pour la re-programmation***
    - **Plusieurs méthodes d'effacement**

Grille Flottante : PROM et EPROM



(a) Device cross-section

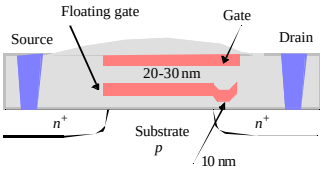


(b) Schematic symbol

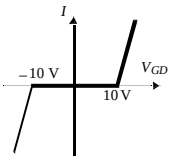
PROM :
Programmation par l'utilisateur
La tension de seuil du transistor est modifiée par l'application d'une tension haute sur la grille extérieure

EPROM :
Effacement « off system » de la programmation par UV, en rendant l'oxyde légèrement conducteur (retour des électrons)
Nombre d'effacements/programmations limité

EEPROM

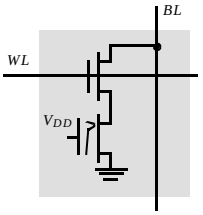


(a) Flotox transistor



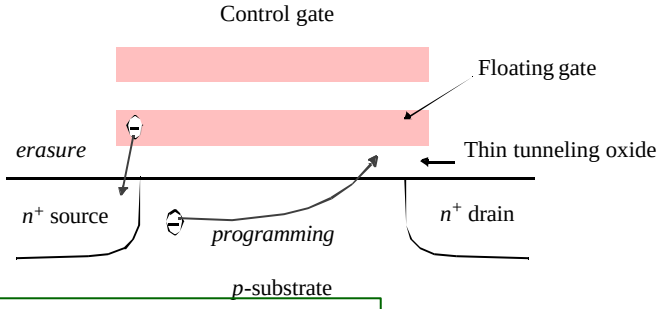
(b) Fowler-Nordheim I-V characteristic

EEPROM : moins denses que EPROM
L'effacement est électrique, par application d'une tension inverse



(c) EEPROM cell during a read operation

Flash EEPROM



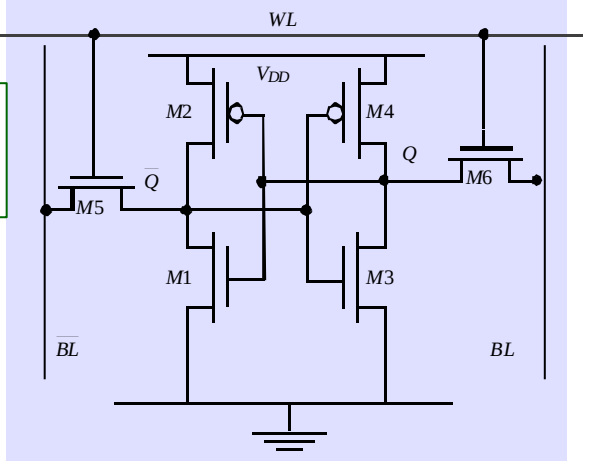
Union de propriétés EPROM et EEPROM :
Programmation type PROM/EPROM
Effacement type EEPROM
Rapidité : les programmations/effacements sont faits pour un ensemble de cellules

Architecture – Le Cœur

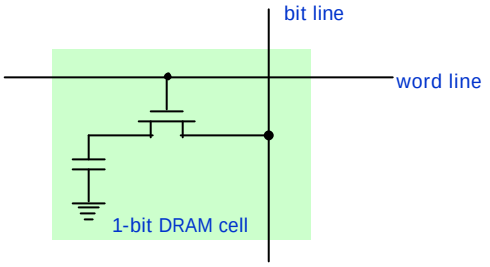
- **RWM (RAM – Mémoires R/W Volatiles)**
  - ***Accès Aléatoire***
    - **Statiques (SRAM)**
    - **Dynamiques (DRAM)**
  - ***Accès Non Aléatoire***
    - **FIFO**

SRAM : Cellule à 6-transistors

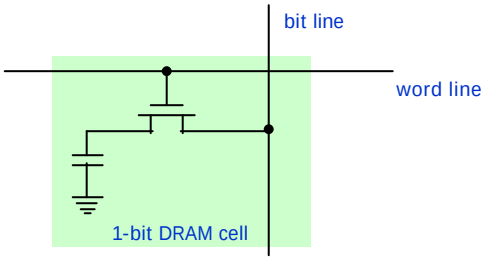
Lecture :
Les BL alimentent un sense amplifier
Ecriture :
~~Les BL sortent d'un buffer~~



DRAM : Cellule à 1 Transistor

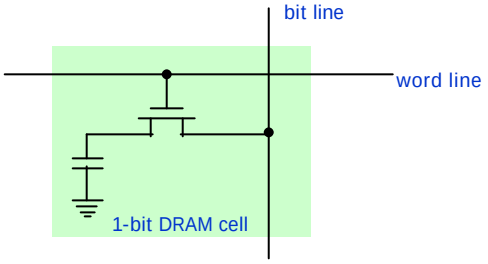


DRAM : Ecriture de “1”



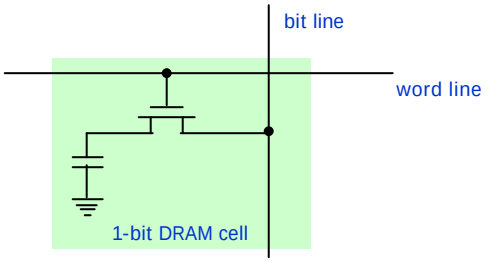
Bit line est mis à “1” et la capacitance se charge à travers le transistor.

DRAM : Ecriture de “0”



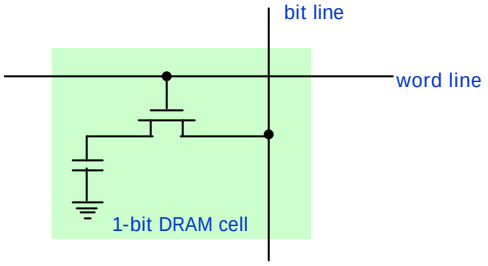
Bit line est mis à “0” et la capacitance se décharge à travers le transistor.

DRAM : Lecture Destructive



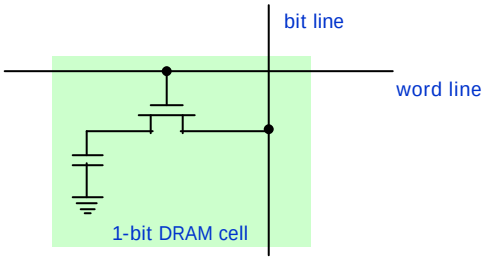
Lecture: bit line est préchargé à une tension entre “1” et “0”, word line est mis à “1”. En fonction de la charge dans la capacité, la valeur de bit line monte ou descend.
Un sense amplifier détecte ce petit changement et positionne la valeur “0” ou “1”.

DRAM : Restauration



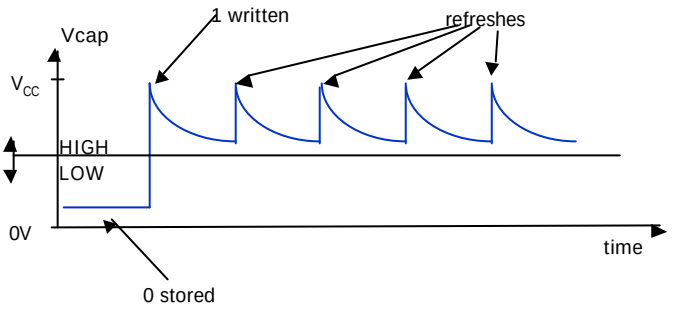
L'opération de lecture décharge la capacité, donc elle doit être suivie d'une opération d'écriture pour restaurer la de la même valeur sur position lue.

DRAM : Memories Qui Oublient …



Problème :
Seule la valeur "0" peut être maintenue indéfiniment.
Si un "1" est chargé, la capacité dissipera jusqu'à la perte de la donnée.

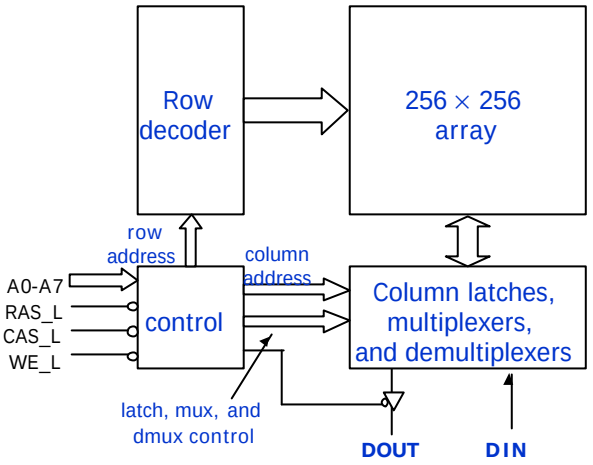
DRAM : Rafraîchissement Périodique



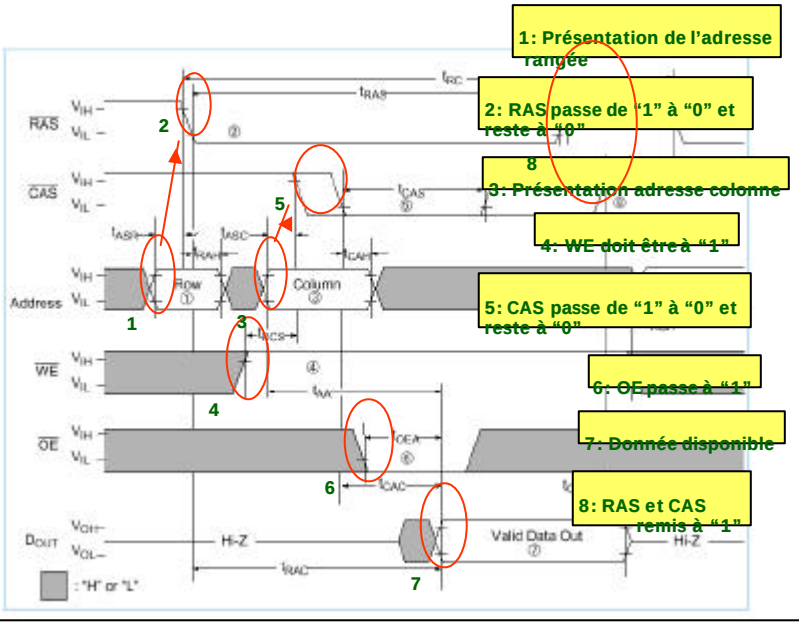
Interface RAM/DRAM

- **DRAM**
  - ***Multiplexage des adresses (MSB, LSB)***
    - **Réduction du nombre de broches**
    - **Signaux supplémentaires (RAS: row-access strobe, CAS:column-access strobe)**
- **SRAM**
  - ***Pas de multiplexage***
    - **Nombre de broches plus important**
    - **Aucun signal externe est nécessaire**

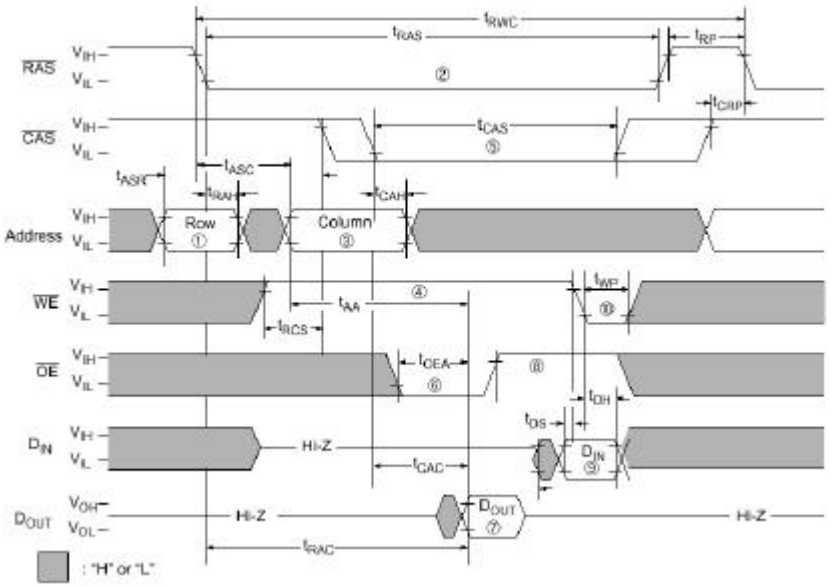
64K ´ 1 DRAM : Structure Interne



Cycle de Lecture DRAM Asynchrone



Cycle d'Ecriture DRAM Asynchrone



Sense Amplifiers

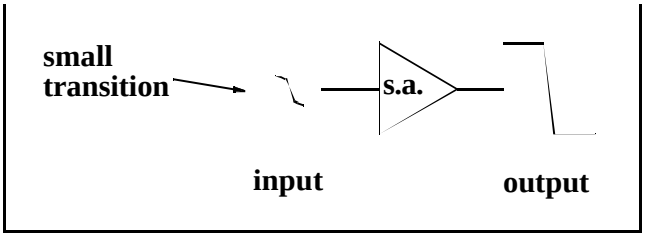
$$t_p = \frac{C \cdot \Delta V}{I_{av}}$$

make ΔV as small as possible

large

small

Idea: Use Sense Amplifer

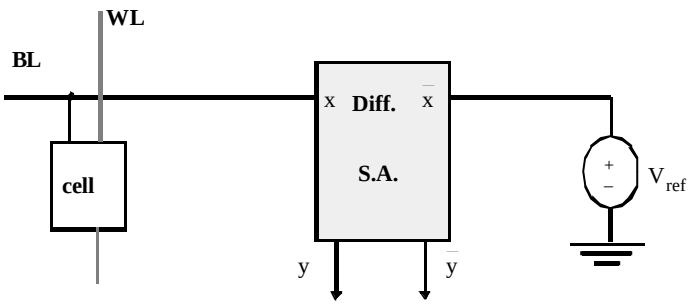


Sense Amplifiers

- **Amplification**
  - ***Essentielle en 1T-DRAM***
- **Augmentation de la performance**
  - ***Compensation du fan-out limité des cellules dans le cœur de la mémoire (accélération de la transition bit-line)***
- **Réduction de la consommation**
  - ***Réduction du swing sur les bit-lines (élimination de la consommation pour chargement/déchargement)***
- **Restauration du signal**
  - ***Mise à pleine échelle des bit-lines après détection (la lecture et le refresh sont liés en 1T-DRAM)***

Sense Amplifiers

Single-to-Differential Conversion

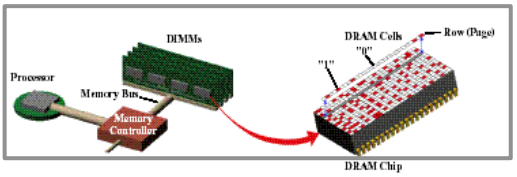


How to make good V_{ref}?

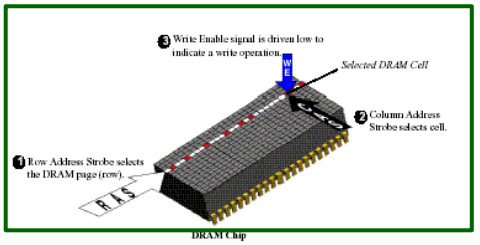
Buffers d'Entrée/Sortie

- **Augmentation de la taille de la mémoire**
  - ***Augmentation de la longueur de word-lines/bit-lines***
    - **Dégradation en performance (diminuée avec partitionnement de la mémoire en blocs)**
      - ***Solution : Buffers***
- **Grande partie de la surface des circuits périphériques dédiée aux drivers**
  - ***Buffers d'adresse + buffers d'entrée/sortie***

Mémoires DRAM



Les DRAMs sont installées sur
-des SIMMs* de 30/72 broches
-des DIMMs* de 144/168 broches



* SIMMs/DIMMs (single/dual inline memory modules)

Mémoires DRAM

- **Premières DRAM : plusieurs cycles pour chaque accès**
  - ***Adresse ligne, adresse colonne, transfert de la donnée sur le bus***
    - **Principale source de retard : transfert des adresses**
  - ***Améliorations***
    - **Mémoires pourvues d'accès en mode rafale (Burst Mode Access)**
      - ***Mémoires asynchrones FPM (Fast Page Mode) et EDO (Extended Data Out)***
      - ***Mémoires SDRAM (Synchronous DRAM)***
      - ***Accès consécutif de 4 sections de 64 bits***
        - Temps d'accès total: x-y-y-y (plutôt que 4 fois x)

Amélioration de l'Accès

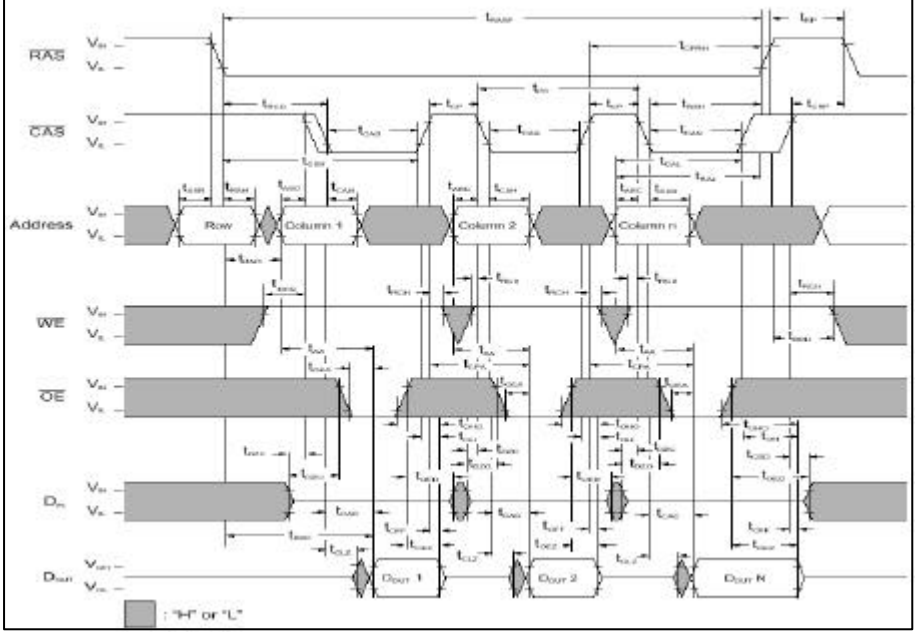
- **Chaque lecture dans une DRAM se traduit par le transfert d'une rangée complète de bits ou word line du coeur de la DRAM vers des sense amplifiers.**
- **Une interface DRAM asynchrone classique sélectionne un certain nombre de ces bits.**
- **Tous les autres bits ont été extraits "inutilement".**

- **Idée de base : profiter des étapes déjà réalisées**

Fast Page Mode DRAMs

- **Dans une DRAM avec Fast Page Mode, une page est définie comme toutes les adresses de la mémoire ayant la même adresse rangée.**
- **Pour une lecture en Fast Page Mmode , toutes les étapes d'un cycle de lecture standard sont effectuées.**
- **Les signaux OE et CAS sont activés (mis à "1"), mais RAS reste à "0" .**
- **Les étapes correspondant à nouvelle adresse colonne, CAS et OE sont effectuées pour chaque position à lire.**

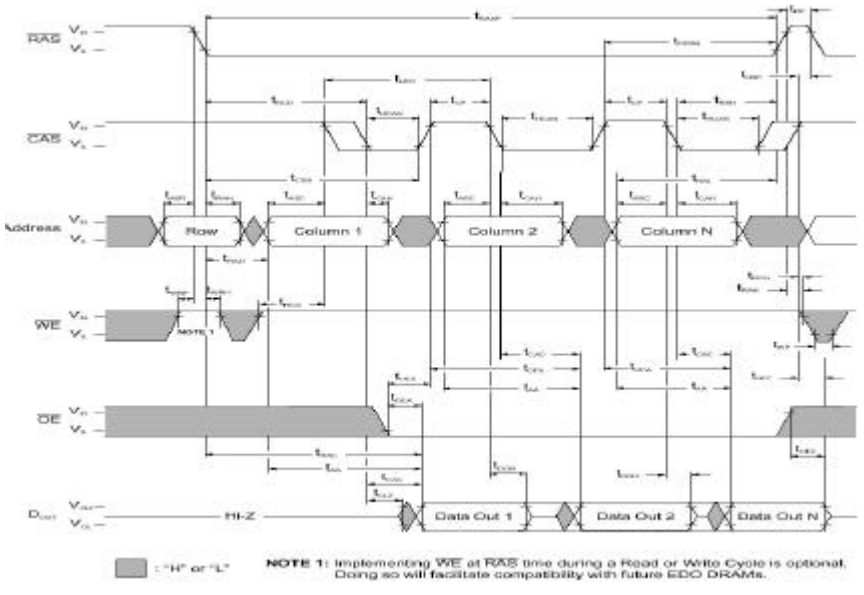
Lecture en Fast Page Mode Read sur DRAM Asynchrone



Enhanced Data Output RAMs (EDO-RAM)

- Le processus de lecture dans une **EDO-RAM** est semblable à celui de la FPM DRAM, mais les drivers de sortie ne sont pas désactivés lors que CAS passe à “1”.
- Cela permet que la donnée relative au cycle de lecture actuel soit présente sur la sortie alors qu’un nouveau cycle commence : les cycles de lecture sont ainsi rendus plus rapides.

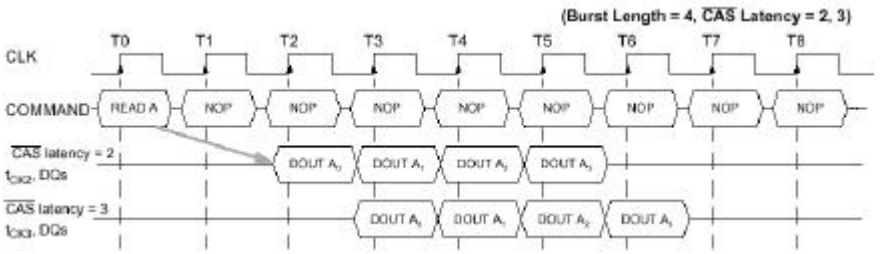
Cycle de Lecture dans une EDO DRAM



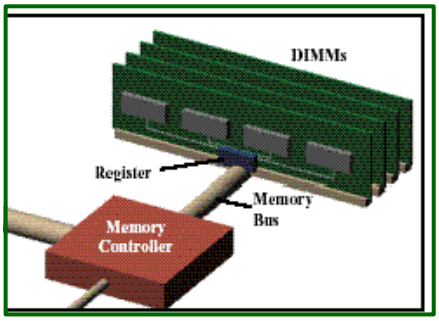
Synchronous DRAMs (SDRAM)

- **Les mémoires Synchronous DRAM (SDRAM) possèdent une entrée d'horloge.**
- **SDRAMs fonctionnent comme FPM et EDO DRAMs, mais les données sortent sur les fronts montants/descendants de l'horloge (c.a.d. ne sont pas commandées par CAS).**
  - **L'horloge interne des SDRAM est d'un ordre de grandeur plus rapide que le temps d'accès pour accès individuels.**
  - **La quantité d'éléments sortants (la taille du burst) est programmable jusque la taille maximale de la rangée.**
- **Bancs de mémoire multiples**
  - **Découpage de la mémoire en 2-4 bancs**
  - **Augmentation de la vitesse aperçue**
- **Adaptation d'impédance des modules (Bufferisation)**

SDRAM Burst Read Cycle



Bufferisation des Modules



Les registres isolent les bus d'adresse/contrôle du reste du système

Le système ne voit pas les charges des plusieurs mémoires dans chaque module (mais seulement celle du registre)

Moins de charge aperçue = possibilité de mettre plus de composants mémoire dans le système

DIMMs bufferisés versus DIMMs non-bufferisés
Non-bufférisés : Plus rapides (pas de logique de bufferisation)
Bufférisés : Capacité de pilotage d'un plus grand nombre de mémoires

DDR SDRAM

- **Les mémoires Double Data Rate (DDR) SDRAM sont des SDRAM qui permettent des transferts sur les deux fronts de l'horloge.**
  - ***La bande passante effective d'une DDR SDRAM est ainsi le double de la bande passante d'une SDRAM pour la même fréquence d'horloge.***
- **Circuits de synchronisation optimisés**
- **Technologie de signalisation différente (SSTL)**

DDR SDRAM – Tableau Récapitulatif

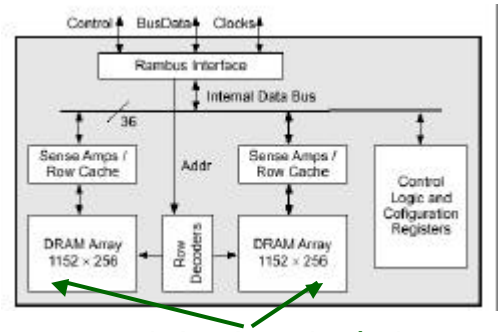
Composant	Type	Module	Bus Speed	Bande Passante
DDR200	PC1600	DDR-I	100 MHz	1.6 GB/s
DDR266	PC2100	DDR-I	133 MHz	2.1 GB/s
DDR333	PC2700	DDR-I	166 MHz	2.7 GB/s
DDR400	PC3200	DDR-II	200 MHz	3.2 GB/s
DDR533	PC4300	DDR-II	266 MHz	4.3 GB/s
DDR667	PC5300	DDR-II	333 MHz	5.3 GB/s
DDR800	PC6400	DDR-II	400 MHz	6.4 GB/s

Rambus

- **Conception système**
- **Possibilité d'opération à hautes fréquences**
- **Trois éléments de base**
  - ***Rambus DRAMs (RDRAMs)***
  - ***Asics Rambus***
  - ***Canaux d'interconnexion Rambus***

Rambus DRAM (RDRAM)

- **Les mémoires Rambus DRAM sont synchrones et permettent le transfert sur les deux fronts de l'horloge. Les bus de commande ligne et colonne sont séparés.**

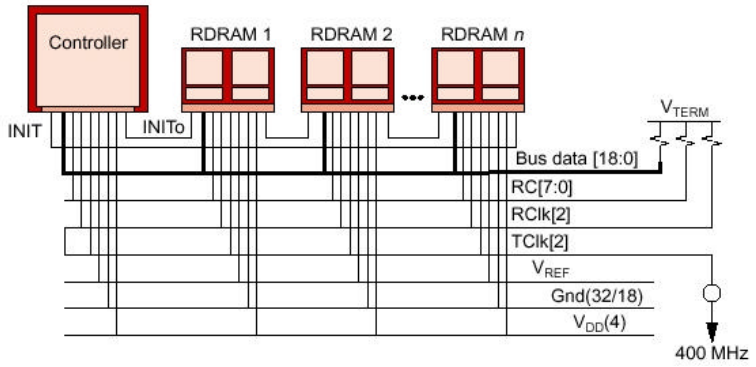


Plusieurs bancs de mémoire

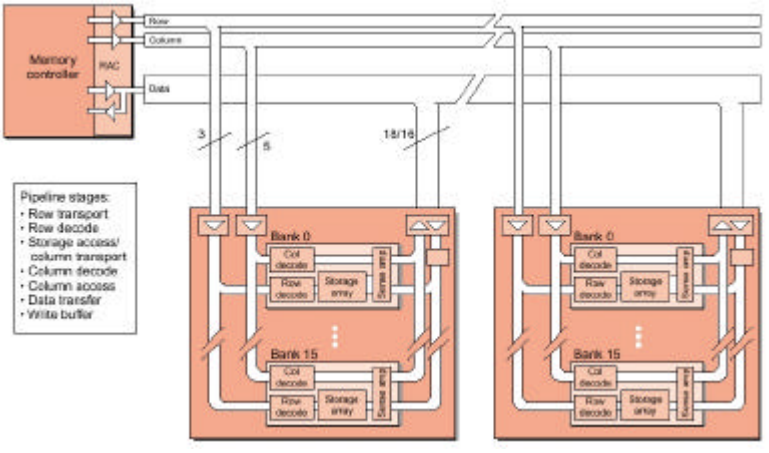
Rambus

- **Les interfaces**
  - ***8 bits (bus contrôle)***
  - ***18 bits (bus données)***
- **Protocole paquet**
  - ***1 paquet transferé en 8 cycles d'horloge***
- **Jusque 32 composants sur un canal mémoire**
  - ***Possibilité d'ajout de composants avec des répéteurs* (repeater chip)**
- **Buffer interne 128-bits**

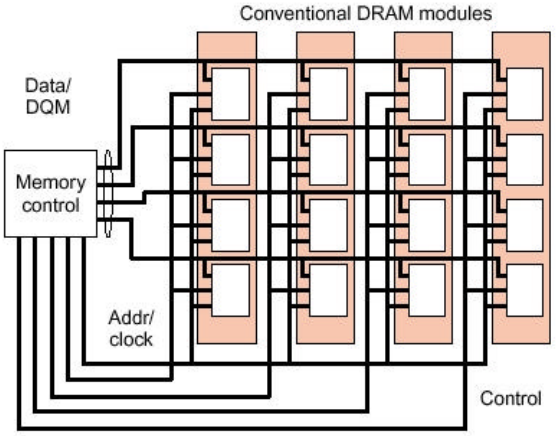
RDRAM Memory Systems



RDRAM : Organisation Interne



SDRAM Memory Systems



Bancs RDRAM ´ Bancs SDRAM

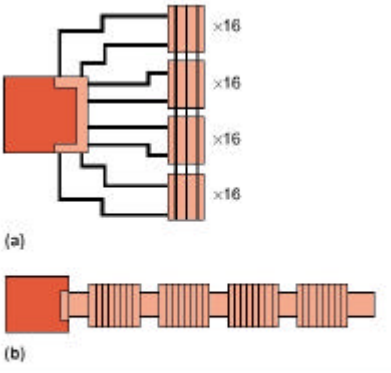
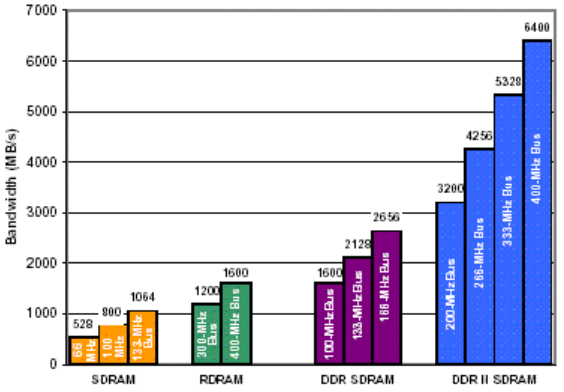


Figure 8. Bank counts: a 32-Mbyte, 64M SDRAM system with four large banks (a) versus a 32-Mbyte, 64M Direct RDRAM system with 32 small banks (b).

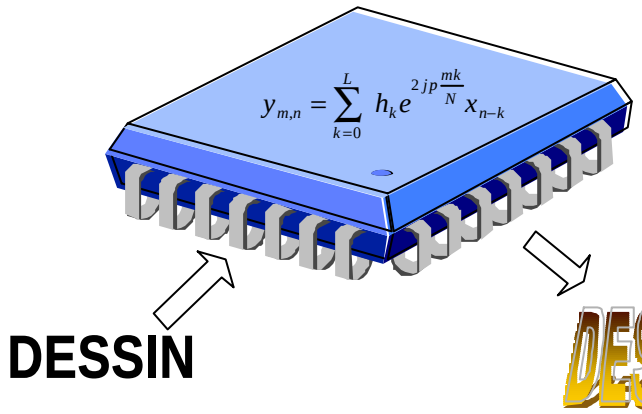
Bandes Passantes



Bibliographie

- **Ces transparents ont été réalisés d'après les sources suivantes**
  - ***Digital Integrated Circuits : A Design Perspective, Jan M. Rabaye, © Copyright 1996 UCB. (livre + jeu de transparents)***
  - ***Technology Brief, « Memory Technology Evolution : An Overview of System Memory Technologies », Compact Computer Corporation, ISSG Technology Communications.***
- **Pour plus d'informations sur les mémoires, voir également :**
  - ***http://www.ece.utexas.edu/~kettler/pcarch.html***
  - ***www.tomshardware.com/mainboard/00q1/000315/rambus-02.html***

Blocs génériques



DESSIN

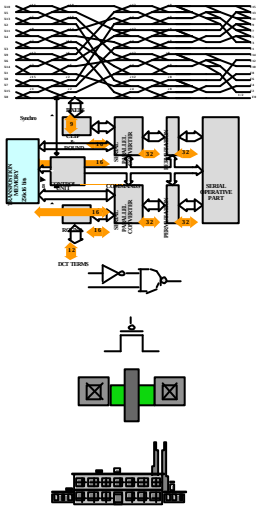
Plan

- Rappels sur la conception ASIC
- Bibliothèques de cellules
- Générateurs de RAM, ROM
- DRAM Embarquées
- Génération de Chemins de Données

Plan

- Rappels sur la conception ASIC
- **Bibliothèques de cellules**
- **Générateurs de RAM, ROM**
- **DRAM Embarquées**
- **Génération de Chemins de Données**

Conception descendante



Algorithme

Architecture

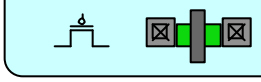
Logique

Transistors

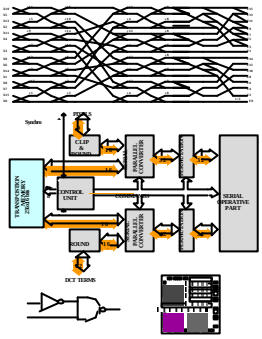
Masques

Fab.

Circuits pré-caractérisés



Bibliothèques de cellules
STm Atmel, Tsmc



L'algorithme sur le silicium

- **Automatisation**

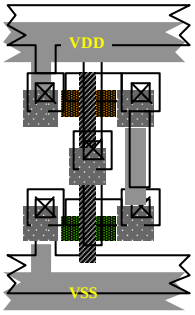
Mais :

- **Fréquences de fonctionnement limitées**

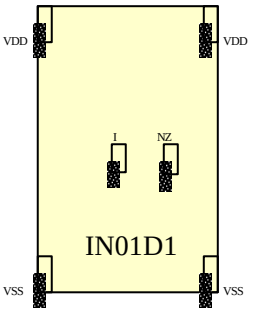
STm Atmel, Tsmc

Notion de cellule standard

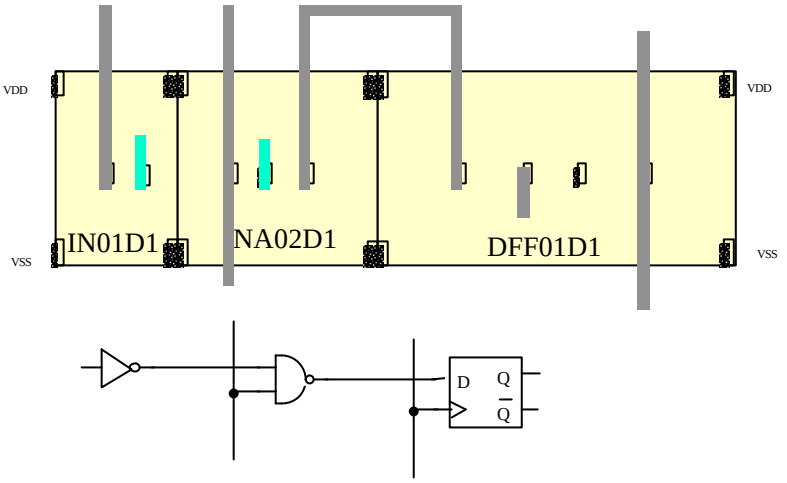
25 objets



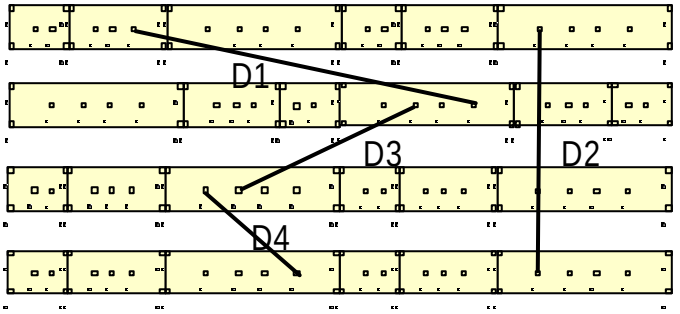
1 objet



Assemblage des cellules

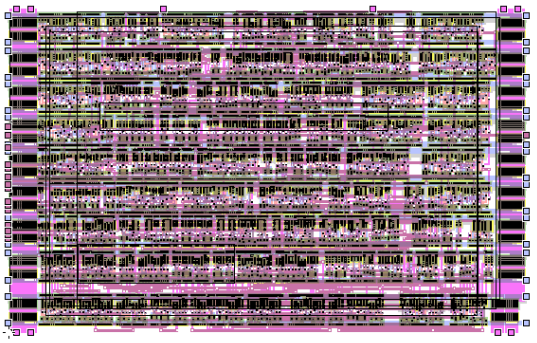


Algorithmes de placement

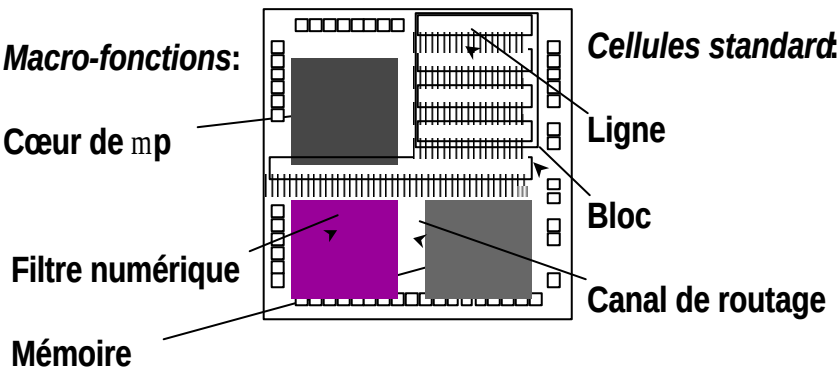


- Minimiser D1+D2+D3+D4+…..
- Heuristiques, recuit simulé
- Critères plus ou moins complexes

Un bloc de cellules standards



Organisation d'un circuit



Exemple de circuit

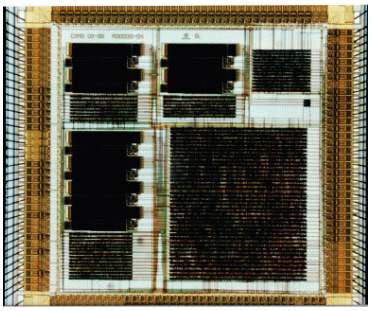


Photo : http://ais.gmd.de/ROB/Graphics/ChipPhoto.gif

Techno : 0.7 µm CMOS.
Fondeur : Mietec Alcatel.
Surface : 6.8 x 6.9 mm^2
Complexité :

- 15k portes
- 1kOctet Mémoire

Plan

- **Rappels sur la conception ASIC**
- **Bibliothèques de cellules**
- **Générateurs de RAM, ROM**
- **DRAM Embarquées**
- **Génération de Chemins de Données**

Vision du concepteur

- **A priori, le concepteur ne connaît pas le contenu de la bibliothèque précaractérisée.**
- **La transcription d'une description (Shéma, VHDL, Verilog…) en assemblage de cellules est prise en charge par l'outil de synthèse.**
- **Mais….**

Vision du concepteur

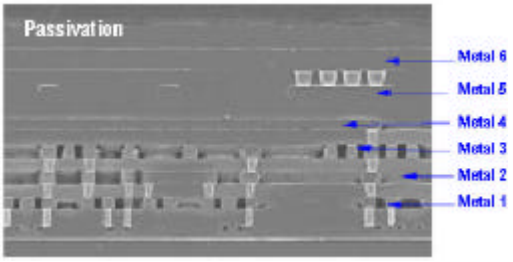
- **En phase de vérification :**
  - ***Les portes « physiques » utilisées sont détaillées dans les chemins critiques***
  - ***Le concepteur chevronné peux évaluer le bon choix de portes réalisé par le synthétiseur***
- **En phase conception :**
  - ***Le concepteur peut restreindre et guider le choix de portes par le synthétiseur***

Caractéristiques des bibliothèques

- Complexité (nombre de portes disponibles)
- Complexité et précision du modèle temporel des portes
- Qualité du modèle de consommation
- Adaptation aux différents outils de CAO du marché (Cadence, Mentor, Avanti….)

Interpréter un document Marketing

ST introduces new 0.18 µm standard cell library



The CB65000 standard cell series uses a high performance, low-voltage, 0.18µm drawn, six metal levels, high density and high speed HCMOS8D process.

- Une bibliothèque est associée à une technologie…
- Pas ou peu de bibliothèques virtuelles…

Interpréter un document Marketing

- Densité d'intégration
  - Après placement et routage final

With an average routed gate density of 85,000 gates/mm², the CB65000 family allows the integration of up to 30 million equivalent gates and is ideal for high-complexity or high-performance devices for computer, telecommunication and consumer products.

- Limitation de la taille des puces

Interpréter un document Marketing

With a gate delay of 35 ps with High Performance transistor and 60 ps with Low Leakage transistor (for a 2-input NAND gate at fan-out 1), the library meets the most demanding speed requirements in telecommunication and computer application designs today

- On ne peut pas optimiser à la fois vitesse et consommation.
- En réalité il y a deux bibliothèques basées sur des compromis différents

Interpréter un document Marketing

- Consommation moyenne d'une porte

Optimized for 1.8V operation, the library features a power consumption of less than 35nW/Gate/MHz (High Performance;fan-out=1) and 25 nW/Gate/MHz (Low Leakage; fan-out=1) at 1.8V.

- Bien que l'on dispose d'une plage de tensions d'alimentation utilisables, la bibliothèque est optimisée pour une tension nominale donnée.

Interpréter un document Marketing

- Il faut communiquer avec l'extérieur du circuit

The I/O buffers can be fully configured for both 1.8V and 3.3V interface options, with several high speed buffer types available. These include: low voltage differential (LVDS) I/Os, PCI, AGP, USB, LVTTL,LVCMOS and SSTL.

- La bibliothèque d'Entrée/Sortie est dissociée. de la bibliothèque de cellules Standards.
- Le concepteur intervient directement dans le choix des cellules d'Entrée/Sortie

Bibliothèques d'entrées/sorties

- Les cellules E/S sont caractérisées en vitesse et consommation au même titre que les cellules logiques
- Les modèles de vitesse et de consommation sont intégrés par les outils de vérification (simulateurs, analyse temporelle…)

Complexité des bibliothèques

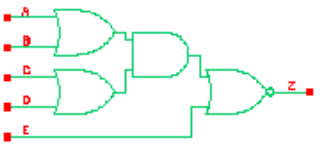
- Exemple de bibliothèque :
  - *293 Cellules combinatoires (nand, nor..)*
  - *88 Cellules séquentielles (bascules D..)*
  - *Une cellule pour l'arbre d'horloge*
- La richesse permet aux synthétiseurs d'optimiser aux mieux les critères de complexité, de vitesse ou de consommation

Complexité : gestion des charges

- 4 Instances du AND à 4 entrées
  - *AN4, AN4P, AN4X3,AN4X4*
- La seule différence provient de l'inverseur de sortie de taille croissante
- Le synthétiseur choisira la version la plus adaptée au contexte d'utilisation…

Complexité : portes CMOS complexes

- La logique CMOS permet de réaliser de façon efficace n'importe quelle fonction de type Somme(Produits..)
- Le synthétiseur peux exploiter de manière efficace l'existence de telles fonctions.
- Exemple : AO13X05



Complexité : variantes d'une bascule

- **Le synthétiseur ne choisi que la bascule strictement nécessaire...**
- **Bascule D**
  - ***Avec ou sans « reset »***
  - ***Avec ou sans « set »***
  - ***Avec ou sans sortie « Q »***
  - ***Avec ou sans sortie « QB »***
- **12 cas possibles….**

Plan

- **Rappels sur la conception ASIC**
- **Bibliothèques de cellules**
- **Générateurs de RAM, ROM**
- **DRAM Embarquées**
- **Génération de Chemins de Données**

Rôle du générateur de RAM, ROM

- **Fournir un bloc RAM ou ROM paramétrable:**
  - ***nombre de mots***
  - ***taille des mots***
  - ***nombre de bus***
  - ***type d'accès***
- **Fournir un modèle simulable pour la vérification**
- **Fournir un modèle physique**
- **Fournir des « spécifications » (vitesse, conso…)**

Alternatives

- **3 solutions possibles**
  - ***Bloc paramétrable synthétisable***

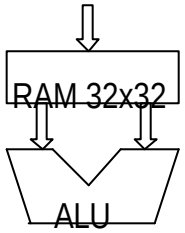
    (description VHDL, Verilog, générateur de netlist….)
  - ***Bloc « cousu-main » paramétrable (dessin dans la technologie cible)***
  - ***Bloc « cousu-main » paramétrable « indépendant techno » (dessin dans une technologie virtuelle)***

Bloc paramétrable synthétisable

- Utiliser une bibliothèque précaractérisée (latchs, D-flipflops)
- Générer une netlist contenant le réseau de bacules, les décodeurs d'adresse, les contrôleurs
- Dans de nombreux cas, le concepteur peut coder lui même le générateur en VHDL ou Verilog (mais peut être utile pour diminuer le temps de conception (FIFO, bancs multiports)
- Pas efficace :
  - *Taille mémoires*
  - *Vitesse de fonctionnement ??*
- Vitesse déterminée par le flot de conception classique

Bloc paramétrable synthétisable

- Exemple : Synopsys DesignWare
- DW_2r_w_a_lat :RAM asynchrone triple ports
- Basée sur des latchs (Max 256 mots de 256 bits)
- Pratique pour faire un banc de registres…



Bloc paramétrable cousu main

- Proposés par le « fondeur »
- Assemblage de cellules dessinées et optimisées par les concepteurs spécialistes du « bas niveau »
- Programme ad-hoc (hors flot de conception) fourni pour créer l'assemblage des cellules
- Génération automatique de diverses vues :
  - Simulation
  - Fantôme (dessin de masques)
  - Data-Sheet

Bloc paramétrable cousu main

- **Difficulté : fournir un modèle de temporel précis**
  - ***Comportements électriques complexes dans une RAM ou une ROM***
  - ***Il n'y a pas que de la logique CMOS (précharge, amplis différentiels…)***
- **Solution 1 (pour le concepteur du générateur): Générer tous les cas possibles et se baser sur des simulations électriques…**
- **Solution 2: Génerer des cas pathologiques et se baser sur un modèle analytique des temps de propagation compatibles avec ces cas pathologiques..**

Bloc paramétrable cousu main

- **Le chemin critique dans une ROM**

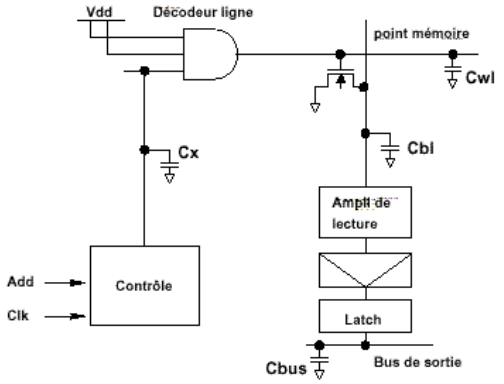


FIG. 4.1 – Schéma simplifié d'un chemin critique pour une ROM

Circuit de test de générateurs de ROM

- **Quelques instances de ROM sortant d'un générateur**

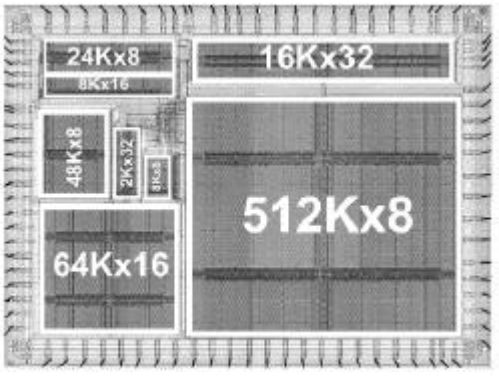


FIG. 2.13 – Testchip avec 8 instances

Remarques:

1/ Surfaces très différentes entre 2kx32 et 8kx8

2/ Facteurs de forme très divers

- **Source : Thèse de Mr Arnaud Turier (2001). (www.lip6.fr/reports)**

Caractéristiques des RAM intégrées

- **Exemple : Techno 0,25µm ATMEL**
  - ***« General Synchronous RAM »***
- **Utilisation : SRAM SYNCHRONE de petite taille / grande vitesse**
- **Taille : de 128 à 144kbits**
- **Nombre de mots : de 32 à 8k**
- **Taille des mots : de 4 à 36 bits**

Configuration	1Kx8	2Kx16	4Kx32
Densité (Kbits/mm^2)	51	58	62
Fréquence(MHz)	314	250	175
Consommation dynamique (mW/MHz)	0,17	0,36	0,73

Caractéristiques des RAM intégrées

- **Exemple : Techno 0,25µm ATMEL (ACT25)**
  - ***« High Range Synchronous RAM »***
- **Utilisation : SRAM SYNCHRONE de grande taille**
- **Taille : de 16K à 2,25M bits**
- **Nombre de mots : de 2 à 32k**
- **Taille des mots : de 8 à 72 bits**

Configuration	8Kx8	16Kx16	32Kx32
Densité (Kbits/mm^2)	80	84	87
Fréquence(MHz)	186	120	71
Consommation dynamique (mW/MHz)	0,29	0,55	1,22

Caractéristiques des RAM intégrées

- **Exemple : Techno 0,25mm ATMEL (ACT25)**
  - ***« General Asynchronous RAM »***
- **Utilisation : SRAM ASYNCHRONE rapide**
- **Taille : de 128 à 128K bits**
- **Nombre de mots : de 16 à 4k**
- **Taille des mots : de 8 à 36 bits**

Configuration	1Kx8	2Kx16	4Kx32
Densité (Kbits/mm^2)	40	40	50
Fréquence(MHz)	510	502	326
Consommation dynamique (mW/MHz)	0,24	0,38	0,63

Caractéristiques des ROM intégrées

- **Exemple : Techno 0,25mm ATMEL (ACT25)**
  - ***« General Synchronous ROM »***
- **Utilisation : ROM SYNCHRONE rapide**
- **Taille : de 256 à 256K bits**
- **Nombre de mots : de 64 à 8k**
- **Taille des mots : de 4 à 64 bits**

Configuration	2Kx8	4Kx16	8Kx32
Densité (Kbits/mm^2)	400	568	669
Fréquence(MHz)	510	502	326
Consommation dynamique (mW/MHz)	0,24	0,38	0,63

Caractéristiques des mémoires intégrées

- **RAM synchrones**
  - ***Facilité d'utilisation, pas de génération de signaux spéciaux non conformes aux techniques de la logique synchrone***
- **RAM asynchrones**
  - ***Grande vitesse, mais signaux de contrôles difficile à générer.***
- **Entrées et sorties généralement séparées**
- **Utilisation systématique de RAM double port**
  - ***Communication entre processus (FIFO…)***

Bloc paramétrable cousu main indépendant de la technologie

- **Proposés par des vendeurs d 'outils de CAO**
- **Assemblage de cellules dessinées et optimisées par les concepteurs spécialistes du « bas niveau » dans une technologie « virtuelle »**
- **Programme ad-hoc (hors flot de conception) fourni pour créer l'assemblage des cellules**
- **Programmes ad-hoc pour caractériser les mémoires dans la technologie fournie par le client**

Bloc paramétrable cousu main indépendant de la technologie

- **Capacité d'adaptation rapide à différentes technologies**
- **Mais…**
  - **Pas résultat non optimal pour une technologie donnée.**
  - **Mais…**
    - Le nombre de fondeurs diminuant, il y a de moins en moins de technologies exotiques…

Plan

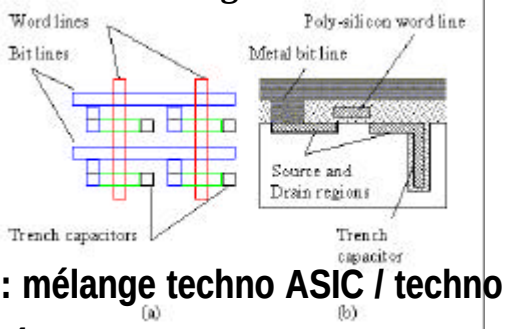
- **Rappels sur la conception ASIC**
- **Bibliothèques de cellules**
- **Générateurs de RAM, ROM**
- **DRAM Embarquées**
- **Génération de Chemins de Données**

Contexte

- DRAM, seule solution pour le stockage de masse
- Technologie DRAM spécifique et différente de la technologie CMOS ASIC
- Les applications VIDEO grand public poussent à la réalisation de DRAM embarquées
- Premières offres concrètes en 1999…

Technologie DRAM spécifique

- point mémoire à 1 transistor
- Capacité de stockage en structure verticale



- Solution : mélange techno ASIC / techno DRAM
- Coût élevé
- Blocs durs (pas de générateurs)

Technologie ASIC CMOS

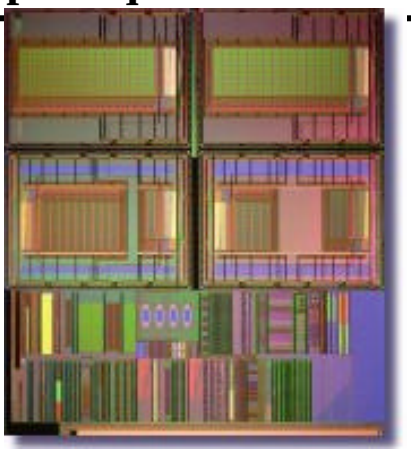
- point mémoire à 3 transistors
- Pas de masque « capacitif » particulier
- 1 grille de transistor pour la capa
- 1 transistor de lecture
- 1 transistor d'écriture
- Pas de lecture destructive

- **Peu dense**
- **Faible coût, générateurs possibles**

DRAM spécifique : IBM

- Technologie 0,15µm SA-27E
- Blocs DRAM : par macro de 1 à 16 Mbit
- Surface :(3,4 + (n-1)*1,1) mm^2 pour n Mbit
- Mode aléatoire (cycle/accès) : 20/13 ns
- Mode Page (cycle/accès) : 6,6/6,6 ns
- Largeur de mots : 256 bits
- Tailles max typiques : 128 Mb

DRAM spécifique : IBM



http://www-3.ibm.com/chips/products/asics/

DRAM en techno ASIC

- **Problématique identique à celle des générateurs de SRAM ou ROM**
- **Des offres existent :**
  - ***Atmos Embedded Memory Solution***
    http://www.atmoscorp.com/index.php
  - ***Dolphin Integration***
    http://www.dolphin.fr

DRAM en techno ASIC

- Caractéristiques en vitesse proches des « vraies » DRAM
- Taux d'intégration plus faible
- Des méthodes d'accès plus simples que la DRAM standard

DRAM en techno ASIC

- Exemple : Dolphin Integration CMOS-DRAM
- Fonctionne comme une mémoire SRAM synchrone
- Pas de bus d'adresse multiplexé
- 1 bus d'adresse
- 1 bus DIN, un bus DOUT
- CLK, READ, WRITE, REFRESH

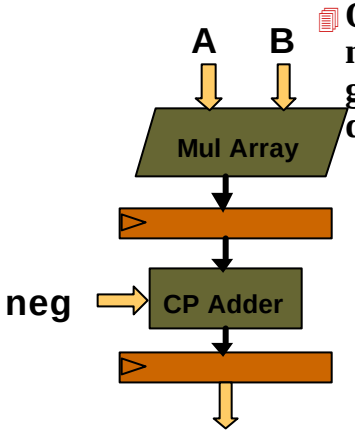
Plan

- Rappels sur la conception ASIC
- Bibliothèques de cellules
- Générateurs de RAM, ROM
- DRAM Embarquées
- Génération de Chemins de Données

Contexte

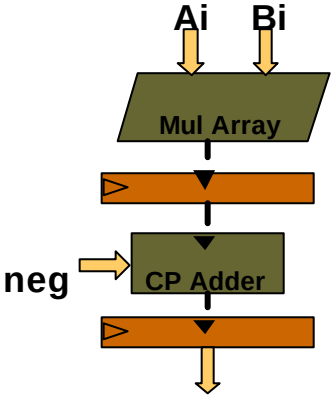
- Les systèmes numériques se découpent en « partie contrôle » / « partie calcul »
- Les parties « calcul » sont
  - *Souvent causes des chemins critiques*
  - *Souvent organisées autour de transferts de données de tailles fixes (8b, 16b, 32b…)*
- Des techniques spécifiques doivent être utilisées pour optimiser ces parties

Idée de départ « cousu-main »



- On peut reproduire au niveau « bit » la structure globale du chemin de données

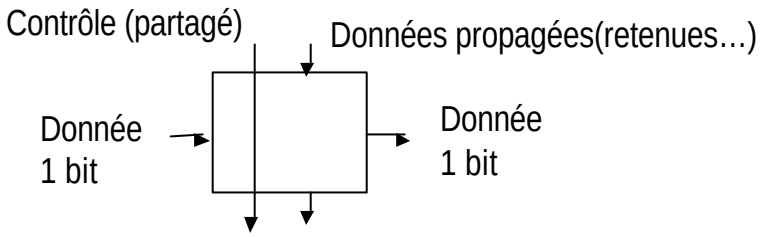
Idée de départ « cousu-main »



- On construit une structure optimale au niveau bit « bit-slice »
- On multiplie ces « bit-slice » pour créer le dessin final

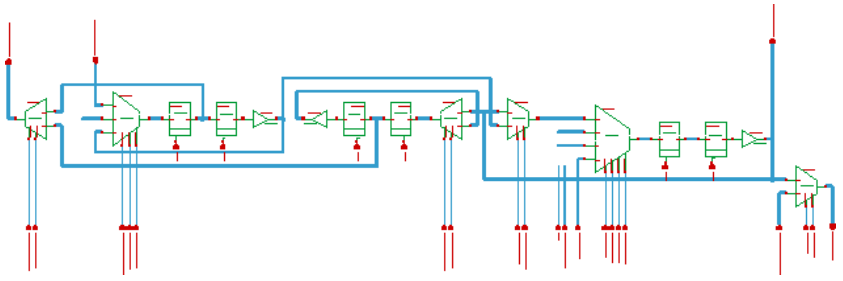
Idée de départ « cousu-main »

- Cellules physiques ad-hoc (bibliothèques)
- Génération par simple « aboutement » des cellules
- Un sens privilégié pour le contrôle
- Un sens (90°) privilégié pour les données



Exemple

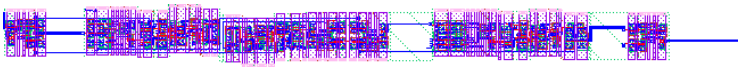
- Source http://www.cs.berkeley.edu/~willyc/rcore/report/n.2.6.html



- Un chemin de données :
- En gras la propagation des données
- En bas les signaux de contrôle

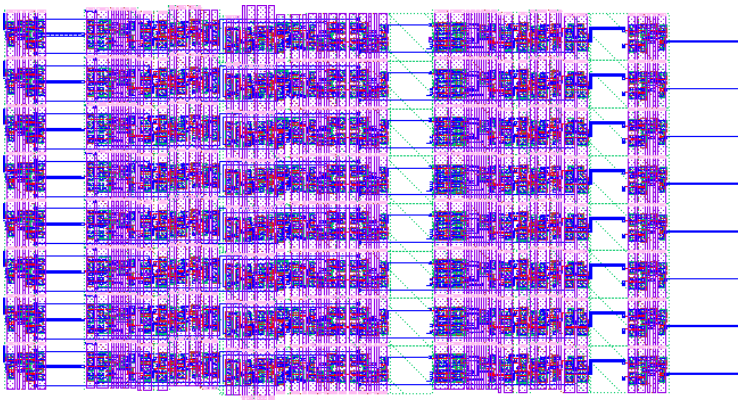
Exemple : la « tranche 1 bit »

- **Source http://www.cs.berkeley.edu/~willyc/rcore/report/n.2.6.html**



• Le compilateur de placement optimise la position des cellules pour minimiser la longueur des fils

Exemple : la « tranche 8 bits »



• Le bloc dur peut être connecté au reste du circuit

Conséquences et contraintes

- **Pas d'indépendance technologique**
- **Outils ad-hoc**
- **Le code VHDL ou VERILOG doit « explicitement » séparer chemin de données et contrôle (le chemin de donnée est une instance en « dur »)**

Solutions non liées à la technologie

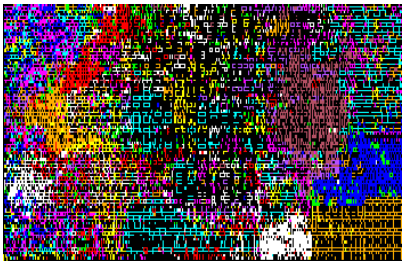
- **Les vendeurs d'outils de CAO proposent des compilateurs de DATAPATH :**
  - ***Extraction des chemins de données dans le code VHDL***
  - ***Proposition de solution « globale » pour chaque opérateur arithmétique (choix de multiplieurs, choix d'additionneurs…)***
  - ***Pas de notion de « bit-slice » : l'outil de placement n'exploite pas la structure globale du chemin données***
- **Exemple :**
  - ***Datapath Compiler (Cadence)***
  - ***DesignWare+Module Compiler (Synopsys)***

Solutions intermédiaires

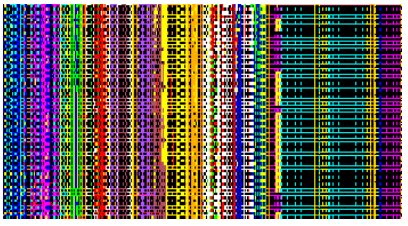
- **Des vendeurs d'outils de CAO proposent des optimiseurs de placement de datapath en cellules standard**
- **Exemple Arcadia Design (outil Mustang) (http://www.arcadiadesign.com)**

Mustang

Placement « classique »



Placement « datapath »



Les couleurs des cellules correspondent aux fonctions
Gain en densité
Gain en vitesse