

ARCHITECTURES ET SYSTEMES A MICROPROCESSEURS
ELEC 288 & ELEC 365
INFORMATIQUE TEMPS REEL ET RESEAUX DE TERRAINS



Intel Xscale PXA 270

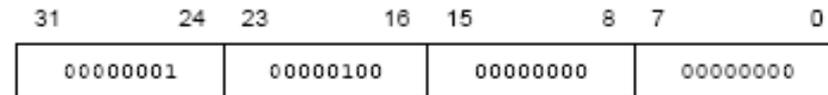
Caractéristiques de la technologie Xscale

- × Année : 2002 - 2004
- × Particularité : Remplace les micro-C&P basés sur Intel StrongARM
Qui remplaçaient les processeurs i860 et i960
- × Architecture : ARM v5TE ISA (2000)
 - les instructions ISA à virgules flottantes
 - + Diverses instructions (43 SIMD pour le 270)
DSP
(Set d'instructions Wireless MMX)
(Set d'instructions Streaming d'Intel (PIII))
 - + 7 stage integer, 8 stage memory Superpipelined RISC
- × Composants : Unités de calculs : Core + CP0 + MAC + CP14 + CP15
Cache de données (32ko) + mini-cache (2ko),
de programme (32ko)+ D&I MMU
Divers buffers : Fill, Write, Pend, Branch Target
Unités de debug + JTAG
- × Familles : IXP – Processeurs pour réseau
PXA – Processeurs pour applications
PXA 210/25x – 26x – 27x – (29x)
IOP – Processeurs d'entrées/sorties
IXC – Processeurs de contrôle

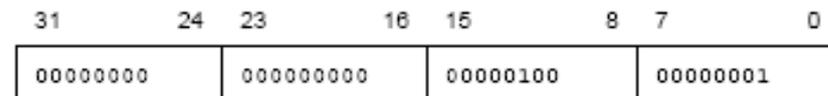
ARM v5TE ISA

- × Traite des données de 8, 16 (mode Thumb) et 32 bits (mode par défaut)
- × Maîtrise à la fois le little (par défaut) et le big Endian (CP15, R1, 7^{ième} bit)
Ex : 1025

- Big Endian



- Little Endian



- × Travaille dans l'un des 7 modes du processeurs :
Utilisateur, Système, Superviseur, Instruction annulée ou non définie, Interruption rapide (FIQ) ou normal (IRQ)
- × Utilise 16 registres de 32 bits : R13 SP, R14 LR, R15 PC
« A current program status register (CPSR) »
« A saved program status register (SPSR) »
CP15 : Identification et contrôle d'opération (Cache, Fault, ...)
CP14 : (Debug, Monitoring)
- × Gère 7 exceptions : Reset, Undefined Instructions, Software Interrupts
Prefetch & Data Abort, Normal & Fast Interrupt

Instructions DSP ajoutées

Instruction	Operation
SMLA _{xy}	$32 \leftarrow 16 \times 16 + 32$
SMLAW _y	$32 \leftarrow 32 \times 16 + 32$
SMLAL _{xy}	$64 \leftarrow 16 \times 16 + 64$
SMUL _{xy}	$32 \leftarrow 16 \times 16$
SMULW _y	$32 \leftarrow 32 \times 16$
QADD	Adds 2 registers & saturates result if overflow occurred
QDADD	Doubles & saturates 1st register, then adds to 2nd reg & saturates
QSUB	Subtracts 2 registers & saturates result if overflow occurred
QDSUB	Doubles & saturates 1st reg, then subtracts 2nd reg & saturates

Autres instructions ajoutées

Instruction	Operation
BKPT	Software Breakpoint
BLX	Branch with Link and Exchange (switch to/from Thumb instruction set)
CLZ	Count Leading Zeroes
LDM/LDR > PC	Load from Memory or Register > PC can cause transfer to Thumb
MAR	Move CP0 40-bit Accumulator to 2 general ARM registers
MRA	Move 2 general ARM registers to CP0 40-bit Accumulator
LDRD	Load 2 general ARM registers from memory
STRD	Store 2 general ARM registers to memory
PLD	Preload Line > Data Caches (abort before any exceptions are taken)

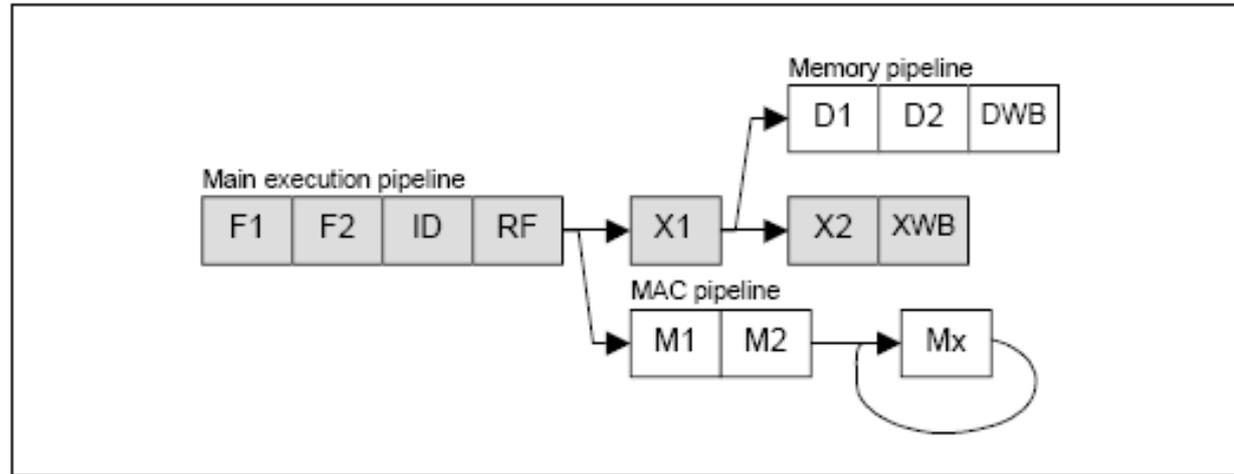
Superpipeline de 6 à 9 étapes

- x « The integer pipe » (7) : Branch Target Buffer (BTB) / Fetch 1, Fetch 2, Decode, Register File / Shift, ALU Execute, State Execute, Integer Writeback
- x « The memory pipe » (8) : Branch Target Buffer (BTB) / Fetch 1, Fetch 2, Decode, Register File / Shift, ALU Execute, Memory Stages Data Cache 1, Data Cache 2, and Data Cache Writeback
- x « The MAC pipe » (6 à 9) : Branch Target Buffer (BTB) / Fetch 1, Fetch 2, Decode, Register File / Shift, MAC stages MAC1, (MAC2, MAC 3, MAC 4,) and Data Cache Writeback

S'il est possible de prédire avec succès les résultats d'un branchement, ces « pipes » améliorent le taux d'exécution d'instructions. En effet, grâce au BTB, aucun temps de latence n'est alors à subir. Cependant, en cas d'échec, 4 à 5 cycles sont perdus. Heureusement, la plupart du temps, BTB réussit ses prévisions.

La longueur a cependant quelques inconvénients (...). A noter que le bus permet le transfert de 32 bits de données à enregistrer et à charger chaque cycle d'horloge. (Fréquence maximale de 600Mhz)

Superpipeline de 6 à 9 étapes (suite)



Pipe / Pipestage	Description	For More Information
Main Execution Pipeline <ul style="list-style-type: none"> • IF1/IF2 • ID • RF • X1 • X2 • XWB 	Handles data processing instructions Instruction Fetch Instruction Decode Register File / Operand Shifter ALU Execute State Execute Write-back	Section 2.2.3 Section 2.2.3.1 Section 2.2.3.2 Section 2.2.3.3 Section 2.2.3.4 Section 2.2.3.5 Section 2.2.3.6
Memory Pipeline <ul style="list-style-type: none"> • D1/D2 • DWB 	Handles load/store instructions Data cache access Data cache writeback	Section 2.2.4 Section 2.2.4.1 Section 2.2.5.1
MAC Pipeline <ul style="list-style-type: none"> • M1-M5 • MWB (not shown) 	Handles all multiply instructions Multiplier stages MAC write-back occurs during M2-M5	Section 2.2.5 Section 2.2.5 Section 2.2.5

Branch Target Buffer

- ×(Dés)activé via le CP15.R1
- 128 entrées dont chacune contient :
 - L'adresse d'une branche d'instructions
 - L'adresse cible de cette même branche
 - Un historique de la branche précédente (4 états : Prise forte/faible ou pas)
 - Si pris : On va chercher la branche à l'adresse cible
 - Si pas pris : On passe à l'instruction séquentielle suivante
- ×Les données associées à une branche sont enregistrées dès la première saisie de la branche. Le slot dans lequel elle se trouve a un historique dit « Strongly Not taken ». Les données sont sans cesse écrasées, ré-écrites.

I&D MMU

- ×(Dés)activation totale possible
- ×Contrôle : la traduction des adresses logiques/physiques
 - les accès (autorisés) à la mémoires
 - l'identification des différentes zones de mémoires et attributs
- ×32 entrées : 0 à 30 premières peuvent être verrouillées
 - Instruction or Data translation look-aside buffer (ITLB or DTLB)
 - Ordonnancement Round-Robin (T=, no priority)
- ×S'il manque une donnée/instruction, une table reprenant ce qu'il y a en mémoire, les autorisations (toutes programmables), ... est invoquée et les informations manquantes sont téléchargées dans le buffer DTLB/ITLB, respectivement. Le MMU s'en enquiert et continue...
- ×S'il l'accès est non-autorisée, une exception est envoyée au processeur.

D&I Cache (32ko)

- ×(Dés)activé et contrôlé via D&I TLB
 - ×Très grande pour contenir des tables/programmes entiers très utilisées
 - ×Permet l'accès par le processeur à la fréquence de ce dernier.
 - ×Réduit donc le temps d'inactivité du processeur
-
- ×32 sets / 32 entrées contenant chacune :
 - « A tag adress »
Huit mots de 32 bits ainsi qu'un bit de parité par mots
 - « A dirty bit » pour les demi-mots au sein d'une ligne (d-cache seulement)
 - Un bit de validité
-
- ×Chaque set peut être (dé)verrouillé, et dans le cas de la d-cache utilisé comme de la mémoire SRAM locale. Les sets non verrouillées sont remplacés via l'ordonnancement Round Robin (voir slide précédente).
-
- ×I-cache : Si une ligne est manquante, on regarde ce qu'il y a dans la file d'attente (à 2 niveaux). L'accès à l'instruction suivante est autorisé si aucune des données requises n'est tributaire des opérations déjà présentes.
-
- ×D-cache : Elle travaille de concert avec « the load and pend buffers » afin d'éviter les “hit-under-miss” qui permettent au processeur d'accéder à des données après un « miss ». Elle travaille également avec « the write buffer » pour les données devant être inscrites en mémoires. (IDEM Mini-cache).

Mini-Data Cache (2ko)

- ×(Dés)activable via les registres des co-processeurs (voir d-cache)
- ×Conçue pour les données changeant très fréquemment (MPEG video, ...) afin de délivrer la d-cache des « data trashing »
- ×32 sets / 2 entrées contenant chacune : voir d-cache
- ×Particularité : Elle ne peut être verrouillée.

Fill Buffer (FB) & Pend Buffer (PB)

- ×Contrôlés via les registres des co-processeurs (vidange).
- ×4 entrées
- ×Travaillent de concert avec le processeur pour contenir les données qui ne peuvent aller dans les diverses caches jusqu'à ce que le contrôleur de bus puisse interagir avec elles.
- ×Travaillent de concert avec les mémoires pour éviter les « hit-under-miss »
Le processeur peut donc subir jusqu'à 4 miss avant de caler.

Write Buffer (WB)

- ×Tjs activés
- ×Contrôlés via les registres des co-processeurs (vidange).
- ×8 entrées des 128 bits chacune
- ×Peut recevoir des données du core, des divers caches (de données)
- ×Retient les données jusqu'à ce que le contrôleur de bus puisse interagir avec
- ×Divers modes de fonctionnement : « WB coalescing » (bit d'attributs)

CP 15

Register	Access	Description
0	Read / Write-ignored	ID
0	Read / Write-ignored	Cache Type
1	Read / Write	Control
1	Read / Write	Auxiliary Control
2	Read / Write	Translation Table Base
3	Read / Write	Domain Access control
4	Unpredictable	Reserved
5	Read / Write	Fault Status
6	Read / Write	Fault Address
7	Read-unpredictable / Write	Cache Operations
8	Read-unpredictable / Write	TLB Operations
9	Read / Write	Cache Lock Down
10	Read / Write	TLB Lock Down
11-12	Unpredictable	Reserved
13	Read / Write	Process ID (PID)
14	Read / Write	Breakpoint Registers
15	Read / Write	Coprocessor Access

CP 14

Register	Access	Description
0	Read / Write	Performance Monitoring: Control Register
1	Read / Write	Performance Monitoring: Clock Counter
2	Read / Write	Performance Monitoring: Event Counter #1
3	Read / Write	Performance Monitoring: Event Counter #2
4-5	Unpredictable	Reserved
6	Read / Write	Core Clock Configuration Register
7	Read / Write	Power Mode Register
8	Read / Write	Software Debug: TX Register
9	Read / Write	Software Debug: RX Register
10	Read / Write	Software Debug: Debug Control and Status Register
11	Read / Write	Software Debug: Trace Buffer Register
12	Read / Write	Software Debug: Checkpoint 0 Register
13	Read / Write	Software Debug: Checkpoint 1 Register
14	Read / Write	Software Debug: TXRX Control Register
15	Unpredictable	Reserved

Multiply-Accumulate CP - CP0

40 bits d'accumulation

- × Diverses sets d'instructions
MAR, MRA
- × Diverses sortes de multiplications possibles
- × Le tout en moins d'un cycle (voir PXA270)

CP Interface :

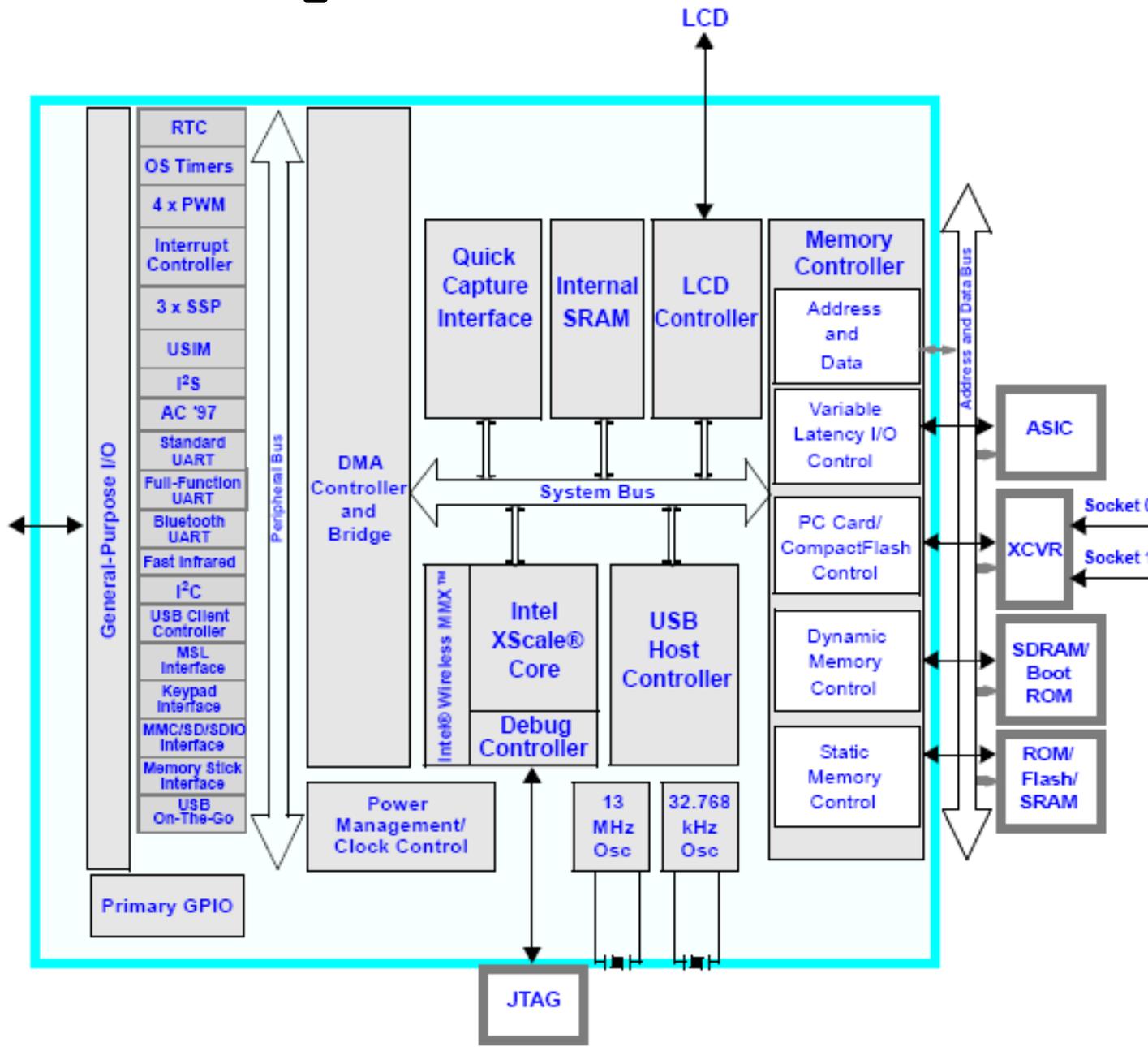
- × Bus de 32 bits pour le transfert des données entre les co-processeurs et les registres du processeur (à la vitesse du processeur)
- × Supporte l'exécution d'instructions en //
Il continue à travailler après commande
Supporte l'exécution couplée au processeur
Il attend

Caractéristiques du processeur PXA 270



- x Année : 2004
- x Famille : PXA 27x surnommée Bulverde
- x Température : de -45° à 80°
- x Fréquence CPU : 312 / 416 / 520 / 624 Mhz
- x Fréquence BUS : Interne de 200 Mhz
Externe de 100 Mhz
- x Nombres de registre : 16 de 64 bits
- x Possibilités sous-jaçantes : 8 additions par cycle*
4 multiplications par cycle*
- x Mémoires : 256 Kib SRAM interne
- x Particularités :
 - Rétrocompatibilité (Intel StrongARM)
 - Intel Wireless MMX
 - Pour le multimédia* – Provient de la famille Pentium
 - Intel Quick Capture
 - Quick View mode – Quick Shot mode – Quick Video mode
 - Intel SpeedStep
 - Ajustement dynamique de la puissance en fct de la demande
 - Sont possibles divers états de tension (0,85 à 1,55 V)
 - Sont possibles divers états de veille
- x Applications : Systèmes embarqués dont les PocketPCs
Dell, Mitac, Asus, Acer, Palm, ...

Block Diagramme



DMA Contrôleur :

× Les données venant de ou allant vers l'extérieur à destination ou à partir de la mémoire centrale sont contrôlées à l'aide de ce système et ce sans l'aide du processeur si ce n'est pour les phases d'initialisation et de conclusion. Il favorise donc la fluidité (-polling)

× Deux modes sont supportés : Fly-by et Flow-Through

× Nombreux périphériques :

I²S, Audio Contrôler'97, UART, USB, PWM, Real Time Clock, I²C, ...

Unité de débogage

- × Elle permet au core de continuer normalement son traitement
- × Le Programme de débogage tourne sur un système extérieur (JTAG)
- × Il lui est permis de stopper l'exécution d'un programme et le cas échéant ré-orienter les prochaines instructions vers une routine
- × Les exceptions de débogage ont aussi cette faculté :(6)
 - Instruction breakpoint,
 - Data breakpoint,
 - Software breakpoint,
 - External debug breakpoint,
 - Exception vector trap,
 - Trace buffer full breakpoint,
 - SOC debug break
- × Une fois l'exécution stoppée, le programme de débogage peut examiner et modifier, si besoin est, l'état du processeur, des co-processeurs et de la mémoire
- × Cela achevé, l'unité relance le programme général

Unité de débogage (suite)

Event	Priority
Reset	1 (highest)
Vector Trap	2
data abort (precise)	3
data bkpt	4
data abort (imprecise)	5
external debug break, trace-buffer full, SOC debug break	6
FIQ	7
IRQ	8
instruction breakpoint	9
pre-fetch abort	10
undef, SWI, software Bkpt	11

- × Elle arbore de nombreux registres :
 - 4 registres Hardware pour les breakpoints d'instructions et de données
 - 1 registre Hardware de contrôle pour les breakpoints de données
(Il sert de masque pour les premiers données)
 - Un buffer arborant 256 entrées permet la capture des adresses, ...
- × Diverses commandes peuvent être utilisées.
 - Citons l'instruction JTAG, LDIC qui une fois lancée, va permettre le téléchargement du progamme « debug handler » dans la mémoire mini-i cache de 2ko, spécialement conçue à cette fin.
(et la raison pour laquelle, nous n'en parlons pas avant)

JTAG

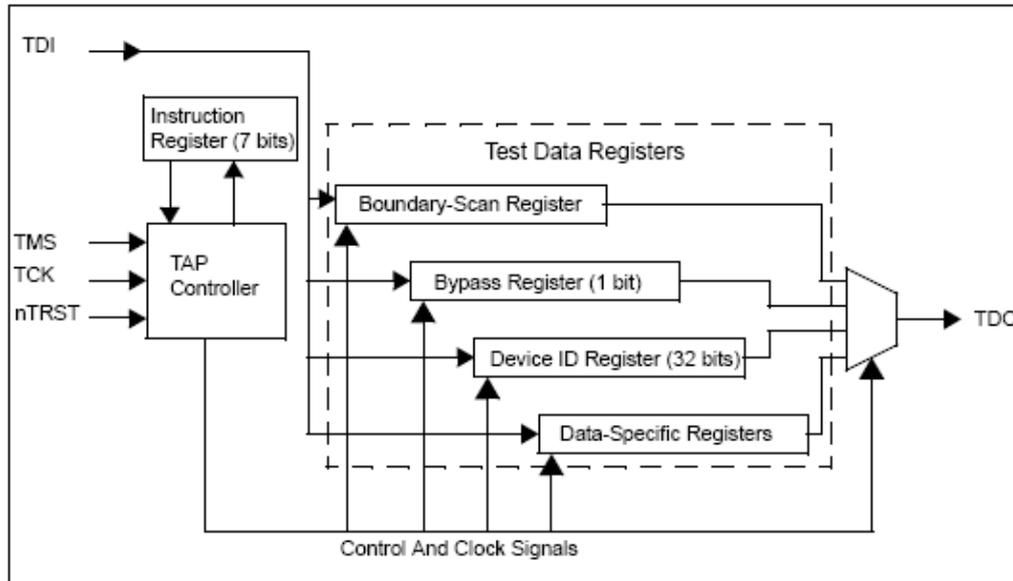
- × Acronyme de Joint Test Action Group
 - × Nom usuel de la norme IEEE 1149.1 intitulée au départ :
 - « Standard Test Access Port and Boundary-Scan Architecture »
 - × Date 1990 + divers ajouts dont le « BS Description Language » en 1994

 - × Permet le test des circuits imprimés, les accès aux différents ports, ...
 - × Permet aussi le test de sous-blocs
 - × Cela en mettant en place une « porte de sortie » bien utile vers l'extérieur
 - × Très utilisé dans l'industrie

 - × Les ports JTAG peuvent être aussi utilisés pour avoir accès à l'unité de débogage qui contrôle le débogage du programme et le monitoring des données.

 - × Consiste en : Un contrôleur de port d'accès de test (TAP) et un de pins
 - Divers registres : Boundary-Scan, By-pass, Device ID
 - Des registres d'instructions et de données
 - Des signaux dédiés : TDI - Test Data In
 - TDO - Test Data Out
 - TCK - Test Clock
 - TMS - Test Mode Select
 - nTRST - Test Reset
- On dénombre généralement 5 pin's. (Protocole série)

JTAG (suite)



TCK : Test Clock – clock input for the TAP controller and instruction and test data registers

TMS : Test Mode Select – controls operation of the TAP controller. The TMS input is pulled high when it is not being driven. TMS is sampled on the rising edge of TCK.

TDI : Test Data In – serial data input to the instruction and test data registers. Data at TDI is sampled on the rising edge of TCK. TDI is pulled high when it is not being driven.

TDO : Test Data Out – serial data output. Data at TDO is clocked out on the falling edge of TCK. It provides an inactive (highimpedance) state during non-shift operations to support parallel connection of TDO outputs at the

board or module level.

nTRST : Test Reset – provides asynchronous initialization of the JTAG test logic. Asserting this pin puts the TAP controller in the *Test-Logic-Reset* state.

An external source must drive nTRST before or at the same time as the hardware nRESET pin for correct TAP controller and device operation.

Dix-sept instructions possibles

Instruction Code	Instruction Type	Instruction Name	Instruction Code	Instruction Type	Instruction Name
0b000_0000	mandatory public	extest	0b000_1010 - 0b000_1111	private	private
0b000_0001	mandatory public	sample/preload	0b001_0000	user defined	dbgtx
0b000_0010	user defined	dbgtx	0b001_0001- 0b011_0101	private	private
0b000_0011	private	private	0b011_0110	user defined	flashload
0b000_0100	optional public	clamp	0b011_0111	user defined	flashprogram
0b000_0101 - 0b000_0110	private	private	0b011_1000 - 0b111_1101	private	private
0b000_0111	user defined	ldic	0b111_1110	optional public	idcode
0b000_1000	optional public	highz	0b111_1111	mandatory public	bypass
0b000_1001	user defined	dcsr			

flashload	0b011_0110	The flashload instruction is for loading values that are programmed into an external flash device. This instruction mimics the sample/preload instruction, but only on a subset of boundary-scan register cells, the flash data register, needed to program an external flash device to reduce flash programming time. Flash values intended for programming are loaded during the Shift-DR state.
flashprogram	0b011_0111	The flashprogram instruction for programming the values into an external flash device. This instruction mimics the extest instruction, but uses only on a subset of boundary-scan register cells, specifically the flash data register. This subset reduces flash programming time. When flashprogram is selected in the Update-IR state, relevant output signal pins are driven by values shifted into the flash data register from a previous flashload instruction. The signals not listed in the flash data register are not affected.
idcode IEEE 1149.1 optional	0b111_1110	The idcode instruction is used with the device identification (ID) register. It connects the ID register between TDI and TDO in the Shift-DR state. When selected, idcode parallel-loads the hard-wired identification code (32 bits) on TDO into the ID register on the rising edge of TCK in the Capture-DR state. NOTE: The ID register is not altered when data is shifted in on TDI.
bypass IEEE 1149.1 required	0b111_1111	The bypass instruction selects the bypass register between TDI and TDO pins in the Shift-DR state, effectively bypassing the processor's test logic. 0b0 is captured in the Capture-DR state. While this instruction is in effect, no other test data registers have any effect on the operation of the system. Test data registers with both test and system functionality perform their system functions when this instruction is selected.

Optimisation des langages de haut niveau

- × Les fonctionnalités d'un système embarqué dépendent fortement des techniques de programmation employée. Plusieurs techniques peuvent être cependant employées.
- × Citons le cas de l'instruction « preload »

Elle permet de précharger des données dans le d-cache ou le mini d-cache sans passer par les registres, réduisant ainsi la pression qu'il peut avoir à leur niveau.

Un bémol est cependant à émettre. En effet, il n'est pas garanti que les données soient bien chargées. En cas d'erreur, le processeur ignorera l'instruction pré-chargée et exécutera la prochaine instruction.

Ceci est particulièrement avantageux dans le cas des listes, ... Un pointeur nul au bout d'une liste ne créera en effet pas d'erreur.

Inséré dans une boucle, il est aisé de remarquer que les données des instructions suivantes seront placées à un certain offset des données de l'instruction en cours. On peut donc prédire leur emplacement. Le nombre d'itérations de « preload » est connu sous le nom de « preload scheduling distance (PSD) ».

« Hot Debug »

- × En raison de l' « Intel XScale® core debug architecture » , une session de débogage engendrera à la connection du débogueur un reset du processeur.
- × Dans de nombreuses applications, un reset est à éviter.
- × « Hot Debug » est un software qui permet d'éviter ce reset. Afin de l'utiliser, on ajoute différentes portions de codes au début de notre programme.

Lorsqu'une telle session est initialisée, un reset n'agit plus. En effet, le processeur reçoit alors l'instruction de télécharger ce qu'il y a dans la mini i-cache, en l'occurrence le debug handler, et le contrôle est redonné au débogueur.

Conclusion

- × La famille Intel Xscale est actuellement l'une des meilleures du marché.
- × Ils existent trois processeurs de la même famille (Bulverde) :
 - 270 : 169 mm² – 356 pins – VF-BGA (32 bits) – Max. 624Mhz
Sans mémoire
 - 271 : 196 mm² – 336 pins – FS-CSP (32 bits) – Max. 416Mhz
32Mb Intel StrataFlash + 32Mb SDRAM basse puissance
 - 272 : 196 mm² – 336 pins – FS-CSP (32 bits) – Max. 520Mhz
64Mb Intel StrataFlash

Nb : IntelStraflash est le nom donné aux mémoires à haute densité et performance de 4^{ème} génération de la technologie MLC. (0,13 µm)

- × Le remplaçant pointe déjà le bout de son nez.
Il s'agit de la famille PXA 29x surnommée Monahans.
Parmi les choses annoncées en 2005 à son propos :
 - Possibilité d'encoder des vidéos hautes définitions
 - 1000 MIPS pour 1.25Ghz vs 800MIPS pour le 624Mhz Bulverde
 - Diverses fonctionnalités graphiques intégrées
 - Annulation du remplaçant au co-processeur graphique 2700G

Bibliographie

- x « Intel® PXA27x Processor Family : Design Guide », Intel corporation, 2005
- x « Intel® PXA27x Processor Family : Optimization Guide », Intel corporation, 2004
- x « Intel XScale® Core : Developer's Manual », Intel corporation, 2003
- x « Intel Xscale PXA 270 : Product Brief », Intel corporation, 2003
- x « Recommended JTAG Circuitry for Debug with Intel® Xscale™ Microarchitecture : Application Note », Intel corporation, 2001
- x « Intel® XScale™ Microarchitecture Programmer Model for Big Endian : Application Note », Intel corporation, 2001
- x « Intel® PXA27x Processor Family Design Check List : Application Note », Intel corporation, 2001
- x « Intel Xscale Microarchitecture : Technical Summary, Intel Corporation, 2000